ANEXO 3

Código Matlab interfaz *simconverter*

Este apéndice contiene el listado del código Matlab comentado del programa, se separarán los trozos de código por ficheros:

- **GUIproyecto.m**

```
% Script that has got the code that controls the behaviour of the GUI

function varargout = GUIproyecto(varargin)
% GUIPROYECTO M-file for GUIproyecto.fig
%      GUIPROYECTO, by itself, creates a new GUIPROYECTO or raises the existing
%      singleton*.
%
%      H = GUIPROYECTO returns the handle to a new GUIPROYECTO or the handle to
%      the existing singleton*.
%
%      GUIPROYECTO('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in GUIPROYECTO.M with the given input arguments.
%
%      GUIPROYECTO('Property','Value',...) creates a new GUIPROYECTO or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before GUIproyecto_OpeningFunction gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to GUIproyecto_OpeningFcn via varargin.
%
%      *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 GUIproyecto

% Last Modified by GUIDE v2.5 05-Dec-2005 14:38:36

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @GUIproyecto_OpeningFcn, ...
                   'gui_OutputFcn',  @GUIproyecto_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin & isstr(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


% --- Executes just before GUIproyecto is made visible.
function GUIproyecto_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
```

```
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUIproyecto (see VARARGIN)

fig = handles.Current;                % Handle to the figure

GUIinit(handles);                     % Iniatilize values of global variables

% Choose default command line output for GUIproyecto
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);


% --- Outputs from this function are returned to the command line.
function varargout = GUIproyecto_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

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

% --- Initialize values of global variables
function GUIinit(handles)

global Ts;                    % Global variables to be initialized
global FS;
global A;
global fsin;
global N;
global sigma;
global sigmajitter;
global bandwidth;
global minsignalband;
global val;
global val2;
global Select;
global modelsimulated;
global VoffsetVrefpn;
global Voffset0;
global nsamples;
global Effort;
global Eps;
global SR;
global Tau;
global Aux1;
global Aux2;
global Aux3;
global Aux4;
global Ao;
global AodB;
global Auto;
global wait;

% Initializing global variables from text boxes, etc.
```

```
Ts = 1/(str2double(get(handles.SamplingFrequency,'String')));
FS = str2double(get(handles.FullScale,'String'));
A = str2double(get(handles.Amplitude,'String'));
fsin = str2double(get(handles.Frequency,'String'));
N = str2double(get(handles.NumberBits,'String'));
sigma = str2double(get(handles.Sigma,'String'));                % Sigma is the typical deviation of
the offset distribution
bandwidth = str2double(get(handles.Bandwidth,'String'));
minsignalband = str2double(get(handles.Minsignalband,'String'));
sigmajitter = str2double(get(handles.Jitter,'String'));
modelsimulated = get(handles.ModelSimulated,'String');
set(handles.ModelSimulated,'String',modelsimulated);
wait = str2double(get(handles.Timewait,'String'));
val = get(handles.Sweep_Popupmenu,'Value');
val2 = get(handles.Sweep2_Popupmenu,'Value');

slider_step(1)=1/9;
slider_step(2)=1/9;
set(handles.Effort_Slider,'sliderstep',slider_step,'Max',10.0,'Min',1.0,'Value',1.0);
Effort = get(handles.Effort_Slider,'Value');


nsamples = str2double(get(handles.Nsamples,'String'));
VoffsetVrefpn = sigma * randn(1);                % Offsets are random numbers with typical
deviation sigma
Voffset0 = sigma * randn(1);
Select = 1;                          % Ramp Input by default


AodB = str2double(get(handles.Gain,'String'));
Eps = str2double(get(handles.Mismatch,'String'));
SR = str2double(get(handles.SR,'String'));
Tau = str2double(get(handles.Tau,'String'));
Aux1 = str2double(get(handles.Aux1,'String'));
Aux2 = str2double(get(handles.Aux2,'String'));
Aux3 = str2double(get(handles.Aux3,'String'));
Aux4 = str2double(get(handles.Aux4,'String'));
Ao = 10^(AodB/20);
Auto=1;                              % Automatic inputs change by default



% --- Executes during object creation, after setting all properties.
function SamplingFrequency_CreateFcn(hObject, eventdata, handles)
% hObject    handle to SamplingFrequency (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes when editing Sampling Freq text box
function SamplingFrequency_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to SamplingFrequency (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of SamplingFrequency as text
%        str2double(get(hObject,'String')) returns contents of SamplingFrequency as a double
global Ts;
global bandwidth;
global fsin;
global minsignalband;
global Auto;
Ts = 1/(str2double(get(hObject,'String')))

% If automatic inputs change is selected, some values change
if(Auto==1)
   bandwidth = 1/(2*Ts);
   set(handles.Bandwidth,'String',bandwidth);
   fsin = 1/(5.1*Ts);
   set(handles.Frequency,'String',fsin);
   minsignalband = 1/(400*Ts);
   set(handles.Minsignalband,'String',minsignalband);
end

% --- Executes during object creation, after setting all properties.
function FullScale_CreateFcn(hObject, eventdata, handles)
% hObject    handle to FullScale (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes when editing FS text box
function FullScale_Callback(hObject, eventdata, handles)
% hObject    handle to FullScale (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of FullScale as text
%        str2double(get(hObject,'String')) returns contents of FullScale as a double
global FS;
global A;
global Auto;
FS = str2double(get(hObject,'String'))

% If automatic inputs change is selected, some values change
if(Auto==1)
   A = FS/2;
   set(handles.Amplitude,'String',A);
end

% --- Executes on button press in DNL_pushbutton.
function DNL_pushbutton_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to DNL_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global DNLCOD;
global axisxlabelVolt;
global axisylabelDNL;

% If figure does not exist, create it
if ~isfield(handles,'PlotFigure') | ~ishandle(handles.PlotFigure),
 handles.PlotFigure = figure('Name','DNL','Visible','on',...
          'NumberTitle','off','HandleVisibility','on','IntegerHandle','off');
 handles.PlotAxes = axes('Parent',handles.PlotFigure);
 guidata(hObject,handles)
end

% Plot the figure, giving name, xlabel, ylabel and grid
plot(DNLCOD(1,:), DNLCOD(2,:),'Parent',handles.PlotAxes)
set(0,'CurrentFigure',handles.PlotFigure)
set(handles.PlotFigure,'Name','DNL')
xlabel(axisxlabelVolt)
ylabel(axisylabelDNL)
grid(handles.PlotAxes,'on')


% --- Executes on button press in Spectrum_pushbutton.
function Spectrum_pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to Spectrum_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global F;
global potencia1;
global axisxlabelFrequency;
global axisylabelSpectrum;

% If figure does not exist, create it
if ~isfield(handles,'PlotFigure') | ~ishandle(handles.PlotFigure),
 handles.PlotFigure = figure('Name','Spectrum','Visible','on',...
          'NumberTitle','off','HandleVisibility','on','IntegerHandle','off');
 handles.PlotAxes = axes('Parent',handles.PlotFigure);
 guidata(hObject,handles)
end

% Plot the figure on logarithmic axes, giving name, xlabel, ylabel and grid
semilogx(F,potencia1,'Parent',handles.PlotAxes);
set(0,'CurrentFigure',handles.PlotFigure)
set(handles.PlotFigure,'Name','Spectrum')
xlabel(axisxlabelFrequency)
ylabel(axisylabelSpectrum)
grid(handles.PlotAxes,'on')


% --- Executes on button press in INL_pushbutton.
function INL_pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to INL_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global INLCOD;
global axisxlabelVolt;
global axisylabelINL;
```

```
% If figure does not exist, create it
if ~isfield(handles,'PlotFigure') | ~ishandle(handles.PlotFigure),
 handles.PlotFigure = figure('Name','INL','Visible','on',...
          'NumberTitle','off','HandleVisibility','on','IntegerHandle','off','Position',[10     40     500
400]);
 handles.PlotAxes = axes('Parent',handles.PlotFigure);
   guidata(hObject,handles)
end

% Plot the figure, giving name, xlabel, ylabel and grid
plot(INLCOD(1,:), INLCOD(2,:),'Parent',handles.PlotAxes)
set(0,'CurrentFigure',handles.PlotFigure)
set(handles.PlotFigure,'Name','INL')
xlabel(axisxlabelVolt)
ylabel(axisylabelINL)
grid(handles.PlotAxes,'on')


% --- Executes on button press in Output_Data_SWI_pushbutton.
function Output_Data_SWI_pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to Output_Data_SWI_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global vector;
global axisxlabelTime;
global axisylabelOutput;

% If figure does not exist, create it
if ~isfield(handles,'PlotFigure') | ~ishandle(handles.PlotFigure),
 handles.PlotFigure = figure('Name','Output','Visible','on',...
          'NumberTitle','off','HandleVisibility','on','IntegerHandle','off');
 handles.PlotAxes = axes('Parent',handles.PlotFigure);
 guidata(hObject,handles)
end

% Plot the figure, giving name, xlabel, ylabel and grid
plot(vector(:,1),vector(:,2),'Parent',handles.PlotAxes)
set(0,'CurrentFigure',handles.PlotFigure)
set(handles.PlotFigure,'Name','Output')
xlabel(axisxlabelTime)
ylabel(axisylabelOutput)
grid(handles.PlotAxes,'on')

% --- Executes during object creation, after setting all properties.
function MaxINL_CreateFcn(hObject, eventdata, handles)
% hObject    handle to MaxINL (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

```
% Not used (output result)
function MaxINL_Callback(hObject, eventdata, handles)
% hObject    handle to MaxINL (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

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


% --- Executes during object creation, after setting all properties.
function MaxDNL_CreateFcn(hObject, eventdata, handles)
% hObject    handle to MaxDNL (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% Not used (output result)
function MaxDNL_Callback(hObject, eventdata, handles)
% hObject    handle to MaxDNL (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

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


% --- Executes during object creation, after setting all properties.
function SNDR_CreateFcn(hObject, eventdata, handles)
% hObject    handle to SNDR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% Not used (output result)
function SNDR_Callback(hObject, eventdata, handles)
% hObject    handle to SNDR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

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

```
% --- Executes during object creation, after setting all properties.
function SNR_CreateFcn(hObject, eventdata, handles)
% hObject    handle to SNR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% Not used (output result)
function SNR_Callback(hObject, eventdata, handles)
% hObject    handle to SNR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

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


% --- Executes during object creation, after setting all properties.
function Messages_Listbox_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Messages_Listbox (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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


% Not used (output messages)
function Messages_Listbox_Callback(hObject, eventdata, handles)
% hObject    handle to Messages_Listbox (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

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


% --- Executes on button press in Evaluate_pushbutton.
function Evaluate_pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to Evaluate_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global Start;
global Stop;
global Points;
```

```
global Start2;
global Stop2;
global Points2;
global val;
global val2;

% Take sweep parameters from text boxes
Start = str2double(get(handles.Start,'String'))
Stop = str2double(get(handles.Stop,'String'))
Points = str2double(get(handles.Points,'String'))
Start2 = str2double(get(handles.Start2,'String'))
Stop2 = str2double(get(handles.Stop2,'String'))
Points2 = str2double(get(handles.Points2,'String'))

% Sweep select (if val2 equals 1 => Single Sweep; Others => Double sweep)
switch val2
  case 1              % Sweep 2 = None => Single Sweep
    switch val
      case 1          % Gain sweep
        EstudioGananciaParamGlobal(Start,Stop,Points,handles,hObject);
      case 2          % Mismatch sweep
        EstudioMismatchEtapa1ParamGlobal(Start,Stop,Points,handles,hObject);
      case 3          % Slew Rate sweep
        EstudioSREtapa1ParamGlobal(Start,Stop,Points,handles,hObject);
      case 4          % Tau sweep
        EstudioTauEtapa1ParamGlobal(Start,Stop,Points,handles,hObject);
      case 5          % Sigma sweep
        EstudioSigmaParamGlobal(Start,Stop,Points,handles,hObject);
      case 6          % Jitter sweep
        EstudioJitterParamGlobal(Start,Stop,Points,handles,hObject);
      case 7          % Sampling Frequency sweep
        EstudioSamplingFreqParamGlobal(Start,Stop,Points,handles,hObject);
      case 8          % Input amplitude sweep
        EstudioAmplitParamGlobal(Start,Stop,Points,handles,hObject);
      case 9          % Full Scale sweep
        EstudioFSParamGlobal(Start,Stop,Points,handles,hObject);
      case 10         % Input frequency sweep
        EstudioFreqParamGlobal(Start,Stop,Points,handles,hObject);
      case 11         % Bandwidth sweep
        EstudioBWParamGlobal(Start,Stop,Points,handles,hObject);
      case 12         % Aux 1 sweep
        EstudioAux1ParamGlobal(Start,Stop,Points,handles,hObject);
      case 13         % Aux 2 sweep
        EstudioAux2ParamGlobal(Start,Stop,Points,handles,hObject);
      case 14         % Aux 3 sweep
        EstudioAux3ParamGlobal(Start,Stop,Points,handles,hObject);
      case 15         % Aux 4 sweep
        EstudioAux4ParamGlobal(Start,Stop,Points,handles,hObject);
    end
  case 3              % Sweep 2 = Mismatch
    switch val
      case 1          % Gain and Mismatch sweep

EstudioGananciaMismatchParamGlobal(Start,Stop,Points,Start2,Stop2,Points2,handles,hObjec
t);
    end
  case 5              % Sweep 2 = Tau
    switch val
      case 3          % Slew Rate and Tau sweep
```

```
EstudioSRTauEtapa1ParamGlobal(Start,Stop,Points,Start2,Stop2,Points2,handles,hObject);
    end
  case 6              % Sweep 2 = Sigma
    switch val
      case 1          % Gain and Sigma sweep

EstudioGananciaSigmaParamGlobal(Start,Stop,Points,Start2,Stop2,Points2,handles,hObject);
      case 2          % Mismatch and Sigma sweep

EstudioMismatchSigmaParamGlobal(Start,Stop,Points,Start2,Stop2,Points2,handles,hObject);
      case 3          % Slew Rate and Sigma sweep

EstudioSRSigmaParamGlobal(Start,Stop,Points,Start2,Stop2,Points2,handles,hObject);
      case 4          % Tau and Sigma sweep

EstudioTauSigmaParamGlobal(Start,Stop,Points,Start2,Stop2,Points2,handles,hObject);
    end
  case 7              % Sweep 2 = Jitter
    switch val
      case 1          % Gain and Jitter sweep

EstudioGananciaJitterParamGlobal(Start,Stop,Points,Start2,Stop2,Points2,handles,hObject);
      case 2          % Mismatch and Jitter sweep

EstudioMismatchJitterParamGlobal(Start,Stop,Points,Start2,Stop2,Points2,handles,hObject);
      case 3          % Slew Rate and Jitter sweep
        EstudioSRJitterParamGlobal(Start,Stop,Points,Start2,Stop2,Points2,handles,hObject);
      case 4          % Tau and Jitter sweep

EstudioTauJitterParamGlobal(Start,Stop,Points,Start2,Stop2,Points2,handles,hObject);
      case 5          % Sigma and Jitter sweep

EstudioSigmaJitterParamGlobal(Start,Stop,Points,Start2,Stop2,Points2,handles,hObject);
    end
  case 8              % Sweep 2 = Sampling Freq
    switch val
      case 1          % Gain and Sampling Frequency sweep

EstudioGananciaSamplingFreqParamGlobal(Start,Stop,Points,Start2,Stop2,Points2,handles,hO
bject);
      case 2          % Mismatch and Sampling Frequency sweep

EstudioMismatchSamplingFreqParamGlobal(Start,Stop,Points,Start2,Stop2,Points2,handles,h
Object);
    end
  case 9              % Sweep 2 = Amplit
    switch val
      case 1          % Gain and input amplitude sweep

EstudioGananciaAmplitParamGlobal(Start,Stop,Points,Start2,Stop2,Points2,handles,hObject);
      case 2          % Mismatch and input amplitude sweep

EstudioMismatchAmplitParamGlobal(Start,Stop,Points,Start2,Stop2,Points2,handles,hObject);
    end
  case 10              % Sweep 2 = FS
    switch val
      case 1          % Gain and Full Scale sweep
```

```
EstudioGananciaFSParamGlobal(Start,Stop,Points,Start2,Stop2,Points2,handles,hObject);
        case 2          % Mismatch and Full Scale sweep

EstudioMismatchFSParamGlobal(Start,Stop,Points,Start2,Stop2,Points2,handles,hObject);
      end
    case 11             % Sweep 2 = Freq
      switch val
        case 1          % Gain and input frequency sweep

EstudioGananciaFreqParamGlobal(Start,Stop,Points,Start2,Stop2,Points2,handles,hObject);
        case 2          % Mismatch and input frequency sweep

EstudioMismatchFreqParamGlobal(Start,Stop,Points,Start2,Stop2,Points2,handles,hObject);
      end
    case 14             % Sweep 2 = Aux 2
      switch val
        case 12         % Aux 1 and Aux 2 sweep

EstudioAux1Aux2ParamGlobal(Start,Stop,Points,Start2,Stop2,Points2,handles,hObject);
      end
    case 15             % Sweep 2 = Aux 3
      switch val
        case 12         % Aux 1 and Aux 3 sweep

EstudioAux1Aux3ParamGlobal(Start,Stop,Points,Start2,Stop2,Points2,handles,hObject);
        case 13         % Aux 2 and Aux 3 sweep

EstudioAux2Aux3ParamGlobal(Start,Stop,Points,Start2,Stop2,Points2,handles,hObject);
      end
    case 16             % Sweep 2 = Aux 4
      switch val
        case 12         % Aux 1 and Aux 4 sweep

EstudioAux1Aux4ParamGlobal(Start,Stop,Points,Start2,Stop2,Points2,handles,hObject);
        case 13         % Aux 2 and Aux 4 sweep

EstudioAux2Aux4ParamGlobal(Start,Stop,Points,Start2,Stop2,Points2,handles,hObject);
        case 14         % Aux 3 and Aux 4 sweep

EstudioAux3Aux4ParamGlobal(Start,Stop,Points,Start2,Stop2,Points2,handles,hObject);
      end
end


% --- Executes during object creation, after setting all properties.
function Gain_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Gain (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
  set(hObject,'BackgroundColor','white');
else
  set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

```
% --- Executes when editing Gain text box
function Gain_Callback(hObject, eventdata, handles)
% hObject    handle to Gain (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Gain as text
%        str2double(get(hObject,'String')) returns contents of Gain as a double
global AodB;
global Ao;
AodB = str2double(get(hObject,'String'))
Ao = 10^(AodB/20)

% --- Executes during object creation, after setting all properties.
function SR_CreateFcn(hObject, eventdata, handles)
% hObject    handle to SR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes when editing SR text box
function SR_Callback(hObject, eventdata, handles)
% hObject    handle to SR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of SR as text
%        str2double(get(hObject,'String')) returns contents of SR as a double
global SR;
SR = str2double(get(hObject,'String'))

% --- Executes during object creation, after setting all properties.
function Mismatch_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Mismatch (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes when editing Mismatch text box
function Mismatch_Callback(hObject, eventdata, handles)
% hObject    handle to Mismatch (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of Mismatch as text
%        str2double(get(hObject,'String')) returns contents of Mismatch as a double
global Mismatch;
Mismatch = str2double(get(hObject,'String'))


% --- Executes on button press in Run_pushbutton.
function Run_pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to Run_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global Gain;
global Mismatch;
global SlewRate;
global TimeConstant;
global Aux1;
global Aux2;
global Aux3;
global Aux4;
global wait;
global sigmajitter;
global Select;
global INLMAX;
global DNLMAX;
global misingcode;
global Sndr;
global Snr;
global Enob;
global axisxlabel;
global axisylabel;
global Timewait;
global go;


% Take single simulation parameters from text boxes
Gain = str2double(get(handles.Gain,'String'));
Mismatch = str2double(get(handles.Mismatch,'String'));
SlewRate = str2double(get(handles.SR,'String'));
TimeConstant = str2double(get(handles.Tau,'String'));
Aux1 = str2double(get(handles.Aux1,'String'));
Aux2 = str2double(get(handles.Aux2,'String'));
Aux3 = str2double(get(handles.Aux3,'String'));
Aux4 = str2double(get(handles.Aux4,'String'));
sigmajitter = str2double(get(handles.Jitter,'String'));
sigma = str2double(get(handles.Sigma,'String'));
Timewait = str2double(get(handles.Timewait,'String'));

% Offsets are random numbers with typical deviation sigma
VoffsetVrefpn = sigma * randn(1);
Voffset0 = sigma * randn(1);

% Calling to the function which controls single simulations


SimulacionNoIdeal(Gain,Mismatch,SlewRate,TimeConstant,sigmajitter,sigma,Aux1,Aux2,Aux3,
Aux4,Timewait,handles,hObject);

% Which results change depends on the input type
```

```
if (Select==1)      % Select==1 equals Ramp Input => Change INL, DNL and MissingCodes
      set(handles.MaxINL,'String',num2str(INLMAX));
      set(handles.MaxDNL,'String',num2str(DNLMAX));
      set(handles.MissingCodes,'String',num2str(misingcode));
else              % Select==0 equals Sine Wave Input => Change SNDR, SNR and ENOB
      set(handles.SNDR,'String',num2str(Sndr));
      set(handles.SNR,'String',num2str(Snr));
      set(handles.ENOB,'String',num2str(Enob));
end


% --- Executes on button press in Esc.
function Esc_Callback(hObject, eventdata, handles)
% hObject    handle to Esc (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% --- Executes during object creation, after setting all properties.

set(handles.Messages_Listbox,'String',char('Simulation has paused'));

pause;


function Start_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Start (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes when editing Start text box
function Start_Callback(hObject, eventdata, handles)
% hObject    handle to Start (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Start as text
%       str2double(get(hObject,'String')) returns contents of Start as a double
global Start;
Start = str2double(get(hObject,'String'))


% --- Executes during object creation, after setting all properties.
function Stop_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Stop (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
```

```
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes when editing Stop text box
function Stop_Callback(hObject, eventdata, handles)
% hObject    handle to Stop (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Stop as text
%        str2double(get(hObject,'String')) returns contents of Stop as a double
global Stop;
Stop = str2double(get(hObject,'String'))


% --- Executes during object creation, after setting all properties.
function Points_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Points (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes when editing Points text box
function Points_Callback(hObject, eventdata, handles)
% hObject    handle to Points (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Points as text
%        str2double(get(hObject,'String')) returns contents of Points as a double
global Points;
Points = str2double(get(hObject,'String'))

% --- Executes during object creation, after setting all properties.
function Tau_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Tau (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes when editing Tau text box
```

```
function Tau_Callback(hObject, eventdata, handles)
% hObject    handle to Tau (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Tau as text
%        str2double(get(hObject,'String')) returns contents of Tau as a double
global Tau;
Tau = str2double(get(hObject,'String'))


% --- Executes during object creation, after setting all properties.
function Start2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Start2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes when editing Start2 text box
function Start2_Callback(hObject, eventdata, handles)
% hObject    handle to Start2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Start2 as text
%        str2double(get(hObject,'String')) returns contents of Start2 as a double
global Start2;
Start2 = str2double(get(hObject,'String'))


% --- Executes during object creation, after setting all properties.
function Stop2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Stop2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes when editing Stop2 text box
function Stop2_Callback(hObject, eventdata, handles)
% hObject    handle to Stop2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of Stop2 as text
%        str2double(get(hObject,'String')) returns contents of Stop2 as a double
global Stop2;
Stop2 = str2double(get(hObject,'String'))


% --- Executes during object creation, after setting all properties.
function Points2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Points2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes when editing Points2 text box
function Points2_Callback(hObject, eventdata, handles)
% hObject    handle to Points2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Points2 as text
%        str2double(get(hObject,'String')) returns contents of Points2 as a double
global Points2;
Points2 = str2double(get(hObject,'String'))


% --- Executes during object creation, after setting all properties.
function Amplitude_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Amplitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes when editing A text box
function Amplitude_Callback(hObject, eventdata, handles)
% hObject    handle to Amplitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Amplitude as text
%        str2double(get(hObject,'String')) returns contents of Amplitude as a double
global A;
A = str2double(get(hObject,'String'))
```

```
% --- Executes during object creation, after setting all properties.
function Frequency_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Frequency (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes when editing Freq text box
function Frequency_Callback(hObject, eventdata, handles)
% hObject    handle to Frequency (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Frequency as text
%        str2double(get(hObject,'String')) returns contents of Frequency as a double
global fsin;
fsin = str2double(get(hObject,'String'))


% --- Executes during object creation, after setting all properties.
function NumberBits_CreateFcn(hObject, eventdata, handles)
% hObject    handle to NumberBits (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes when editing N text box
function NumberBits_Callback(hObject, eventdata, handles)
% hObject    handle to NumberBits (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of NumberBits as text
%        str2double(get(hObject,'String')) returns contents of NumberBits as a double
global N;
N = str2double(get(hObject,'String'))


% --- Executes on button press in SineWaveInput_Radiobutton.
function SineWaveInput_Radiobutton_Callback(hObject, eventdata, handles)
% hObject    handle to SineWaveInput_Radiobutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of SineWaveInput_Radiobutton

% If this button turns on, RampInput will be turned off
off = [handles.RampInput_Radiobutton];
mutual_exclude(off)
global Select;
Select=0;               % Sine Wave Input


% --- Executes on button press in RampInput_Radiobutton.
function RampInput_Radiobutton_Callback(hObject, eventdata, handles)
% hObject    handle to RampInput_Radiobutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of RampInput_Radiobutton

% If this button turns on, SineWaveInput will be turned off
off= [handles.SineWaveInput_Radiobutton];
mutual_exclude(off)
global Select;
Select=1;               % Ramp Input


% --- Function that ensures mutual exclusion between a group of radiobuttons
function mutual_exclude(off)
set(off,'Value',0)


% --- Executes during object creation, after setting all properties.
function Jitter_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Jitter (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes when editing Jitter text box
function Jitter_Callback(hObject, eventdata, handles)
% hObject    handle to Jitter (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Jitter as text
%        str2double(get(hObject,'String')) returns contents of Jitter as a double
global sigmajitter;
sigmajitter = str2double(get(hObject,'String'))


% --- Executes during object creation, after setting all properties.
```

```
function MissingCodes_CreateFcn(hObject, eventdata, handles)
% hObject    handle to MissingCodes (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% Not used (output result)
function MissingCodes_Callback(hObject, eventdata, handles)
% hObject    handle to MissingCodes (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

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


% --- Executes during object creation, after setting all properties.
function Sigma_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Sigma (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes when editing Sigma text box
function Sigma_Callback(hObject, eventdata, handles)
% hObject    handle to Sigma (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Sigma as text
%        str2double(get(hObject,'String')) returns contents of Sigma as a double
global sigma;
global VoffsetVrefpn;
global Voffset0;
sigma = str2double(get(hObject,'String'))
VoffsetVrefpn = sigma * randn(1);
Voffset0 = sigma * randn(1);


% --- Executes on button press in Output_Data_RI_pushbutton.
function Output_Data_RI_pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to Output_Data_RI_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
global vector;
global vectorcode;
global axisxlabelVolt;
global axisylabelOutput;

% If figure does not exist, create it
if ~isfield(handles,'PlotFigure') | ~ishandle(handles.PlotFigure),
 handles.PlotFigure = figure('Name','Output','Visible','on',...
          'NumberTitle','off','HandleVisibility','on','IntegerHandle','off');
 handles.PlotAxes = axes('Parent',handles.PlotFigure);
 guidata(hObject,handles)
end

% Plot the figure, giving name, xlabel, ylabel and grid
plot(vectorcode(:,1),vector(:,2),'Parent',handles.PlotAxes)
set(0,'CurrentFigure',handles.PlotFigure)
set(handles.PlotFigure,'Name','Output')
xlabel(axisxlabelVolt)
ylabel(axisylabelOutput)
grid(handles.PlotAxes,'on')


% --- Executes during object creation, after setting all properties.
function ENOB_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ENOB (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% Not used (output result)
function ENOB_Callback(hObject, eventdata, handles)
% hObject    handle to ENOB (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

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


% --- Executes during object creation, after setting all properties.
function Bandwidth_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Bandwidth (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
```

```
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes when editing BW text box
function Bandwidth_Callback(hObject, eventdata, handles)
% hObject    handle to Bandwidth (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Bandwidth as text
%        str2double(get(hObject,'String')) returns contents of Bandwidth as a double
global bandwidth;
bandwidth = str2double(get(hObject,'String'))


% --- Executes during object creation, after setting all properties.
function Minsignalband_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Minsignalband (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes when editing MinSB text box
function Minsignalband_Callback(hObject, eventdata, handles)
% hObject    handle to Minsignalband (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Minsignalband as text
%        str2double(get(hObject,'String')) returns contents of Minsignalband as a double
global minsignalband;
minsignalband = str2double(get(hObject,'String'))


% --- Executes during object creation, after setting all properties.
function ModelSimulated_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ModelSimulated (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes when editing Model Simulated text box
```

```
function ModelSimulated_Callback(hObject, eventdata, handles)
% hObject    handle to ModelSimulated (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of ModelSimulated as text
%        str2double(get(hObject,'String')) returns contents of ModelSimulated as a double
global modelsimulated;
modelsimulated = get(handles.ModelSimulated,'String')


% --- Executes during object creation, after setting all properties.
function Nsamples_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Nsamples (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes when editing Nsamples text box
function Nsamples_Callback(hObject, eventdata, handles)
% hObject    handle to Nsamples (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Nsamples as text
%        str2double(get(hObject,'String')) returns contents of Nsamples as a double
global nsamples;
nsamples = str2double(get(hObject,'String'))


% --- Executes during object creation, after setting all properties.
function Sweep2_Popupmenu_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Sweep2_Popupmenu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes on selection change in Sweep2_Popupmenu.
function Sweep2_Popupmenu_Callback(hObject, eventdata, handles)
% hObject    handle to Sweep2_Popupmenu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: contents = get(hObject,'String') returns Sweep2_Popupmenu contents as cell array
%        contents{get(hObject,'Value')} returns selected item from Sweep2_Popupmenu
global val2;
global Start2;
global Stop2;
global Points2;
val2 = get(hObject,'Value');

% Automatic values on Sweep 2 parameters depends on the option
% selected on Sweep 2
switch val2
  case 2                % Sweep 2 = Gain
    Start2=40;
    Stop2=90;
    Points2=21;
    set(handles.Start2,'String',Start2);
    set(handles.Stop2,'String',Stop2);
    set(handles.Points2,'String',Points2);
  case 3                % Sweep 2 = Mismatch
    Start2=-0.2;
    Stop2=0.2;
    Points2=41;
    set(handles.Start2,'String',Start2);
    set(handles.Stop2,'String',Stop2);
    set(handles.Points2,'String',Points2);
  case 4                % Sweep 2 = SR
    Start2=14e8;
    Stop2=38e8;
    Points2=25;
    set(handles.Start2,'String',Start2);
    set(handles.Stop2,'String',Stop2);
    set(handles.Points2,'String',Points2);
  case 5                % Sweep 2 = Tau
    Start2=0.01e-8;
    Stop2=2.01e-8;
    Points2=21;
    set(handles.Start2,'String',Start2);
    set(handles.Stop2,'String',Stop2);
    set(handles.Points2,'String',Points2);
  case 6                % Sweep 2 = Sigma
    Start2=1e-3;
    Stop2=1e-2;
    Points2=11;
    set(handles.Start2,'String',Start2);
    set(handles.Stop2,'String',Stop2);
    set(handles.Points2,'String',Points2);
  case 7                % Sweep 2 = Jitter
    Start2=1e-8;
    Stop2=1e-12;
    Points2=41;
    set(handles.Start2,'String',Start2);
    set(handles.Stop2,'String',Stop2);
    set(handles.Points2,'String',Points2);
  case 8                % Sweep 2 = Sampling Freq
    Start2=2e8;
    Stop2=2e6;
    Points2=21;
    set(handles.Start2,'String',Start2);
    set(handles.Stop2,'String',Stop2);
```

```
    set(handles.Points2,'String',Points2);
  case 9                    % Sweep 2 = Amplit
    Start2=0.1;
    Stop2=1;
    Points2=21;
    set(handles.Start2,'String',Start2);
    set(handles.Stop2,'String',Stop2);
    set(handles.Points2,'String',Points2);
  case 10                   % Sweep 2 = FS
    Start2=1;
    Stop2=3;
    Points2=21;
    set(handles.Start2,'String',Start2);
    set(handles.Stop2,'String',Stop2);
    set(handles.Points2,'String',Points2);
  case 11                   % Sweep 2 = Freq
    Start2=0.5e6;
    Stop2=4e6;
    Points2=21;
    set(handles.Start2,'String',Start2);
    set(handles.Stop2,'String',Stop2);
    set(handles.Points2,'String',Points2);
  case 12                   % Sweep 2 = BW
    Start2=5e6;
    Stop2=15e6;
    Points2=21;
    set(handles.Start2,'String',Start2);
    set(handles.Stop2,'String',Stop2);
    set(handles.Points2,'String',Points2);
  case 13                   % Sweep 2 = Aux 1
    Start2=1e-2;
    Stop2=1e-3;
    Points2=21;
    set(handles.Start2,'String',Start2);
    set(handles.Stop2,'String',Stop2);
    set(handles.Points2,'String',Points2);
   case 14                  % Sweep 2 = Aux 2
    Start2=1e-2;
    Stop2=1e-3;
    Points2=21;
    set(handles.Start2,'String',Start2);
    set(handles.Stop2,'String',Stop2);
    set(handles.Points2,'String',Points2);
  case 15                   % Sweep 2 = Aux 3
    Start2=1e-2;
    Stop2=1e-3;
    Points2=21;
    set(handles.Start2,'String',Start2);
    set(handles.Stop2,'String',Stop2);
    set(handles.Points2,'String',Points2);
  case 16                   % Sweep 2 = Aux 4
    Start2=1e-2;
    Stop2=1e-3;
    Points2=21;
    set(handles.Start2,'String',Start2);
    set(handles.Stop2,'String',Stop2);
    set(handles.Points2,'String',Points2);
end
```

```
% --- Executes during object creation, after setting all properties.
function Aux1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Aux1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes when editing Aux 1 text box
function Aux1_Callback(hObject, eventdata, handles)
% hObject    handle to Aux1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Aux1 as text
%        str2double(get(hObject,'String')) returns contents of Aux1 as a double
global Aux1;
Aux1 = str2double(get(hObject,'String'))


% --- Executes during object creation, after setting all properties.
function Aux3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Aux3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes when editing Aux 3 text box
function Aux3_Callback(hObject, eventdata, handles)
% hObject    handle to Aux3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Aux3 as text
%        str2double(get(hObject,'String')) returns contents of Aux3 as a double
global Aux3;
Aux3 = str2double(get(hObject,'String'))


% --- Executes during object creation, after setting all properties.
function Aux2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Aux2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes when editing Aux 2 text box
function Aux2_Callback(hObject, eventdata, handles)
% hObject    handle to Aux2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Aux2 as text
%        str2double(get(hObject,'String')) returns contents of Aux2 as a double
global Aux2;
Aux2 = str2double(get(hObject,'String'))


% --- Executes during object creation, after setting all properties.
function Aux4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Aux4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes when editing Aux 4 text box
function Aux4_Callback(hObject, eventdata, handles)
% hObject    handle to Aux4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Aux4 as text
%        str2double(get(hObject,'String')) returns contents of Aux4 as a double
global Aux4;
Aux4 = str2double(get(hObject,'String'))


% --- Executes on button press in RampInput_pushbutton.
function RampInput_pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to RampInput_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global siginput;
global axisxlabelTime;
global axisylabelInput;
```

```
% If figure does not exist, create it
if ~isfield(handles,'PlotFigure') | ~ishandle(handles.PlotFigure),
 handles.PlotFigure = figure('Name','Input','Visible','on',...
          'NumberTitle','off','HandleVisibility','on','IntegerHandle','off');
 handles.PlotAxes = axes('Parent',handles.PlotFigure);
 guidata(hObject,handles)
end

% Plot the figure, giving name, xlabel, ylabel and grid
plot(siginput(:,1),siginput(:,2),'Parent',handles.PlotAxes)
set(0,'CurrentFigure',handles.PlotFigure)
set(handles.PlotFigure,'Name','Input')
xlabel(axisxlabelTime)
ylabel(axisylabelInput)
grid(handles.PlotAxes,'on')


% --- Executes on button press in SineWaveInput_pushbutton.
function SineWaveInput_pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to SineWaveInput_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global siginput;
global axisxlabelTime;
global axisylabelInput;

% If figure does not exist, create it
if ~isfield(handles,'PlotFigure') | ~ishandle(handles.PlotFigure),
 handles.PlotFigure = figure('Name','Input','Visible','on',...
          'NumberTitle','off','HandleVisibility','on','IntegerHandle','off');
 handles.PlotAxes = axes('Parent',handles.PlotFigure);
 guidata(hObject,handles)
end

% Plot the figure, giving name, xlabel, ylabel and grid
plot(siginput(:,1),siginput(:,2),'Parent',handles.PlotAxes)
set(0,'CurrentFigure',handles.PlotFigure)
set(handles.PlotFigure,'Name','Input')
xlabel(axisxlabelTime)
ylabel(axisylabelInput)
grid(handles.PlotAxes,'on')


% Not used (output result)
function Yes_Monotonic_Radiobutton_Callback(hObject, eventdata, handles)
% hObject    handle to Yes_Monotonic_Radiobutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of Yes_Monotonic_Radiobutton


% Not used (output result)
function No_Monotonic_Radiobutton_Callback(hObject, eventdata, handles)
% hObject    handle to No_Monotonic_Radiobutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of No_Monotonic_Radiobutton
```

```
% --- Executes on button press in HoldOn_pushbutton.
function HoldOn_pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to HoldOn_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% If figure exists, set it current and hold on graphics
if isfield(handles,'PlotFigure') & ishandle(handles.PlotFigure),
   set(0,'CurrentFigure',handles.PlotFigure);
   hold on;
end


% --- Executes on button press in HoldOff_pushbutton.
function HoldOff_pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to HoldOff_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% If figure exists, set it current and hold off graphics
if isfield(handles,'PlotFigure') & ishandle(handles.PlotFigure),
   set(0,'CurrentFigure',handles.PlotFigure);
   hold off;
end


% --- Executes during object creation, after setting all properties.
function Sweep_Popupmenu_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Sweep_Popupmenu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes on selection change in Sweep_Popupmenu.
function Sweep_Popupmenu_Callback(hObject, eventdata, handles)
% hObject    handle to Sweep_Popupmenu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns Sweep_Popupmenu contents as cell array
%        contents{get(hObject,'Value')} returns selected item from Sweep_Popupmenu
global val;
global Start;
global Stop;
global Points;
val = get(hObject,'Value');

% Automatic values on Sweep parameters depends on the option
% selected on Sweep
```

```
switch val
  case 1              % Sweep = Gain
    Start=40;
    Stop=90;
    Points=21;
    set(handles.Start,'String',Start);
    set(handles.Stop,'String',Stop);
    set(handles.Points,'String',Points);
  case 2              % Sweep = Mismatch
    Start=-0.2;
    Stop=0.2;
    Points=41;
    set(handles.Start,'String',Start);
    set(handles.Stop,'String',Stop);
    set(handles.Points,'String',Points);
  case 3              % Sweep = SR
    Start=14e8;
    Stop=38e8;
    Points=25;
    set(handles.Start,'String',Start);
    set(handles.Stop,'String',Stop);
    set(handles.Points,'String',Points);
  case 4              % Sweep = Tau
    Start=0.01e-8;
    Stop=2.01e-8;
    Points=21;
    set(handles.Start,'String',Start);
    set(handles.Stop,'String',Stop);
    set(handles.Points,'String',Points);
  case 5              % Sweep = Sigma
    Start=1e-3;
    Stop=1e-2;
    Points=11;
    set(handles.Start,'String',Start);
    set(handles.Stop,'String',Stop);
    set(handles.Points,'String',Points);
  case 6              % Sweep = Jitter
    Start=1e-8;
    Stop=1e-12;
    Points=41;
    set(handles.Start,'String',Start);
    set(handles.Stop,'String',Stop);
    set(handles.Points,'String',Points);
  case 7              % Sweep = Sampling Freq
    Start=2e8;
    Stop=2e6;
    Points=21;
    set(handles.Start,'String',Start);
    set(handles.Stop,'String',Stop);
    set(handles.Points,'String',Points);
  case 8              % Sweep = Amplit
    Start=0.1;
    Stop=1;
    Points=21;
    set(handles.Start,'String',Start);
    set(handles.Stop,'String',Stop);
    set(handles.Points,'String',Points);
  case 9              % Sweep = FS
    Start=1;
```

```
    Stop=3;
    Points=21;
    set(handles.Start,'String',Start);
    set(handles.Stop,'String',Stop);
    set(handles.Points,'String',Points);
  case 10                % Sweep = Freq
    Start=0.5e6;
    Stop=4e6;
    Points=21;
    set(handles.Start,'String',Start);
    set(handles.Stop,'String',Stop);
    set(handles.Points,'String',Points);
  case 11                % Sweep = BW
    Start=5e6;
    Stop=15e6;
    Points=21;
    set(handles.Start,'String',Start);
    set(handles.Stop,'String',Stop);
    set(handles.Points,'String',Points);
  case 12                % Sweep = Aux 1
    Start=1e-2;
    Stop=1e-3;
    Points=21;
    set(handles.Start,'String',Start);
    set(handles.Stop,'String',Stop);
    set(handles.Points,'String',Points);
  case 13                % Sweep = Aux 2
    Start=1e-2;
    Stop=1e-3;
    Points=21;
    set(handles.Start,'String',Start);
    set(handles.Stop,'String',Stop);
    set(handles.Points,'String',Points);
  case 14                % Sweep = Aux 3
    Start=1e-2;
    Stop=1e-3;
    Points=21;
    set(handles.Start,'String',Start);
    set(handles.Stop,'String',Stop);
    set(handles.Points,'String',Points);
  case 15                % Sweep = Aux 4
    Start=1e-2;
    Stop=1e-3;
    Points=21;
    set(handles.Start,'String',Start);
    set(handles.Stop,'String',Stop);
    set(handles.Points,'String',Points);
end


% Not used----------------------------------------------------------
function file_menu_Callback(hObject, eventdata, handles)
% hObject    handle to file_menu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% Not used----------------------------------------------------------
function menu_file_new_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to menu_file_new (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes on selecting menu File->New->Model
function file_new_model_Callback(hObject, eventdata, handles)
% hObject    handle to file_new_model (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global modelsimulated;

% If is not opened, open file "current.mdl"
if  isempty(find_system('Name','current')),
   open_system('current')
   figure(handles.Current)
end

% Write on Model Simulated text box the complete route of "current.mdl"
modelsimulated = [];
modelsimulated = strcat(pwd,'\current.mdl');

set(handles.ModelSimulated,'String',modelsimulated);


% --- Executes on selecting menu File->Open->Model
function file_open_model_Callback(hObject, eventdata, handles)
% hObject    handle to file_open_model (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global Ts;                      % Global variables to be initialized
global FS;
global A;
global fsin;
global N;
global sigma;
global sigmajitter;
global bandwidth;
global minsignalband;
global val;
global val2;
global Select;
global modelsimulated;
global VoffsetVrefpn;
global Voffset0;
global nsamples;
global Effort;
global Eps;
global SR;
global Tau;
global Aux1;
global Aux2;
global Aux3;
global Aux4;
global Ao;
global AodB;
global Auto;
global load_ex;
```

```
global load_example;

load_example = 0;
load_example = load_ex;
if (load_ex == 1)
   load_ex = 0;
   Ts = handles.Ts;
   FS = handles.FS;
   N = handles.N;
   bandwidth = handles.bandwidth;
   A = handles.A;
   fsin = handles.fsin;
   nsamples = handles.nsamples;
   minsignalband = handles.minsignalband;
   AodB = handles.AodB;
   sigmajitter = handles.sigmajitter;
   SR = handles.SR;
   Tau = handles.Tau;
   Eps = handles.Eps;
   sigma = handles.sigma;
   Aux1 = handles.Aux1;
   Aux2 = handles.Aux2;
   Aux3 = handles.Aux3;
   Aux4 = handles.Aux4;
   modelsimulated = handles.model;

   handles = load('Struct_gui');
   guidata(hObject,handles);

   set(handles.ModelSimulated,'String',modelsimulated);
   set(handles.SamplingFrequency,'String',1/Ts);
   set(handles.FullScale,'String',FS);
   set(handles.NumberBits,'String',N);
   set(handles.Bandwidth,'String',bandwidth);
   set(handles.Amplitude,'String',A);
   set(handles.Frequency,'String',fsin);
   set(handles.Nsamples,'String',nsamples);
   set(handles.Minsignalband,'String',minsignalband);
   set(handles.Gain,'String',AodB);
   set(handles.Jitter,'String',sigmajitter);
   set(handles.SR,'String',SR);
   set(handles.Tau,'String',Tau);
   set(handles.Mismatch,'String',Eps);
   set(handles.Sigma,'String',sigma);
   set(handles.Aux1,'String',Aux1);
   set(handles.Aux2,'String',Aux2);
   set(handles.Aux3,'String',Aux3);
   set(handles.Aux4,'String',Aux4);
   close('library')
else
   % Open a dialog box for selecting the file to be opened
   [filename, pathname] = uigetfile( ...
      {'*.mdl', 'All Model and Library Files (*.mdl)'; ...
      '*.*','All Files (*.*)'}, ...
      'Select Model');

   % If "Cancel" is selected then return
   if isequal([filename,pathname],[0,0])
      return
```

```
   % Otherwise construct the fullfilename and Check and load the file.
   else
      File = fullfile(pathname,filename);

      handles.LastFIle = File;
      guidata(hObject,handles)
      open_system(File);

   % Write on Model Simulated text box the complete route of the file
      modelsimulated = [];
      modelsimulated = File;
      set(handles.ModelSimulated,'String',modelsimulated);
      set(handles.Gain,'String',AodB);
      set(handles.SamplingFrequency,'String',1/Ts);

   end
end

% --- Executes on selecting menu File->Open->Library
function file_open_library_Callback(hObject, eventdata, handles)
% hObject    handle to file_open_library (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global handles_gui;
handles_gui = handles;
save Struct_gui  -struct handles_gui;
clear;
library;

% Add Library directory to the path
%diract=pwd;
%cd Library;
%dirlib=pwd;
%path(path,dirlib)

% Open a dialog box for selecting the file to be opened,
% starting in Library directory
%[filename, pathname] = uigetfile( ...
%    {'*.mdl', 'All Model and Library Files (*.mdl)'; ...
%       '*.*','All Files (*.*)'}, ...
%    'Select Library');
%cd ..;
% If "Cancel" is selected then return
%if isequal([filename,pathname],[0,0])
%    return
% Otherwise construct the fullfilename and Check and load the file.
%else
%    File = fullfile(pathname,filename);

%    handles.LastFIle = File;
%    guidata(hObject,handles)
%    open_system(File);
%end


% Not used-----------------------------------------------------------
function environment_menu_Callback(hObject, eventdata, handles)
% hObject    handle to environment_menu (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes on selecting menu Environment->Load
function menu_environment_load_Callback(hObject, eventdata, handles)
% hObject    handle to menu_environment_load (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global modelsimulated;
global Ts;
global FS;
global N;
global bandwidth;
global minsignalband;
global fsin;
global A;
global nsamples;
global AodB;
global Mismatch;
global SR;
global Tau;
global sigma;
global sigmajitter;
global Aux1;
global Aux2;
global Aux3;
global Aux4;
global val;
global val2;
global Start;
global Stop;
global Points;
global Start2;
global Stop2;
global Points2;

diract=pwd;
cd States;

% Open a dialog box for selecting the file to be run,
% starting in States directory
[filename, pathname] = uigetfile( ...
   {'*.m','M-Files (*.m)'}, ...
   'Select Environment File');
cd ..;

% If "Cancel" is selected then return
if isequal([filename,pathname],[0,0])
   return
% Otherwise construct the fullfilename and Check and run the file.
else
   File = fullfile(pathname,filename);

   handles.LastFIle = File;
   guidata(hObject,handles)
   run(File);
end
```

```
% --- Executes on selecting menu Environment->Save
function menu_environment_save_Callback(hObject, eventdata, handles)
% hObject    handle to menu_environment_save (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

diract=pwd;
cd States;

% Open a dialog box for selecting the name of the file in which
% the actual state will be saved, starting in States directory
[filename, pathname] = uiputfile( ...
   {'*.m','M-Files (*.m)'}, ...
   'Save as');
cd ..;
% If "Cancel" is selected then return
if isequal([filename,pathname],[0,0])
   return
% Otherwise construct the fullfilename and Check and save the file.
else
   File = fullfile(pathname,filename);

   handles.LastFIle = File;
   guidata(hObject,handles)

% Open file with write permission
   fid=fopen(File,'w');

% Write all the code for the m-file that saves the code which
% will be run when "load" option would be selected, and so
% the actual state would be loaded
   fprintf(fid,'Ts=1/');
   fprintf(fid,get((handles.SamplingFrequency),'String'));
   fprintf(fid,';\n');
   fprintf(fid,'set(handles.SamplingFrequency,''String'',1/Ts)\n');
   fprintf(fid,'FS=');
   fprintf(fid,get((handles.FullScale),'String'));
   fprintf(fid,';\n');
   fprintf(fid,'set(handles.FullScale,''String'',FS)\n');
   fprintf(fid,'N=');
   fprintf(fid,get((handles.NumberBits),'String'));
   fprintf(fid,';\n');
   fprintf(fid,'set(handles.NumberBits,''String'',N)\n');
   fprintf(fid,'bandwidth=');
   fprintf(fid,get((handles.Bandwidth),'String'));
   fprintf(fid,';\n');
   fprintf(fid,'set(handles.Bandwidth,''String'',bandwidth)\n');
   fprintf(fid,'A=');
   fprintf(fid,get((handles.Amplitude),'String'));
   fprintf(fid,';\n');
   fprintf(fid,'set(handles.Amplitude,''String'',A)\n');
   fprintf(fid,'fsin=');
   fprintf(fid,get((handles.Frequency),'String'));
   fprintf(fid,';\n');
   fprintf(fid,'set(handles.Frequency,''String'',fsin)\n');
   fprintf(fid,'nsamples=');
   fprintf(fid,get((handles.Nsamples),'String'));
   fprintf(fid,';\n');
```

```
fprintf(fid,'set(handles.Nsamples,''String'',nsamples)\n');
fprintf(fid,'minsignalband=');
fprintf(fid,get((handles.Minsignalband),'String'));
fprintf(fid,';\n');
fprintf(fid,'set(handles.Minsignalband,''String'',minsignalband)\n');
fprintf(fid,'AodB=');
fprintf(fid,get((handles.Gain),'String'));
fprintf(fid,';\n');
fprintf(fid,'set(handles.Gain,''String'',AodB)\n');
fprintf(fid,'Eps=');
fprintf(fid,get((handles.Mismatch),'String'));
fprintf(fid,';\n');
fprintf(fid,'set(handles.Mismatch,''String'',Eps)\n');
fprintf(fid,'SR=');
fprintf(fid,get((handles.SR),'String'));
fprintf(fid,';\n');
fprintf(fid,'set(handles.SR,''String'',SR)\n');
fprintf(fid,'Tau=');
fprintf(fid,get((handles.Tau),'String'));
fprintf(fid,';\n');
fprintf(fid,'set(handles.Tau,''String'',Tau)\n');
fprintf(fid,'sigmajitter=');
fprintf(fid,get((handles.Jitter),'String'));
fprintf(fid,';\n');
fprintf(fid,'set(handles.Jitter,''String'',sigmajitter)\n');
fprintf(fid,'sigma=');
fprintf(fid,get((handles.Sigma),'String'));
fprintf(fid,';\n');
fprintf(fid,'set(handles.Sigma,''String'',sigma)\n');
fprintf(fid,'Aux1=');
fprintf(fid,get((handles.Aux1),'String'));
fprintf(fid,';\n');
fprintf(fid,'set(handles.Aux1,''String'',Aux1)\n');
fprintf(fid,'Aux2=');
fprintf(fid,get((handles.Aux2),'String'));
fprintf(fid,';\n');
fprintf(fid,'set(handles.Aux2,''String'',Aux2)\n');
fprintf(fid,'Aux3=');
fprintf(fid,get((handles.Aux3),'String'));
fprintf(fid,';\n');
fprintf(fid,'set(handles.Aux3,''String'',Aux3)\n');
fprintf(fid,'Aux4=');
fprintf(fid,get((handles.Aux4),'String'));
fprintf(fid,';\n');
fprintf(fid,'set(handles.Aux4,''String'',Aux4)\n');
fprintf(fid,'val=');
fprintf(fid,'%d',get((handles.Sweep_Popupmenu),'Value'));
fprintf(fid,';\n');
fprintf(fid,'set(handles.Sweep_Popupmenu,''Value'',val)\n');
fprintf(fid,'val2=');
fprintf(fid,'%d',get((handles.Sweep2_Popupmenu),'Value'));
fprintf(fid,';\n');
fprintf(fid,'set(handles.Sweep2_Popupmenu,''Value'',val2)\n');
fprintf(fid,'Start=');
fprintf(fid,get((handles.Start),'String'));
fprintf(fid,';\n');
fprintf(fid,'set(handles.Start,''String'',Start)\n');
fprintf(fid,'Stop=');
fprintf(fid,get((handles.Stop),'String'));
```

```
   fprintf(fid,';\n');
   fprintf(fid,'set(handles.Stop,''String'',Stop)\n');
   fprintf(fid,'Points=');
   fprintf(fid,get((handles.Points),'String'));
   fprintf(fid,';\n');
   fprintf(fid,'set(handles.Points,''String'',Points)\n');
   fprintf(fid,'Start2=');
   fprintf(fid,get((handles.Start2),'String'));
   fprintf(fid,';\n');
   fprintf(fid,'set(handles.Start2,''String'',Start2)\n');
   fprintf(fid,'Stop2=');
   fprintf(fid,get((handles.Stop2),'String'));
   fprintf(fid,';\n');
   fprintf(fid,'set(handles.Stop2,''String'',Stop2)\n');
   fprintf(fid,'Points2=');
   fprintf(fid,get((handles.Points2),'String'));
   fprintf(fid,';\n');
   fprintf(fid,'set(handles.Points2,''String'',Points2)\n');

% Close file
   fclose(fid);
end


% Not used-----------------------------------------------------------
function help_menu_Callback(hObject, eventdata, handles)
% hObject    handle to help_menu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes on selecting menu Help->User Guide
function menu_help_userguide_Callback(hObject, eventdata, handles)
% hObject    handle to menu_help_userguide (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Open User Guide (pdf format)
open('ManualdeUsuario.pdf')


% ------- Executes on selecting menu Help->User Guide
function menu_help_userguide2_Callback(hObject, eventdata, handles)
% hObject    handle to menu_help_userguide2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Open User Guide Advance(pdf format)
open('ManualdeUsuarioAvanzado.pdf')


% --- Executes on selecting menu Environment->Save----------------
function menu_help_about_Callback(hObject, eventdata, handles)
% hObject    handle to menu_help_about (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Create the character array for the msgbox
aboutmsg={'SimConverter ver 2.0','','Authors:','','Daniel Falcon Medina','Isabel Vacas
```

```
Páez','Fernando Munoz Chavero'};

% Display the msgbox, which will be modal
msgbox(aboutmsg,'About','modal')


% --- Executes during object creation, after setting all properties.
function Effort_Slider_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Effort_Slider (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background, change
%       'usewhitebg' to 0 to use default.  See ISPC and COMPUTER.
usewhitebg = 1;
if usewhitebg
   set(hObject,'BackgroundColor',[.9 .9 .9]);
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes on slider movement.
function Effort_Slider_Callback(hObject, eventdata, handles)
% hObject    handle to Effort_Slider (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range of slider
global Effort;
Effort=get(hObject,'Value')


% --- Executes on button press in Yes_Auto_Radiobutton.
function Yes_Auto_Radiobutton_Callback(hObject, eventdata, handles)
% hObject    handle to Yes_Auto_Radiobutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of Yes_Auto_Radiobutton

% If this button turns on, No_Auto will be turned off
off= [handles.No_Auto_Radiobutton];
mutual_exclude(off)
global Auto;
Auto=1;                 % Yes_Auto


% --- Executes on button press in No_Auto_Radiobutton.
function No_Auto_Radiobutton_Callback(hObject, eventdata, handles)
% hObject    handle to No_Auto_Radiobutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of No_Auto_Radiobutton

% If this button turns on, Yes_Auto will be turned off
off= [handles.Yes_Auto_Radiobutton];
```

```
mutual_exclude(off)
global Auto;
Auto=0;                         % No_Auto

% Not used----------------------------------------------------------
function menu_file_open_Callback(hObject, eventdata, handles)
% hObject    handle to menu_file_open (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


function edit32_Callback(hObject, eventdata, handles)
% hObject    handle to edit32 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit32 as text
%        str2double(get(hObject,'String')) returns contents of edit32 as a double


% --- Executes during object creation, after setting all properties.
function edit32_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit32 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


function Timewait_Callback(hObject, eventdata, handles)
% hObject    handle to Timewait (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

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

global wait;
wait = str2double(get(hObject,'String'))

% --- Executes during object creation, after setting all properties.
function Timewait_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Timewait (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% 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
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```