School of Electronics and Communication Engineering
Academic Year 2020 - 2021

Mini Project Report - Trimester VI

Natural Sampling Theorem

Analog Communication

Student details:

Sr. No	PRN	Student Name	Contact No	Email Id
1	1032191005	Rohan Habu	8983117939	haburohan@gmail.com
2	1032191053	Mehul Verma	8454097831	mehul2001.mit@gmail.com
3	1032191177	Vishal Singh	8287676489	vs978886@gmail.com

Circuit Diagram

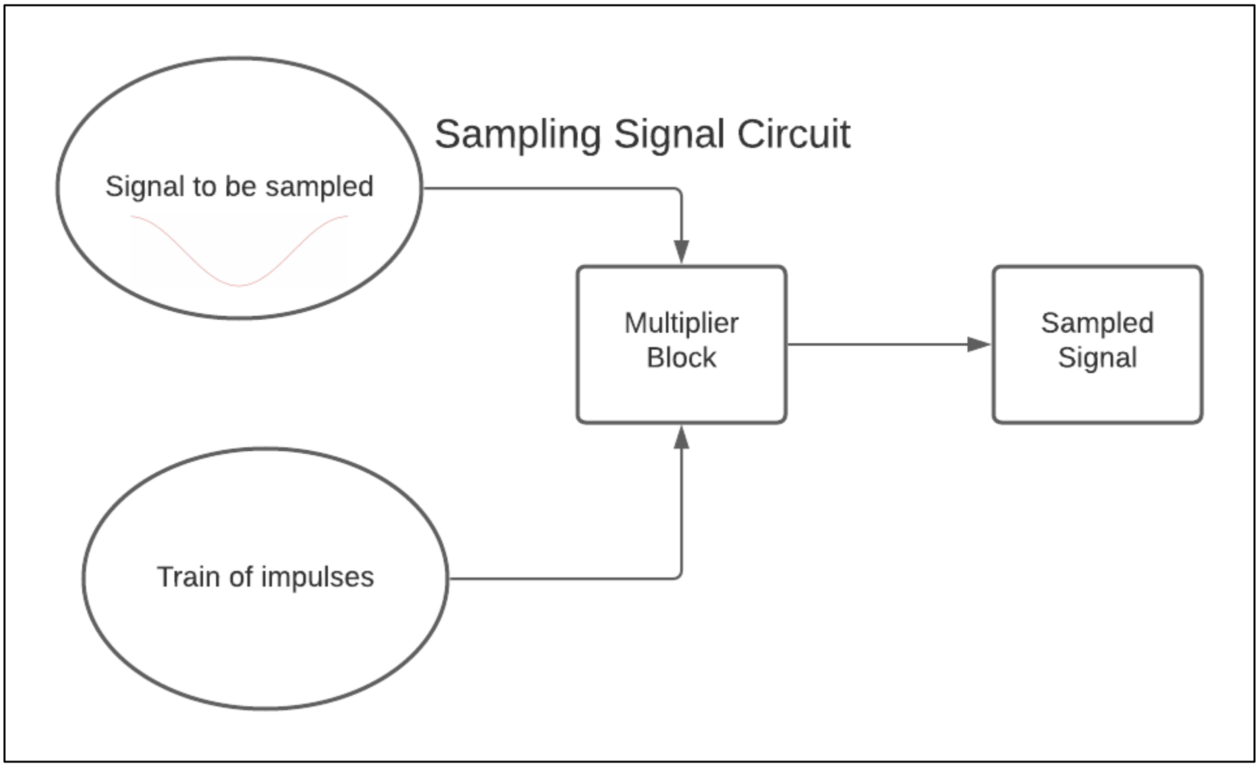
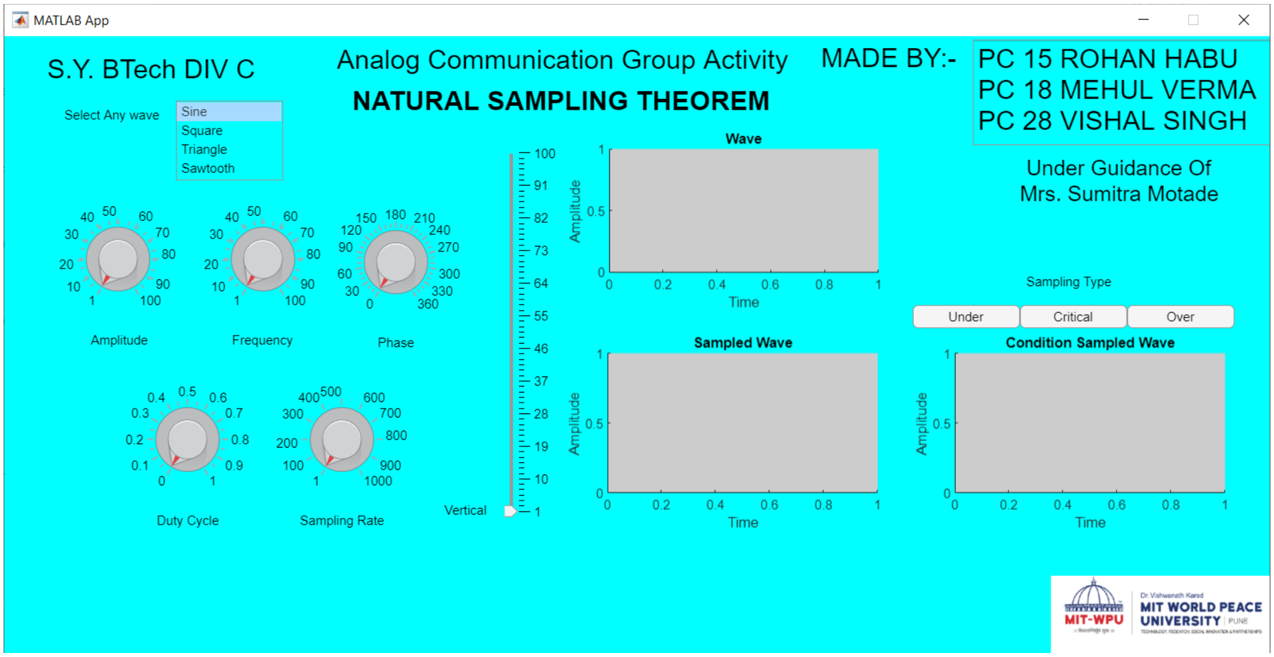


Figure 1: Sampling Circuit

GUI Layout Screenshot



S.Y. BTech DIV C

Analog Communication Group Activity

NATURAL SAMPLING THEOREM

MADE BY:-

PC 15 ROHAN HABU
PC 18 MEHUL VERMA
PC 28 VISHAL SINGH

Under Guidance Of
Mrs. Sumitra Motade

Introduction

- This GUI implements the concept of natural sampling process.
- Natural sampling process includes multiplication of impulse signal to the original signal to be sampled.
- The pulse width is negligible.
- Thus, at the output, straight lines called samples are received.
- There are also other techniques like sample and hold and flat top sampling.
- Practically the width can’t be zero.
- So, there is some portion present but it is unnoticeable by naked eyes.
- In this process, we take the impulse signal and its pulse width is 0% percent of the amplitude of impulse in MATLAB.
- In this manner a train of impulses can be generated at different frequencies.
- Suppose consider an example of a variable sine wave.
- It can reach up to highest amplitude of 5V and highest frequency of 10Hz.
- So, we need to generate an impulse of amplitude 5V minimum or slightly greater than this to properly sample the wave aligning on the Y Axis.
- In the train of impulses, we need to carefully select the frequency of the impulses occurring.
- According to the impulse/ sampling frequency selection, the sampling process can be distributed in to following categories:
    - Under sampling.
    - Critical sampling (Nyquist sampling).
    - Over sampling.
- Here, oversampling is the most useful case to sample the input signal.
- Critical sampling is the point at which you just get the nature of the wave that is fed in to the circuit.
- After critical sampling case, the oversampling case begins and the sampled wave gets more closer to the wave at the input.
- In under sampling, the frequency bands of the impulses overlap with each other.
- This changes the nature of the signal at the output.
- Also, due to overlapping of bands, it is common to 2 adjacent signals and due to this, a phenomenon, or an error in case of communication occurs which is called as aliasing.
- It means that a band is common for 2 signals at the same time which disturbs the nature of the signal during sampling process.
- Let fs=sampling frequency.
- fm=maximum frequency of the signal at the input.
- Under sampling condition:

  fs<fm

- Critical/Nyquist sampling:

  fs=2fm

- Oversampling condition:

  fs>>2fm

- So, to avoid aliasing effect and the disturbance in the output of the signal, oversampling method is used everywhere.

Description of the GUI

So, the GUI has ability of generating the sine, square, triangular and saw tooth types of waves. There is a choice given to the user for selecting an appropriate wave to generate and perform sampling operation on the desired sampling rate. User also can select the required amplitude, frequency, phase, and duty cycle for the waves. Also, the sampling rate can be chosen by the choice of the user. The maximum limit for amplitude, frequency and phase in the GUI is 100.

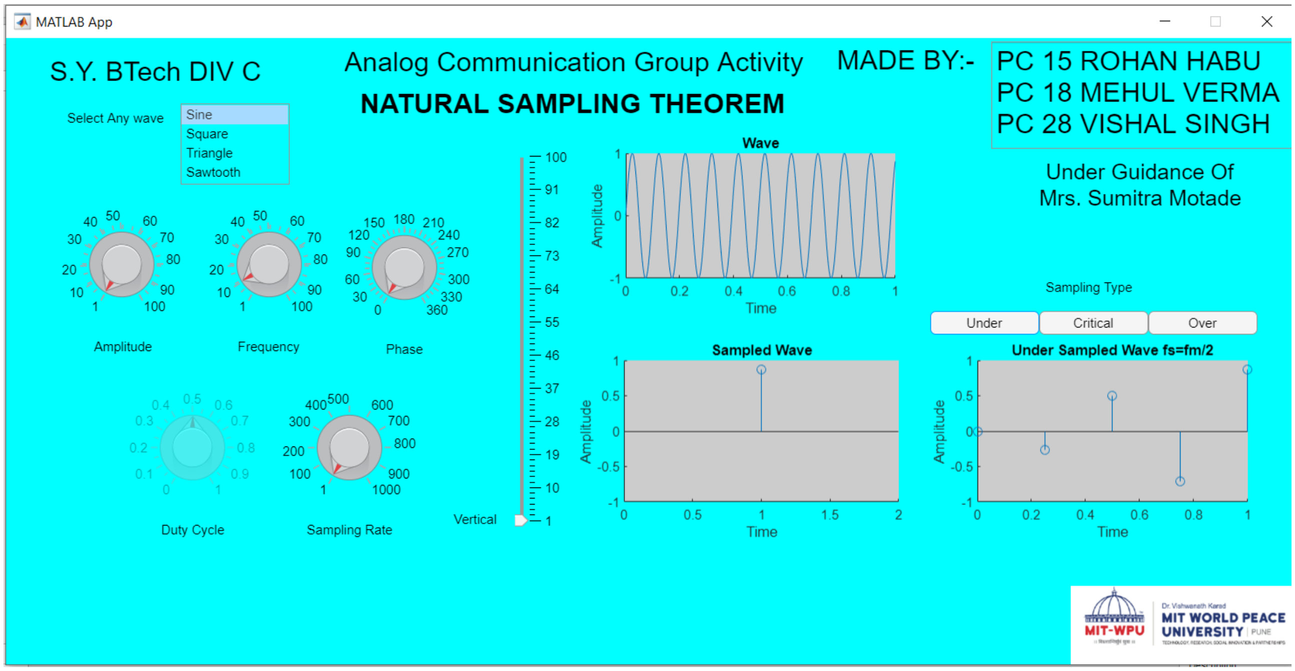
Applications

Sampling theorem has various applications in real life: -

1. Mobile phones and laptops for communication.
2. Bluetooth speakers.
3. Television broadcasting on internet.
4. To study the signal in discrete domain and remove discontinuities in the signal.
5. Used in smart wearables (smart watches, fitness bands, etc.)

OUTPUTS

SINE WAVE UNDER SAMPLED



MATLAB App

S.Y. BTech DIV C

Analog Communication Group Activity

NATURAL SAMPLING THEOREM

MADE BY:-

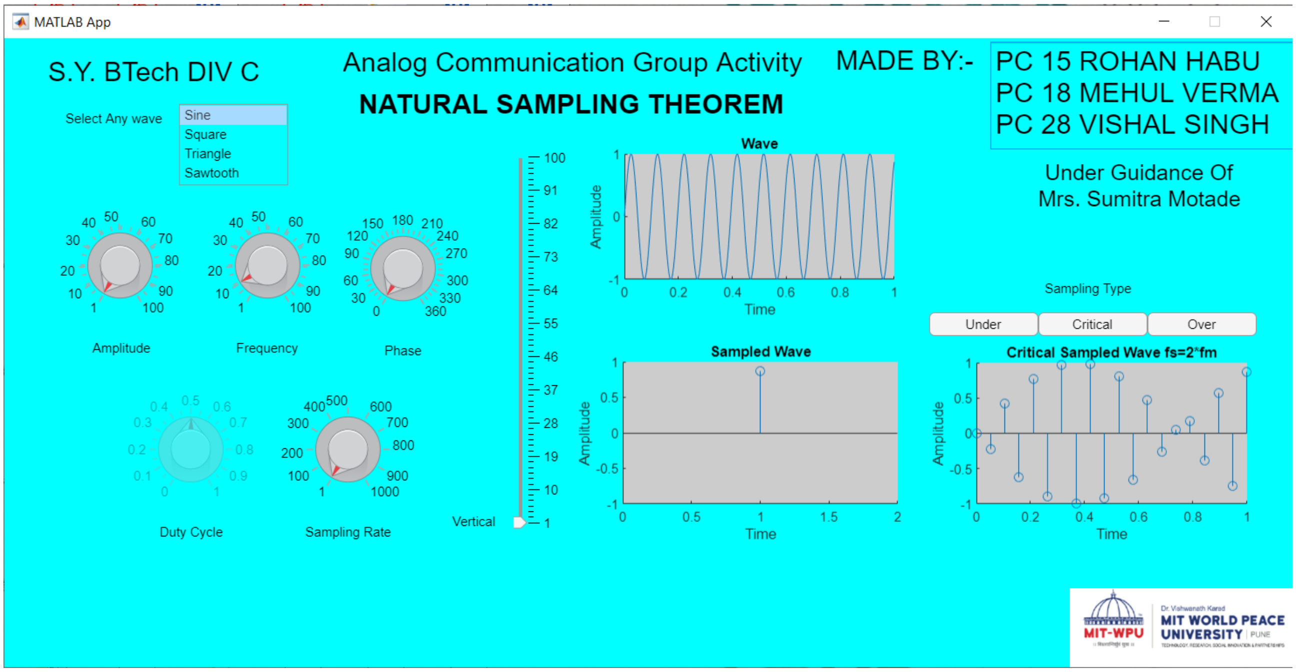
PC 15 ROHAN HABU
PC 18 MEHUL VERMA
PC 28 VISHAL SINGH

Under Guidance Of
Mrs. Sumitra Motade

Select Any wave: Sine

Sampling Type: Under

SINE WAVE CRITICALLY SAMPLED



MATLAB App

S.Y. BTech DIV C

Analog Communication Group Activity

NATURAL SAMPLING THEOREM

MADE BY:-

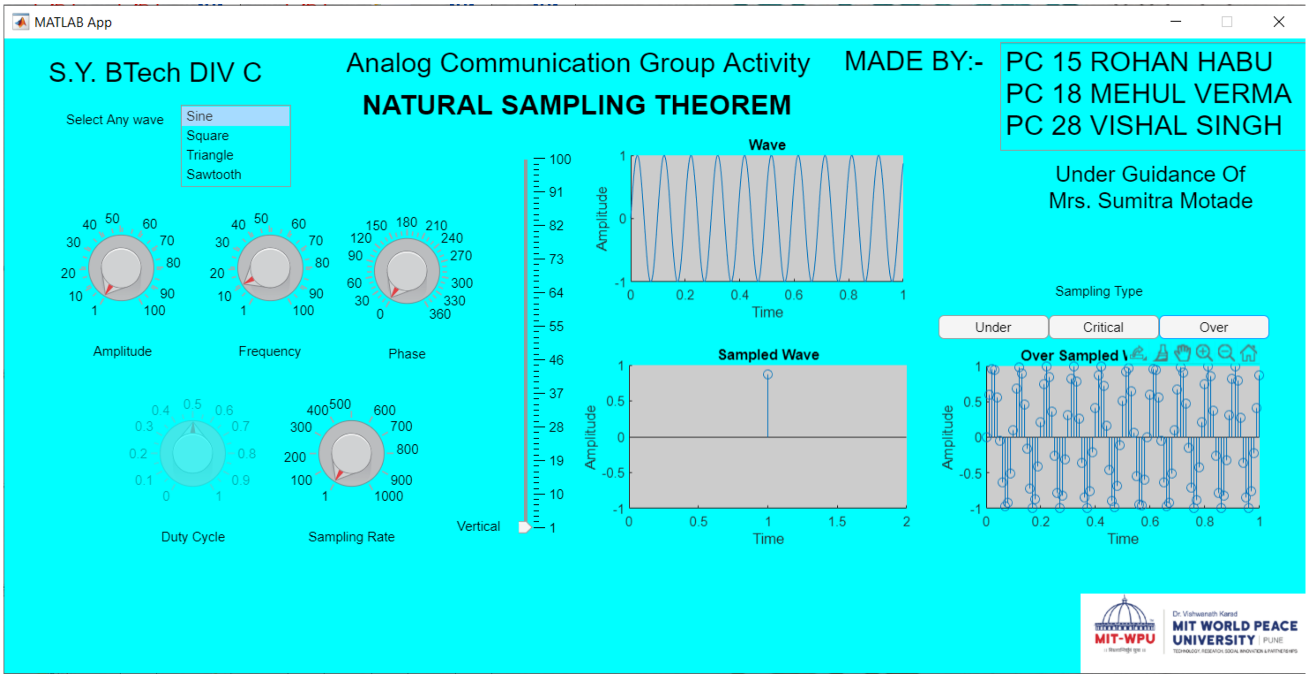
PC 15 ROHAN HABU
PC 18 MEHUL VERMA
PC 28 VISHAL SINGH

Under Guidance Of
Mrs. Sumitra Motade

Select Any wave: Sine

Sampling Type: Critical

SINE WAVE OVER SAMPLED



MATLAB App

S.Y. BTech DIV C

Analog Communication Group Activity

NATURAL SAMPLING THEOREM

MADE BY:-

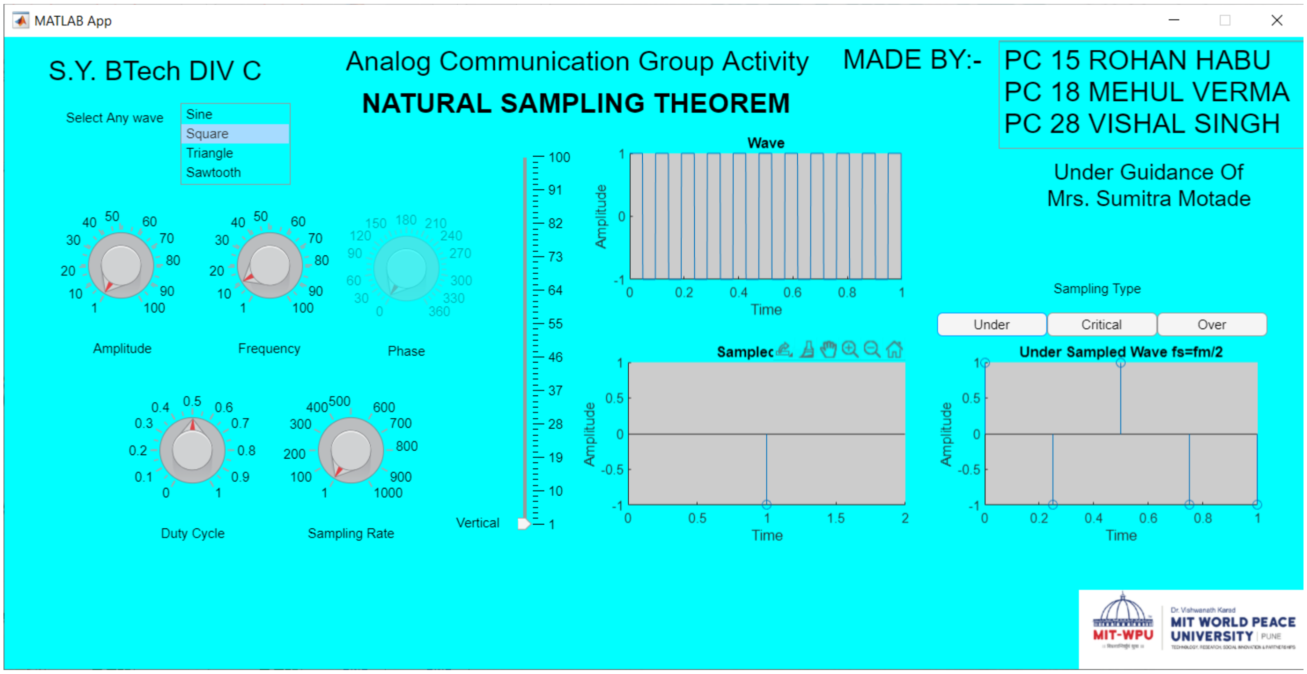
PC 15 ROHAN HABU
PC 18 MEHUL VERMA
PC 28 VISHAL SINGH

Under Guidance Of
Mrs. Sumitra Motade

Select Any wave: Sine

Sampling Type: Over

SQUARE WAVE UNDER SAMPLED



MATLAB App

S.Y. BTech DIV C

Analog Communication Group Activity

NATURAL SAMPLING THEOREM

MADE BY:-

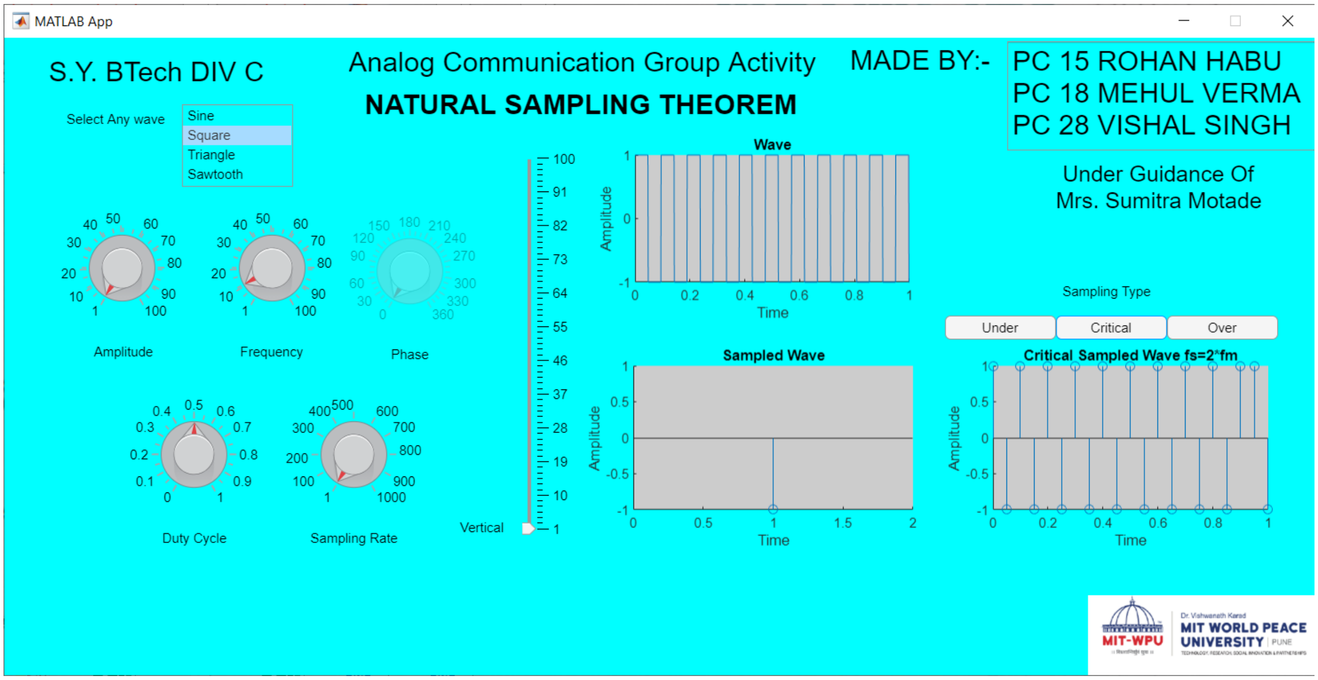
PC 15 ROHAN HABU
PC 18 MEHUL VERMA
PC 28 VISHAL SINGH

Under Guidance Of
Mrs. Sumitra Motade

Select Any wave: Square

Sampling Type: Under

SQUARE WAVE CRITICALLY SAMPLED



MATLAB App

S.Y. BTech DIV C

Analog Communication Group Activity

NATURAL SAMPLING THEOREM

MADE BY:-

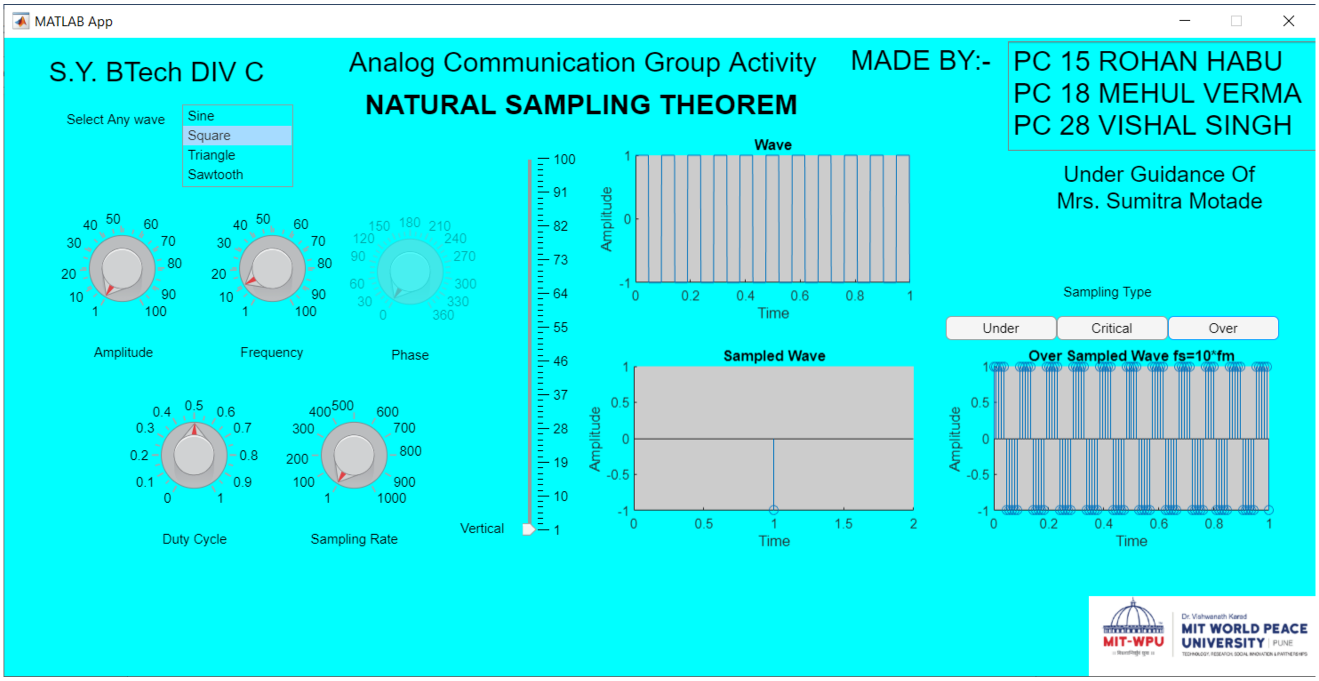
PC 15 ROHAN HABU
PC 18 MEHUL VERMA
PC 28 VISHAL SINGH

Under Guidance Of
Mrs. Sumitra Motade

Select Any wave: Sine / [✓ Square] / Triangle / Sawtooth

Sampling Type: Under / Critical / Over

SQUARE WAVE OVER SAMPLED



MATLAB App

S.Y. BTech DIV C

Analog Communication Group Activity

NATURAL SAMPLING THEOREM

MADE BY:-

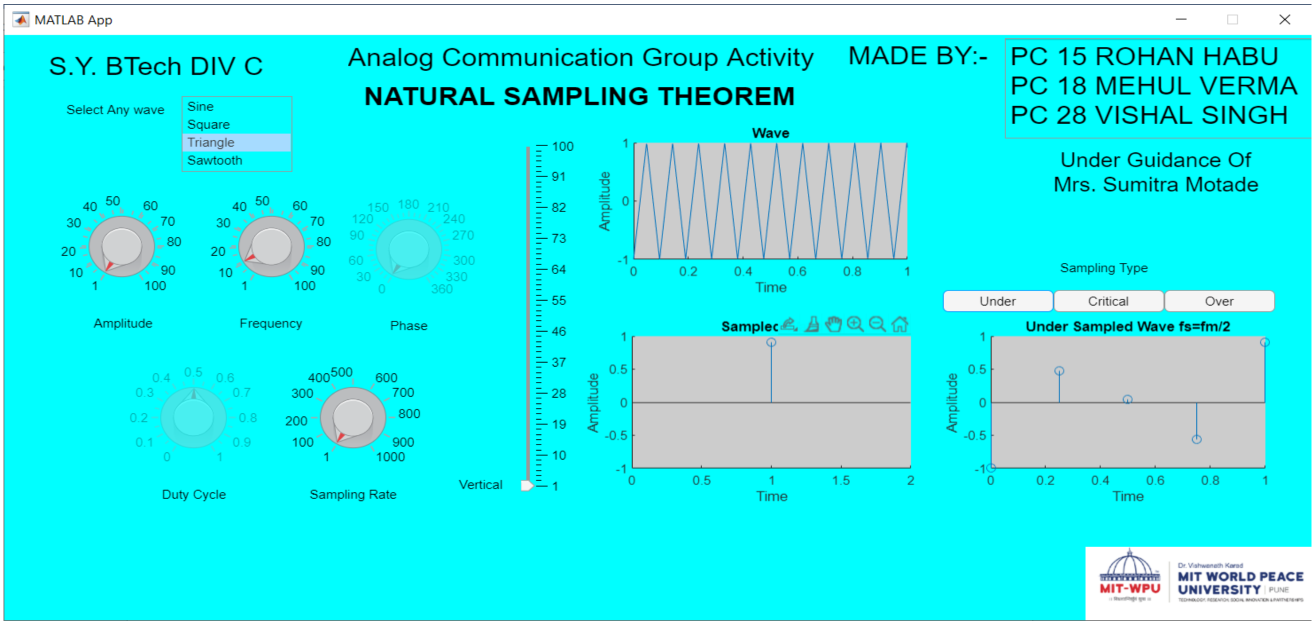
PC 15 ROHAN HABU
PC 18 MEHUL VERMA
PC 28 VISHAL SINGH

Under Guidance Of
Mrs. Sumitra Motade

Select Any wave: Sine / [✓ Square] / Triangle / Sawtooth

Sampling Type: Under / Critical / Over

TRIANGULAR WAVE UNDER SAMPLED



MATLAB App

S.Y. BTech DIV C

Analog Communication Group Activity

NATURAL SAMPLING THEOREM

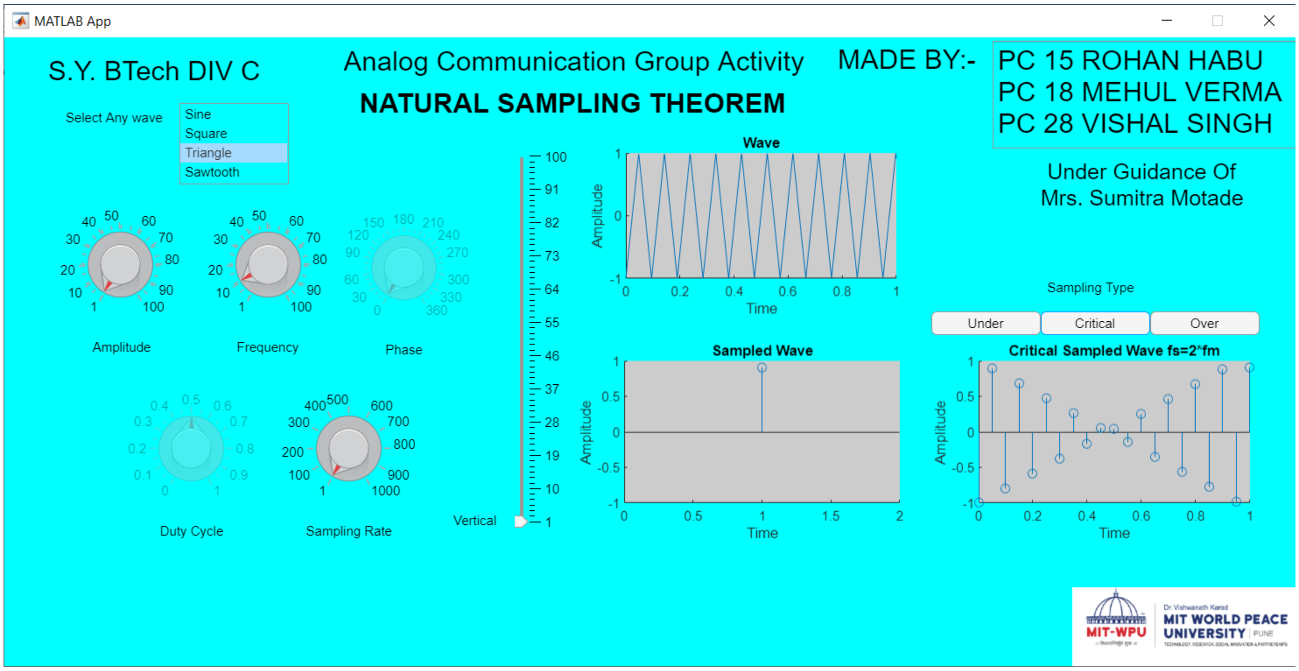
MADE BY:-

PC 15 ROHAN HABU
PC 18 MEHUL VERMA
PC 28 VISHAL SINGH

Under Guidance Of
Mrs. Sumitra Motade

Select Any wave: Triangle

TRIANGULAR WAVE CRITICALLY SAMPLED



MATLAB App

S.Y. BTech DIV C

Analog Communication Group Activity

NATURAL SAMPLING THEOREM

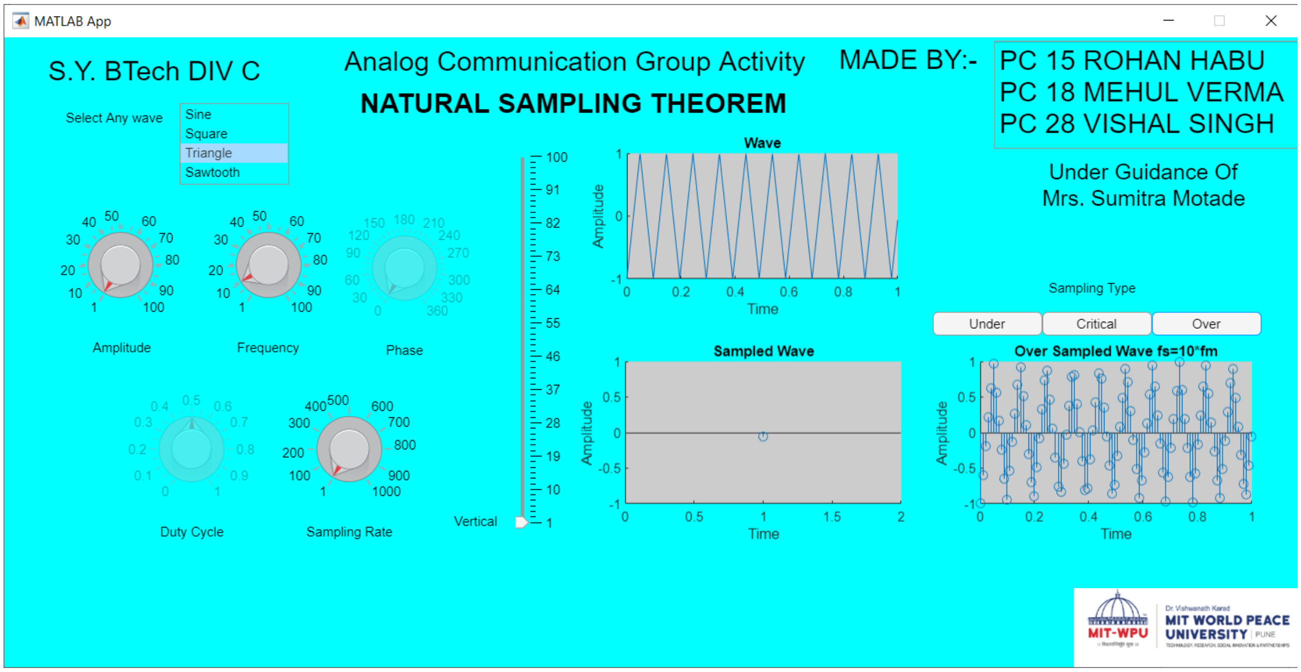
MADE BY:-

PC 15 ROHAN HABU
PC 18 MEHUL VERMA
PC 28 VISHAL SINGH

Under Guidance Of
Mrs. Sumitra Motade

Select Any wave: Triangle

TRIANGULAR WAVE OVER SAMPLED



MATLAB App

S.Y. BTech DIV C

Analog Communication Group Activity

NATURAL SAMPLING THEOREM

MADE BY:-

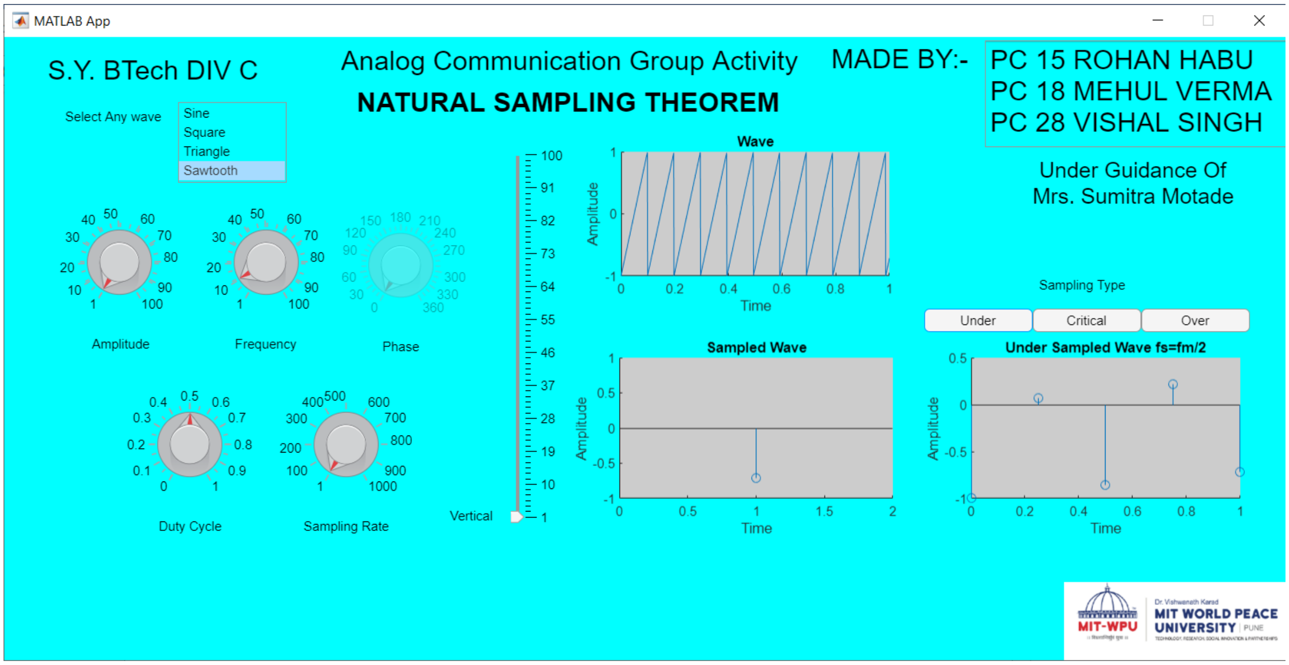
PC 15 ROHAN HABU
PC 18 MEHUL VERMA
PC 28 VISHAL SINGH

Under Guidance Of
Mrs. Sumitra Motade

Select Any wave: Sine, Square, Triangle (selected), Sawtooth

Sampling Type: Under | Critical | Over

SAW TOOTH WAVE UNDER SAMPLED



MATLAB App

S.Y. BTech DIV C

Analog Communication Group Activity

NATURAL SAMPLING THEOREM

MADE BY:-

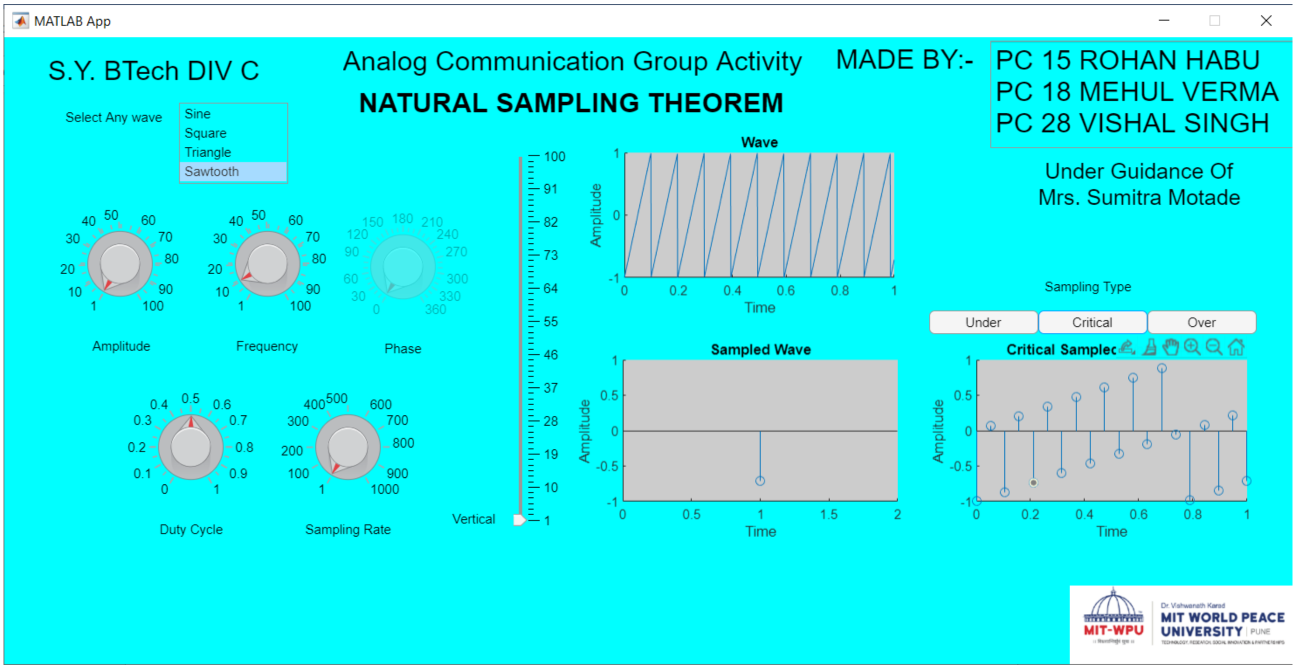
PC 15 ROHAN HABU
PC 18 MEHUL VERMA
PC 28 VISHAL SINGH

Under Guidance Of
Mrs. Sumitra Motade

Select Any wave: Sine, Square, Triangle, Sawtooth (selected)

Sampling Type: Under | Critical | Over

SAW TOOTH WAVE CRITICALLY SAMPLED



MATLAB App

S.Y. BTech DIV C

Analog Communication Group Activity

NATURAL SAMPLING THEOREM

MADE BY:-

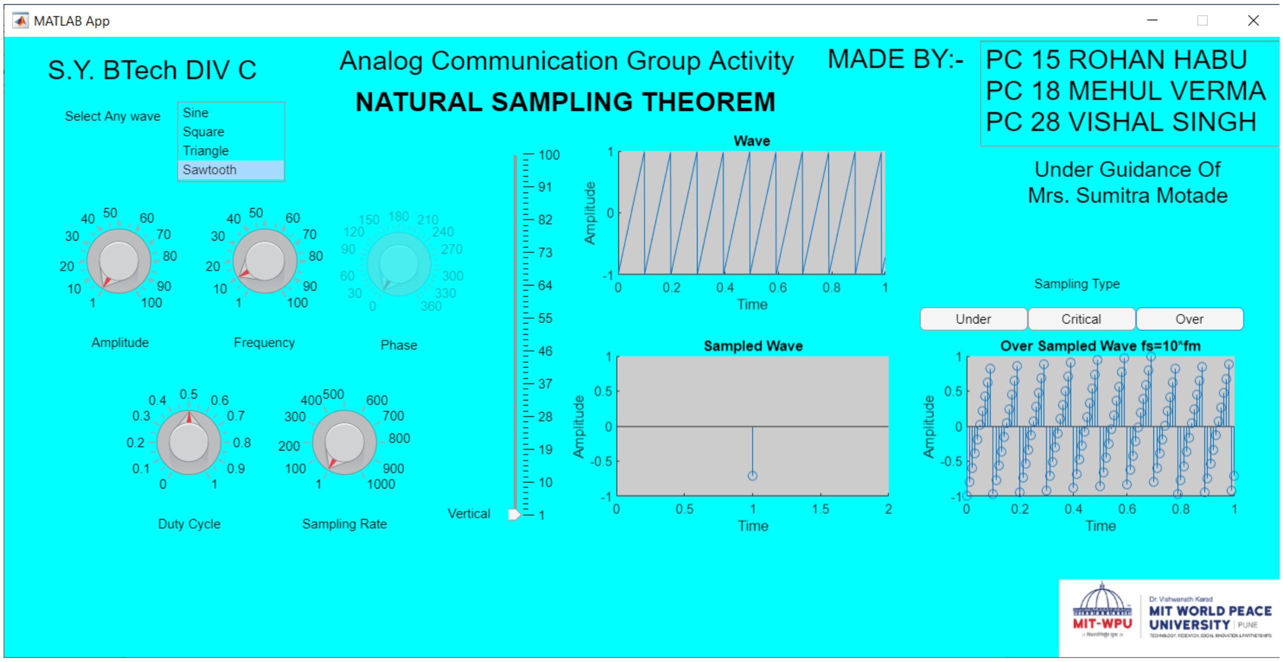
PC 15 ROHAN HABU
PC 18 MEHUL VERMA
PC 28 VISHAL SINGH

Under Guidance Of
Mrs. Sumitra Motade

Select Any wave: Sine, Square, Triangle, Sawtooth (selected)

Sampling Type: Under, Critical, Over

SAW TOOTH WAVE OVER SAMPLED



MATLAB App

S.Y. BTech DIV C

Analog Communication Group Activity

NATURAL SAMPLING THEOREM

MADE BY:-

PC 15 ROHAN HABU
PC 18 MEHUL VERMA
PC 28 VISHAL SINGH

Under Guidance Of
Mrs. Sumitra Motade

Select Any wave: Sine, Square, Triangle, Sawtooth (selected)

Sampling Type: Under, Critical, Over

PROGRAM

```
classdef samplingGUI < matlab.apps.AppBase

  % Properties that correspond to app components
  properties (Access = public)
    UIFigure                                 matlab.ui.Figure
    AmplitudeKnobLabel                       matlab.ui.control.Label
    AmplitudeKnob                            matlab.ui.control.Knob
    FrequencyKnobLabel                       matlab.ui.control.Label
    FrequencyKnob                            matlab.ui.control.Knob
    PhaseKnobLabel                           matlab.ui.control.Label
    PhaseKnob                                matlab.ui.control.Knob
    DutyCycleKnobLabel                       matlab.ui.control.Label
    DutyCycleKnob                            matlab.ui.control.Knob
    SamplingRateKnobLabel                    matlab.ui.control.Label
    SamplingRateKnob                         matlab.ui.control.Knob
    SelectAnywaveListBoxLabel                matlab.ui.control.Label
    SelectAnywaveListBox                     matlab.ui.control.ListBox
    VerticalSliderLabel                      matlab.ui.control.Label
    VerticalSlider                           matlab.ui.control.Slider
    Image                                    matlab.ui.control.Image
    MADEBYTextAreaLabel                      matlab.ui.control.Label
    MADEBYTextArea                           matlab.ui.control.TextArea
    UnderButton                              matlab.ui.control.Button
    CriticalButton                           matlab.ui.control.Button
    OverButton                               matlab.ui.control.Button
    SamplingTypeLabel                        matlab.ui.control.Label
    AnalogCommunicationGroupActivityLabel    matlab.ui.control.Label
    UnderGuidanceOfLabel                     matlab.ui.control.Label
    MrsSumitraMotadeLabel                    matlab.ui.control.Label
    NATURALSAMPLINGTHEOREMLabel              matlab.ui.control.Label
    SYBTechDIVCLabel                         matlab.ui.control.Label
    UIAxes                                   matlab.ui.control.UIAxes
    UIAxes2                                  matlab.ui.control.UIAxes
    UIAxes3                                  matlab.ui.control.UIAxes
  End
```

```
properties (Access = private)
    time
    samplingTime
    sampleRate
    sineamp % Description
    sinefre
    sineph
    sinewave
    sqamp
    sqfre
    sqduty
    sqwave
    triamp
    trifre
    triwave
    sawamp
    sawfre
    sawduty
    sawwave
    sampledSine
    sampledSq
    sampledTri
    sampledSaw
    undersampledAmp
    undersampledFre
    undersampledPh
    undersampledDc
    undersampledSine
    undersampledSq
    undersampledTri
    undersampledSaw
    undersamplingtime
    criticalsampledAmp
    criticalsampledFre
    criticalsampledPh
    criticalsampledDc
    CriticalsampledSine
    criticalsampledSq
    criticalsampledTri
    CriticalsampledSaw
    criticalsamplingtime
    oversampledAmp
    OversampledFre
```

```
  oversampledPh
  OversampledDc
  oversampledSine
  oversampledSq
  oversampledTri
  oversampledSaw
  oversamplingtime
End
% Callbacks that handle component events
methods (Access = private)
  % Code that executes after component creation
  function startupFcn(app)
    app.SamplingRateKnob.Value=1;
    app.sampleRate=app.SamplingRateKnob.Value;
    app.samplingTime=linspace(0,1,app.sampleRate);
  end
  % Value changed function: SelectAnywaveListBox
  function SelectAnywaveListBoxValueChanged(app, event)
    value = app.SelectAnywaveListBox.Value;
    switch value
      case 'Sine'
      app.VerticalSlider.Value=1;
      app.sineamp=1;
      app.sinefre=1;
      app.sineph=0;
      app.time=linspace(0,1,1000);
      app.sinewave=app.sineamp*sin(2*pi*app.time*app.sinefre+app.sineph);
      app.sampledSine=app.sineamp*sin(2*pi*app.sinefre*app.samplingTime+app.sineph);
      plot(app.UIAxes,app.time,app.sinewave);
      stem(app.UIAxes2,app.samplingTime,app.sampledSine);
      ylim(app.UIAxes,[-1,1]);
      ylim(app.UIAxes2,[-1,1]);
      app.DutyCycleKnob.Value=0.5;
      app.DutyCycleKnob.Enable=0;
      app.AmplitudeKnob.Value=1;
      app.FrequencyKnob.Value=1;
      app.PhaseKnob.Enable=1;
      app.PhaseKnob.Value=0;
      case 'Square'
      app.VerticalSlider.Value=1;
      app.sqamp=1;
      app.sqfre=1;
      app.time=linspace(0,1,1000);
      app.sqduty=0.5*100;
app.sqwave=app.sqamp*square(2*pi*app.sqfre*app.time);
app.sampledSq=app.sqamp*square(2*pi*app.sqfre*app.samplingTime,app.sqduty);
plot(app.UIAxes,app.time,app.sqwave);
stem(app.UIAxes2,app.samplingTime,app.sampledSq);
ylim(app.UIAxes,[-1,1]);
ylim(app.UIAxes2,[-1,1]);
app.DutyCycleKnob.Enable=1;
app.DutyCycleKnob.Value=0.5;
app.AmplitudeKnob.Value=1;
app.FrequencyKnob.Value=1;
app.PhaseKnob.Enable=0;
app.PhaseKnob.Value=0;
case 'Triangle'
app.VerticalSlider.Value=1;
app.trifre = 1;
app.triamp=1;
app.time =linspace(0,1,1000);
app.triwave = app.triamp*sawtooth(2*pi*app.trifre*app.time,1/2);
app.sampledTri = app.triamp*sawtooth(2*pi*app.samplingTime*app.trifre,1/2);
plot(app.UIAxes,app.time,app.triwave);
stem(app.UIAxes2,app.samplingTime,app.sampledTri);
ylim(app.UIAxes,[-1,1]);
ylim(app.UIAxes2,[-1,1]);
app.DutyCycleKnob.Enable=0;
app.DutyCycleKnob.Value=0.5;
app.AmplitudeKnob.Value=1;
app.FrequencyKnob.Value=1;
app.PhaseKnob.Enable=0;
app.PhaseKnob.Value=0;
case 'Sawtooth'
app.VerticalSlider.Value=1;
app.sawfre=1;
app.time=linspace(0,1,1000);
app.sawduty=1;
app.sawamp=1;
app.sawwave=app.sawamp*sawtooth(2*pi*app.sawfre*app.time);
app.sampledSaw=app.sawamp*sawtooth(2*pi*app.sawfre*app.samplingTime,app.sawduty);
plot(app.UIAxes,app.time,app.sawwave);
stem(app.UIAxes2,app.samplingTime,app.sampledSaw);
ylim(app.UIAxes,[-1,1]);
ylim(app.UIAxes2,[-1,1]);
app.DutyCycleKnob.Enable=1;
app.DutyCycleKnob.Value=0.5;
app.AmplitudeKnob.Value=1;
```

```
      app.FrequencyKnob.Value=1;
      app.PhaseKnob.Enable=0;
      app.PhaseKnob.Value=0;
   end
end
% Value changing function: AmplitudeKnob
function AmplitudeKnobValueChanging(app, event)
   value = event.Value;
   switch app.SelectAnywaveListBox.Value
      case 'Sine'
         app.sineamp=value;
         app.sinewave=app.sineamp*sin(2*pi*app.time*app.sinefre+app.sineph);
         app.sampledSine=app.sineamp*sin(2*pi*app.sinefre*app.samplingTime+app.sineph);
         plot(app.UIAxes,app.time,app.sinewave);
         stem(app.UIAxes2,app.samplingTime,app.sampledSine);
      case 'Square'
         app.sqamp=value;
         app.sqwave=app.sqamp*square(2*pi*app.sqfre*app.time);
         app.sampledSq=app.sqamp*square(2*pi*app.sqfre*app.samplingTime,app.sqduty);
         plot(app.UIAxes,app.time,app.sqwave);
         stem(app.UIAxes2,app.samplingTime,app.sampledSq);
      case 'Triangle'
         app.triamp=value;
         app.triwave = app.triamp*sawtooth(2*pi*app.trifre*app.time,1/2);
         app.sampledTri = app.triamp*sawtooth(2*pi*app.samplingTime*app.trifre,1/2);
         plot(app.UIAxes,app.time,app.triwave);
         stem(app.UIAxes2,app.samplingTime,app.sampledTri);
      case 'Sawtooth'
         app.sawamp=value;
         app.sawwave=app.sawamp*sawtooth(2*pi*app.sawfre*app.time);
         app.sampledSaw=app.sawamp*sawtooth(2*pi*app.sawfre*app.samplingTime,app.sawduty);
         plot(app.UIAxes,app.time,app.sawwave);
         stem(app.UIAxes2,app.samplingTime,app.sampledSaw);
   end
end
% Value changing function: FrequencyKnob
function FrequencyKnobValueChanging(app, event)
   value = event.Value;
   switch app.SelectAnywaveListBox.Value
      case 'Sine'
         app.sinefre=value;
         app.sinewave=app.sineamp*sin(2*pi*app.time*app.sinefre+app.sineph);
         app.sampledSine=app.sineamp*sin(2*pi*app.sinefre*app.samplingTime+app.sineph);
         plot(app.UIAxes,app.time,app.sinewave);
```

```
          stem(app.UIAxes2,app.samplingTime,app.sampledSine);
        case 'Square'
          app.sqfre=value;
          app.sqwave=app.sqamp*square(2*pi*app.sqfre*app.time);
          app.sampledSq=app.sqamp*square(2*pi*app.sqfre*app.samplingTime,app.sqduty);
          plot(app.UIAxes,app.time,app.sqwave);
          stem(app.UIAxes2,app.samplingTime,app.sampledSq);
        case 'Triangle'
          app.trifre=value;
          app.triwave = app.triamp*sawtooth(2*pi*app.trifre*app.time,1/2);
          app.sampledTri = app.triamp*sawtooth(2*pi*app.samplingTime*app.trifre,1/2);
          plot(app.UIAxes,app.time,app.triwave);
          stem(app.UIAxes2,app.samplingTime,app.sampledTri);
        case 'Sawtooth'
          app.sawfre=value;
          app.sawwave=app.sawamp*sawtooth(2*pi*app.sawfre*app.time);
         app.sampledSaw=app.sawamp*sawtooth(2*pi*app.sawfre*app.samplingTime,app.sawduty);
          plot(app.UIAxes,app.time,app.sawwave);
          stem(app.UIAxes2,app.samplingTime,app.sampledSaw);
      end
    end
    % Value changing function: PhaseKnob
    function PhaseKnobValueChanging(app, event)
      value = event.Value;
      switch app.SelectAnywaveListBox.Value
        case 'Sine'
          app.sineph=value;
          app.sinewave=app.sineamp*sin(2*pi*app.time*app.sinefre+app.sineph);
          app.sampledSine=app.sineamp*sin(2*pi*app.sinefre*app.samplingTime+app.sineph);
          plot(app.UIAxes,app.time,app.sinewave);
          stem(app.UIAxes2,app.samplingTime,app.sampledSine);
      end
    end
% Value changing function: DutyCycleKnob
    function DutyCycleKnobValueChanging(app, event)
      value = event.Value;
      switch app.SelectAnywaveListBox.Value
        case 'Square'
          app.sqduty=value*100;
          app.sqwave=app.sqamp*square(2*pi*app.sqfre*app.time,app.sqduty);
          app.sampledSq=app.sqamp*square(2*pi*app.sqfre*app.samplingTime,app.sqduty);
          plot(app.UIAxes,app.time,app.sqwave);
          stem(app.UIAxes2,app.samplingTime,app.sampledSq);
        case 'Sawtooth'
```

```
          app.sawduty=value;
          app.sawwave=app.sawamp*sawtooth(2*pi*app.sawfre*app.time,app.sawduty);
         app.sampledSaw=app.sawamp*sawtooth(2*pi*app.sawfre*app.samplingTime,app.sawduty);
          plot(app.UIAxes,app.time,app.sawwave);
          stem(app.UIAxes2,app.samplingTime,app.sampledSaw);
      end
    end
% Value changing function: SamplingRateKnob
    function SamplingRateKnobValueChanging(app, event)
      value = event.Value;
      app.sampleRate=value;
      switch app.SelectAnywaveListBox.Value
        case 'Sine'
          app.samplingTime=0:1/value:1;
          app.sampledSine=app.sineamp*sin(2*pi*app.sinefre*app.samplingTime+app.sineph);
          stem(app.UIAxes2,app.samplingTime,app.sampledSine);
        case 'Square'
          app.samplingTime=0:1/value:1;
          app.sampledSq=app.sqamp*square(2*pi*app.sqfre*app.samplingTime,app.sqduty);
          stem(app.UIAxes2,app.samplingTime,app.sampledSq);
        case 'Triangle'
          app.samplingTime=0:1/value:1;
          app.sampledTri = app.triamp*sawtooth(2*pi*app.samplingTime*app.trifre,1/2);
          stem(app.UIAxes2,app.samplingTime,app.sampledTri);
        case 'Sawtooth'
          app.samplingTime=0:1/value:1;
         app.sampledSaw=app.sawamp*sawtooth(2*pi*app.sawfre*app.samplingTime,app.sawduty);
          stem(app.UIAxes2,app.samplingTime,app.sampledSaw);
      end
    end
   % Value changing function: VerticalSlider
    function VerticalSliderValueChanging(app, event)
      changingValue = event.Value;
      ylim(app.UIAxes,[-changingValue,changingValue]);
      ylim(app.UIAxes2,[-changingValue,changingValue]);
    end
    % Button pushed function: UnderButton
    function UnderButtonPushed(app, event)
      value=app.SelectAnywaveListBox.Value;
       switch value
        case 'Sine'
        app.undersamplingtime=linspace(0,1,app.sinefre/2);
        app.undersampledSine=app.sineamp*sin(2*pi*app.sinefre*app.undersamplingtime
+(app.sineph));
```

```
        stem(app.UIAxes3,app.undersamplingtime,app.undersampledSine);
        title(app.UIAxes3,'Under Sampled Wave fs=fm/2');
        case 'Square'
        app.undersamplingtime=linspace(0,1,app.sqfre/2);
       app.undersampledSq=app.sqamp*square(2*pi*app.sqfre*app.undersamplingtime,app.sqduty);
        stem(app.UIAxes3,app.undersamplingtime,app.undersampledSq);
        title(app.UIAxes3,'Under Sampled Wave fs=fm/2');
        case 'Triangle'
        app.undersamplingtime=linspace(0,1,app.trifre/2);
        app.undersampledTri = app.triamp*sawtooth(2*pi*app.undersamplingtime*app.trifre,1/2);
        stem(app.UIAxes3,app.undersamplingtime,app.undersampledTri);
        title(app.UIAxes3,'Under Sampled Wave fs=fm/2');
        case 'Sawtooth'
        app.undersamplingtime=linspace(0,1,app.sawfre/2);
app.undersampledSaw=app.sawamp*sawtooth(2*pi*app.sawfre*app.undersamplingtime,app.sawduty)
;
        stem(app.UIAxes3,app.undersamplingtime,app.undersampledSaw);
        title(app.UIAxes3,'Under Sampled Wave fs=fm/2');
      end
    end
  % Button pushed function: CriticalButton
    function CriticalButtonPushed(app, event)
       value=app.SelectAnywaveListBox.Value;
       switch value
        case 'Sine'
        app.criticalsamplingtime=linspace(0,1,2*app.sinefre);

app.criticalsampledSine=app.sineamp*sin(2*pi*app.sinefre*app.criticalsamplingtime+(app.sineph));
        stem(app.UIAxes3,app.criticalsamplingtime,app.criticalsampledSine);
        title(app.UIAxes3,'Critical Sampled Wave fs=2*fm');
        case 'Square'
        app.criticalsamplingtime=linspace(0,1,2*app.sqfre);
      app.criticalsampledSq=app.sqamp*square(2*pi*app.sqfre*app.criticalsamplingtime,app.sqduty);
        stem(app.UIAxes3,app.criticalsamplingtime,app.criticalsampledSq);
        title(app.UIAxes3,'Critical Sampled Wave fs=2*fm');
        case 'Triangle'
        app.criticalsamplingtime=linspace(0,1,2*app.trifre);
        app.criticalsampledTri = app.triamp*sawtooth(2*pi*app.criticalsamplingtime*app.trifre,1/2);
        stem(app.UIAxes3,app.criticalsamplingtime,app.criticalsampledTri);
        title(app.UIAxes3,'Critical Sampled Wave fs=2*fm');
        case 'Sawtooth'
        app.criticalsamplingtime=linspace(0,1,app.sawfre*2);

app.criticalsampledSaw=app.sawamp*sawtooth(2*pi*app.sawfre*app.criticalsamplingtime,app.sawdut
y);
        stem(app.UIAxes3,app.criticalsamplingtime,app.criticalsampledSaw);
        title(app.UIAxes3,'Critical Sampled Wave fs=2*fm');
      end
    end
% Button pushed function: OverButton
    function OverButtonPushed(app, event)
            value=app.SelectAnywaveListBox.Value;
      switch value
        case 'Sine'
        app.oversamplingtime=linspace(0,1,10*app.sinefre);
        app.oversampledSine=app.sineamp*sin(2*pi*app.sinefre*app.oversamplingtime
+(app.sineph));
        stem(app.UIAxes3,app.oversamplingtime,app.oversampledSine);
        title(app.UIAxes3,'Over Sampled Wave fs=10*fm');
        case 'Square'
        app.oversamplingtime=linspace(0,1,10*app.sqfre);
        app.oversampledSq=app.sqamp*square(2*pi*app.sqfre*app.oversamplingtime,app.sqduty);
        stem(app.UIAxes3,app.oversamplingtime,app.oversampledSq);
        title(app.UIAxes3,'Over Sampled Wave fs=10*fm');
        case 'Triangle'
        app.oversamplingtime=linspace(0,1,10*app.trifre);
        app.oversampledTri = app.triamp*sawtooth(2*pi*app.oversamplingtime*app.trifre,1/2);
        stem(app.UIAxes3,app.oversamplingtime,app.oversampledTri);
        title(app.UIAxes3,'Over Sampled Wave fs=10*fm');
        case 'Sawtooth'
        app.oversamplingtime=linspace(0,1,10*app.sawfre);

app.oversampledSaw=app.sawamp*sawtooth(2*pi*app.sawfre*app.oversamplingtime,app.sawduty);
        stem(app.UIAxes3,app.oversamplingtime,app.oversampledSaw);
        title(app.UIAxes3,'Over Sampled Wave fs=10*fm');
      end
    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.Color = [0 1 1];
      app.UIFigure.Position = [100 100 1180 574];
      app.UIFigure.Name = 'MATLAB App';
      app.UIFigure.Resize = 'off';
      % Create AmplitudeKnobLabel
      app.AmplitudeKnobLabel = uilabel(app.UIFigure);
      app.AmplitudeKnobLabel.HorizontalAlignment = 'center';
```

```
        app.AmplitudeKnobLabel.Position = [79 283 59 22];
        app.AmplitudeKnobLabel.Text = 'Amplitude';
        % Create AmplitudeKnob
        app.AmplitudeKnob = uiknob(app.UIFigure, 'continuous');
        app.AmplitudeKnob.Limits = [1 100];
        app.AmplitudeKnob.MajorTicks = [1 10 20 30 40 50 60 70 80 90 100];
        app.AmplitudeKnob.MajorTickLabels = {'1', '10', '20', '30', '40', '50', '60', '70', '80', '90', '100'};
        app.AmplitudeKnob.ValueChangingFcn = createCallbackFcn(app,
@AmplitudeKnobValueChanging, true);
        app.AmplitudeKnob.Position = [77 339 60 60];
        app.AmplitudeKnob.Value = 1;
 % Create FrequencyKnobLabel
        app.FrequencyKnobLabel = uilabel(app.UIFigure);
        app.FrequencyKnobLabel.HorizontalAlignment = 'center';
        app.FrequencyKnobLabel.Position = [211 283 62 22];
        app.FrequencyKnobLabel.Text = 'Frequency';
        % Create FrequencyKnob
        app.FrequencyKnob = uiknob(app.UIFigure, 'continuous');
        app.FrequencyKnob.Limits = [1 100];
        app.FrequencyKnob.MajorTicks = [1 10 20 30 40 50 60 70 80 90 100];
        app.FrequencyKnob.ValueChangingFcn = createCallbackFcn(app,
@FrequencyKnobValueChanging, true);
        app.FrequencyKnob.Position = [212 339 60 60];
        app.FrequencyKnob.Value = 1;
        % Create PhaseKnobLabel
        app.PhaseKnobLabel = uilabel(app.UIFigure);
        app.PhaseKnobLabel.HorizontalAlignment = 'center';
        app.PhaseKnobLabel.Position = [347 280 39 22];
        app.PhaseKnobLabel.Text = 'Phase';
       % Create PhaseKnob
        app.PhaseKnob = uiknob(app.UIFigure, 'continuous');
        app.PhaseKnob.Limits = [0 360];
        app.PhaseKnob.MajorTicks = [0 30 60 90 120 150 180 210 240 270 300 330 360];
        app.PhaseKnob.ValueChangingFcn = createCallbackFcn(app, @PhaseKnobValueChanging, true);
        app.PhaseKnob.Position = [336 336 60 60];
        % Create DutyCycleKnobLabel
        app.DutyCycleKnobLabel = uilabel(app.UIFigure);
        app.DutyCycleKnobLabel.HorizontalAlignment = 'center';
        app.DutyCycleKnobLabel.Position = [141 115 63 22];
        app.DutyCycleKnobLabel.Text = 'Duty Cycle';
% Create DutyCycleKnob
        app.DutyCycleKnob = uiknob(app.UIFigure, 'continuous');
        app.DutyCycleKnob.Limits = [0 1];
        app.DutyCycleKnob.MajorTicks = [0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1];
```

```
    app.DutyCycleKnob.ValueChangingFcn = createCallbackFcn(app,
@DutyCycleKnobValueChanging, true);
    app.DutyCycleKnob.Position = [142 171 60 60];
    % Create SamplingRateKnobLabel
    app.SamplingRateKnobLabel = uilabel(app.UIFigure);
    app.SamplingRateKnobLabel.HorizontalAlignment = 'center';
    app.SamplingRateKnobLabel.Position = [274 115 84 22];
    app.SamplingRateKnobLabel.Text = 'Sampling Rate';
    % Create SamplingRateKnob
    app.SamplingRateKnob = uiknob(app.UIFigure, 'continuous');
    app.SamplingRateKnob.Limits = [1 1000];
    app.SamplingRateKnob.MajorTicks = [1 100 200 300 400 500 600 700 800 900 1000];
    app.SamplingRateKnob.ValueChangingFcn = createCallbackFcn(app,
@SamplingRateKnobValueChanging, true);
    app.SamplingRateKnob.MinorTicks = [];
    app.SamplingRateKnob.Position = [286 171 60 60];
    app.SamplingRateKnob.Value = 1;
    % Create SelectAnywaveListBoxLabel
    app.SelectAnywaveListBoxLabel = uilabel(app.UIFigure);
    app.SelectAnywaveListBoxLabel.HorizontalAlignment = 'right';
    app.SelectAnywaveListBoxLabel.Position = [52 492 94 22];
    app.SelectAnywaveListBoxLabel.Text = 'Select Any wave';
    % Create SelectAnywaveListBox
    app.SelectAnywaveListBox = uilistbox(app.UIFigure);
    app.SelectAnywaveListBox.Items = {'Sine', 'Square', 'Triangle', 'Sawtooth'};
    app.SelectAnywaveListBox.ValueChangedFcn = createCallbackFcn(app,
@SelectAnywaveListBoxValueChanged, true);
    app.SelectAnywaveListBox.BackgroundColor = [0 1 1];
    app.SelectAnywaveListBox.Position = [161 442 100 74];
    app.SelectAnywaveListBox.Value = 'Sine';
    % Create VerticalSliderLabel
    app.VerticalSliderLabel = uilabel(app.UIFigure);
    app.VerticalSliderLabel.HorizontalAlignment = 'right';
    app.VerticalSliderLabel.Position = [406 124 45 22];
    app.VerticalSliderLabel.Text = 'Vertical';
    % Create VerticalSlider
    app.VerticalSlider = uislider(app.UIFigure);
    app.VerticalSlider.Limits = [1 100];
    app.VerticalSlider.Orientation = 'vertical';
    app.VerticalSlider.ValueChangingFcn = createCallbackFcn(app, @VerticalSliderValueChanging,
true);
    app.VerticalSlider.Position = [472 133 3 334];
    app.VerticalSlider.Value = 1;
    % Create Image
    app.Image = uiimage(app.UIFigure);
```

```
        app.Image.Position = [977 -13 205 100];
        app.Image.ImageSource = 'Capture.PNG';
  % Create MADEBYTextAreaLabel
        app.MADEBYTextAreaLabel = uilabel(app.UIFigure);
        app.MADEBYTextAreaLabel.BackgroundColor = [0 1 1];
        app.MADEBYTextAreaLabel.HorizontalAlignment = 'right';
        app.MADEBYTextAreaLabel.FontSize = 25;
        app.MADEBYTextAreaLabel.Position = [757 540 132 31];
        app.MADEBYTextAreaLabel.Text = 'MADE BY:-';
        % Create MADEBYTextArea
        app.MADEBYTextArea = uitextarea(app.UIFigure);
        app.MADEBYTextArea.FontSize = 25;
        app.MADEBYTextArea.BackgroundColor = [0 1 1];
        app.MADEBYTextArea.Position = [904 475 278 98];
        app.MADEBYTextArea.Value = {'PC 15 ROHAN HABU'; 'PC 18 MEHUL VERMA'; 'PC 28 VISHAL SINGH'};
        % Create UnderButton
        app.UnderButton = uibutton(app.UIFigure, 'push');
        app.UnderButton.ButtonPushedFcn = createCallbackFcn(app, @UnderButtonPushed, true);
        app.UnderButton.Position = [848 304 100 22];
        app.UnderButton.Text = 'Under';
        % Create CriticalButton
        app.CriticalButton = uibutton(app.UIFigure, 'push');
        app.CriticalButton.ButtonPushedFcn = createCallbackFcn(app, @CriticalButtonPushed, true);
        app.CriticalButton.Position = [948 304 100 22];
        app.CriticalButton.Text = 'Critical';
  % Create OverButton
        app.OverButton = uibutton(app.UIFigure, 'push');
        app.OverButton.ButtonPushedFcn = createCallbackFcn(app, @OverButtonPushed, true);
        app.OverButton.Position = [1048 304 100 22];
        app.OverButton.Text = 'Over';
        % Create SamplingTypeLabel
        app.SamplingTypeLabel = uilabel(app.UIFigure);
        app.SamplingTypeLabel.Position = [954 337 86 22];
        app.SamplingTypeLabel.Text = 'Sampling Type';
        % Create AnalogCommunicationGroupActivityLabel
        app.AnalogCommunicationGroupActivityLabel = uilabel(app.UIFigure);
        app.AnalogCommunicationGroupActivityLabel.FontSize = 25;
        app.AnalogCommunicationGroupActivityLabel.Position = [311 539 434 31];
        app.AnalogCommunicationGroupActivityLabel.Text = 'Analog Communication Group Activity';
        % Create UnderGuidanceOfLabel
        app.UnderGuidanceOfLabel = uilabel(app.UIFigure);
        app.UnderGuidanceOfLabel.FontSize = 20;
        app.UnderGuidanceOfLabel.Position = [954 443 177 24];
        app.UnderGuidanceOfLabel.Text = 'Under Guidance Of';
        % Create MrsSumitraMotadeLabel
        app.MrsSumitraMotadeLabel = uilabel(app.UIFigure);
        app.MrsSumitraMotadeLabel.FontSize = 20;
        app.MrsSumitraMotadeLabel.Position = [948 419 192 24];
        app.MrsSumitraMotadeLabel.Text = 'Mrs. Sumitra Motade';
       % Create NATURALSAMPLINGTHEOREMLabel
        app.NATURALSAMPLINGTHEOREMLabel = uilabel(app.UIFigure);
        app.NATURALSAMPLINGTHEOREMLabel.HorizontalAlignment = 'center';
        app.NATURALSAMPLINGTHEOREMLabel.FontSize = 25;
        app.NATURALSAMPLINGTHEOREMLabel.FontWeight = 'bold';
        app.NATURALSAMPLINGTHEOREMLabel.Position = [311 500 419 31];
        app.NATURALSAMPLINGTHEOREMLabel.Text = 'NATURAL SAMPLING THEOREM';
        % Create SYBTechDIVCLabel
        app.SYBTechDIVCLabel = uilabel(app.UIFigure);
        app.SYBTechDIVCLabel.FontSize = 25;
        app.SYBTechDIVCLabel.Position = [41 530 200 31];
        app.SYBTechDIVCLabel.Text = 'S.Y. BTech DIV C';
        % Create UIAxes
        app.UIAxes = uiaxes(app.UIFigure);
        title(app.UIAxes, 'Wave')
        xlabel(app.UIAxes, {'Time'; ''})
        ylabel(app.UIAxes, 'Amplitude')
        zlabel(app.UIAxes, 'Z')
        app.UIAxes.GridLineStyle = '--';
        app.UIAxes.Color = [0.8 0.8 0.8];
        app.UIAxes.GridColor = [0 0 1];
        app.UIAxes.MinorGridColor = [1 0.0745 0.651];
        app.UIAxes.Position = [525 304 300 185];
    % Create UIAxes2
        app.UIAxes2 = uiaxes(app.UIFigure);
        title(app.UIAxes2, 'Sampled Wave')
        xlabel(app.UIAxes2, 'Time')
        ylabel(app.UIAxes2, 'Amplitude')
        zlabel(app.UIAxes2, 'Z')
        app.UIAxes2.Color = [0.8 0.8 0.8];
        app.UIAxes2.Position = [524 115 300 185];
        % Create UIAxes3
        app.UIAxes3 = uiaxes(app.UIFigure);
        title(app.UIAxes3, 'Condition Sampled Wave')
        xlabel(app.UIAxes3, 'Time')
        ylabel(app.UIAxes3, 'Amplitude')
        zlabel(app.UIAxes3, 'Z')
        app.UIAxes3.Color = [0.8 0.8 0.8];
        app.UIAxes3.Position = [848 115 300 185];
```

```
            % 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 = samplingGUI
 % 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
```

REFERENCES

www.sciencedirect.com

www.tutorialspoint.com

www.mathworks.com

www.cnx.org

www.musicweb.ucsd.edu