**Project Report**

On

**Signal Operations App/GUI Development**

Siddhant Verma (67, 11810064)

Suyash Soni (63, 11811205)

Ameya Uplanchi (66, 11811163)

Sawan Damase (74, 11920059)

Guided by: Dr. M. S. Deshpande

**Course: Signals and Systems**

**2019-20 (Semester-I)**

**S. Y. B. Tech**



Department of Electronics and Telecommunication Engineering

BRACT’S

Vishwakarma Institute of Technology, Pune- 411037

**Signal Operations App/GUI Development**

**Abstract:** A signal, comprises of a set of information expressed as a function of any number of independent variables, that can be given as an input to a system, or derived as output from the system, to realize its true practical utility. The signal we derive out of a complex system might not always be in the form we want ,therefore being well acquainted with some **basic signal operations** may come really handy to enhance the understandability and applicability of signals. The mathematical transformation from one signal to another can be expressed as:

 Where, Y(t) represents the modified signal derived from the original signal

X(t) where having only one independent variable t.

**Project mapping to COs and POs**

|  |  |  |
| --- | --- | --- |
| **CO** | **CO Statements** | **Correlation** |
| **CO1** | Perform operations on dependent and independent variable of one dimensional signals | **Substantial** |
| **CO2** | Synthesize the signal using elementary signals | **Substantial** |
| **CO3** | Classify the systems and determine response of given CT/DT LTI system to any arbitrary input using convolution integral/sum | **NA** |
| **CO4** | Analyze the given CT deterministic signal in spectral domain using Fourier series/transform. | **NA** |
| **CO5** | Apply sampling theorem to obtain a discrete time signal from a continuous signal and to find the spectral components of the discrete-time signal using discrete Fourier transform (DFT). | **NA** |
| **CO6** | Analyze the given LTI systems using Laplace transform. | **NA** |

|  |  |  |
| --- | --- | --- |
| **PO** | **PO Statements** | **Correlation** |
| **PO1** | Engineering knowledge | **Substantial** |
| **PO2** | Problem analysis: | **Moderate** |
| **PO3** | Design/development of solutions | **Moderate** |
| **PO4** | Conduct investigations of complex problems | **Substantial** |
| **PO5** | Modern tool usage | **Substantial** |
| **PO6** | The engineer and society | **Moderate** |
| **PO7** | Environment and sustainability | **Moderate** |
| **PO8** | Ethics | **Moderate** |
| **PO9** | Individual and team work | **Substantial** |
| **PO10** | Communication | **Substantial** |
| **PO11** | Project management and finance | **Moderate** |
| **PO12** | Life-long learning | **Moderate** |

Correlation Levels: - 1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

**Introduction:**

The **basic set of signal operations** can be broadly classified as below:

## **Basic Signal Operations Performed on Dependent Variables:**

In this transformation, only the quadrature axis values are modified i.e magnitude of the signal changes, with no effects on the horizontal axis values or periodicity of signals like.

1. Amplitude scaling of signals.
2. Addition of signals.
3. Multiplication of signals.
4. Differentiation of signals.

## **Basic Signal Operations Performed on Dependent Variables**

This is exactly the opposite of the above mentioned case, here the periodicity of the signal is varied by modifying the horizontal axis values, while the amplitude or the strength remains constant. These are:-

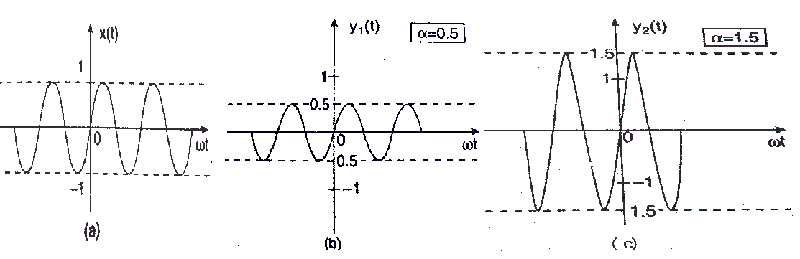
1. Time scaling of signals
2. Reflection of signals
3. Time-shifting of signals.

**Objectives:**

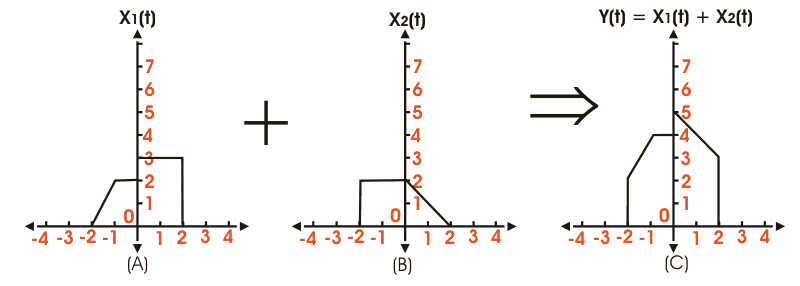
1. Objective of this app is to perform different operations on signals and obtain desired output on monitor.
2. Be able to describe signals mathematically and understand how to perform mathematical operations on signals. The operations should include operations on the dependent as well as independent variable

**Methodology:**

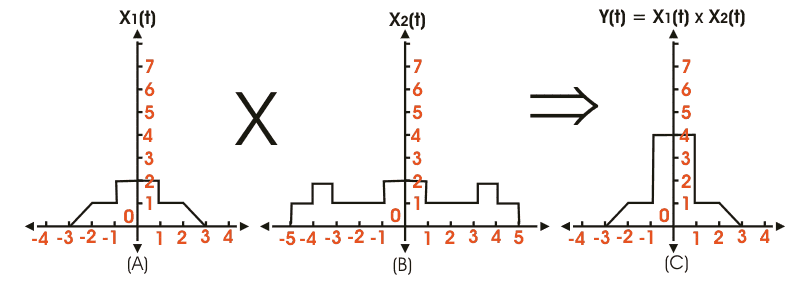
* 1. **Amplitude Scaling of Signals:**

Amplitude scaling is a very basic operation performed on signals to vary its strength. It can be mathematically represented asY(t)=**α**X(t).  
Here, **α** is the scaling factor, where:  
α1→signalisamplified.  
  
This is illustrated in the diagram, where the signal is attenuated when α = 0.5 in fig (b) and amplified when α = 1.5 as in fig (c).

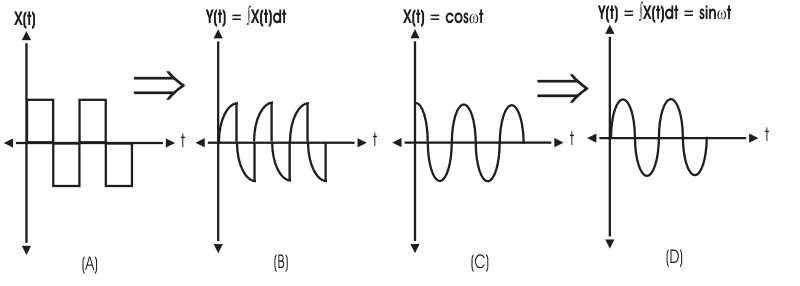
* 1. **Addition of Signals:**

This particular operation involves the addition of amplitude of two or more signals at each instance of time or any other independent variables which are common between the signals. Addition of signals is illustrated in the diagram below, where X1(t) and X2(t) are two time dependent signals, performing the additional operation on them we get,  
  


### **Multiplication of Signals**

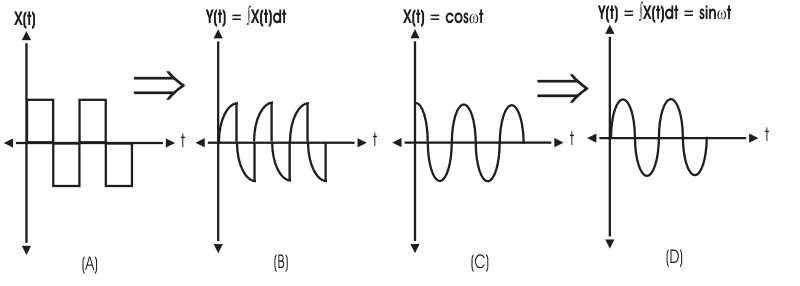
Like addition multiplication of signals also falls under the category of basic signal operations. Here multiplication of amplitude of two or more signals at each instance of time or any other independent variables is done which are common between the signals. The resultant signal we get has values equal to the product of amplitude of the parent signals for each instance of time. Multiplication of signals is illustrated in the diagram below, where X1(t) and X2(t) are two time dependent signals, on whom after performing the multiplication operation we get,  
  


## **Differentiation of Signals**

  
For differentiation of signals, it must be noted that this operation is only applicable for only continuous signals, as a discrete function cannot be differentiated. The modified signal we get on differentiation has tangential values of the parent signal at all instance of time. Mathematically it can be expressed as:-  
  
Differentiation of a standard square and sine wave is shown in the figure below.

### **Integration of Signals**

Like differentiation, integration of signals is also applicable to only continuous time signals. The limits of integration will be from – ∞ to present instance of time t. It is mathematically expressed as,  
  
Integration of some continuous time signals is shown in the diagram below.

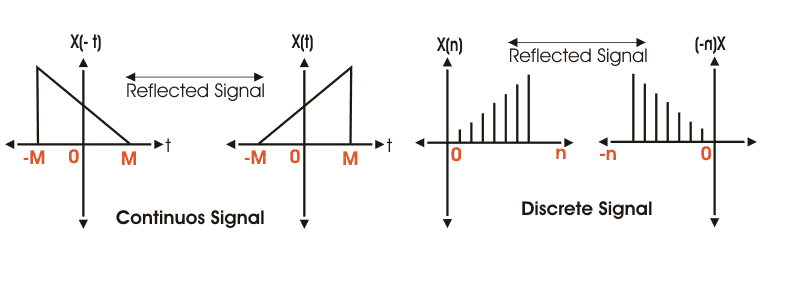


### **Time Scaling of Signals:**

Time scaling of signals of signals involves the modification of a periodicity of the signal, keeping its amplitude constant. Its mathematically expressed as,  
  
Where, X(t) is the original signal, and β is the scaling factor.  
If β > 1 implies, the signal is compressed and β

### **Time reversal of Signals:**

Reversal of signal is a very interesting operation applicable on both continuous and discrete signals. Here in this case the vertical axis acts as the mirror, and the transformed image obtained is exactly the mirror image of the parent signal.  
It can be defined as Y(t) = X(- t) Where, X(t) is the original signal.  
But if the reflected signal X(- t) = X(t); then its called an even signal.  
Where as when X(- t) = − X(t); then its known as an odd signal.

Its explained diagrammatically as,:  


1. **Time Shifting of Signals:**

Time shifting of signals is probably the most important one, and most widely used amongst all **basic signal operations**. Its generally used to fast-forward or delay a signal, as is necessary in most practical circumstances. Time shifting is mathematically expressed as,  
  
Where, X(t) is the original signal, and t0 represents the shift in time.  
For a signal X(t) if the position shift t0> 0. Then the signal is said to be right shifted or delayed.

**System Implementation:**

classdef app2 < matlab.apps.AppBase

% Properties that correspond to app components

properties (Access = public)

UIFigure matlab.ui.Figure

GridLayout matlab.ui.container.GridLayout

LeftPanel matlab.ui.container.Panel

ContinousTimeLabel\_2 matlab.ui.control.Label

SinusoidalWaveLabel matlab.ui.control.Label

FSEditFieldLabel matlab.ui.control.Label

FSEditField matlab.ui.control.NumericEditField

PlotButton matlab.ui.control.Button

CosineWaveLabel matlab.ui.control.Label

FSEditField\_2Label matlab.ui.control.Label

FSEditField\_2 matlab.ui.control.NumericEditField

GenerateButton matlab.ui.control.Button

PhaseChangeButton matlab.ui.control.Button

FreqChangeButton matlab.ui.control.Button

AmplitudeScalingButton\_4 matlab.ui.control.Button

AmplitudeScalingButton\_5 matlab.ui.control.Button

FreqchangeButton matlab.ui.control.Button

PhasechangeButton matlab.ui.control.Button

OperationsLabel\_3 matlab.ui.control.Label

OperationsLabel\_2 matlab.ui.control.Label

ContinousTimeLabel matlab.ui.control.Label

triangularWaveButton matlab.ui.control.Button

AmplitudeScalingButton matlab.ui.control.Button

TimeShiftingcompButton matlab.ui.control.Button

TimeShiftingAdvanceButton matlab.ui.control.Button

TimeFoldingButton matlab.ui.control.Button

ShiftingScalingButton matlab.ui.control.Button

SliderLabel matlab.ui.control.Label

Slider matlab.ui.control.Slider

TimeScalingButton matlab.ui.control.Button

CenterPanel matlab.ui.container.Panel

UIAxes matlab.ui.control.UIAxes

AmplitudeScalingButton\_6 matlab.ui.control.Button

TimeShiftingcompButton\_4 matlab.ui.control.Button

TimeScalingButton\_4 matlab.ui.control.Button

TimeShiftingadvButton\_2 matlab.ui.control.Button

TimeFoldingButton\_4 matlab.ui.control.Button

CreateasignalButton matlab.ui.control.Button

GenerateyourdesiredsignalLabel matlab.ui.control.Label

SelectFIleButton matlab.ui.control.Button

ClcClearallButton matlab.ui.control.Button

RightPanel matlab.ui.container.Panel

RampButton matlab.ui.control.Button

UnitStepButton matlab.ui.control.Button

ImpulseButton matlab.ui.control.Button

Slider2Label matlab.ui.control.Label

Slider2 matlab.ui.control.Slider

DiscreteTimeLabel matlab.ui.control.Label

AmplitudeScalingButton\_3 matlab.ui.control.Button

TimeShiftingcompButton\_3 matlab.ui.control.Button

TimeShiftingadvButton matlab.ui.control.Button

AmplitudeScalingButton\_10 matlab.ui.control.Button

TimeShiftingadvButton\_5 matlab.ui.control.Button

TimeShiftingcompButton\_7 matlab.ui.control.Button

TimeShiftingButton matlab.ui.control.Button

AmplitudeScalingButton\_11 matlab.ui.control.Button

TimeShiftingadvButton\_6 matlab.ui.control.Button

OperationsUnitStepLabel matlab.ui.control.Label

OperationsImpulseLabel\_2 matlab.ui.control.Label

Slider3Label matlab.ui.control.Label

Slider3 matlab.ui.control.Slider

RampSignalButton matlab.ui.control.Button

UnitStepSignalButton matlab.ui.control.Button

ImpulseSignalButton matlab.ui.control.Button

end

% Properties that correspond to apps with auto-reflow

properties (Access = private)

onePanelWidth = 576;

twoPanelWidth = 768;

end

methods (Access = private)

function [x] = y(t)

x1 = t+1

x2 = -t+1

x = x1.\*(-1<t & t<=0)+x2.\*(0<t & t<=1);

end

end

methods (Access = public)

function r = ramp(x)

r = max(0,x)

end

end

% Callbacks that handle component events

methods (Access = private)

% Code that executes after component creation

function startupFcn(app)

end

% Changes arrangement of the app based on UIFigure width

function updateAppLayout(app, event)

currentFigureWidth = app.UIFigure.Position(3);

if(currentFigureWidth <= app.onePanelWidth)

% Change to a 3x1 grid

app.GridLayout.RowHeight = {725, 725, 725};

app.GridLayout.ColumnWidth = {'1x'};

app.CenterPanel.Layout.Row = 1;

app.CenterPanel.Layout.Column = 1;

app.LeftPanel.Layout.Row = 2;

app.LeftPanel.Layout.Column = 1;

app.RightPanel.Layout.Row = 3;

app.RightPanel.Layout.Column = 1;

elseif (currentFigureWidth > app.onePanelWidth && currentFigureWidth <= app.twoPanelWidth)

% Change to a 2x2 grid

app.GridLayout.RowHeight = {725, 725};

app.GridLayout.ColumnWidth = {'1x', '1x'};

app.CenterPanel.Layout.Row = 1;

app.CenterPanel.Layout.Column = [1,2];

app.LeftPanel.Layout.Row = 2;

app.LeftPanel.Layout.Column = 1;

app.RightPanel.Layout.Row = 2;

app.RightPanel.Layout.Column = 2;

else

% Change to a 1x3 grid

app.GridLayout.RowHeight = {'1x'};

app.GridLayout.ColumnWidth = {215, '1x', 242};

app.LeftPanel.Layout.Row = 1;

app.LeftPanel.Layout.Column = 1;

app.CenterPanel.Layout.Row = 1;

app.CenterPanel.Layout.Column = 2;

app.RightPanel.Layout.Row = 1;

app.RightPanel.Layout.Column = 3;

end

end

% Button pushed function: AmplitudeScalingButton

function AmplitudeScalingButtonPushed(app, event)

t = -10:0.01:10; %Time vector

value\_amplitude = app.Slider.Value;

y2 = value\_amplitude\*y(t); %Amplitude scaling

plot(app.UIAxes , t , y2)

grid(app.UIAxes)

end

% Button pushed function: TimeShiftingcompButton

function TimeShiftingcompButtonPushed(app, event)

t = -10:0.01:10;

value\_amplitude = app.Slider.Value;

y3 = y(t-value\_amplitude); %Time shifting

plot(app.UIAxes , t , y3)

grid(app.UIAxes)

end

% Button pushed function: TimeShiftingAdvanceButton

function TimeShiftingAdvanceButtonPushed(app, event)

t = -10:0.01:10; %Time vector

value\_amplitude = app.Slider.Value;

y4 = y(t+value\_amplitude)

plot(app.UIAxes , t , y4)

grid(app.UIAxes)

end

% Button pushed function: TimeScalingButton

function TimeScalingButtonPushed(app, event)

t = -10:0.01:10; %Time vector

value\_amplitude = app.Slider.Value;

y5 = y(value\_amplitude\*t); %Time scaling ( Com[ression)

plot(app.UIAxes , t , y5)

grid(app.UIAxes)

end

% Button pushed function: TimeFoldingButton

function TimeFoldingButtonPushed(app, event)

t = -10:0.01:10; %Time vector

value\_amplitude = app.Slider.Value;

y7 = y(value\_amplitude\*t); %Time folding

plot(app.UIAxes , t , y7)

grid(app.UIAxes)

end

% Button pushed function: ShiftingScalingButton

function ShiftingScalingButtonPushed(app, event)

t = -10:0.01:10; %Time vector

value\_amplitude = app.Slider.Value;

y8 = y(value\_amplitude\*t -value\_amplitude); %Shiftings and scaling

plot(app.UIAxes , t , y8)

grid(app.UIAxes)

end

% Button pushed function: triangularWaveButton

function triangularWaveButtonPushed(app, event)

t = -10:0.01:10; %Time vector

value\_amplitude = app.Slider.Value;

y1 = y(t); %Orignal Signal

plot(app.UIAxes , t , y1)

hold(app.UIAxes)

grid(app.UIAxes)

end

% Callback function

function SquareWaveButton\_2Pushed(app, event)

s=0:.001:1;

f=3;

sq=10\*0.5\*(square(2\*pi\*f\*s)+1);

plot(app.UIAxes , s , sq)

end

% Value changed function: Slider

function SliderValueChanged(app, event)

value = app.Slider.Value;

end

% Callback function

function AmplitudeScalingButton\_2Pushed(app, event)

s=0:.001:1;

f=3;

sq=10\*0.5\*(square(2\*pi\*f\*s)+1);

sq2 = 2\*sq

plot(app.UIAxes , s , sq2)

end

% Callback function

function ClearallButtonPushed(app, event)

clear all;

end

% Callback function

function TimeShiftingcompButton\_2Pushed(app, event)

s=-10:10;

f=3;

sq=10\*0.5\*(square(2\*pi\*f\*s)+1);

sq3 = sq(s-2)

plot(app.UIAxes , s , sq2)

end

% Callback function

function TimeShiftingadvButtonPushed(app, event)

s=10:.01:20 ; %Time vector

value\_amplitude = app.Slider.Value;

f=2;

l=1000\*0.5\*(square(2\*pi\*f\*s)+1);

l3 = value\_amplitude(s-2); %Amplitude scaling

plot(app.UIAxes , s, l3)

hold(app.UIAxes)

end

% Callback function

function TimeScalingButton\_2Pushed(app, event)

s=0:.001:1;

f=3;

sq=10\*0.5\*(square(2\*pi\*f\*s)+1);

sq2 = sq(2\*sq)

plot(app.UIAxes , s , sq2)

end

% Callback function

function TimeFoldingButton\_2Pushed(app, event)

s=10:.01:20 ; %Time vector

value\_amplitude = app.Slider.Value;

f=2;

l=1000\*0.5\*(square(2\*pi\*f\*s)+1);

l5 = value\_amplitude(-1\*s); %Amplitude scaling

plot(app.UIAxes , s, l5)

hold(app.UIAxes)

end

% Callback function

function ShiftingScalingButton\_2Pushed(app, event)

s=10:.01:20 ; %Time vector

value\_amplitude = app.Slider.Value;

f=2;

l=1000\*0.5\*(square(2\*pi\*f\*s)+1);

l6 = value\_amplitude(2\*s-3); %Amplitude scaling

plot(app.UIAxes , s, l1)

hold(app.UIAxes)

end

% Button pushed function: SelectFIleButton

function SelectFIleButtonPushed(app, event)

[file,path] = uigetfile('\*.m');

if isequal(file,0)

disp('User selected Cancel');

else

disp(['User selected ', fullfile(path,file)]);

end

end

% Callback function

function dtEditFieldValueChanged(app, event)

value = app.dtEditField.Value;

dt = app.dtEditField.Value;

end

% Callback function

function DurationEditFieldValueChanged(app, event)

end

% Callback function

function dt2EditFieldValueChanged(app, event)

end

% Callback function

function DTsignalButtonPushed(app, event)

value = app.nEditField.Value;

n = app.nEditField;

f\_one = 2\*0.5.^n.\* heaviside(n+2)

stem(app.UIAxes , f\_one , n)

end

% Callback function

function LowerLimitEditFieldValueChanged(app, event)

value = app.LowerLimitEditField.Value;

n\_value = app.LowerLimitEditField.Value;

end

% Callback function

function HigherLimitEditFieldValueChanged(app, event)

value = app.HigherLimitEditField.Value;

n\_value2 = app.HigherLimitEditField.Value;

end

% Callback function

function nEditFieldValueChanged(app, event)

value = app.nEditField.Value;

n\_n2 = app.nEditField;

end

% Callback function

function EditFieldValueChanged(app, event)

value = app.FSEditField.Value;

n\_n2 = app.nEditField;

end

% Button pushed function: RampButton

function RampButtonPushed(app, event)

value = app.Slider.Value;

n\_1 = app.Slider.Value;

value = app.Slider2.Value;

n\_2 = app.Slider2.Value;

n=n\_1:n\_2; % define index n

ramp=n.\*(n>=0); % define a ramp

stem(app.UIAxes , n,ramp)

end

% Value changed function: Slider2

function Slider2ValueChanged(app, event)

value = app.Slider2.Value;

dt\_1 = app.Slider2.Value;

end

% Button pushed function: UnitStepButton

function UnitStepButtonPushed(app, event)

value = app.Slider.Value;

n\_1 = app.Slider.Value;

value = app.Slider2.Value;

n\_2 = app.Slider2.Value;

n=n\_1:n\_2; % specify index n

step\_sig=(n>=0); % define the unit step sequence

stem(app.UIAxes , n, step\_sig ) % plot the unit step sequence

end

% Button pushed function: ImpulseButton

function ImpulseButtonPushed(app, event)

value = app.Slider.Value;

n\_1 = app.Slider.Value;

value = app.Slider2.Value;

n\_2 = app.Slider2.Value;

n=n\_1:n\_2; % specify index n

delta=(n==0); % define the delta sequence

stem(app.UIAxes , n,delta) % plot the delta sequence

end

% Value changed function: FSEditField

function FSEditFieldValueChanged(app, event)

value = app.FSEditField.Value;

end

% Button pushed function: PlotButton

function PlotButtonPushed(app, event)

value = app.FSEditField.Value;

Fs = app.FSEditField.Value;

Ts=1/Fs; % sampling interval

t=0:Ts:0.1; % sampling instants

x=sin(2\*pi\*50\*t); % signal vector

plot(app.UIAxes , t,x);

end

% Button pushed function: GenerateButton

function GenerateButtonPushed(app, event)

value = app.FSEditField.Value;

Fs1 = app.FSEditField\_2.Value;

Ts=1/Fs1; % sampling interval

t=0:Ts:0.1; % sampling instants

x=cos(2\*pi\*50\*t); % signal vector

plot(app.UIAxes , t,x);

end

% Button pushed function: RampSignalButton

function RampSignalButtonPushed(app, event)

value = app.Slider.Value;

n\_1 = app.Slider.Value;

dt = .001;

duration = n\_1;

t4 = -1: dt : 0-dt;

t5 = 0:dt:duration-dt;

t6 = [t4 , t5];

x4=[zeros(1 , length(t4)) t5];

plot(app.UIAxes , t6 , x4 )

end

% Button pushed function: UnitStepSignalButton

function UnitStepSignalButtonPushed(app, event)

value = app.Slider.Value;

n\_1 = app.Slider.Value;

dt = .001;

duration =n\_1;

t7 = -1: dt : 0-dt;

t8 = 0:dt:duration-dt;

t9 = [t7 , t8];

x5=[zeros(1 , length(t7)) ones(1 , length(t8))];

plot(app.UIAxes , t9,x5)

end

% Button pushed function: ImpulseSignalButton

function ImpulseSignalButtonPushed(app, event)

duration = 2;

dt = 0.001

t10 = -1\*duration:dt:duration-dt;

impulse = t10==0;

plot(app.UIAxes , t10 , impulse)

end

% Button pushed function: AmplitudeScalingButton\_4

function AmplitudeScalingButton\_4Pushed(app, event)

value = app.Slider.Value;

n = app.Slider2.Value;

value = app.FSEditField.Value;

Fs = app.FSEditField.Value;

Ts=1/Fs; % sampling interval

t=0:Ts:0.1; % sampling instants

x=sin(2\*pi\*50\*t); % signal vector

x3 = n\*x

plot(app.UIAxes , t,x3);

end

% Button pushed function: FreqChangeButton

function FreqChangeButtonPushed(app, event)

value = app.Slider.Value;

n = app.Slider2.Value;

n2 = app.Slider2.Value;

value = app.FSEditField.Value;

Fs = app.FSEditField.Value;

Ts=1/Fs; % sampling interval

t=0:Ts:0.1; % sampling instants

x=sin(2\*pi\*n2\*t); % signal vector

plot(app.UIAxes , t,x);

end

% Button pushed function: PhaseChangeButton

function PhaseChangeButtonPushed(app, event)

value = app.Slider.Value;

phase1 = app.Slider2.Value;

value = app.FSEditField.Value;

Fs = app.FSEditField.Value;

Ts=1/Fs; % sampling interval

t=0:Ts:0.1; % sampling instants

x=sin(2\*pi\*50\*t+phase1); % signal vector

plot(app.UIAxes , t,x);

end

% Value changed function: FSEditField\_2

function FSEditField\_2ValueChanged(app, event)

value = app.FSEditField\_2.Value;

Fs1 = app.FSEditField\_2.Value;

end

% Button pushed function: AmplitudeScalingButton\_5

function AmplitudeScalingButton\_5Pushed(app, event)

value = app.Slider2.Value;

n = app.Slider2.Value;

value = app.FSEditField.Value;

Fs = app.FSEditField.Value;

Ts=1/Fs; % sampling interval

t=0:Ts:0.1; % sampling instants

x=cos(2\*pi\*50\*t); % signal vector

x11 = n\*x

plot(app.UIAxes , t ,x11);

end

% Button pushed function: FreqchangeButton

function FreqchangeButtonPushed(app, event)

value = app.Slider2.Value;

n = app.Slider2.Value;

value = app.FSEditField.Value;

Fs = app.FSEditField.Value;

Ts=1/Fs; % sampling interval

t=0:Ts:0.1; % sampling instants

x=cos(2\*pi\*n\*t); % signal vector

plot(app.UIAxes , t ,x);

end

% Button pushed function: CreateasignalButton

function CreateasignalButtonPushed(app, event)

Value = app.Slider.Value;

n1 = app.Slider.Value : -2;

n2 = app.Slider2.Value : 2;

n3 = app.Slider3.Value : 10;

n = [ n1 n2 n3];

t = length(n)

x = [ ones(1,length(n1)) 2\*ones(1 , length(n2)) zeros(1 ,length(n3))]

plot(app.UIAxes , n , x )

end

% Button pushed function: ClcClearallButton

function ClcClearallButtonPushed(app, event)

end

% Button pushed function: AmplitudeScalingButton\_6

function AmplitudeScalingButton\_6Pushed(app, event)

Value = app.Slider.Value;

n1 = app.Slider.Value : -2;

n2 = app.Slider2.Value : 2;

n3 = app.Slider3.Value : 10;

n = [ n1 n2 n3];

t = length(n)

x = [ ones(1,length(n1)) 2\*ones(1 , length(n2)) zeros(1 ,length(n3))]

x2 = 2\*x

plot(app.UIAxes , n , x2 )

end

% Button pushed function: TimeShiftingcompButton\_4

function TimeShiftingcompButton\_4Pushed(app, event)

Value = app.Slider.Value;

n1 = app.Slider.Value : -2;

n2 = app.Slider2.Value : 2;

n3 = app.Slider3.Value : 10;

n = [ n1 n2 n3];

t = length(n)

x = [ ones(1,length(n1)) 2\*ones(1 , length(n2)) zeros(1 ,length(n3))]

x5 = x(n1\*x)

plot(app.UIAxes , n , x5 )

end

% Callback function

function AmplitudeScalingButton\_7Pushed2(app, event)

value = app.Slider.Value;

n\_1 = app.Slider.Value;

n = app.Slider2.Value

dt = .001;

duration = n\_1;

t4 = -1: dt : 0-dt;

t5 = 0:dt:duration-dt;

t6 = [t4 , t5];

x4=[zeros(1 , length(t4)) t5];

x5 =n\*x4

plot(app.UIAxes , t6 , x5 )

end

% Callback function

function AmplitudeScalingButton\_8Pushed(app, event)

t = -10:0.01:10; %Time vector

value\_amplitude = app.Slider.Value;

y2 = value\_amplitude\*y(t); %Amplitude scaling

plot(app.UIAxes , t , y2)

grid(app.UIAxes)

end

% Callback function

function TimeShiftingcompButton\_5Pushed(app, event)

t = -10:0.01:10;

value\_amplitude = app.Slider.Value;

y3 = y(t-value\_amplitude); %Time shifting

plot(app.UIAxes , t , y3)

grid(app.UIAxes)

end

% Callback function

function TimeShiftingcompButton\_2Pushed2(app, event)

value = app.Slider.Value;

n\_1 = app.Slider.Value;

n = app.Slider2.Value

dt = .001;

duration = n\_1;

t4 = -1: dt : 0-dt;

t5 = 0:dt:duration-dt;

t6 = [t4 , t5];

x4=[zeros(1 , length(t4)) t5];

x5 = x4(t6-n)

plot(app.UIAxes , t6 , x5 )

end

% Button pushed function: AmplitudeScalingButton\_3

function AmplitudeScalingButton\_3Pushed(app, event)

value = app.Slider.Value;

n\_1 = app.Slider.Value;

value = app.Slider2.Value;

n\_2 = app.Slider2.Value;

n1 = app.Slider2.Value

n=n\_1:n\_2; % define index n

ramp=n.\*(n>=0); % define a ramp

n2 = n1\*ramp

stem(app.UIAxes , n2,ramp)

end

% Button pushed function: TimeShiftingcompButton\_3

function TimeShiftingcompButton\_3Pushed(app, event)

value = app.Slider.Value;

n\_1 = app.Slider.Value;

value = app.Slider2.Value;

n\_2 = app.Slider2.Value;

n1 = app.Slider2.Value

n=n\_1:n\_2; % define index n

ramp=n.\*(n>=0); % define a ramp

n2 = n1\*ramp-n

stem(app.UIAxes , n2,ramp)

end

% Button pushed function: TimeShiftingadvButton

function TimeShiftingadvButtonPushed2(app, event)

value = app.Slider.Value;

n\_1 = app.Slider.Value;

value = app.Slider2.Value;

n\_2 = app.Slider2.Value;

n1 = app.Slider2.Value

n=n\_1:n\_2; % define index n

ramp=n.\*(n>=0); % define a ramp

n2 = n1\*ramp+n

stem(app.UIAxes , n2,ramp)

end

% Callback function

function TimeScalingButton\_3Pushed(app, event)

value = app.Slider.Value;

n\_1 = app.Slider.Value;

value = app.Slider2.Value;

n\_2 = app.Slider2.Value;

n1 = app.Slider2.Value

n=n\_1:n\_2; % define index n

ramp=n.\*(n>=0); % define a ramp

n2 = n1\*(n\*ramp)

stem(app.UIAxes , n2,ramp)

end

% Callback function

function TimeFoldingButton\_3Pushed(app, event)

value = app.Slider.Value;

n\_1 = app.Slider.Value;

value = app.Slider2.Value;

n\_2 = app.Slider2.Value;

n1 = app.Slider2.Value

n=n\_1:n\_2; % define index n

ramp=n.\*(n>=0); % define a ramp

n2 = n1\*(-n\*ramp)

stem(app.UIAxes , n2,ramp)

end

% Button pushed function: AmplitudeScalingButton\_10

function AmplitudeScalingButton\_10Pushed(app, event)

value = app.Slider.Value;

n\_1 = app.Slider.Value;

value = app.Slider2.Value;

n\_2 = app.Slider2.Value;

n=n\_1:n\_2; % specify index n

step\_sig=(n>=0); % define the unit step sequence

x2 = n\_1\*step\_sig

stem(app.UIAxes , n, x2 ) % plot the unit step sequence

end

% Button pushed function: TimeShiftingcompButton\_7

function TimeShiftingcompButton\_7Pushed(app, event)

value = app.Slider.Value;

n\_1 = app.Slider.Value;

value = app.Slider2.Value;

n\_2 = app.Slider2.Value;

n=n\_1:n\_2; % specify index n

step\_sig=(n>=0); % define the unit step sequence

x2 = step\_sig(step\_sig-n\_1)

stem(app.UIAxes , n, x2 ) % plot the unit step sequence

end

% Button pushed function: TimeShiftingadvButton\_5

function TimeShiftingadvButton\_5Pushed(app, event)

value = app.Slider.Value;

n\_1 = app.Slider.Value;

value = app.Slider2.Value;

n\_2 = app.Slider2.Value;

n=n\_1:n\_2; % specify index n

step\_sig=(n>=0); % define the unit step sequence

x2 = n(step\_sig+n\_1)

stem(app.UIAxes , n, x2 ) % plot the unit step sequence

end

% Button pushed function: AmplitudeScalingButton\_11

function AmplitudeScalingButton\_11Pushed(app, event)

value = app.Slider.Value;

n\_1 = app.Slider.Value;

value = app.Slider2.Value;

n\_2 = app.Slider2.Value;

n=n\_1:n\_2; % specify index n

delta=(n==0); % define the delta sequence

x1 = delta\*n\_1

stem(app.UIAxes , n,x1) % plot the delta sequence

end

% Button pushed function: TimeShiftingButton

function TimeShiftingButtonPushed(app, event)

value = app.Slider.Value;

n\_1 = app.Slider.Value;

value = app.Slider2.Value;

n\_2 = app.Slider2.Value;

n=n\_1:n\_2; % specify index n

delta=(n==0); % define the delta sequence

n2 = n\_1\*delta+n

stem(app.UIAxes , n,n2) % plot the delta sequence

end

% Button pushed function: TimeShiftingadvButton\_6

function TimeShiftingadvButton\_6Pushed(app, event)

value = app.Slider.Value;

n\_1 = app.Slider.Value;

value = app.Slider2.Value;

n\_2 = app.Slider2.Value;

n=n\_1:n\_2; % specify index n

delta=(n==0); % define the delta sequence

x2 = n(delta+n\_1)

stem(app.UIAxes , n,x2) % plot the delta sequence

end

end

% Component initialization

methods (Access = private)

% Create UIFigure and components

function createComponents(app)

% Create UIFigure and hide until all components are created

app.UIFigure = uifigure('Visible', 'off');

app.UIFigure.AutoResizeChildren = 'off';

app.UIFigure.Position = [100 100 1226 725];

app.UIFigure.Name = 'UI Figure';

app.UIFigure.SizeChangedFcn = createCallbackFcn(app, @updateAppLayout, true);

% Create GridLayout

app.GridLayout = uigridlayout(app.UIFigure);

app.GridLayout.ColumnWidth = {215, '1x', 242};

app.GridLayout.RowHeight = {'1x'};

app.GridLayout.ColumnSpacing = 0;

app.GridLayout.RowSpacing = 0;

app.GridLayout.Padding = [0 0 0 0];

app.GridLayout.Scrollable = 'on';

% Create LeftPanel

app.LeftPanel = uipanel(app.GridLayout);

app.LeftPanel.Layout.Row = 1;

app.LeftPanel.Layout.Column = 1;

% Create ContinousTimeLabel\_2

app.ContinousTimeLabel\_2 = uilabel(app.LeftPanel);

app.ContinousTimeLabel\_2.FontSize = 14;

app.ContinousTimeLabel\_2.Position = [52 692 103 22];

app.ContinousTimeLabel\_2.Text = 'Continous Time';

% Create SinusoidalWaveLabel

app.SinusoidalWaveLabel = uilabel(app.LeftPanel);

app.SinusoidalWaveLabel.HorizontalAlignment = 'center';

app.SinusoidalWaveLabel.FontSize = 14;

app.SinusoidalWaveLabel.Position = [42 552 109 37];

app.SinusoidalWaveLabel.Text = 'Sinusoidal Wave';

% Create FSEditFieldLabel

app.FSEditFieldLabel = uilabel(app.LeftPanel);

app.FSEditFieldLabel.HorizontalAlignment = 'right';

app.FSEditFieldLabel.Position = [8 519 25 22];

app.FSEditFieldLabel.Text = 'FS';

% Create FSEditField

app.FSEditField = uieditfield(app.LeftPanel, 'numeric');

app.FSEditField.ValueChangedFcn = createCallbackFcn(app, @FSEditFieldValueChanged, true);

app.FSEditField.Position = [48 519 100 22];

% Create PlotButton

app.PlotButton = uibutton(app.LeftPanel, 'push');

app.PlotButton.ButtonPushedFcn = createCallbackFcn(app, @PlotButtonPushed, true);

app.PlotButton.Position = [154 518 54 23];

app.PlotButton.Text = 'Plot';

% Create CosineWaveLabel

app.CosineWaveLabel = uilabel(app.LeftPanel);

app.CosineWaveLabel.FontSize = 14;

app.CosineWaveLabel.Position = [47 360 109 27];

app.CosineWaveLabel.Text = 'Cosine Wave';

% Create FSEditField\_2Label

app.FSEditField\_2Label = uilabel(app.LeftPanel);

app.FSEditField\_2Label.HorizontalAlignment = 'right';

app.FSEditField\_2Label.Position = [7 328 25 22];

app.FSEditField\_2Label.Text = 'FS';

% Create FSEditField\_2

app.FSEditField\_2 = uieditfield(app.LeftPanel, 'numeric');

app.FSEditField\_2.ValueChangedFcn = createCallbackFcn(app, @FSEditField\_2ValueChanged, true);

app.FSEditField\_2.Position = [37 328 106 22];

% Create GenerateButton

app.GenerateButton = uibutton(app.LeftPanel, 'push');

app.GenerateButton.ButtonPushedFcn = createCallbackFcn(app, @GenerateButtonPushed, true);

app.GenerateButton.Position = [144 328 61 22];

app.GenerateButton.Text = 'Generate';

% Create PhaseChangeButton

app.PhaseChangeButton = uibutton(app.LeftPanel, 'push');

app.PhaseChangeButton.ButtonPushedFcn = createCallbackFcn(app, @PhaseChangeButtonPushed, true);

app.PhaseChangeButton.Position = [17 466 164 22];

app.PhaseChangeButton.Text = 'Phase Change';

% Create FreqChangeButton

app.FreqChangeButton = uibutton(app.LeftPanel, 'push');

app.FreqChangeButton.ButtonPushedFcn = createCallbackFcn(app, @FreqChangeButtonPushed, true);

app.FreqChangeButton.Position = [48 397 123 23];

app.FreqChangeButton.Text = 'Freq Change';

% Create AmplitudeScalingButton\_4

app.AmplitudeScalingButton\_4 = uibutton(app.LeftPanel, 'push');

app.AmplitudeScalingButton\_4.ButtonPushedFcn = createCallbackFcn(app, @AmplitudeScalingButton\_4Pushed, true);

app.AmplitudeScalingButton\_4.Position = [52 437 111 22];

app.AmplitudeScalingButton\_4.Text = 'Amplitude Scaling';

% Create AmplitudeScalingButton\_5

app.AmplitudeScalingButton\_5 = uibutton(app.LeftPanel, 'push');

app.AmplitudeScalingButton\_5.ButtonPushedFcn = createCallbackFcn(app, @AmplitudeScalingButton\_5Pushed, true);

app.AmplitudeScalingButton\_5.Position = [12 262 98 22];

app.AmplitudeScalingButton\_5.Text = 'Amplitude Scaling';

% Create FreqchangeButton

app.FreqchangeButton = uibutton(app.LeftPanel, 'push');

app.FreqchangeButton.ButtonPushedFcn = createCallbackFcn(app, @FreqchangeButtonPushed, true);

app.FreqchangeButton.Position = [110 260 100 22];

app.FreqchangeButton.Text = 'Freq.change';

% Create PhasechangeButton

app.PhasechangeButton = uibutton(app.LeftPanel, 'push');

app.PhasechangeButton.Position = [8 214 98 22];

app.PhasechangeButton.Text = 'Phase change';

% Create OperationsLabel\_3

app.OperationsLabel\_3 = uilabel(app.LeftPanel);

app.OperationsLabel\_3.Position = [61 283 65 32];

app.OperationsLabel\_3.Text = 'Operations';

% Create OperationsLabel\_2

app.OperationsLabel\_2 = uilabel(app.LeftPanel);

app.OperationsLabel\_2.Position = [62 488 65 22];

app.OperationsLabel\_2.Text = 'Operations';

% Create ContinousTimeLabel

app.ContinousTimeLabel = uilabel(app.LeftPanel);

app.ContinousTimeLabel.HorizontalAlignment = 'center';

app.ContinousTimeLabel.FontSize = 14;

app.ContinousTimeLabel.Position = [20 182 173 24];

app.ContinousTimeLabel.Text = 'Continous Time';

% Create triangularWaveButton

app.triangularWaveButton = uibutton(app.LeftPanel, 'push');

app.triangularWaveButton.ButtonPushedFcn = createCallbackFcn(app, @triangularWaveButtonPushed, true);

app.triangularWaveButton.Position = [9 148 188 22];

app.triangularWaveButton.Text = 'triangular Wave';

% Create AmplitudeScalingButton

app.AmplitudeScalingButton = uibutton(app.LeftPanel, 'push');

app.AmplitudeScalingButton.ButtonPushedFcn = createCallbackFcn(app, @AmplitudeScalingButtonPushed, true);

app.AmplitudeScalingButton.Position = [12 107 99 22];

app.AmplitudeScalingButton.Text = 'Amplitude Scaling';

% Create TimeShiftingcompButton

app.TimeShiftingcompButton = uibutton(app.LeftPanel, 'push');

app.TimeShiftingcompButton.ButtonPushedFcn = createCallbackFcn(app, @TimeShiftingcompButtonPushed, true);

app.TimeShiftingcompButton.Position = [110 102 99 22];

app.TimeShiftingcompButton.Text = 'Time Shifting(comp.)';

% Create TimeShiftingAdvanceButton

app.TimeShiftingAdvanceButton = uibutton(app.LeftPanel, 'push');

app.TimeShiftingAdvanceButton.ButtonPushedFcn = createCallbackFcn(app, @TimeShiftingAdvanceButtonPushed, true);

app.TimeShiftingAdvanceButton.Position = [9 60 99 22];

app.TimeShiftingAdvanceButton.Text = 'Time Shifting Advance';

% Create TimeFoldingButton

app.TimeFoldingButton = uibutton(app.LeftPanel, 'push');

app.TimeFoldingButton.ButtonPushedFcn = createCallbackFcn(app, @TimeFoldingButtonPushed, true);

app.TimeFoldingButton.Position = [17 18 99 22];

app.TimeFoldingButton.Text = 'Time Folding';

% Create ShiftingScalingButton

app.ShiftingScalingButton = uibutton(app.LeftPanel, 'push');

app.ShiftingScalingButton.ButtonPushedFcn = createCallbackFcn(app, @ShiftingScalingButtonPushed, true);

app.ShiftingScalingButton.Position = [115 18 99 22];

app.ShiftingScalingButton.Text = 'Shifting & Scaling';

% Create SliderLabel

app.SliderLabel = uilabel(app.LeftPanel);

app.SliderLabel.HorizontalAlignment = 'right';

app.SliderLabel.Position = [7 653 36 22];

app.SliderLabel.Text = 'Slider';

% Create Slider

app.Slider = uislider(app.LeftPanel);

app.Slider.Limits = [-10 10];

app.Slider.ValueChangedFcn = createCallbackFcn(app, @SliderValueChanged, true);

app.Slider.Position = [64 662 133 3];

% Create TimeScalingButton

app.TimeScalingButton = uibutton(app.LeftPanel, 'push');

app.TimeScalingButton.ButtonPushedFcn = createCallbackFcn(app, @TimeScalingButtonPushed, true);

app.TimeScalingButton.Position = [111 60 99 22];

app.TimeScalingButton.Text = 'Time Scaling';

% Create CenterPanel

app.CenterPanel = uipanel(app.GridLayout);

app.CenterPanel.Layout.Row = 1;

app.CenterPanel.Layout.Column = 2;

% Create UIAxes

app.UIAxes = uiaxes(app.CenterPanel);

title(app.UIAxes, 'Signal Operations')

xlabel(app.UIAxes, 'X')

ylabel(app.UIAxes, 'Y')

app.UIAxes.PlotBoxAspectRatio = [1 1.27560975609756 1];

app.UIAxes.Position = [7 123 757 595];

% Create AmplitudeScalingButton\_6

app.AmplitudeScalingButton\_6 = uibutton(app.CenterPanel, 'push');

app.AmplitudeScalingButton\_6.ButtonPushedFcn = createCallbackFcn(app, @AmplitudeScalingButton\_6Pushed, true);

app.AmplitudeScalingButton\_6.Position = [100 58 112 22];

app.AmplitudeScalingButton\_6.Text = 'Amplitude Scaling';

% Create TimeShiftingcompButton\_4

app.TimeShiftingcompButton\_4 = uibutton(app.CenterPanel, 'push');

app.TimeShiftingcompButton\_4.ButtonPushedFcn = createCallbackFcn(app, @TimeShiftingcompButton\_4Pushed, true);

app.TimeShiftingcompButton\_4.Position = [477 60 112 22];

app.TimeShiftingcompButton\_4.Text = 'Time Shifting(comp.)';

% Create TimeScalingButton\_4

app.TimeScalingButton\_4 = uibutton(app.CenterPanel, 'push');

app.TimeScalingButton\_4.Position = [231 58 100 22];

app.TimeScalingButton\_4.Text = 'Time Scaling';

% Create TimeShiftingadvButton\_2

app.TimeShiftingadvButton\_2 = uibutton(app.CenterPanel, 'push');

app.TimeShiftingadvButton\_2.Position = [345 60 105 22];

app.TimeShiftingadvButton\_2.Text = 'Time Shifting(adv.)';

% Create TimeFoldingButton\_4

app.TimeFoldingButton\_4 = uibutton(app.CenterPanel, 'push');

app.TimeFoldingButton\_4.Position = [615 60 100 22];

app.TimeFoldingButton\_4.Text = 'Time Folding';

% Create CreateasignalButton

app.CreateasignalButton = uibutton(app.CenterPanel, 'push');

app.CreateasignalButton.ButtonPushedFcn = createCallbackFcn(app, @CreateasignalButtonPushed, true);

app.CreateasignalButton.Position = [588 18 153 31];

app.CreateasignalButton.Text = 'Create a signal';

% Create GenerateyourdesiredsignalLabel

app.GenerateyourdesiredsignalLabel = uilabel(app.CenterPanel);

app.GenerateyourdesiredsignalLabel.Position = [279 102 182 22];

app.GenerateyourdesiredsignalLabel.Text = 'Generate your desired signal';

% Create SelectFIleButton

app.SelectFIleButton = uibutton(app.CenterPanel, 'push');

app.SelectFIleButton.ButtonPushedFcn = createCallbackFcn(app, @SelectFIleButtonPushed, true);

app.SelectFIleButton.Position = [15 7 197 31];

app.SelectFIleButton.Text = 'Select FIle';

% Create ClcClearallButton

app.ClcClearallButton = uibutton(app.CenterPanel, 'push');

app.ClcClearallButton.ButtonPushedFcn = createCallbackFcn(app, @ClcClearallButtonPushed, true);

app.ClcClearallButton.Position = [15 93 100 22];

app.ClcClearallButton.Text = 'Clc/Clear all';

% Create RightPanel

app.RightPanel = uipanel(app.GridLayout);

app.RightPanel.Layout.Row = 1;

app.RightPanel.Layout.Column = 3;

% Create RampButton

app.RampButton = uibutton(app.RightPanel, 'push');

app.RampButton.ButtonPushedFcn = createCallbackFcn(app, @RampButtonPushed, true);

app.RampButton.Position = [17 657 100 22];

app.RampButton.Text = 'Ramp';

% Create UnitStepButton

app.UnitStepButton = uibutton(app.RightPanel, 'push');

app.UnitStepButton.ButtonPushedFcn = createCallbackFcn(app, @UnitStepButtonPushed, true);

app.UnitStepButton.Position = [137 657 100 22];

app.UnitStepButton.Text = 'Unit Step';

% Create ImpulseButton

app.ImpulseButton = uibutton(app.RightPanel, 'push');

app.ImpulseButton.ButtonPushedFcn = createCallbackFcn(app, @ImpulseButtonPushed, true);

app.ImpulseButton.Position = [17 617 100 22];

app.ImpulseButton.Text = 'Impulse';

% Create Slider2Label

app.Slider2Label = uilabel(app.RightPanel);

app.Slider2Label.HorizontalAlignment = 'right';

app.Slider2Label.Position = [15 575 43 22];

app.Slider2Label.Text = 'Slider2';

% Create Slider2

app.Slider2 = uislider(app.RightPanel);

app.Slider2.Limits = [-10 10];

app.Slider2.ValueChangedFcn = createCallbackFcn(app, @Slider2ValueChanged, true);

app.Slider2.Position = [79 584 150 3];

% Create DiscreteTimeLabel

app.DiscreteTimeLabel = uilabel(app.RightPanel);

app.DiscreteTimeLabel.FontSize = 14;

app.DiscreteTimeLabel.Position = [95 689 136 29];

app.DiscreteTimeLabel.Text = 'Discrete Time';

% Create AmplitudeScalingButton\_3

app.AmplitudeScalingButton\_3 = uibutton(app.RightPanel, 'push');

app.AmplitudeScalingButton\_3.ButtonPushedFcn = createCallbackFcn(app, @AmplitudeScalingButton\_3Pushed, true);

app.AmplitudeScalingButton\_3.Position = [7 513 109 22];

app.AmplitudeScalingButton\_3.Text = 'Amplitude Scaling';

% Create TimeShiftingcompButton\_3

app.TimeShiftingcompButton\_3 = uibutton(app.RightPanel, 'push');

app.TimeShiftingcompButton\_3.ButtonPushedFcn = createCallbackFcn(app, @TimeShiftingcompButton\_3Pushed, true);

app.TimeShiftingcompButton\_3.Position = [130 513 106 22];

app.TimeShiftingcompButton\_3.Text = 'Time Shifting(comp.)';

% Create TimeShiftingadvButton

app.TimeShiftingadvButton = uibutton(app.RightPanel, 'push');

app.TimeShiftingadvButton.ButtonPushedFcn = createCallbackFcn(app, @TimeShiftingadvButtonPushed2, true);

app.TimeShiftingadvButton.Position = [7 470 115 22];

app.TimeShiftingadvButton.Text = 'Time Shifting(adv.)';

% Create AmplitudeScalingButton\_10

app.AmplitudeScalingButton\_10 = uibutton(app.RightPanel, 'push');

app.AmplitudeScalingButton\_10.ButtonPushedFcn = createCallbackFcn(app, @AmplitudeScalingButton\_10Pushed, true);

app.AmplitudeScalingButton\_10.Position = [2 397 101 22];

app.AmplitudeScalingButton\_10.Text = 'Amplitude Scaling';

% Create TimeShiftingadvButton\_5

app.TimeShiftingadvButton\_5 = uibutton(app.RightPanel, 'push');

app.TimeShiftingadvButton\_5.ButtonPushedFcn = createCallbackFcn(app, @TimeShiftingadvButton\_5Pushed, true);

app.TimeShiftingadvButton\_5.Position = [5 362 96 22];

app.TimeShiftingadvButton\_5.Text = 'Time Shifting(adv.)';

% Create TimeShiftingcompButton\_7

app.TimeShiftingcompButton\_7 = uibutton(app.RightPanel, 'push');

app.TimeShiftingcompButton\_7.ButtonPushedFcn = createCallbackFcn(app, @TimeShiftingcompButton\_7Pushed, true);

app.TimeShiftingcompButton\_7.Position = [125 397 100 22];

app.TimeShiftingcompButton\_7.Text = 'Time Shifting(comp.)';

% Create TimeShiftingButton

app.TimeShiftingButton = uibutton(app.RightPanel, 'push');

app.TimeShiftingButton.ButtonPushedFcn = createCallbackFcn(app, @TimeShiftingButtonPushed, true);

app.TimeShiftingButton.Position = [131 260 100 22];

app.TimeShiftingButton.Text = 'Time Shifting';

% Create AmplitudeScalingButton\_11

app.AmplitudeScalingButton\_11 = uibutton(app.RightPanel, 'push');

app.AmplitudeScalingButton\_11.ButtonPushedFcn = createCallbackFcn(app, @AmplitudeScalingButton\_11Pushed, true);

app.AmplitudeScalingButton\_11.Position = [5 262 102 22];

app.AmplitudeScalingButton\_11.Text = 'Amplitude Scaling';

% Create TimeShiftingadvButton\_6

app.TimeShiftingadvButton\_6 = uibutton(app.RightPanel, 'push');

app.TimeShiftingadvButton\_6.ButtonPushedFcn = createCallbackFcn(app, @TimeShiftingadvButton\_6Pushed, true);

app.TimeShiftingadvButton\_6.Position = [8 228 108 22];

app.TimeShiftingadvButton\_6.Text = 'Time Shifting(adv.)';

% Create OperationsUnitStepLabel

app.OperationsUnitStepLabel = uilabel(app.RightPanel);

app.OperationsUnitStepLabel.Position = [69 429 122 22];

app.OperationsUnitStepLabel.Text = 'Operations(Unit Step)';

% Create OperationsImpulseLabel\_2

app.OperationsImpulseLabel\_2 = uilabel(app.RightPanel);

app.OperationsImpulseLabel\_2.Position = [69 293 114 22];

app.OperationsImpulseLabel\_2.Text = 'Operations(Impulse)';

% Create Slider3Label

app.Slider3Label = uilabel(app.RightPanel);

app.Slider3Label.HorizontalAlignment = 'right';

app.Slider3Label.Position = [5 184 43 22];

app.Slider3Label.Text = 'Slider3';

% Create Slider3

app.Slider3 = uislider(app.RightPanel);

app.Slider3.Limits = [-10 10];

app.Slider3.Position = [69 193 150 3];

% Create RampSignalButton

app.RampSignalButton = uibutton(app.RightPanel, 'push');

app.RampSignalButton.ButtonPushedFcn = createCallbackFcn(app, @RampSignalButtonPushed, true);

app.RampSignalButton.Position = [5 102 175 22];

app.RampSignalButton.Text = 'Ramp Signal';

% Create UnitStepSignalButton

app.UnitStepSignalButton = uibutton(app.RightPanel, 'push');

app.UnitStepSignalButton.ButtonPushedFcn = createCallbackFcn(app, @UnitStepSignalButtonPushed, true);

app.UnitStepSignalButton.Position = [7 72 174 22];

app.UnitStepSignalButton.Text = 'Unit Step Signal';

% Create ImpulseSignalButton

app.ImpulseSignalButton = uibutton(app.RightPanel, 'push');

app.ImpulseSignalButton.ButtonPushedFcn = createCallbackFcn(app, @ImpulseSignalButtonPushed, true);

app.ImpulseSignalButton.Position = [7 37 164 22];

app.ImpulseSignalButton.Text = 'Impulse Signal';

% Show the figure after all components are created

app.UIFigure.Visible = 'on';

end

end

% App creation and deletion

methods (Access = public)

% Construct app

function app = app2

% Create UIFigure and components

createComponents(app)

% Register the app with App Designer

registerApp(app, app.UIFigure)

% Execute the startup function

runStartupFcn(app, @startupFcn)

if nargout == 0

clear app

end

end

% Code that executes before app deletion

function delete(app)

% Delete UIFigure when app is deleted

delete(app.UIFigure)

end

end

end

**Applications:**

Signal Operations applications include audio and speech processing, sonar, radar and other sensor array processing, spectral density estimation, statistical signal processing, digital image processing, data compression, video coding, audio coding, image compression, signal processing for telecommunications, control systems, biomedical engineering, and seismology, among others. Digital signal processing and analog signal processing are subfields of signal processing.

The application of digital computation to signal processing allows for many advantages over analog processing in many applications, such as error detection and correction in transmission as well as data compression. Digital signal processing is also fundamental to digital technology, such as digital telecommunication and wireless communications

1. Time-shifting is an important operation that is used in many signal-processing applications. For example, a time-delayed version of the signal is used when performing autocorrelation.
2. Basically, when we perform time scaling, we change the rate at which the signal is sampled. Changing the sampling rate of a signal is employed in the field of speech processing. A particular example of this would be a time-scaling-algorithm-based system developed to read text to the visually impaired.
3. Time reversal is an important preliminary step when computing the convolution of signals: one signal is kept in its original state while the other is mirror-image and slid along the former signal to obtain the result. Time-reversal operations, therefore, are useful in various image-processing procedures, such as edge detection.

**Results:**

**Conclusion:**

This app presents an analysis of the above performed operations on signals: addition, multiplication, amplitude scaling, time shifting, time scaling, and time reversal. In addition, we briefly went over practical applications for these operations.

**References:**

1. http://electrical4u.com/basicsignaloperationshttp:/
2. www.tutorialspoint.com/signals\_and\_systems/signals\_basic\_operations.htm
3. http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-01sc-introduction-to-electrical-engineering-and-computer-science-i-spring-2011/unit-2-signals-and-systems/designing-control-systems/MIT6\_01SCS11\_chap05.pdf
4. http://in.mathworks.com/
5. https://www.allaboutcircuits.com/technical-articles/basic-signal-operations-in-dsp-time-shifting-time-scaling-and-time-reversal/