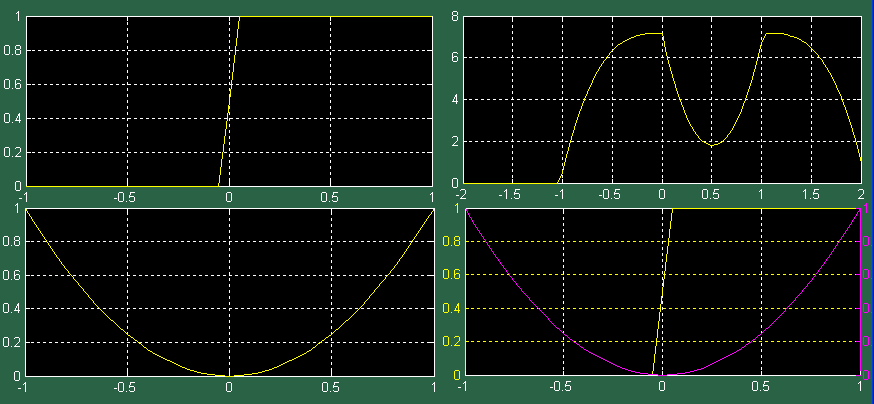
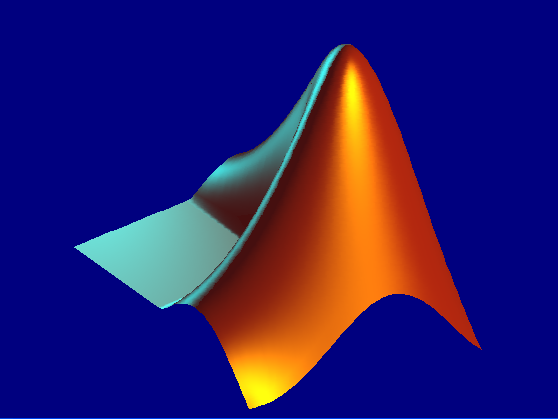
**CONVOLUTION IN CONTINUOUS AND DISCRETE TIME DOMAIN   
(PROJECT SIGNALS AND SYSTEMS USING MATLAB GUIDE)**

**GROUP MEMBERS**

***ADIL ABDUL QAYYUM 1352-FET/BSEE/F10 8/4***

***MUHAMMAD TAYYAB SHER 1444-FET/BSEE/F10 8/4***

***SHAHID ALI MURTZA 1437-FET/BSEE/F10 8/4*  
SUBMITTED TO  
 *ENGINEER MUHAMMAD ASAD  
 (IIUI)***

**WHAT IS CONVOLUTION?**A very general way of representing an LTI mapping from an input signal x to an output signal y is through convolution of the input with the system impulse response. Convolution is a powerful tool for determining the output of any LTI system to *any* input.  
 **Discrete Time Convolution:**

If the input to a system is x[n] and the impulse response of that system is h[n], then we can determine the output of the system, y[n], from the superposition sum defined as:

* Mathematical Convolution is defined as:

**(Convolution is Commutative)**

* If x[n] has nx number of elements and h[n] nh number of elements then y[n] must have
* Suppose x[n] is defined for and h[n] is defined for then

y[n] will be defined for

* In other words y[n]=0 for and
* If we consider the convolution in overlapping regions:   
  For and **NO OVERLAPPING**.   
  Let for

**If B > C**

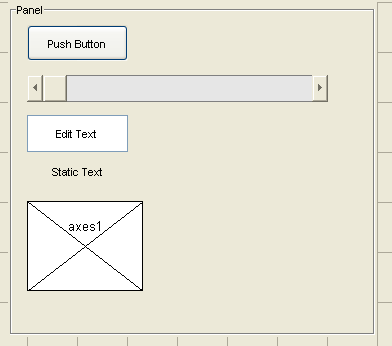
1st Semi Overlapping region is:   
 Full Overlapping region is: C  
 2nd Semi Overlapping region is: B  
**If B < C**

1st Semi Overlapping region is:   
 Full Overlapping region is:   
 2nd Semi Overlapping region is:

* Limits for k in each overlapping region are calculated as under

**If bx-ax < bh-ah**   
 1st Semi Overlapping region is:  
  
 Full Overlapping region is:   
  
 2nd Semi Overlapping region is:   
  
**If bx-ax > bh-ah**  
 1st Semi Overlapping region is:  
  
 Full Overlapping region is:   
  
 2nd Semi Overlapping region is:

**GUIDE:**

This is a project of **Signals and Systems** based on GUIDE [MATLAB].   
Feature used in this GUI are as under:

1. **Unicontrols:**

|  |  |
| --- | --- |
| **Unicontrol** | **Icon** |
| Panel |  |
| Push Button |  |
| Edit Text |  |
| Static Text |  |
| Slider |  |
| Axes |  |

**b) Matlab Functions:**

|  |  |
| --- | --- |
| **FUNCTION NAME** | **PURPOSE FOR WHITHCH IT IS USED IN GUI** |
| get() | Get the handles properties of unicontrol via tags |
| set() | Set the handles properties of unicontrol via tags |
| eval() | Evaluate the expression in a variable. |
| reshape() | To resize a matrix with same number of elements. |
| length() | To calculate the length of an array or vector. |
| fliplr() | For folding of a signal. |
| plotyy() | Two plots with different y-axes. |
| plot() | Plot a continuous time signal. |
| stem() | To plot a discrete time signal. |
| text() | Text on plot window/axis. |
| axis() | To set axis or handles of specific axis. |
| grid() | To set grid on or off. |
| colordef black | Set black background color of any axis. |
| helpdlg() | GUI Help dialogue box. |
| errordlg() | GUI Error dialogue box. |
| imread() | For Image Read(IIUI LOGO) |
| imshow() | To Show IIUI LOGO. |
| date | For current date on GUI window. |
| pause | For time delay in seconds. |
| wavread() | To read a .wav audio file. |
| soundsc() | To play a background music. |
| global | To define a variable as global in scope. |
| num2str() | To convert a string into a number. |
| str2double() | To convert a string into a number (double). |
| strcat() | To concatenate strings. |
| **refresh** | Refresh window |

**c) Unicontrols in detailed Used:**

|  |  |  |  |
| --- | --- | --- | --- |
| **PURPOSE NAME** | **Style** | **TAG NAME** | **Property Used** |
| CONVOLVE | Push Button | con |  |
| OL | Push Button | OL |  |
| FL | Push Button | FL |  |
| Animate | Push Button | Animate |  |
| Help | Push Button | Help |  |
| To go to discrete | Push Button | discrete |  |
| IIUI LOGO | Axis | iiui |  |
| PLOT X(t) | Axis | xx |  |
| PLOT H(t) | Axis | hh |  |
| PLOT CONVOLUTION | Axis | yy1 |  |
| PLOT FlipSig+OLapping+Animation | Axis | yy2 |  |
| Slider | Slider | SLDR |  |
| Input x(t) | Edit | xt |  |
| Input h(t) | Edit | ht |  |
| Minimum t for x(t) | Edit | txmin |  |
| Max t for x(t) | Edit | txmax |  |
| Min t for h(t) | Edit | thmin |  |
| Max t for h(t) | Edit | thmax |  |
| Slider Set Value | Edit | SV |  |
| Date | Text | date |  |
| Slider Limits Min | Text | SI |  |
| Slider Limit Max | Text | SM |  |
| For other Titles And Headings | Text | **NOT USED** |  |
| Panels | Panel | **NOT USED** |  |
|  |  |  |  |

**User Defined Function for Convolution:**

|  |
| --- |
| function [y m]=convolution\_(x,nx,h,nh)  m=nx(1)+nh(1):nh(end)+nx(end);  Lx=length(x);  Lh=length(h);  H=[h zeros(1,Lx)]'\*ones(1,Lx);  H=H(1:end-Lx); %%(Lx+Lh-1) by (Lx)elements Matrix  H=reshape(H,Lx+Lh-1,Lx);  y=(H\*x')';  end |
| function [y m]=convolution\_c(x,nx,h,nh, increment)  m=(nx(1)+nh(1)):increment:(nx(end)+nh(end));  Lx=length(x);  Lh=length(h);  H=[h zeros(1,Lx)]'\*ones(1,Lx);  H=H(1:end-Lx);  H=reshape(H,Lx+Lh-1,Lx);  y=(H\*x')'\*increment;  end |
| This is achieved by using the fact that **y=Hx** where H is a matrix known as toeplitz matrix. The built-in for toeplitz matrix in matlab is toeplitz (),  T = toeplitz(c, r)  T = toeplitz(r)  But here, this function is not used in convolution\_c.m/ convolution\_.m. |

**Default Lines To be executed:**

|  |
| --- |
| X=imread('iiui.jpg');  imshow(X,'parent',handles.iiui)  con\_Callback(hObject, eventdata, handles)  OL\_Callback(hObject, eventdata, handles)  set(handles.date,'String',date)  %[x f]=wavread('background.wav');  % soundsc(x,f) |

The above lines of code include

1. Showing IIUI LOGO
2. Showing Current Date
3. Playing Background Music
4. Call Functions con\_Callback and OL\_Callback to default values set on GUI.

**Push Buttons:**

1. **CONVOLVE**

This calls the functioncon\_Callback (hObject, eventdata, handles).  
CODE ADDED IS AS UNDER:

|  |
| --- |
| global inc  colordef black  axis(handles.xx);  inc=0.1;  global ax bx ah bh yx yh yy  ax=eval(get(handles.txmin,'String'));  bx=eval(get(handles.txmax,'String'));  t=ax:inc:bx;  t1=t;  yx=eval(get(handles.xt,'String'));  plot(handles.xx,t,yx)  grid(handles.xx,'on')  %axes(handles.hh)  ah=eval(get(handles.thmin,'String'));  bh=eval(get(handles.thmax,'String'));  t=ah:inc:bh;  t2=t;  yh=eval(get(handles.ht,'String'));  plot(handles.hh,t,yh)  grid(handles.hh,'on')  %axes(handles.yy1)  [yy m]=convolution\_c(yx,t1,yh,t2);  plot(handles.yy1,m,yy)  grid(handles.yy1,'on') |

1. **OL**

This calls the functionOL\_Callback (hObject, eventdata, handles).  
CODE ADDED IS AS UNDER:

|  |
| --- |
| con\_Callback(hObject, eventdata, handles)  global ax bx ah bh  colordef black  axis(handles.yy2);  cla(handles.yy2,'reset');  axis off  ax\_=get(handles.txmin,'String');s  bx\_=get(handles.txmax,'String');  ah\_=get(handles.thmin,'String');  bh\_=get(handles.thmax,'String');  INT1=strcat('y(t) =\int\_{',ax\_,'}^{','t- ',ah\_,'}x(\tau)h(t-\tau)d\tau ');  if bx-ax <= bh-ah  INT2=strcat('y(t) =\int\_{',ax\_,'}^{',bx\_,'}x(\tau)h(t-\tau)d\tau ');  else  INT2=strcat('y(t) =\int\_{',ah\_,'}^{',bh\_,'}x(\tau)h(t-\tau)d\tau ');  end  INT3=strcat('y(t) =\int\_{','t- ',bh\_,'}^{',bx\_,'}x(\tau)h(t-\tau)d\tau ');  A=strcat(' t <',num2str(ax+ah));  text(0.2,0.9,['\fontsize{14}y(t)=0 ' A])  B=num2str(ax+bh);  C=num2str(bx+ah);  D=strcat(' t >',num2str(bx+bh));  if ax+bh<ah+bx % B<C  text(0.2,0.7,...  ['\fontsize{14}' INT1 ' ' num2str(ax+ah) '\leqt<' B])  text(0.2,0.5,...  ['\fontsize{14}' INT2 ' ' B '\leqt\leq ' C])  text(0.2,0.3,...  ['\fontsize{14}' INT3 ' ' C '< t\leq ' num2str(bx+bh)])  end  if ax+bh>ah+bx % B>C  text(0.2,0.7,...  ['\fontsize{14}' INT1 ' ' num2str(ax+ah) '\leqt<' C])  text(0.2,0.5,...  ['\fontsize{14}' INT2 ' ' C '\leqt\leq' B])  text(0.2,0.3,...  ['\fontsize{14}' INT3 ' ' B '< t\leq ' num2str(bx+bh)])  end  if (ax+bh)==(ah+bx)% B=C  text(0.2,0.7,...  ['\fontsize{14}' INT1 ' ' num2str(ax+ah) '\leqt< ' C])  text(0.2,0.5,...  ['\fontsize{14}' INT2 ' t=' C])  text(0.2,0.3,...  ['\fontsize{14}' INT3 ' ' B '< t\leq ' num2str(bx+bh)])  end  text(0.2,0.1,['\fontsize{14}y(t)=0 ' D]) |

1. **FL**This calls the functionFL\_Callback (hObject, eventdata, handles).  
   CODE ADDED IS AS UNDER:

|  |
| --- |
| colordef black  con\_Callback(hObject, eventdata, handles)  global ax bx ah bh inc  t=ax:inc:bx;  global tx\_ x\_ th\_ h\_  tx\_=t;  x\_=eval(get(handles.xt,'String'));  t=ah:inc:bh;  h\_=eval(get(handles.ht,'String'));  t=-fliplr(t);  th\_=t;  h\_=fliplr(h\_);  plotyy(handles.yy2,tx\_,x\_,th\_,h\_)  grid(handles.yy2,'on') |

1. **Animate**This calls the functionAnimate\_Callback(hObject, eventdata, handles).  
   CODE ADDED IS AS UNDER:

|  |
| --- |
| con\_Callback(hObject, eventdata, handles)  global ax ah bx bh inc yy  a=ax+ah;  b=bx+bh;  t=a:inc:b;  k=0;  for i=a:inc:b  k=k+1;  plot(handles.yy1,t,yy,i,yy(k),'d')  grid(handles.yy1,'on')  set(handles.SLDR,'Value',i);  pause(0.1)  SLDR\_Callback(hObject, eventdata, handles)  end |

1. **Help**

This calls the functionHelp\_Callback(hObject, eventdata, handles).  
CODE ADDED IS AS UNDER:

|  |
| --- |
| helpdlg({'Help:','COVOLVE: For convolution results',...  'OL: For Overlapping Regions',' FL: For Flip Version of signal',...  'Animate:For Animating convolution overlapping'},'GUI HELP')  pause(1)  winopen('PROJECT REPORT\_.docx') |

1. **Go To Discrete**This executes following code.

function discrete\_Callback(hObject, eventdata, handles)

refresh

Sig\_project\_Disc

**Edits:**

1. **Slider Value:**

A call to SV\_Callback(hObject, eventdata, handles).  
Code Added.

|  |
| --- |
| S\_V=eval(get(handles.SV,'String'));  con\_Callback(hObject, eventdata, handles)  global ax bx ah bh  S=strcat('Min: ',num2str(ax+ah),' Max: ',num2str(bx+bh));  if ax+ah>S\_V ||bx+bh<S\_V  errordlg({'Please Set The Value of Slider Within limits.',... S},'Slider Value Error')  return  end  set(handles.SLDR,'Value' ,eval(get(handles.SV,'String')))  SLDR\_Callback(hObject, eventdata, handles) |

1. **Other Edits:**

|  |  |  |
| --- | --- | --- |
| Input x(t) | Edit | xt |
| Input h(t) | Edit | ht |
| Minimum t for x(t) | Edit | txmin |
| Max t for x(t) | Edit | txmax |
| Min t for h(t) | Edit | thmin |
| Max t for h(t) | Edit | thmax |

All these edits call the following code.

|  |
| --- |
| con\_Callback(hObject, eventdata, handles)  OL\_Callback(hObject, eventdata, handles) |

**SLIDER:**

Code Added is following: SLDR\_Callback()

|  |
| --- |
| FL\_Callback(hObject, eventdata, handles)  global tx\_ x\_ th\_ h\_ ax bx ah bh  set(handles.SI,'String',num2str(ax+ah))  set(handles.SM,'String',num2str(bx+bh))  set(handles.SLDR,'Min' ,ax+ah,'Max',bx+bh)  V=get(handles.SLDR,'Value');  set(handles.SV,'String',V), th\_1=th\_+V;  plotyy(handles.yy2,tx\_,x\_,th\_1,h\_)  grid(handles.yy2,'on') |

