Technical University of Cluj-Napoca
Faculty of Electronics, Telecommunications and Information Technology

Circuits with Operational Amplifiers

Written by: Zaicanu Raluca-Maria

Group: e_2021

Year: 2023

Year of study: II

Teachers: Mihaela Cirlugea and Paul Farago

Table of Contents

Introduction

MATLAB, which stands for Matrix Laboratory, is a programming language and environment that was first developed by MathWorks in the 1970s. It quickly became a popular tool among engineers and scientists for its ability to perform complex mathematical calculations and data analysis.

One of the key features of MATLAB is its focus on matrices and arrays as the primary data structure. This makes it well-suited for tasks such as linear algebra, signal processing, and image analysis. Additionally, MATLAB includes a wide range of built-in functions for mathematical operations, as well as a large library of toolboxes that provide specialized functionality for specific fields.

In addition to its mathematical capabilities, MATLAB also has a robust set of visualization tools. This makes it easy to create plots and graphics that help to communicate data and results.

For this project to be able to highlight most of the features of MATLAB, the theme has been generally chosen: “Circuits with Operational Amplifiers”. An operational amplifier (Op-amp) is a type of electronic amplifier that is designed to amplify weak signals in a circuit. It is a DC-coupled, high-gain voltage amplifier that can be used to amplify a small input voltage to a larger output voltage. Op-amps are widely used in electronic circuits for a variety of applications such as amplifiers, active filters, oscillators, and analog-to-digital converters. They are also widely used in control systems, signal processing and other applications where a high gain and a high input impedance are required. Op-amps are typically packaged in a small integrated circuit (IC) package and can be powered by a single voltage supply.

Theoretical presentation

In this project, there are presented three types of simple circuits: Simple comparator, Hysteresis Comparator and Amplifier with OpAmp, that showcase the use of the Operational Amplifier, with the help of plots and pictures and also some simple problems.

For all the circuits named above, there are voltage plots (Vout as a function of Vin). The operational amplifier has two DC voltage supplies(usually +VPS and -VPS), which give the high and the low values of Vout. The AC voltage supplies (V^+ and V^-) determine if Vout should be low or high according to:

- If $V^+-V^- > 0$, Vout=VoHigh
- If $V^+-V^- < 0$, Vout=VoLow

Therefore, the output signal is meant to always be a square signal. These exact caracteristics are found in the simple comparator circuit. For simpler models V^+ is Vin (or VI) and V^- is connected to ground (in case of a non inverting circuit). As for the other two circuits, feedback is added.

The hysteresis comparator has a positive feedback , which means it has a connection between V^+ and Vout (or Vo) and two series-connected resistors(R1 between VI and V^+ and R2 between V^+ and Vo). Because of this, the threshold voltage appears(Vth), which is usually calculated as such for the non-inverting circuit:

- Vth=-(R1/R2)Vo

When VI reaches the value +Vth, Vo will become VoHigh and when it reaches -Vth Vo will be VoLow, creating the same square signal as the simple comparator for Vo, but shifted to the right. In case of an inverting hysteresis comparator, Vo will be flipped upside-down.

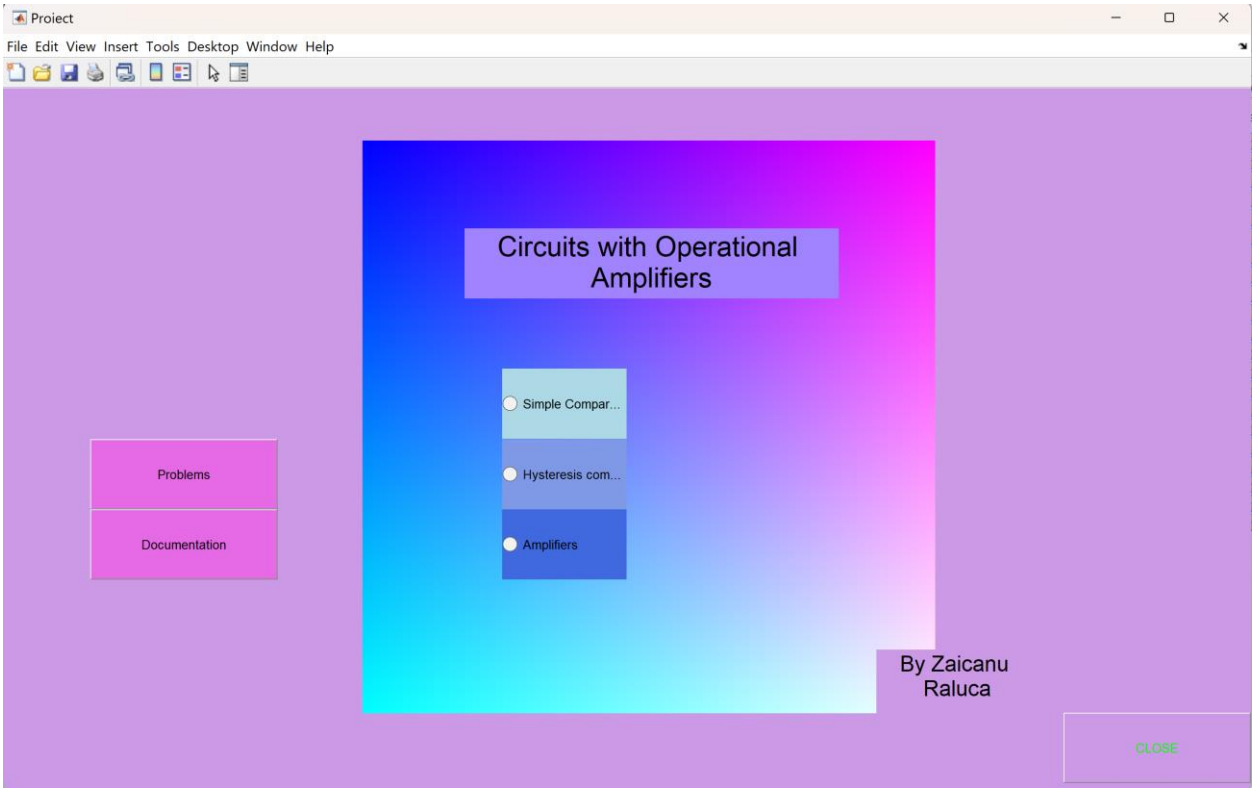
The amplifier has a negative feedback, because V^- is linked to Vo and there are two resistors situated just as above, but on the negative feedback loop. This feedback forces V^+-V^- to always be equal to 0, so that the formula for Vo is:

- Vo=(R1+R2)VI/R1

The gain of the amplifier (Av) can be calculated as Vo/VI, therefore the signal has the same shape as the input signal, but amplified.

Experimental part

When you first open the project, the main page looks like this:



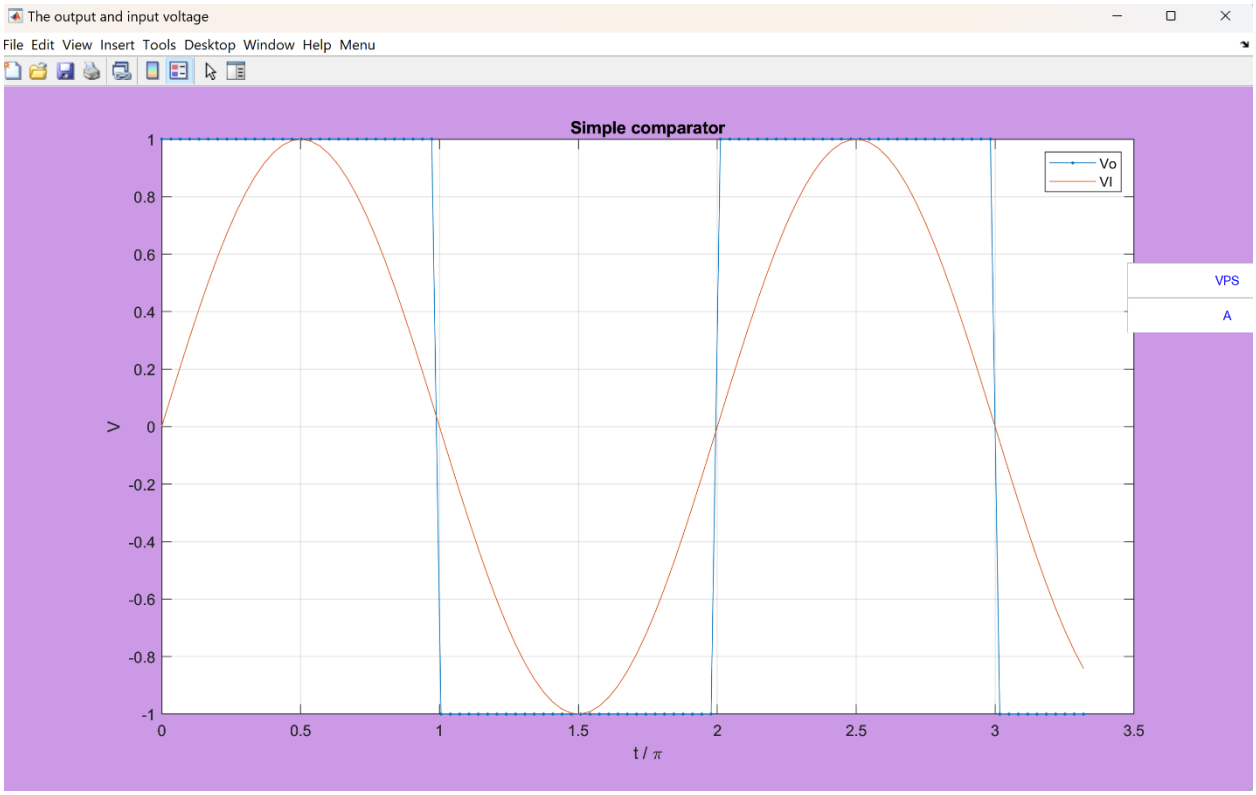
There are two text objects for the title and name, three pushbuttons for “close” , “Problems” and “documentation”(which opens this .pdf file) and three radiobuttons, which help the project expand further.

The gradient behind the radiobuttons is actually an image created by this code:

```
sz = [400 400];
gradx = repmat(linspace(0,1,sz(2)),[sz(1) 1]);% matrix of size sz(1) x sz(2) where each element is a value between 0 and 1 that increases linearly from left to right

grady = repmat(linspace(0,1,sz(1)).',[1 sz(2)]);% matrix of size sz(1) x sz(2) where each element is a value between 0 and 1 that increases linearly from top to bottom
bk = zeros(sz);%2D array of size sz filled with zeros
wh = ones(sz);%2D array of size sz filled with ones


grp3 = cat(3,gradx,grady,wh);% concatenates the 3 previously created 2D arrays along the 3rd dimension using the cat function resulting in 3D array grp3 with size sz(1) x sz(2) x 3
```

The image will have gradients of red, green and blue. The red channel will have the same values as gradx, the green channel will have the same values as grady and blue channel will have all ones.

If the Simple Comparator is clicked on, the Vo(VI) plot of the circuit will appear:



It has two edit objects, VPS changes the amplitude of Vo and A changes the amplitude of VI. The function for plotting this is:

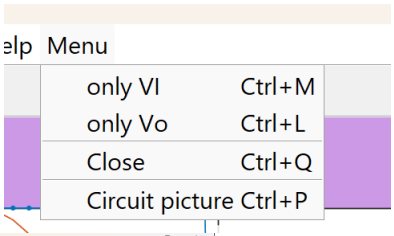
```
function grafic(VPS,A)
t = linspace(0,3*pi+1)';% independent variable t,defined as a linspace of values from
0 to 3*pi+1
x = VPS*square(t);%Vo

plot(t/pi,x,'.-',t/pi,A*sin(t)) % Plot the dependent variable x as a function of
t/pi, using '.-' as the marker style and plot A*sin(t)(VI) as a function of t/pi on
the same graph
xlabel('t / \pi')%axis labels
ylabel('V')
grid on
legend('Vo','VI')%legend
title('Simple comparator')%title
```

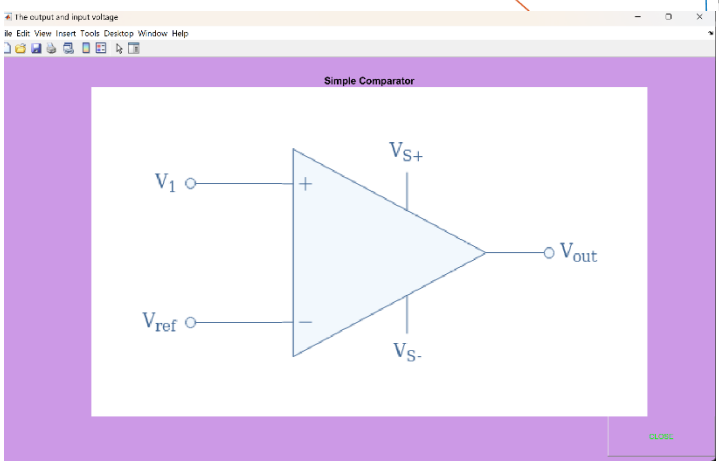
The edit objects are written as such:

```
GO_e=uicontrol('Style','edit',...
    'Units','normalized',...
    'Position',[0.9 0.7 0.16 0.05],...
    'foregroundcolor','b',...
    'String','VPS',...
    'Callback',['VPS=','str2num(get(gco,''String''));grafic(VPS,A);']);%sets VPS to
the numerical value of the string of the current object

GO_f=uicontrol('Style','edit',...
    'Units','normalized',...
    'Position',[0.9 0.65 0.16 0.05],...
    'foregroundcolor','b',...
    'String','A',...
    'Callback','A=str2num(get(gco,''string''));grafic(VPS,A);'); ]);%sets A to the
numerical value of the string of the current object
```

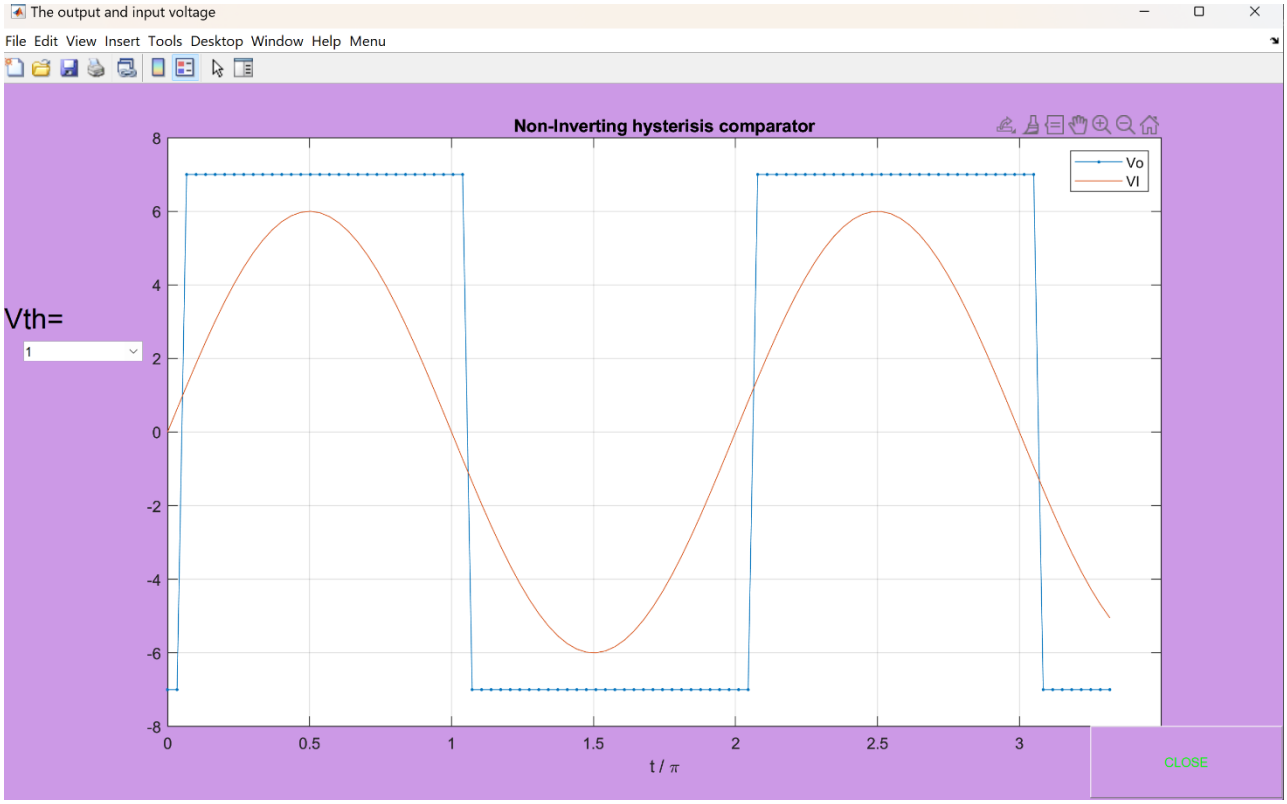
Above the plot there is a custom menu:

Which gives the user four options: to only see VI or Vo plotted, to close the figure and to show the circuit picture:





For the hysteresis comparator, the plot differs:



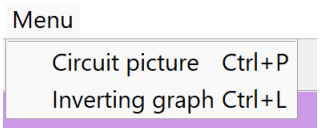
Vo is shifted to the right so that it changes when VI is approximately equal to Vth, which can be chosen from a popupmenu. This is done by changing x:

```
x = VPS*square(t-Vth*0.18);% Vth*0.18 is an approximation that unfortunately only
works(with small errors) for A=6 and VPS=7
```

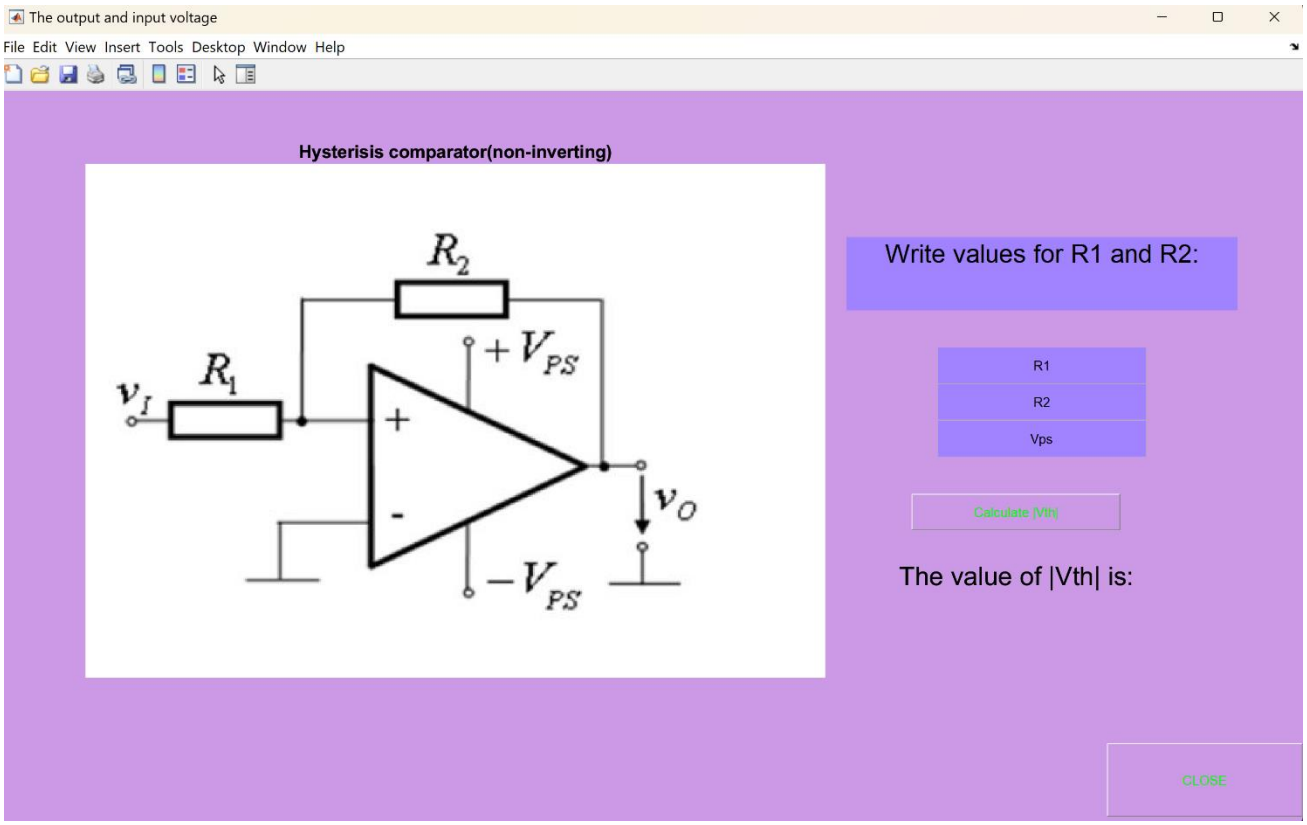
As for the popupmenu:

```
popup = uicontrol('Style', 'popup',...
    'String', {'1','2','3','4','5'},...
    'Position', [20 340 100 50],...
    'Callback', @setVth);


function setVth(source,callbackdata)
Vth=str2double(source.String{source.Value});% the str2double function converts the
string value of the "source" object into a numerical value and assigns it to the
variable "Vth"

graphHis2(Vth)%plots the graph with the new Vth
end
```

The menu above has two options:



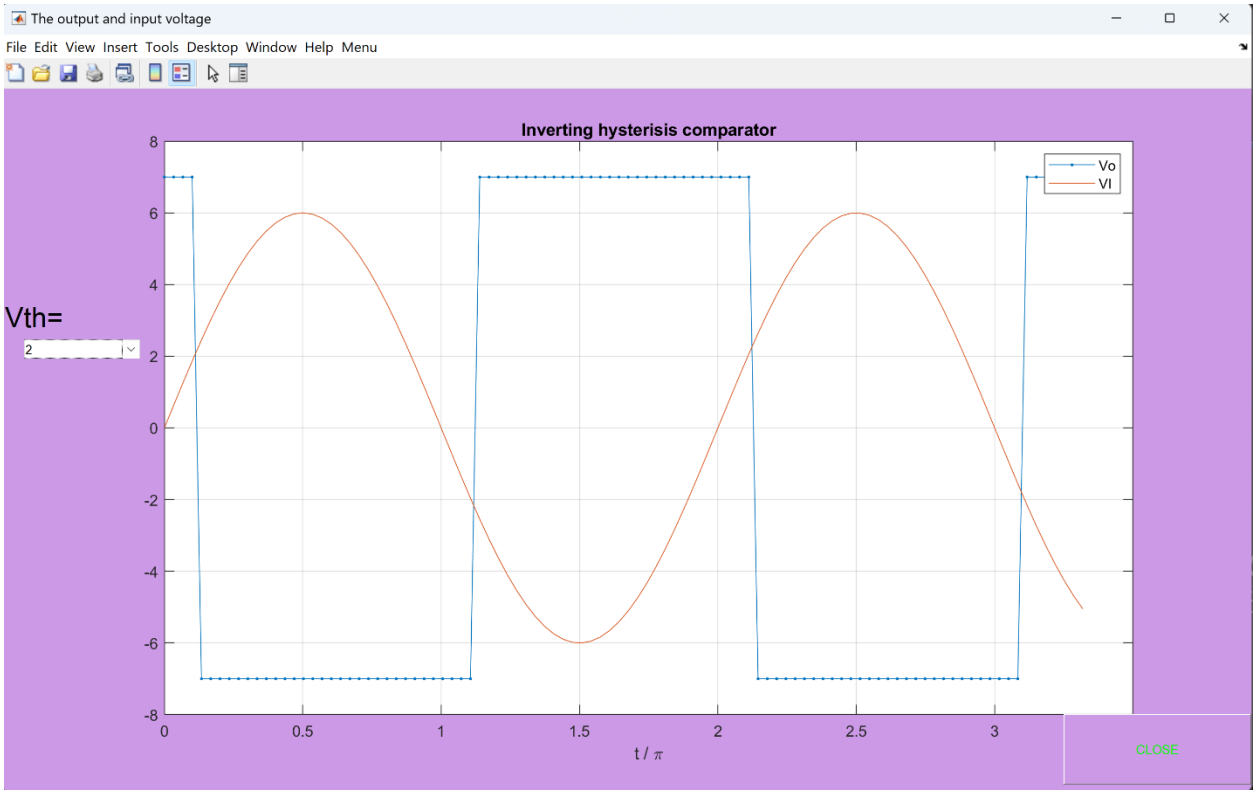
By clicking on “Circuit picture” there also appear three edit objects for R1, R2 and Vps and a pushbutton “Calculate |Vth|”(it calculates the value of Vth and displays it):



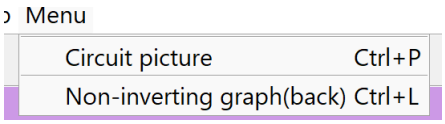

```
%text button

GO_t=uicontrol('Style','text',...
    'Units','normalized',...
    'FontSize',16,...
    'Position',[0.65 0.7 0.3 0.1],...
    'backgroundcolor','[0.63 0.51 1]',...
    'foregroundcolor','black',...
    'String','Write values for R1 and R2:');

%edit object for R1
Go_x=uicontrol('Style','edit',...
    'Units','normalized',...
    'Position',[0.72 0.6 0.16 0.05],...
    'foregroundcolor','black',...
     'backgroundcolor','[0.63 0.51 1]',...
    'String','R1');

%edit object for R2
Go_y= uicontrol('Style','edit',...
     'Units','normalized',...
```

```
    'Position',[0.72 0.55 0.16 0.05],...
    'foregroundcolor','black',...
     'backgroundcolor','[0.63 0.51 1]',...
    'String','R2');

%edit object for Vps
Go_z=uicontrol('Style','edit',...
    'Units','normalized',...
    'Position',[0.72 0.5 0.16 0.05],...
    'foregroundcolor','black',...
     'backgroundcolor','[0.63 0.51 1]',...
    'String','Vps');

%other text object
 h = text(1.1, 0.2, 'The value of |Vth| is:', 'Units', 'normalized',...
     'FontSize',16);

%pushbutton
GO_cal=uicontrol('Style','pushbutton',...
    'Units','normalized',...
    'Position',[0.7 0.4 0.16 0.05],...
    'backgroundcolor','[0.8 0.6 0.9]',...
    'foregroundcolor','g',...
    'String','Calculate |Vth|',...
    'Callback',{@calculate_callback_Vth, Go_x, Go_y, Go_z,h});%has the function
@calculate_callback_Vth as callback, but also the other elements are additional
inputs that are passed to the callback function so it can access the values of the
edit objects


function calculate_callback_Vth(hObject, eventdata, Go_x, Go_y, Go_z,h)
%the variables take the numerical values of the edit objects
    R1 = str2double(get(Go_x,'String'));
    R2 = str2double(get(Go_y,'String'));
    Vps = str2double(get(Go_z,'String'));


       Vth=(R1/R2)*Vps; %calculates Vth
      set(h, 'String', ['Vth=',num2str(Vth)]);%changes the String of the text object
to the value of Vth. In case not all variables are numerical it will "say": Vth=NaN,
an improvement coud be made so that it would show an error message

  end
```

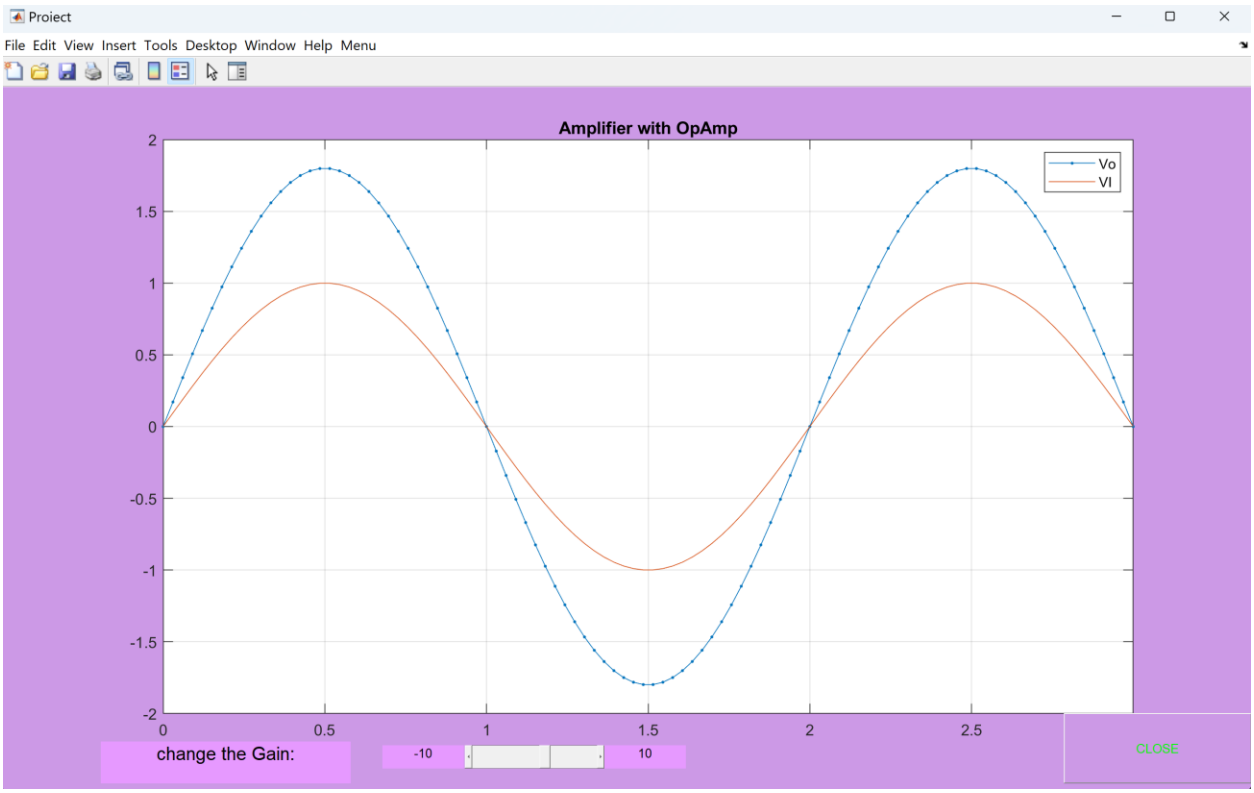
By clicking on “Inverting graph”, there will appear the same plot, but with Vo inverted (`x = -VPS*square(t-Vth*0.18);`):



It also has a menu with a circuit picture and a “back” option:

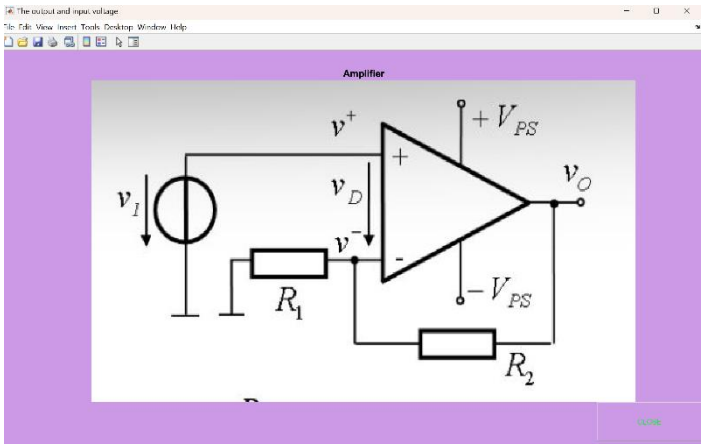


The plot for the amplifier changes completely, the output signal no longer being a square one, but an amplification of the sinusoidal input voltage(`X= Av*sin(t)`)and the gain can be modified by the user with the help of a slider object which goes from -10 to 10:

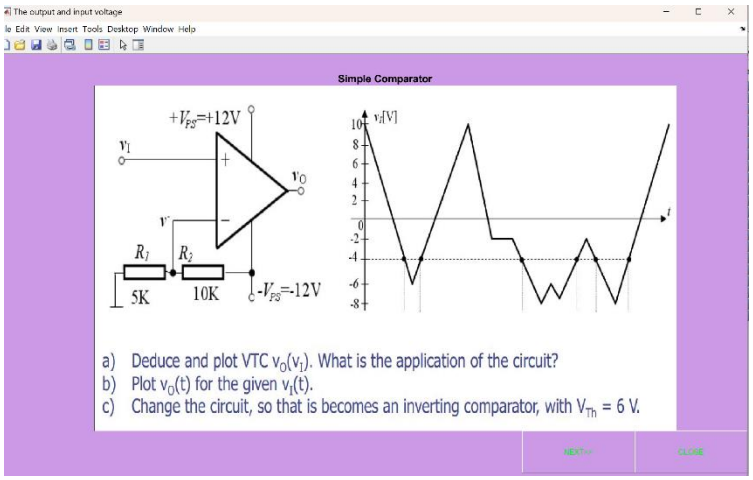


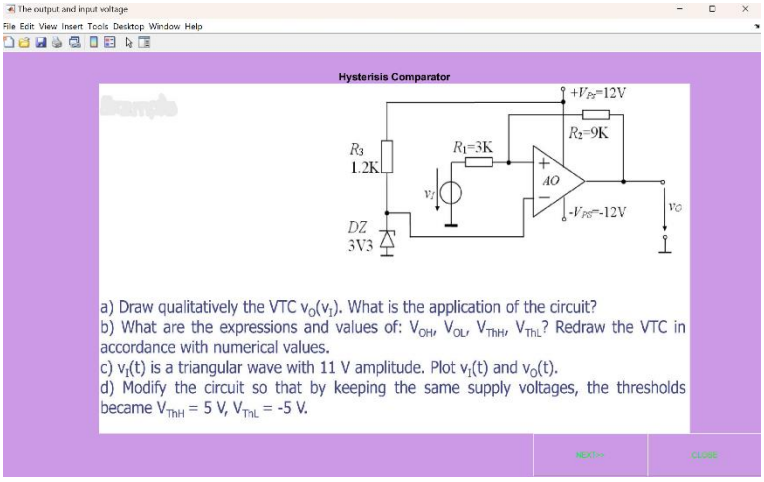

```
%the slider object

sld = uicontrol('Style', 'slider',...

   'Min',-10,'Max',10,'Value',1,...
   'Position', [400 20 120 20],...
   'Callback', @slider_callback);

function slider_callback(source,callbackdata)
val = source.Value;%gets the value from the slider object

graficAmplif(val);%plots the graph with the new value of the gain
end
```

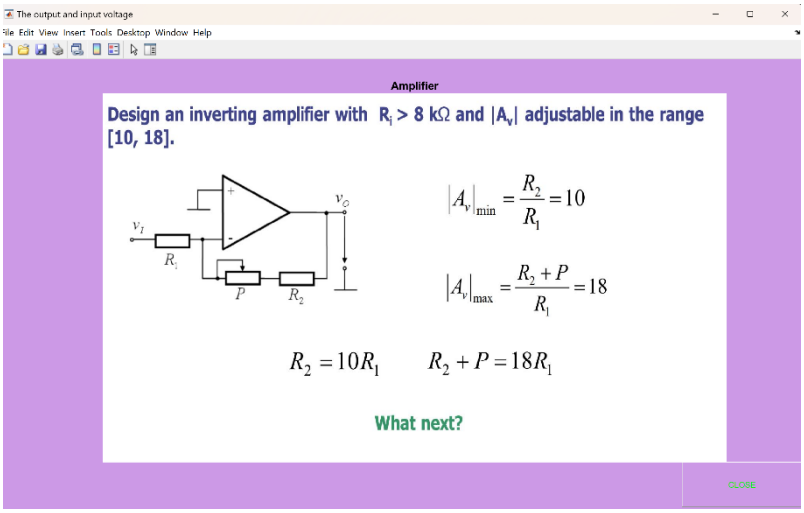
The menu also contains a circuit picture:



By pressing the pushbutton "Problems", three images will appear with simple exercises with the presented circuits. They have a "next" and a "close" button(the last one closes all three figures):







Conclusion

The aim of this project is to show some simple and very useful circuits that have as the main component the Operational Amplifier so that the user can experiment with the plots for visual and better understanding of the functionality of this very important circuit component.

For future development, some other circuits could be added to this project and it could also include some other kind of plots (maybe VTC's). Also, numerous problems could be solved with the help of MATLAB, therefore it could include more exercises with user-introduced parameters in the future.

Bibliography

- Electronic Devices - Laura-Nicoleta IVANCIU- for the circuit photos and information
- https://ch.mathworks.com/help/matlab/
- CAG laboratories
- "MATLAB for Students"-by Mihaela Cirlugea, Paul Farago

Appendix

%creation of the gradient picture

```
sz = [400 400];
gradx = repmat(linspace(0,1,sz(2)),[sz(1) 1]);
grady = repmat(linspace(0,1,sz(1)).',[1 sz(2)]);
bk = zeros(sz);
wh = ones(sz);


grp3 = cat(3,gradx,grady,wh);

imshow(grp3)
```

%The main page

```
Fig=figure('Name','Proiect',...
    'Units','normalized',...
    'Position',[.1 .1 .7 .7],...
    'NumberTitle','off','color',[0.8 0.6 0.9] );



GO_t=uicontrol('Style','text',...
    'Units','normalized',...
    'FontSize',18,...
    'Position',[0.37 0.7 0.3 0.1],...
    'backgroundcolor','[0.63 0.51 1]',...
    'foregroundcolor','black',...
    'String','Circuits with Operational Amplifiers');

imshow(grp3)

GO_t2=uicontrol('Style','text',...
    'Units','normalized',...
    'FontSize',14,...
    'Position',[0.7 0.1 0.13 0.1],...
    'backgroundcolor','[0.8 0.6 0.9]',...
    'foregroundcolor','black',...
    'String','By Zaicanu Raluca');

GO_f=uicontrol('Style','pushbutton',...
    'Units','normalized',...
    'Position',[0.07 0.4 0.15 0.1],...
    'backgroundcolor','[0.9, 0.41, 0.9]',...
    'String','Problems',...
    'Callback','P1');

GO_e=uicontrol('Style','pushbutton',...
    'Units','normalized',...
    'Position',[0.07 0.3 0.15 0.1],...
    'backgroundcolor','[0.9, 0.41, 0.9]',...
```

```
    'String','Documentation',...
    'Callback','P1');



GO_p=uicontrol('Style','pushbutton',...
    'Units','normalized',...
    'Position',[0.85 0.01 0.15 0.1],...
    'backgroundcolor','[0.8 0.6 0.9]',...
    'foregroundcolor','g',...
    'String','CLOSE',...
    'Callback','close');



GO_rb1=uicontrol('Style','radiobutton',...
    'Units','normalized',...
    'Position',[0.4 0.5 .1 .1],...
    'backgroundcolor','[0.68 0.85 0.90]',...
    'String',' Simple Comparator',...
    'Callback','graphVIVo') ;



GO_rb2=uicontrol('Style','radiobutton',...
    'Units','normalized',...
    'Position',[0.4 0.4 .1 .1],...
    'backgroundcolor','[0.5 0.6 0.9]',...
    'String',' Hysteresis comparators',...
    'Callback','hysVIVo') ;

GO_rb3=uicontrol('Style','radiobutton',...
    'Units','normalized',...
    'Position',[0.4 0.3 .1 .1],...
    'backgroundcolor','[0.25 0.41 0.88]',...
    'String',' Amplifiers',...
    'Callback','AmplificatorVIVo') ;
```

%the simple comparator main figure

```
Fig=figure('Name','The output and input voltage',...
    'Units','normalized',...
    'Position',[.1 .1 .7 .7],...
    'NumberTitle','off','color',[0.8 0.6 0.9] );

h = uimenu('Label','Menu');

uimenu(h,'Label','only VI','Callback','graphVI',...
    'Accelerator','m');
```

```
uimenu(h,'Label','only Vo','Callback','graphVo',...
    'Accelerator','L');

uimenu(h,'Label','Close','Callback','close',...
    'Separator','on','Accelerator','Q');

uimenu(h,'Label','Circuit picture','Callback','SimpleComparator',...
    'Separator','on','Accelerator','p');
VPS=1;
A=1;


grafic(VPS,A);

GO_e=uicontrol('Style','edit',...
    'Units','normalized',...
    'Position',[0.9 0.7 0.16 0.05],...
    'foregroundcolor','b',...
    'String','VPS',...
    'Callback',['VPS=','str2num(get(gco,''String''));grafic(VPS,A);']);

GO_f=uicontrol('Style','edit',...
    'Units','normalized',...
    'Position',[0.9 0.65 0.16 0.05],...
    'foregroundcolor','b',...
    'String','A',...
    'Callback','A=str2num(get(gco,''string''));grafic(VPS,A);');
```

%the function of the plot for the simple comparator

```
function grafic(VPS,A)


t = linspace(0,3*pi+1)';
x = VPS*square(t);

plot(t/pi,x,'.-',t/pi,A*sin(t))
xlabel('t / \pi')
ylabel('V')
grid on
legend('Vo','VI')
title('Simple comparator')
end
```

%figure with only VI plot

```
Fig=figure('Name','The input voltage',...
    'Units','normalized',...
    'Position',[.1 .1 .7 .7],...
    'NumberTitle','off','color',[0.8 0.6 0.9] );
```

```
closing();
graficVI(A)



GO_e=uicontrol('Style','edit',...
    'Units','normalized',...
    'Position',[0.9 0.65 0.16 0.05],...
    'foregroundcolor','b',...
    'String','A',...
    'Callback','A=str2num(get(gco,''string''));graficVI(A);');
```

%function for only VI plot

```
function graficVI(A)

t = linspace(0,3*pi)';
plot(t/pi,A*sin(t));
grid on
```

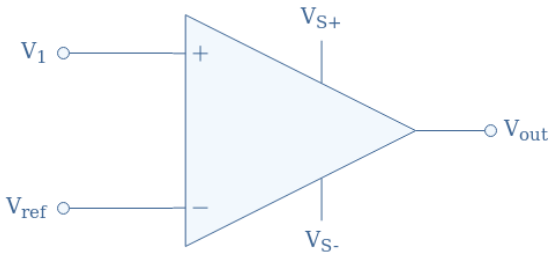
%figure for only Vo

```
Fig=figure('Name','The output voltage',...
    'Units','normalized',...
    'Position',[.1 .1 .7 .7],...
    'NumberTitle','off','color',[0.8 0.6 0.9] );


closing();

graficVo(VPS,A)

GO_e=uicontrol('Style','edit',...
    'Units','normalized',...
    'Position',[0.9 0.7 0.16 0.05],...
    'foregroundcolor','b',...
    'String','VPS',...
    'Callback',['VPS=','str2num(get(gco,''String''));graficVo(VPS,A);']);
```

%function for only Vo

```
function graficVo(VPS,A)

t = linspace(0,3*pi)';
x = VPS*square(t);
plot(t/pi,x)
```

```
grid on
```



%function that shows the picture of simple comparator

```
function SimpleComparator
 Fig=figure('Name','The output and input voltage',...
     'Units','normalized',...
     'Position',[.1 .1 .7 .7],...
     'NumberTitle','off','color',[0.8 0.6 0.9] );


i=imread('comparatorp1.png');
image(i)
axis off;
title('Simple Comparator');
closing();
```

%main figure for hysteresis comparator

```
Fig=figure('Name','The output and input voltage',...
    'Units','normalized',...
    'Position',[.1 .1 .7 .7],...
    'NumberTitle','off','color',[0.8 0.6 0.9] );


graphHis2(1);

GO_t=uicontrol('Style','text',...
    'Units','normalized',...
    'FontSize',18,...
    'Position',[0.001 0.6 0.05 0.1],...
    'backgroundcolor','[0.8 0.6 0.9]',...
    'foregroundcolor','black',...
    'String','Vth=');


popup = uicontrol('Style', 'popup',...
    'String', {'1','2','3','4','5'},...
    'Position', [20 340 100 50],...
    'Callback', @setVth);

closing();

h = uimenu('Label','Menu');

uimenu(h,'Label','Circuit picture','Callback','HysComparator',...
    'Separator','on','Accelerator','p');
```

```
uimenu(h,'Label','Inverting graph','Callback','hysInv_VIVo',...
    'Accelerator','L');
```

%function for the plot hysteresis

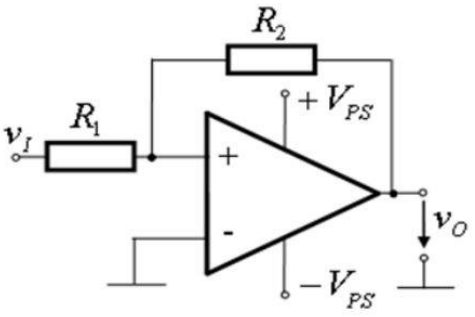
```
     t = linspace(0,3*pi+1)';

A=6;
VPS=7;


    x = VPS*square(t-Vth*0.18);

  plot(t/pi,x,'.-',t/pi,A*sin(t))
    xlabel('t / \pi')
    grid on
    legend('Vo','VI')
 title('Non-Inverting hysterisis comparator')
 end
```

%callback function for popup

```
function setVth(source,callbackdata)
Vth=str2double(source.String{source.Value});

graphHis2(Vth)
end
```

%function to show the picture and solve the exercise

```
function HysComparator
 Fig=figure('Name','The output and input
voltage',...
     'Units','normalized',...
     'Position',[.1 .1 .7 .7],...
     'NumberTitle','off','color',[0.8 0.6 0.9] );
closing();

imshow('hysim.jpg');
i=gca;

set(i, 'Position', [0.05 0.2 0.6 0.7])
axis off;
title('Hysterisis comparator(non-inverting)');

GO_t=uicontrol('Style','text',...
    'Units','normalized',...
    'FontSize',16,...
    'Position',[0.65 0.7 0.3 0.1],...
```



```
    'backgroundcolor','[0.63 0.51 1]',...
    'foregroundcolor','black',...
    'String','Write values for R1 and R2:');

Go_x=uicontrol('Style','edit',...
    'Units','normalized',...
    'Position',[0.72 0.6 0.16 0.05],...
    'foregroundcolor','black',...
     'backgroundcolor','[0.63 0.51 1]',...
    'String','R1');


Go_y= uicontrol('Style','edit',...
     'Units','normalized',...
    'Position',[0.72 0.55 0.16 0.05],...
    'foregroundcolor','black',...
     'backgroundcolor','[0.63 0.51 1]',...
    'String','R2');

Go_z=uicontrol('Style','edit',...
    'Units','normalized',...
    'Position',[0.72 0.5 0.16 0.05],...
    'foregroundcolor','black',...
     'backgroundcolor','[0.63 0.51 1]',...
    'String','Vps');

 h = text(1.1, 0.2, 'The value of |Vth| is:', 'Units', 'normalized',...
     'FontSize',16);


GO_cal=uicontrol('Style','pushbutton',...
    'Units','normalized',...
    'Position',[0.7 0.4 0.16 0.05],...
    'backgroundcolor','[0.8 0.6 0.9]',...
    'foregroundcolor','g',...
    'String','Calculate |Vth|',...
    'Callback',{@calculate_callback_Vth, Go_x, Go_y, Go_z,h});
```

%the callback function

```
function calculate_callback_Vth(hObject, eventdata, Go_x, Go_y, Go_z,h)
    R1 = str2double(get(Go_x,'String'));
    R2 = str2double(get(Go_y,'String'));
    Vps = str2double(get(Go_z,'String'));


       Vth=(R1/R2)*Vps;
      set(h, 'String', ['Vth=',num2str(Vth)]);

  end
```

%the main figure for inverting hysteresis comparator

```
Fig=figure('Name','The output and input voltage',...
    'Units','normalized',...
    'Position',[.1 .1 .7 .7],...
    'NumberTitle','off','color',[0.8 0.6 0.9] );


graphHisInv(1);

GO_t=uicontrol('Style','text',...
    'Units','normalized',...
    'FontSize',18,...
    'Position',[0.001 0.6 0.05 0.1],...
    'backgroundcolor','[0.8 0.6 0.9]',...
    'foregroundcolor','black',...
    'String','Vth=');


popup = uicontrol('Style', 'popup',...
    'String', {'1','2','3','4','5'},...
    'Position', [20 340 100 50],...
    'Callback', @setVthInv);


% GO_p=uicontrol('Style','pushbutton',...
%     'Units','normalized',...
%     'Position',[0.85 0.01 0.15 0.1],...
%     'backgroundcolor','[0.8 0.6 0.9]',...
%     'foregroundcolor','g',...
%     'String','CLOSE',...
%     'Callback','close');

closing();

h = uimenu('Label','Menu');

uimenu(h,'Label','Circuit picture','Callback','HysComparator_Inv',...
    'Separator','on','Accelerator','p');

uimenu(h,'Label','Non-inverting graph(back)','Callback','hysVIVo',...
    'Separator','on','Accelerator','l');
```

%function callback

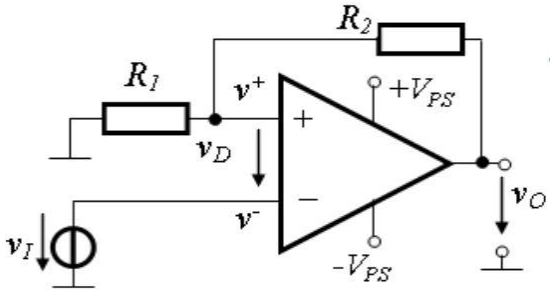
```
function setVthInv(source,callbackdata)
Vth=str2double(source.String{source.Value});

graphHisInv(Vth)
end
```

%function for the inverting plot

```
function graphHisInv(Vth)

     t = linspace(0,3*pi+1)';

A=6;
VPS=7;


    x = -VPS*square(t-Vth*0.18);

  plot(t/pi,x,'.-',t/pi,A*sin(t))
    xlabel('t / \pi')
    grid on
    legend('Vo','VI')
 title('Inverting hysterisis comparator')
 end
```

%function to show the picture for the inverting circuit

```
function HysComparator_Inv
 Fig=figure('Name','The output and input voltage',...
     'Units','normalized',...
     'Position',[.1 .1 .7 .7],...
     'NumberTitle','off','color',[0.8 0.6 0.9] );
closing();

i=imread('HysInv.jpg');
image(i)
axis off;
title('Hysterisis comparator(inverting)');
```



%function for close button

```
function closing()

GO_p=uicontrol('Style','pushbutton',...
    'Units','normalized',...
    'Position',[0.85 0.01 0.15 0.1],...
    'backgroundcolor','[0.8 0.6 0.9]',...
    'foregroundcolor','g',...
    'String','CLOSE',...
    'Callback','close');
```

%main figure for the amplifier

```
Fig=figure('Name','Proiect',...
    'Units','normalized',...
    'Position',[.1 .1 .7 .7],...
    'NumberTitle','off','color',[0.8 0.6 0.9] );

graficAmplif(0);

h = uimenu('Label','Menu');

uimenu(h,'Label','Circuit picture','Callback','Amplifier',...
    'Separator','on','Accelerator','p');


sld = uicontrol('Style', 'slider',...
   'Min',-10,'Max',10,'Value',1,...
   'Position', [400 20 120 20],...
   'Callback', @slider_callback);

 uicontrol('Style','text',...
      'Position', [329 20 70 20],...
      'backgroundcolor',[0.9 0.6 1],...
     'foregroundcolor','black',...
     'String',num2str(get(sld,'Min')));

 uicontrol('Style','text',...
     'Position', [520 20 70 20],...
     'backgroundcolor',[0.9 0.6 1],...
     'foregroundcolor','black',...
     'String',num2str(get(sld,'Max')));

 GO_t=uicontrol('Style','text',...
    'Units','normalized',...
    'FontSize',12,...
    'Position',[0.08 0.01 0.2 0.06],...
    'backgroundcolor','[0.9 0.6 1]',...
    'foregroundcolor','black',...
    'String','change the Gain:');


 closing();
```

%function for the plot

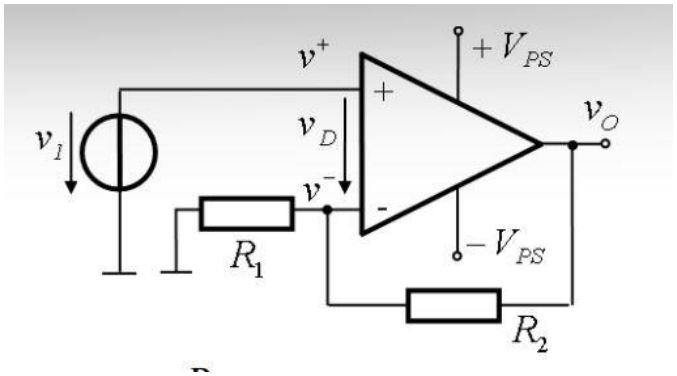
```
function graficAmplif(Av)

t = linspace(0,3*pi)';
plot(t/pi,Av*sin(t),'.-',t/pi,sin(t));
legend('Vo','VI')
grid on
title('Amplifier with OpAmp')
```

%slider callback

```
function slider_callback(source,callbackdata)
val = source.Value;

graficAmplif(val);
end
```

%displaying picture

```
unction Amplifier
 Fig=figure('Name','The output and input voltage',...
     'Units','normalized',...
     'Position',[.1 .1 .7 .7],...
     'NumberTitle','off','color',[0.8 0.6 0.9] );
closing();

i=imread('amplifier.jpg');
image(i)
axis off;
title('Amplifier');
```



%figure for the first problem

```
Fig=figure('Name','The output and input voltage',...
     'Units','normalized',...
     'Position',[.1 .1 .7 .7],...
     'NumberTitle','off','color',[0.8 0.6 0.9] );


i=imread('p1Comp.jpg');
image(i)
axis off;
title('Simple Comparator');
closing();

GO_p=uicontrol('Style','pushbutton',...
    'Units','normalized',...
    'Position',[0.7 0.01 0.15 0.1],...
    'backgroundcolor','[0.8 0.6 0.9]',...
    'foregroundcolor','g',...
    'String','NEXT>>',...
    'Callback','P2');
```

%figure for the second problem

```
Fig=figure('Name','The output and input voltage',...
     'Units','normalized',...
     'Position',[.1 .1 .7 .7],...
     'NumberTitle','off','color',[0.8 0.6 0.9] );


i=imread('p2Hys.jpg');
image(i)
axis off;
title('Hysterisis Comparator');
closing();

GO_p=uicontrol('Style','pushbutton',...
    'Units','normalized',...
    'Position',[0.7 0.01 0.15 0.1],...
    'backgroundcolor','[0.8 0.6 0.9]',...
    'foregroundcolor','g',...
    'String','NEXT>>',...
    'Callback','P3');
```

%and for the third

```
Fig=figure('Name','The output and input voltage',...
     'Units','normalized',...
     'Position',[.1 .1 .7 .7],...
     'NumberTitle','off','color',[0.8 0.6 0.9] );


i=imread('p3Ampl.jpg');
image(i)
axis off;
title('Amplifier');
GO_p=uicontrol('Style','pushbutton',...
    'Units','normalized',...
    'Position',[0.85 0.01 0.15 0.1],...
    'backgroundcolor','[0.8 0.6 0.9]',...
    'foregroundcolor','g',...
    'String','CLOSE',...
    'Callback','close,close,close'...
    );
```