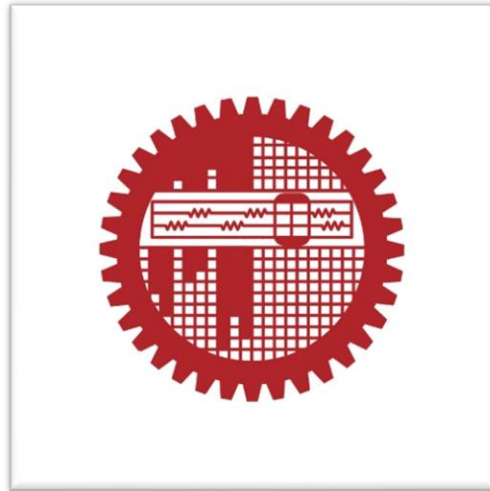


BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY



Department of Electrical and Electronic Engineering

Course No.: EEE 212

Course Title: Numerical Technique Laboratory

“Signal Representation Simulation: Rectification, Clipping, & Clamping”

Prepared By:

Farhan Hamid (S201706175)

Subah Karnine(S201706174)

Level:02 Term:01

Group No:06

Section: C2

Objectives:

Throughout the entire project, the following goals were set as targets to meet using the Matlab software:

1. Derivation of the output signal of a half wave rectifier for a given input sinusoidal signal under ideal conditions.
2. Derivation of the output signal of a full wave rectifier for a given input sinusoidal signal under ideal conditions.
3. Showing the clipped waveform of a given sinusoidal signal and a certain clipping constant.
4. Showing the clamped waveform of a given sinusoidal signal and a certain clamping constant.
5. Peak to peak ripple voltage for given capacitor and load resistance value.

Theory:

Half Wave & Full Wave Rectifier:

For the input sinusoidal signal amplitude, frequency and phase angle have to be known. The input voltage here is,

$$V_{in} = A \sin(2\pi f + \theta)$$

The half wave rectifier cancels out any negative values of V_{in} , leaving only the positive sides. The full wave rectifier on the other hand turns the negative portion into positive, turning the whole waveform unidirectional.

Clipper and Clamper:

A clipper circuit takes a constant value and clips the portion of the waveform which is greater than it. A clamper takes a dc value and adds it to the given signal.

Peak to Peak Ripple Voltage:

Here a capacitor is added parallel to the load resistance. For Half wave rectifier the peak to peak ripple voltage is given by,

$$V_r = V_p / (RCf)$$

For full wave rectifier the time period is halved. The peak to peak ripple voltage is given by,

$$V_r = V_p / (2RCf)$$

Instruction:

- 1) At the beginning a prompt asks the user what he/she wishes to do. Selecting the desired action will take the user to the relevant GUI.
- 2) **Half Wave-Full Wave Rectifier:** The user must put the necessary information e.g. amplitude, frequency, phase of the source voltage. A list box is there to choose either half wave rectification or full wave rectification. Clicking on output will give the user the desired graph and results.

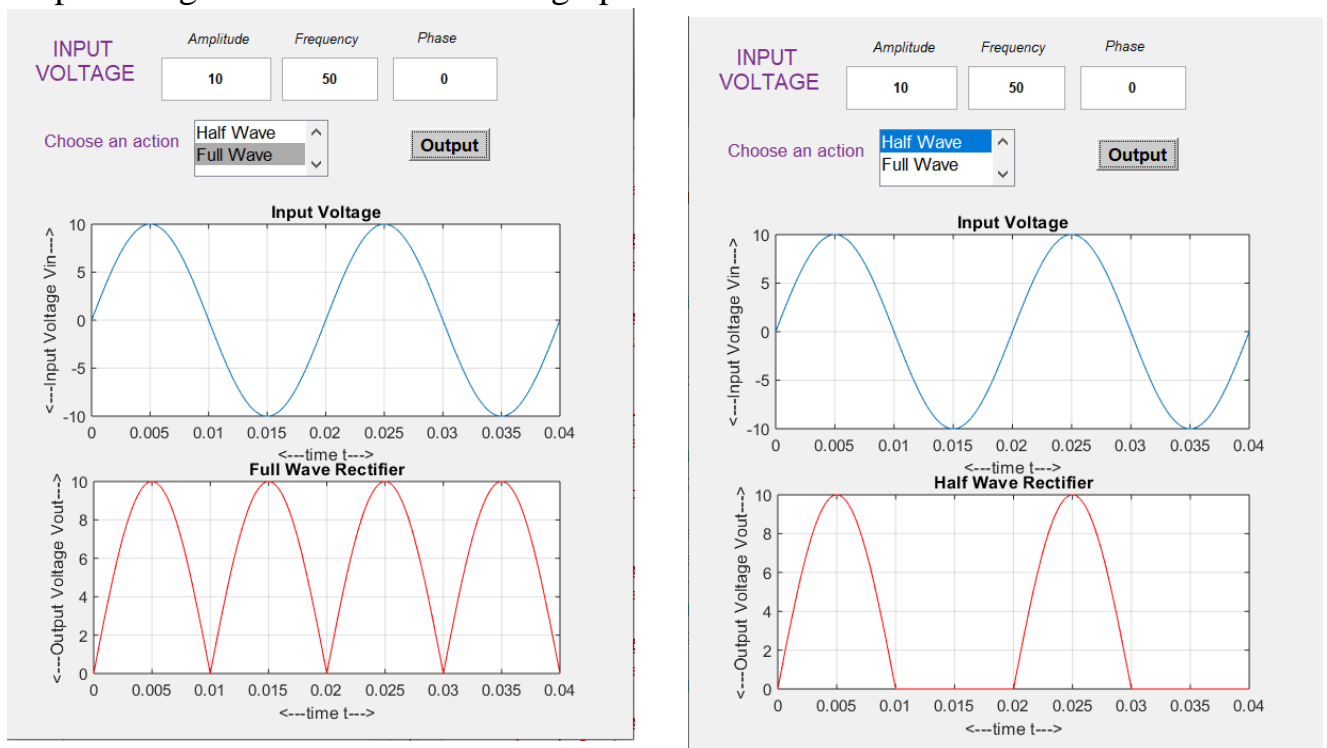


Figure: GUI of Half Wave-Full wave rectifier

- 3) **Clipper-Clamper Circuit:** The user must give the necessary input voltage parameters. A clipping constant is required for clipping and a clamping constant is required for clamping. The corresponding graphs will show the desired output. **In addition**, the output vs input voltage graph for the given clipper and clamper is also shown in the side.

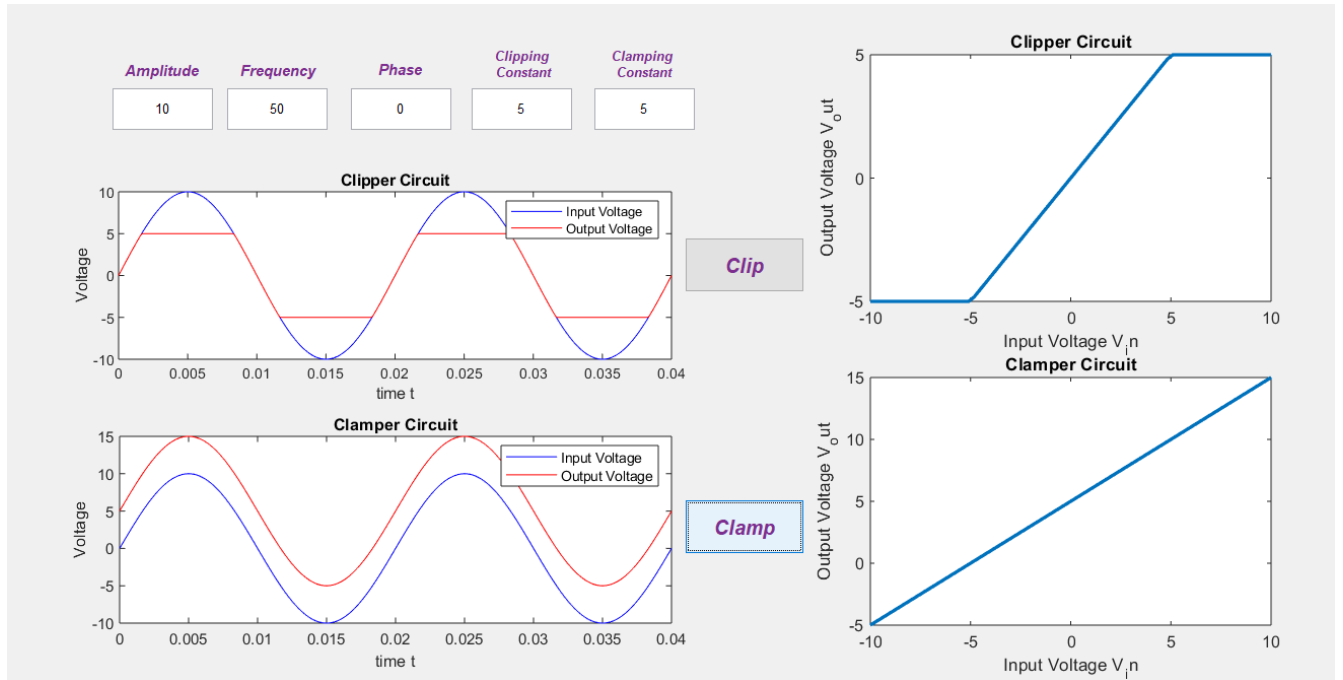


Figure: GUI of Clipper-Clamper Circuit

- 4) **Ripple:** The user must give the input voltage parameters and the value of the capacitor and load voltage connected in parallel. The output corresponding to half wave and full wave rectification will show in the output box.

The GUI for the Peak to peak ripple voltage calculation includes the following components:

- Input Parameters:**
 - Amplitude: 10
 - Frequency: 50
 - Phase: 0
 - Capacitor: 2
 - Load: 50
- Rectification Type:** A "Choose" dropdown menu with options "Half Wave" and "Full Wave".
- Output:** A button labeled "Output".
- Results:**
 - P-P Ripple Voltage: 0.002

Figure: GUI of Peak to peak ripple voltage

Matlab Codes:

Main Code

```
close all
clear all
clc

choose= menu('    What do you wish to do?    ', ' Wave
Rectifier ', ' Clipper-Clamper ', 'Peak to peak Ripple
Voltage');
switch choose
    case 1
        half_full
    case 2
        clipNclamp2
    case 3
        ripple
end
```

Half Wave Full Wave Rectifier

```
function varargout = half_full(varargin)
% HALF_FULL MATLAB code for half_full.fig
%     HALF_FULL, by itself, creates a new HALF_FULL or
raises the existing
%     singleton*.
%
%     H = HALF_FULL returns the handle to a new HALF_FULL
or the handle to
%     the existing singleton*.
%
%     HALF_FULL('CALLBACK',hObject,eventData,handles,...)
calls the local
%     function named CALLBACK in HALF_FULL.M with the
given input arguments.
%
%     HALF_FULL('Property','Value',...) creates a new
HALF_FULL or raises the
%     existing singleton*. Starting from the left,
property value pairs are
%     applied to the GUI before half_full_OpeningFcn gets
called. An
```

```

% unrecognized property name or invalid value makes
property application
% stop. All inputs are passed to half_full_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
half_full

% Last Modified by GUIDE v2.5 07-Sep-2019 05:35:52

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @half_full_OpeningFcn,
                  ...
                  'gui_OutputFcn',    @half_full_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

% --- Executes just before half_full is made visible.
function half_full_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 half_full (see
VARARGIN)

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

% Update handles structure
guidata(hObject, handles);
end

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

% --- Outputs from this function are returned to the
command line.
function varargout = half_full_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;
end

function amp_Callback(hObject, eventdata, handles)
% hObject     handle to amp (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 amp as
text
%         str2double(get(hObject,'String')) returns contents
of amp as a double
end
```

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

```
function amp_CreateFcn(hObject, eventdata, handles)
% hObject    handle to amp (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
end
```

```
function phase_Callback(hObject, eventdata, handles)
% hObject    handle to phase (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 phase as
text
%         str2double(get(hObject,'String')) returns contents
of phase as a double
end
```

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

```
function phase_CreateFcn(hObject, eventdata, handles)
% hObject    handle to phase (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
end

function freq_Callback(hObject, eventdata, handles)
% hObject handle to freq (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 freq as
text
% str2double(get(hObject,'String')) returns contents
of freq as a double
end

% --- Executes during object creation, after setting all
properties.
function freq_CreateFcn(hObject, eventdata, handles)
% hObject handle to freq (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
end

```

```
end
```

```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton1 (see GCBO)
% eventdata    reserved - to be defined in a future version
of MATLAB
% handles      structure with handles and user data (see
GUIDATA)
amp=str2num(get(handles.amp,'String'));
freq=str2num(get(handles.freq,'String'));
ph=str2num(get(handles.phase,'String'));
t=0:0.0001:2*(1/freq);
V_in=amp*sin(2*pi*freq*t+ph);
axes(handles.axes1);
plot(t,V_in)
grid on;
xlabel('<---time t--->');
ylabel('<---Input Voltage Vin--->');
title('Input Voltage');

a=get(handles.listbox1,'Value');
switch a
    case 1
        for i=1:length(V_in)
            if V_in(i)>0
                V_out(i)=V_in(i);
            else
                V_out(i)=0;
            end
        end
        axes(handles.axes2);
        plot(t,V_out,'r')
        grid on
        xlabel('<---time t--->');
        ylabel('<---Output Voltage Vout--->');
        title('Half Wave Rectifier');

    case 2
        V_out=abs(V_in);
        axes(handles.axes2);
        plot(t,V_out,'r')
```

```

        grid on
        xlabel('<---time t--->');
        ylabel('<---Output Voltage Vout--->');
        title('Full Wave Rectifier');
end
end

% --- Executes on selection change in listbox1.
function listbox1_Callback(hObject, eventdata, handles)
% hObject      handle to listbox1 (see GCBO)
% eventdata    reserved - to be defined in a future version
of MATLAB
% handles      structure with handles and user data (see
GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
listbox1 contents as cell array
%           contents{get(hObject,'Value')} returns selected
item from listbox1
end

% --- Executes during object creation, after setting all
properties.
function listbox1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to listbox1 (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
end

```



```

        'gui_OpeningFcn',
@clipNclamp2_OpeningFcn, ...
        'gui_OutputFcn',
@clipNclamp2_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


% --- Executes just before clipNclamp2 is made visible.
function clipNclamp2_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 clipNclamp2 (see
VARARGIN)

% Choose default command line output for clipNclamp2
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes clipNclamp2 wait for user response (see
UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the
command line.

```

```

function varargout = clipNclamp2_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;


function amp_Callback(hObject, eventdata, handles)
% hObject      handle to amp (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 amp as
text
%           str2double(get(hObject,'String')) returns contents
of amp as a double


% --- Executes during object creation, after setting all
properties.
function amp_CreateFcn(hObject, eventdata, handles)
% hObject      handle to amp (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function freq_Callback(hObject, eventdata, handles)
% hObject      handle to freq (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 freq as
text
%           str2double(get(hObject,'String')) returns contents
of freq as a double

% --- Executes during object creation, after setting all
properties.
function freq_CreateFcn(hObject, eventdata, handles)
% hObject      handle to freq (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function ph_Callback(hObject, eventdata, handles)
% hObject      handle to ph (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 ph as
text

```

```
%          str2double(get(hObject,'String')) returns contents
of ph as a double
```

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

```
function ph_CreateFcn(hObject, eventdata, handles)
% hObject      handle to ph (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function clip_Callback(hObject, eventdata, handles)
% hObject      handle to clip (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 clip as
text
```

```
%          str2double(get(hObject,'String')) returns contents
of clip as a double
```

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

```
function clip_CreateFcn(hObject, eventdata, handles)
% hObject      handle to clip (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function clamp_Callback(hObject, eventdata, handles)
% hObject      handle to clamp (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 clamp as
text
```

```
%      str2double(get(hObject,'String')) returns contents
of clamp as a double
```

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

```
function clamp_CreateFcn(hObject, eventdata, handles)
% hObject      handle to clamp (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on button press in pushbutton1.
```

```
function pushbutton1_Callback(hObject, eventdata, handles)
```

```

% hObject      handle to pushbutton1 (see GCBO)
% eventdata    reserved - to be defined in a future version
of MATLAB
% handles      structure with handles and user data (see
GUIDATA)
amp=str2num(get(handles.amp,'String'));
freq=str2num(get(handles.freq,'String'));
ph=str2num(get(handles.ph,'String'));
t=0:0.0001:2*(1/freq);
V_in=amp*sin(2*pi*freq*t+ph);
clip=str2num(get(handles.clip,'String'));
clip=abs(clip);
    for i=1:length(V_in)
        if abs(amp)>abs(clip)
            if V_in(i)>0
                if V_in(i)<clip
                    V_out(i)=V_in(i);
                else
                    V_out(i)=clip;
                end
            else if V_in(i)<0
                if V_in(i)>(-clip)
                    V_out(i)=V_in(i);
                else
                    V_out(i)=-clip;
                end
            end
        end
    end
    else
        V_out(i)=V_in(i);
    end
end

axes(handles.axes1);
plot(t,V_in,'b')
hold on
plot(t,V_out,'r')
xlabel('time t')
ylabel('Voltage')
title('Clipper Circuit')
legend('Input Voltage','Output Voltage')
axes(handles.axes3);
plot(V_in,V_out,'linewidth',2)

```

```

xlabel('Input Voltage V_in')
ylabel('Output Voltage V_out')
title('Clipper Circuit')

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton2 (see GCBO)
% eventdata    reserved - to be defined in a future version
of MATLAB
% handles      structure with handles and user data (see
GUIDATA)
amp=str2num(get(handles.amp, 'String'));
freq=str2num(get(handles.freq, 'String'));
ph=str2num(get(handles.ph, 'String'));
t=0:0.0001:2*(1/freq);
V_in=amp*sin(2*pi*freq*t+ph);
clamp=str2num(get(handles.clamp, 'String'));
V_out=clamp+V_in;
axes(handles.axes2);
plot(t,V_in, 'b')
hold on
plot(t,V_out, 'r')
xlabel('time t')
ylabel('Voltage')
title('Clamper Circuit')
legend('Input Voltage', 'Output Voltage')
axes(handles.axes4);
plot(V_in,V_out, 'linewidth', 2)
xlabel('Input Voltage V_in')
ylabel('Output Voltage V_out')
title('Clamper Circuit')

```

Ripple Voltage

```

function varargout = ripple(varargin)
% RIPPLE MATLAB code for ripple.fig
%     RIPPLE, by itself, creates a new RIPPLE or raises
the existing
%     singleton*.
%
%     H = RIPPLE returns the handle to a new RIPPLE or the
handle to

```

```

%         the existing singleton*.
%
%         RIPPLE('CALLBACK', hObject, eventData, handles, ...)
calls the local
%         function named CALLBACK in RIPPLE.M with the given
input arguments.
%
%         RIPPLE('Property','Value',...) creates a new RIPPLE
or raises the
%         existing singleton*. Starting from the left,
property value pairs are
%         applied to the GUI before ripple_OpeningFcn gets
called. An
%         unrecognized property name or invalid value makes
property application
%         stop. All inputs are passed to ripple_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 ripple

% Last Modified by GUIDE v2.5 08-Sep-2019 01:28:58

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @ripple_OpeningFcn,
                  ...
                  'gui_OutputFcn',   @ripple_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

% --- Executes just before ripple is made visible.
function ripple_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 ripple (see
VARARGIN)

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

% Update handles structure
guidata(hObject, handles);

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

% --- Outputs from this function are returned to the
command line.
function varargout = ripple_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;

```

```

function amp_Callback(hObject, eventdata, handles)
% hObject      handle to amp (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 amp as
text
%           str2double(get(hObject,'String')) returns contents
of amp as a double

% --- Executes during object creation, after setting all
properties.
function amp_CreateFcn(hObject, eventdata, handles)
% hObject      handle to amp (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function freq_Callback(hObject, eventdata, handles)
% hObject      handle to freq (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 freq as
text
%           str2double(get(hObject,'String')) returns contents
of freq as a double

```

```
% --- Executes during object creation, after setting all
properties.
function freq_CreateFcn(hObject, eventdata, handles)
% hObject    handle to freq (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function ph_Callback(hObject, eventdata, handles)
% hObject    handle to ph (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 ph as
text
%         str2double(get(hObject,'String')) returns contents
of ph as a double
```

```
% --- Executes during object creation, after setting all
properties.
function ph_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ph (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton1 (see GCBO)
% eventdata    reserved - to be defined in a future version
of MATLAB
% handles      structure with handles and user data (see
GUIDATA)
V_p=str2num(get(handles.amp,'String'));
f=str2num(get(handles.freq,'String'));
C=abs(str2num(get(handles.cap,'String')));
R=abs(str2num(get(handles.R,'String')));
V_p=abs(V_p);
a=get(handles.listbox1,'Value');
switch a
    case 1
        V_r=double(V_p/(C*f*R));
    case 2
        V_r=double(V_p/(2*C*f*R));
end
set(handles.Vr,'String',V_r);

```

```

function Vr_Callback(hObject, eventdata, handles)
% hObject      handle to Vr (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 Vr as
text

```



```
%          str2double(get(hObject,'String')) returns contents
of Vr as a double
```

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

```
function Vr_CreateFcn(hObject, eventdata, handles)
% hObject      handle to Vr (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function cap_Callback(hObject, eventdata, handles)
% hObject      handle to cap (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 cap as
text
```

```
%          str2double(get(hObject,'String')) returns contents
of cap as a double
```

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

```
function cap_CreateFcn(hObject, eventdata, handles)
% hObject      handle to cap (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in listbox1.
function listbox1_Callback(hObject, eventdata, handles)
% hObject      handle to listbox1 (see GCBO)
% eventdata    reserved - to be defined in a future version
of MATLAB
% handles      structure with handles and user data (see
GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
listbox1 contents as cell array
%       contents{get(hObject,'Value')} returns selected
item from listbox1

% --- Executes during object creation, after setting all
properties.
function listbox1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to listbox1 (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function R_Callback(hObject, eventdata, handles)

```

```

% hObject      handle to R (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 R as
text
%           str2double(get(hObject,'String')) returns contents
of R as a double

% --- Executes during object creation, after setting all
properties.
function R_CreateFcn(hObject, eventdata, handles)
% hObject      handle to R (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 && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```