School of Electronics and Communication Engineering

Academic Year 2020 - 2021

Mini Project Report - Trimester VI

Title of the Project: Sampling & Aliasing

Course Name: Analog Communication

Name of Students:

Sr. No.	PRN No.	Name of Student	Contact No.	Email ID
1.	1032191471	Piyush Pamnani	7509245660	piyushpamnani46@gmail.com
2.	1032191501	Ruchin Shroff	8793514266	ruchin.shroff@gmail.com
3.	1032191519	Vijay Choudhary	9982143695	vkc66183@gmail.com

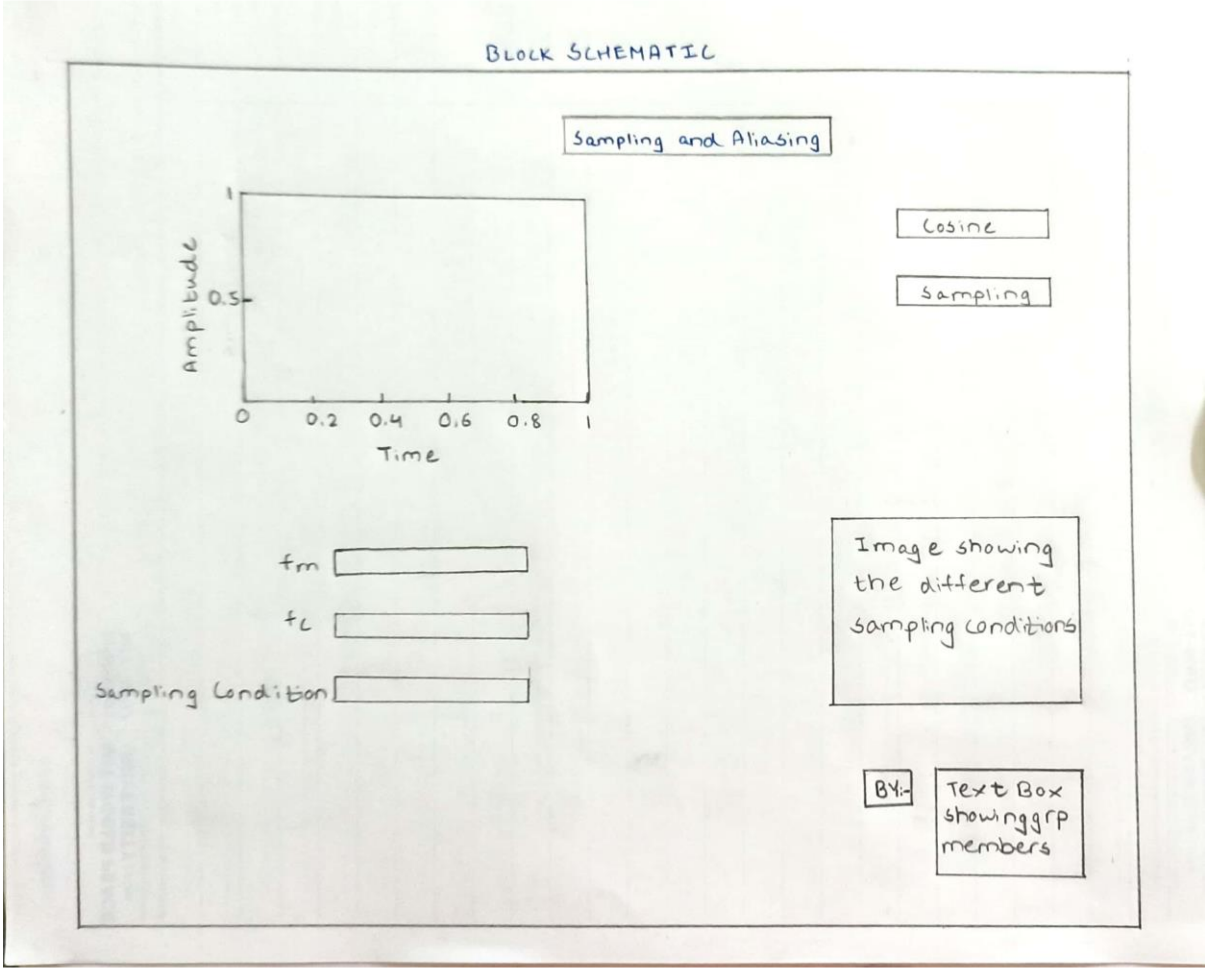
Introduction:

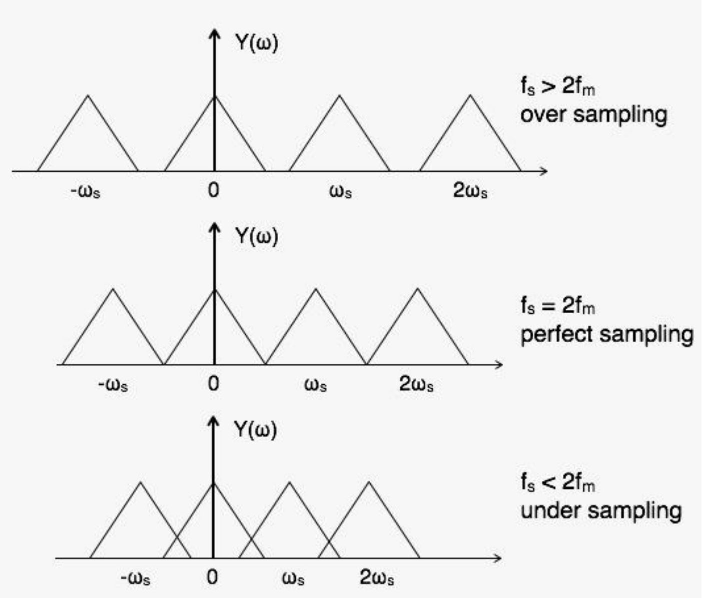
The process of transmitting signals in the form of pulses (discontinuous signals) by using special techniques is called “sampling”.
The signal is sampled at regular intervals such that each sample is proportional to the amplitude of signal at that instant.

The Sampling Theorem states that a signal can be exactly reproduced if it is sampled at a frequency F, where F is greater than twice the maximum frequency in the signal. When the signal is converted back into a continuous time signal, it will exhibit a phenomenon called aliasing. Aliasing is the presence of unwanted components in the reconstructed signal.

Aliasing is a potential problem whenever an analog signal is point sampled to convert it into a digital signal. Aliasing happens whenever an analog signal is not sampled at a high enough frequency.

Block Schematic and Explanation:





Sampling Theorem:

Sampling is defined as, “The process of measuring the instantaneous values of continuous-time signal in a discrete form.”

Sample is a piece of data taken from the whole data which is continuous in the time domain.

When a source generates an analog signal and if that has to be digitized, having 1s and 0s i.e., High or Low, the signal has to be discretized in time. This discretization of analog signal is called as Sampling.

The following figure indicates a continuous-time signal x tt and a sampled signal x_s tt. When x tt is multiplied by a periodic impulse train, the sampled signal x_s tt is obtained.

To discretize the signals, the gap between the samples should be fixed. That gap can be termed as a sampling period T_s.

Sampling Frequency=1/Ts=fs

Where,

- Ts is the sampling time
- fs is the sampling frequency or the sampling rate

Sampling frequency is the reciprocal of the sampling period. This sampling frequency, can be simply called as Sampling rate. The sampling rate denotes the number of samples taken per second, or for a finite set of values.

For an analog signal to be reconstructed from the digitized signal, the sampling rate should be highly considered. The rate of sampling should be such that the data in the message signal should neither be lost nor it should get over-lapped. Hence, a rate was fixed for this, called as Nyquist rate.

The transformation of signals into the frequency domain is performed by the Fourier transformation, which essentially reformulates the signal into a cosine function space. If, for example, we transform the sine function into the frequency domain, this results in a peak at the frequency of the sine function. Due to the symmetry of the Fourier transform for real values, there are two peaks on both sides of the ordinate (y) axis. More details on the frequency-domain-based interpretation can be found in Glassner.
If we transform the continuous representation into a discrete representation, we need to take samples of the continuous sine function to measure its characteristics.

Since the sine function has the periodicity T=2π/w (or the frequency of w/2π), this sampling frequency would be also T.

If we increase the sampling speed to half of the periodicity of the continuous function, the minimum demand of the sampling theorem, we can recover the correct characteristics of the sine function, as it can be seen in. However, depending on what exact position in the period T of the original function we take the sample, we recover different amplitudes of the original signal. In an unfortunate case, we always sample the zero crossing of the sine function. In this case, the characteristics could be correctly recovered, but the amplitude of the signal was recovered in an unfortunate way, so we are back with a constant signal. Overall, sampling at a rate satisfying the sampling theorem does not guarantee that the full signal strength is reconstructed, although higher sampling rates usually approach the original strength.

Nyquist Rate:

The main basis in signal theory is the sampling theorem that is credited to Nyquist —who first formulated the theorem in 1928.

The sampling theorem essentially says that a signal has to be sampled at least with twice the frequency of the original signal. Since signals and their respective speed can be easier expressed by frequencies, most explanations of artifacts are based on their representation in the frequency domain. The sampling frequency required by the sampling theorem is called the Nyquist frequency.

$F_s = 2f_m$ is also called as Nyquist rate.
A continuous time signal can be completely represented in its samples & recovered back if sampling frequency f_s is greater than or equal to the twice of highest frequency component of message signal.
Nyquist criterion ensures an analog signal to be reconstructed completely from a set of uniformly spaced discrete time samples.

Aliasing:

Aliasing can be referred to as "the phenomenon of a high-frequency component in the spectrum of a signal, taking on the identity of a low-frequency component in the spectrum of its sampled version."

The corrective measures taken to reduce the effect of Aliasing are −

- In the transmitter section of PCM, a low pass anti-aliasing filter is employed, before the sampler, to eliminate the high frequency components, which are unwanted.
- The signal which is sampled after filtering, is sampled at a rate slightly higher than the Nyquist rate.

This choice of having the sampling rate higher than Nyquist rate, also helps in the easier design of the reconstruction filter at the receiver.

Types of Sampling:

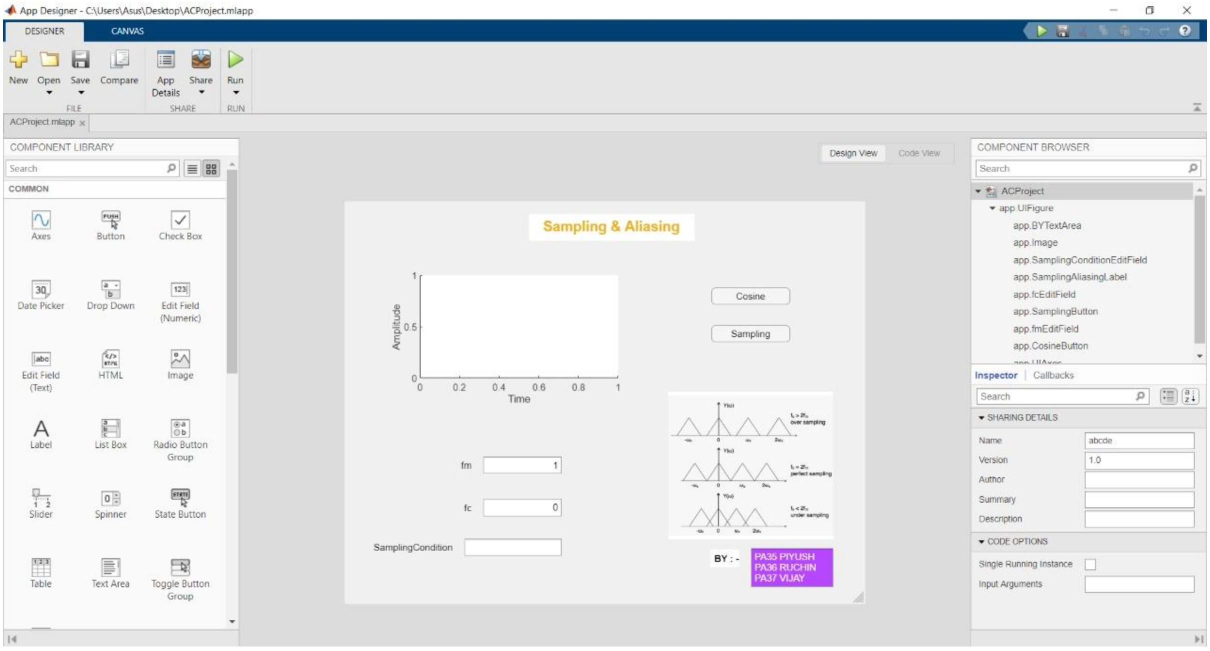
Impulse or Ideal Sampling: Impulse at each sampling instant
Natural Sampling: A pulse of short width with varying amplitude
Flat Top Sampling: Sample and hold, like natural but with single amplitude value

Applications of Sampling Theorem:

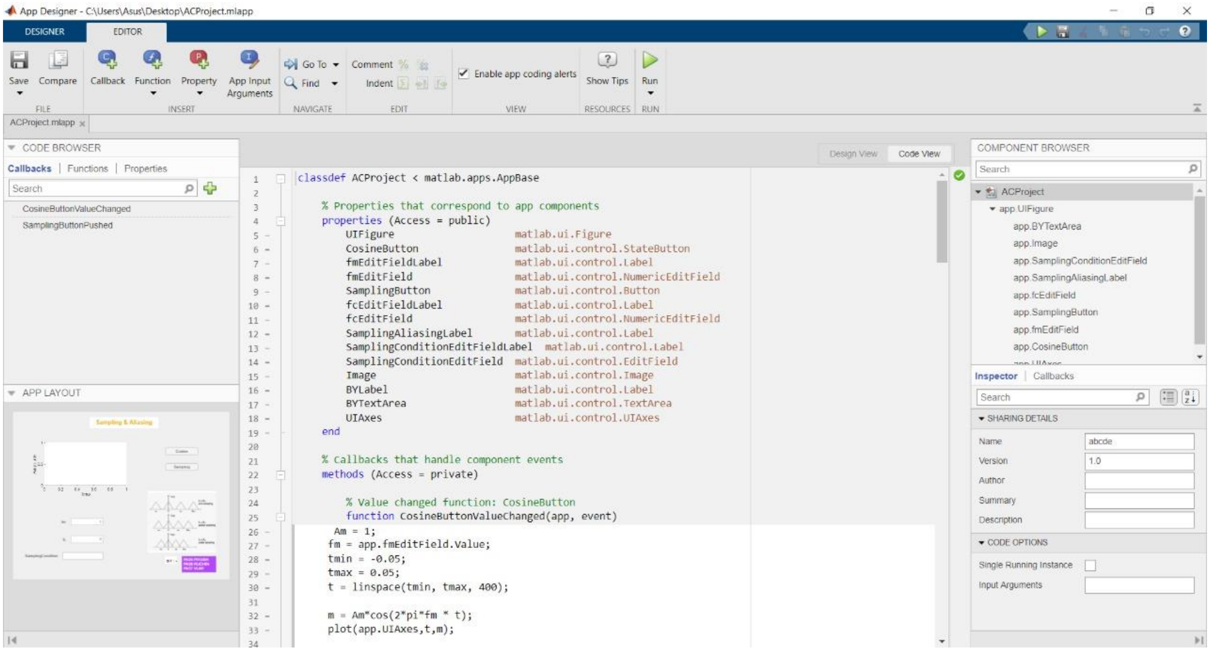
Signal Analysis
Signal Processing
Signal Transmission

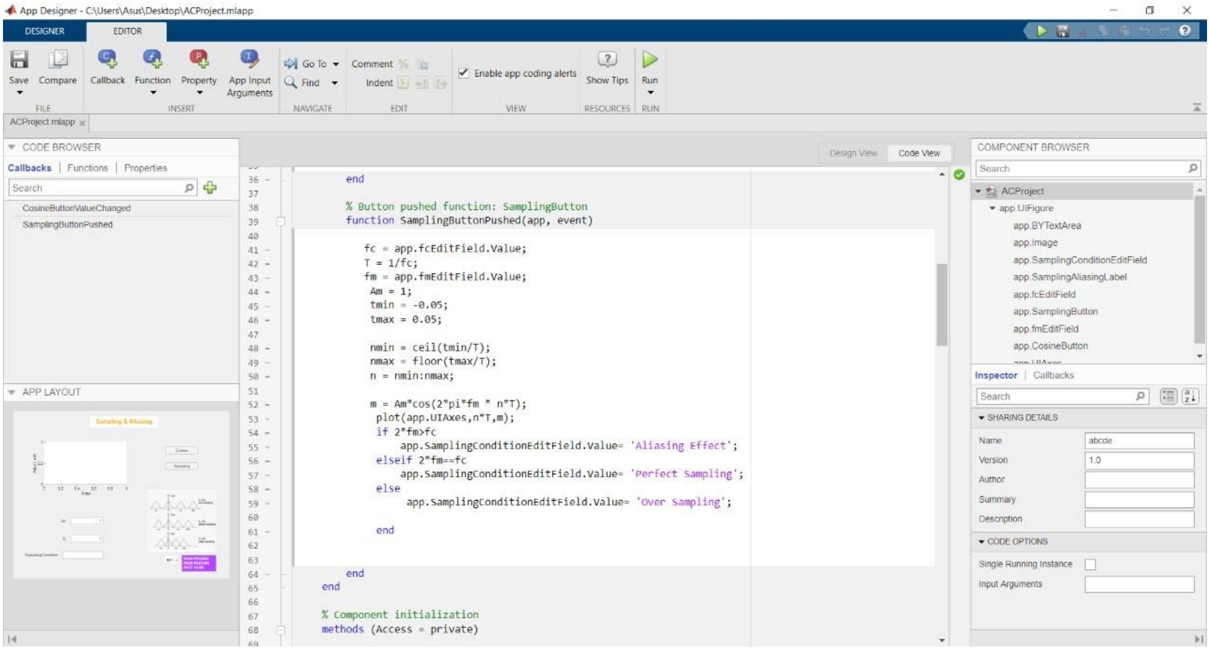
Screen shots:

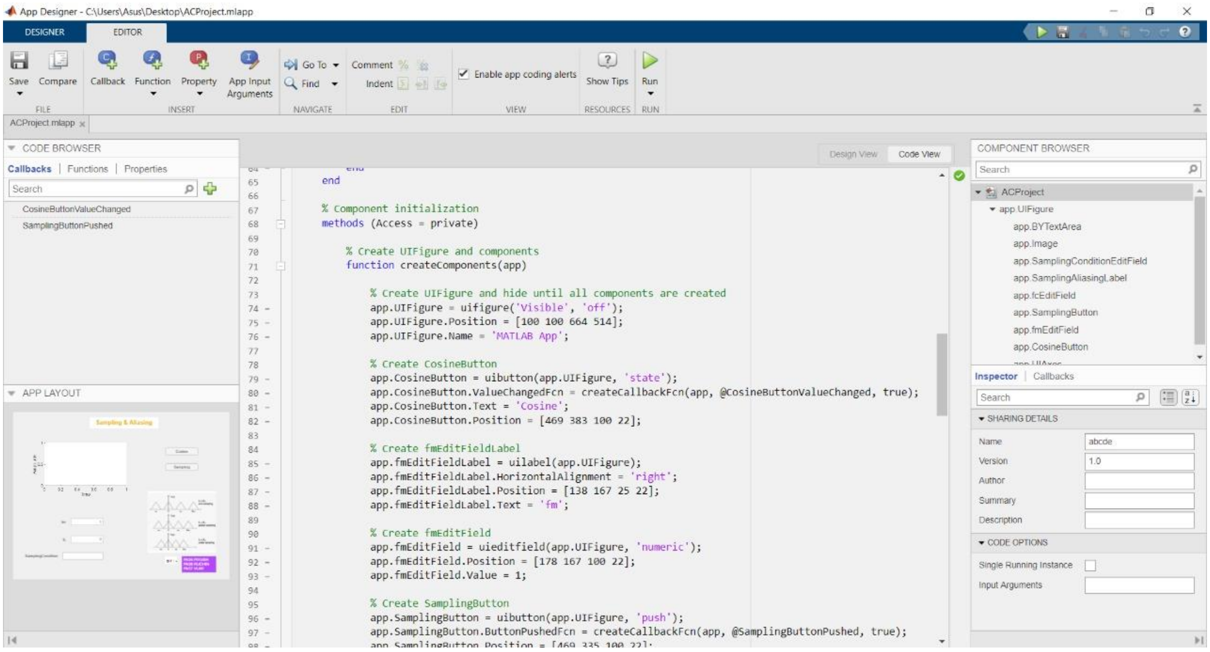
MATLAB GUI:

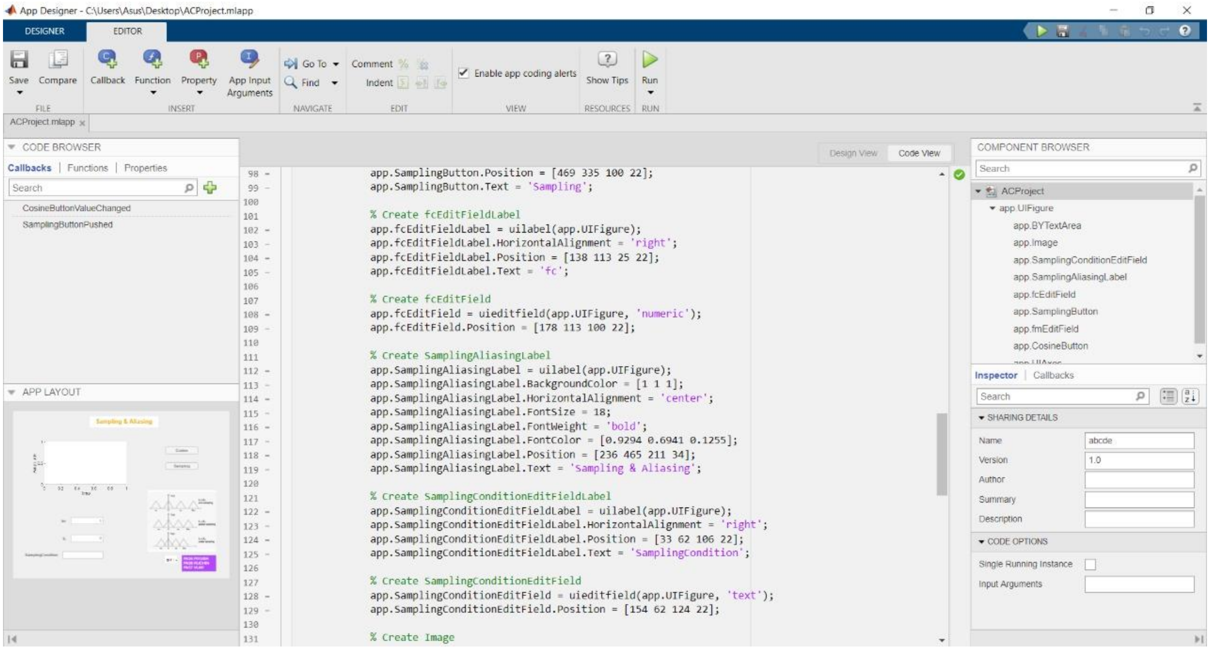

App Designer - C:\Users\Asus\Desktop\ACProject.mlapp
Sampling & Aliasing
Amplitude
Time
Cosine
Sampling
fm
fc
SamplingCondition
BY : -
PA35 PIYUSH
PA36 RUCHIN
PA37 VIJAY

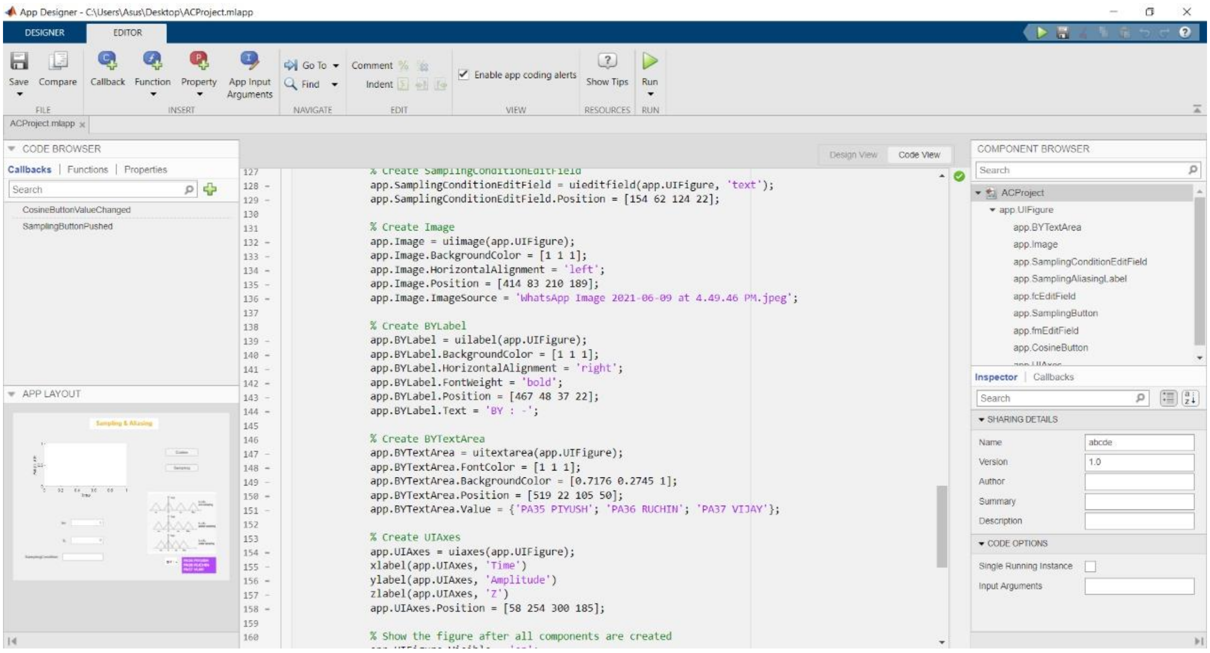
CODE:

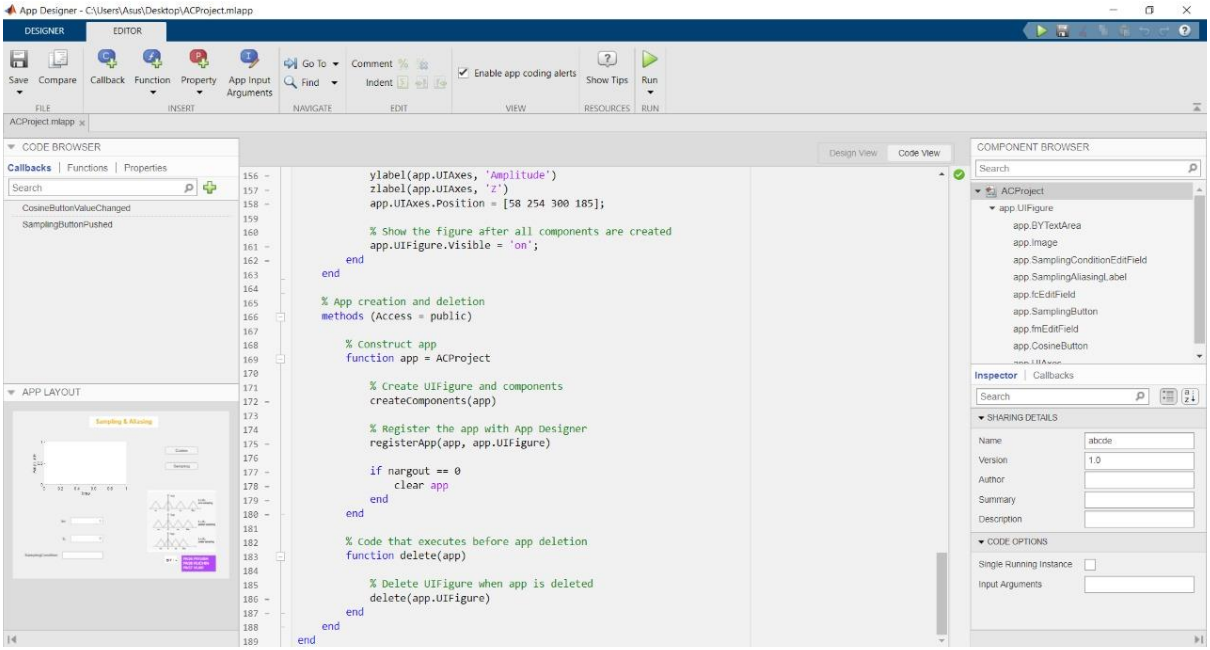

App Designer - C:\Users\Asus\Desktop\ACProject.mlapp
classdef ACProject < matlab.apps.AppBase
% Properties that correspond to app components
properties (Access = public)
UIFigure matlab.ui.Figure
CosineButton matlab.ui.control.StateButton
fmEditFieldLabel matlab.ui.control.Label
fmEditField matlab.ui.control.NumericEditField
SamplingButton matlab.ui.control.Button
fcEditFieldLabel matlab.ui.control.Label
fcEditField matlab.ui.control.NumericEditField
SamplingAliasingLabel matlab.ui.control.Label
SamplingConditionEditFieldLabel matlab.ui.control.Label
SamplingConditionEditField matlab.ui.control.EditField
Image matlab.ui.control.Image
BYLabel matlab.ui.control.Label
BYTextArea matlab.ui.control.TextArea
UIAxes matlab.ui.control.UIAxes
end
% Callbacks that handle component events
methods (Access = private)
% Value changed function: CosineButton
function CosineButtonValueChanged(app, event)
Am = 1;
fm = app.fmEditField.Value;
tmin = -0.05;
tmax = 0.05;
t = linspace(tmin, tmax, 400);
m = Am*cos(2*pi*fm * t);
plot(app.UIAxes,t,m);


App Designer - C:\Users\Asus\Desktop\ACProject.mlapp
DESIGNER
EDITOR
Save
Compare
Callback
Function
Property
App Input Arguments
Go To
Find
Comment
Indent
Enable app coding alerts
Show Tips
Run
FILE
INSERT
NAVIGATE
EDIT
VIEW
RESOURCES
RUN
ACProject.mlapp
CODE BROWSER
Callbacks
Functions
Properties
Search
CosineButtonValueChanged
SamplingButtonPushed
APP LAYOUT
Design View
Code View
end
% Button pushed function: SamplingButton
function SamplingButtonPushed(app, event)
fc = app.fcEditField.Value;
T = 1/fc;
fm = app.fmEditField.Value;
Am = 1;
tmin = -0.05;
tmax = 0.05;
nmin = ceil(tmin/T);
nmax = floor(tmax/T);
n = nmin:nmax;
m = Am*cos(2*pi*fm * n*T);
plot(app.UIAxes,n*T,m);
if 2*fm>fc
app.SamplingConditionEditField.Value= 'Aliasing Effect';
elseif 2*fm==fc
app.SamplingConditionEditField.Value= 'Perfect Sampling';
else
app.SamplingConditionEditField.Value= 'Over Sampling';
end
end
end
% Component initialization
methods (Access = private)
COMPONENT BROWSER
Search
ACProject
app.UIFigure
app.BYTextArea
app.Image
app.SamplingConditionEditField
app.SamplingAliasingLabel
app.fcEditField
app.SamplingButton
app.fmEditField
app.CosineButton
Inspector
Callbacks
SHARING DETAILS
Name
abcde
Version
1.0
Author
Summary
Description
CODE OPTIONS
Single Running Instance
Input Arguments

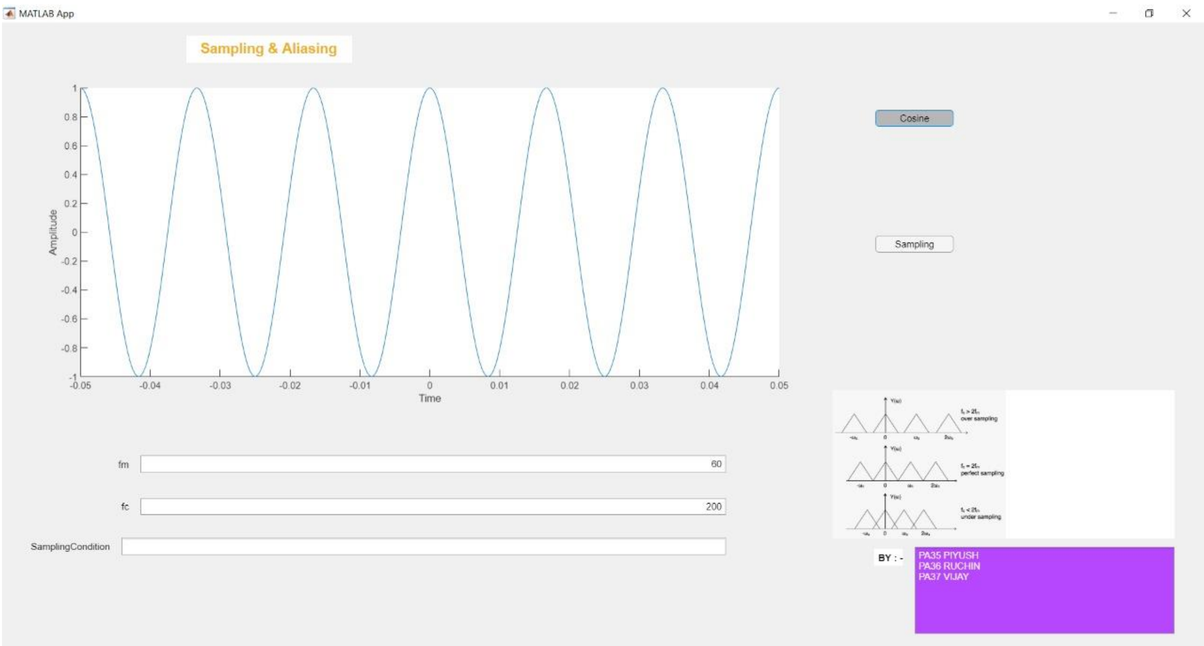

App Designer - C:\Users\Asus\Desktop\ACProject.mlapp
DESIGNER
EDITOR
Save
Compare
Callback
Function
Property
App Input Arguments
Go To
Find
Comment
Indent
Enable app coding alerts
Show Tips
Run
FILE
INSERT
NAVIGATE
EDIT
VIEW
RESOURCES
RUN
ACProject.mlapp
CODE BROWSER
Callbacks
Functions
Properties
Search
CosineButtonValueChanged
SamplingButtonPushed
APP LAYOUT
Design View
Code View
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.Position = [100 100 664 514];
app.UIFigure.Name = 'MATLAB App';
% Create CosineButton
app.CosineButton = uibutton(app.UIFigure, 'state');
app.CosineButton.ValueChangedFcn = createCallbackFcn(app, @CosineButtonValueChanged, true);
app.CosineButton.Text = 'Cosine';
app.CosineButton.Position = [469 383 100 22];
% Create fmEditFieldLabel
app.fmEditFieldLabel = uilabel(app.UIFigure);
app.fmEditFieldLabel.HorizontalAlignment = 'right';
app.fmEditFieldLabel.Position = [138 167 25 22];
app.fmEditFieldLabel.Text = 'fm';
% Create fmEditField
app.fmEditField = uieditfield(app.UIFigure, 'numeric');
app.fmEditField.Position = [178 167 100 22];
app.fmEditField.Value = 1;
% Create SamplingButton
app.SamplingButton = uibutton(app.UIFigure, 'push');
app.SamplingButton.ButtonPushedFcn = createCallbackFcn(app, @SamplingButtonPushed, true);
COMPONENT BROWSER
Search
ACProject
app.UIFigure
app.BYTextArea
app.Image
app.SamplingConditionEditField
app.SamplingAliasingLabel
app.fcEditField
app.SamplingButton
app.fmEditField
app.CosineButton
Inspector
Callbacks
SHARING DETAILS
Name
abcde
Version
1.0
Author
Summary
Description
CODE OPTIONS
Single Running Instance
Input Arguments


App Designer - C:\Users\Asus\Desktop\ACProject.mlapp
app.SamplingButton.Position = [469 335 100 22];
app.SamplingButton.Text = 'Sampling';
% Create fcEditFieldLabel
app.fcEditFieldLabel = uilabel(app.UIFigure);
app.fcEditFieldLabel.HorizontalAlignment = 'right';
app.fcEditFieldLabel.Position = [138 113 25 22];
app.fcEditFieldLabel.Text = 'fc';
% Create fcEditField
app.fcEditField = uieditfield(app.UIFigure, 'numeric');
app.fcEditField.Position = [178 113 100 22];
% Create SamplingAliasingLabel
app.SamplingAliasingLabel = uilabel(app.UIFigure);
app.SamplingAliasingLabel.BackgroundColor = [1 1 1];
app.SamplingAliasingLabel.HorizontalAlignment = 'center';
app.SamplingAliasingLabel.FontSize = 18;
app.SamplingAliasingLabel.FontWeight = 'bold';
app.SamplingAliasingLabel.FontColor = [0.9294 0.6941 0.1255];
app.SamplingAliasingLabel.Position = [236 465 211 34];
app.SamplingAliasingLabel.Text = 'Sampling & Aliasing';
% Create SamplingConditionEditFieldLabel
app.SamplingConditionEditFieldLabel = uilabel(app.UIFigure);
app.SamplingConditionEditFieldLabel.HorizontalAlignment = 'right';
app.SamplingConditionEditFieldLabel.Position = [33 62 106 22];
app.SamplingConditionEditFieldLabel.Text = 'SamplingCondition';
% Create SamplingConditionEditField
app.SamplingConditionEditField = uieditfield(app.UIFigure, 'text');
app.SamplingConditionEditField.Position = [154 62 124 22];
% Create Image


App Designer - C:\Users\Asus\Desktop\ACProject.mlapp
app.SamplingConditionEditField = uieditfield(app.UIFigure, 'text');
app.SamplingConditionEditField.Position = [154 62 124 22];
% Create Image
app.Image = uiimage(app.UIFigure);
app.Image.BackgroundColor = [1 1 1];
app.Image.HorizontalAlignment = 'left';
app.Image.Position = [414 83 210 189];
app.Image.ImageSource = 'WhatsApp Image 2021-06-09 at 4.49.46 PM.jpeg';
% Create BYLabel
app.BYLabel = uilabel(app.UIFigure);
app.BYLabel.BackgroundColor = [1 1 1];
app.BYLabel.HorizontalAlignment = 'right';
app.BYLabel.FontWeight = 'bold';
app.BYLabel.Position = [467 48 37 22];
app.BYLabel.Text = 'BY : -';
% Create BYTextArea
app.BYTextArea = uitextarea(app.UIFigure);
app.BYTextArea.FontColor = [1 1 1];
app.BYTextArea.BackgroundColor = [0.7176 0.2745 1];
app.BYTextArea.Position = [519 22 105 50];
app.BYTextArea.Value = {'PA35 PIYUSH'; 'PA36 RUCHIN'; 'PA37 VIJAY'};
% Create UIAxes
app.UIAxes = uiaxes(app.UIFigure);
xlabel(app.UIAxes, 'Time')
ylabel(app.UIAxes, 'Amplitude')
zlabel(app.UIAxes, 'Z')
app.UIAxes.Position = [58 254 300 185];
% Show the figure after all components are created



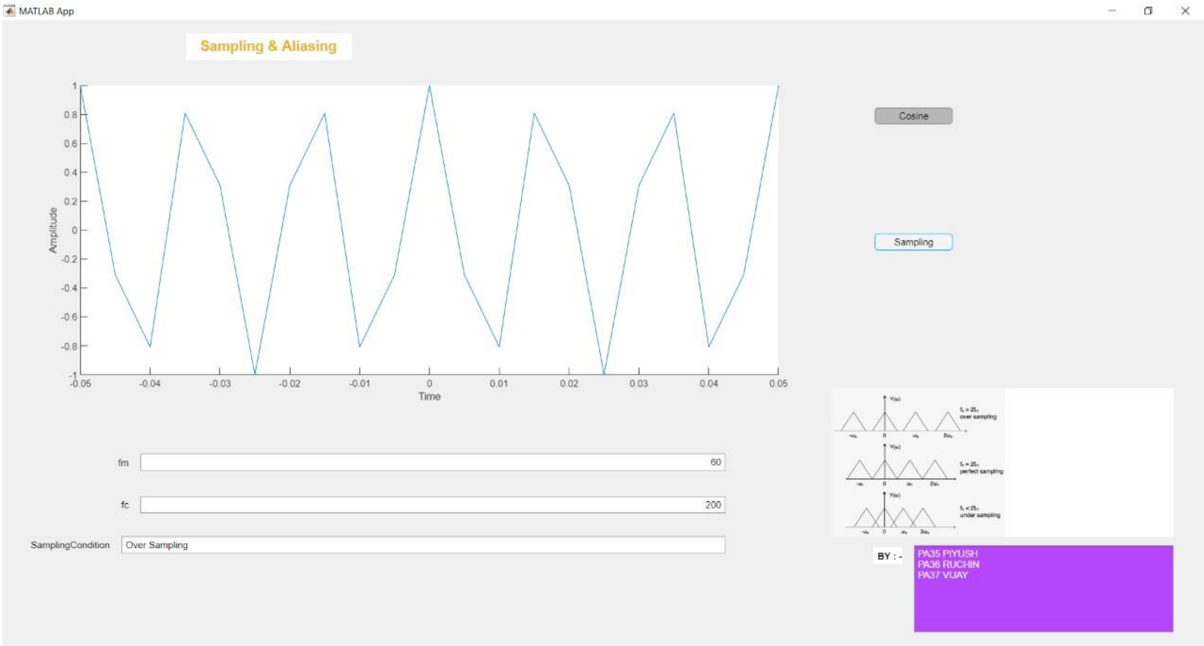
OUTPUT:



MATLAB App
Sampling & Aliasing
Amplitude
Time
Cosine
Sampling
fm
60
fc
200
SamplingCondition
BY : -
PA35 PIYUSH
PA36 RUCHIN
PA37 VIJAY

Case I:

$F_c > 2F_m$: Over Sampling

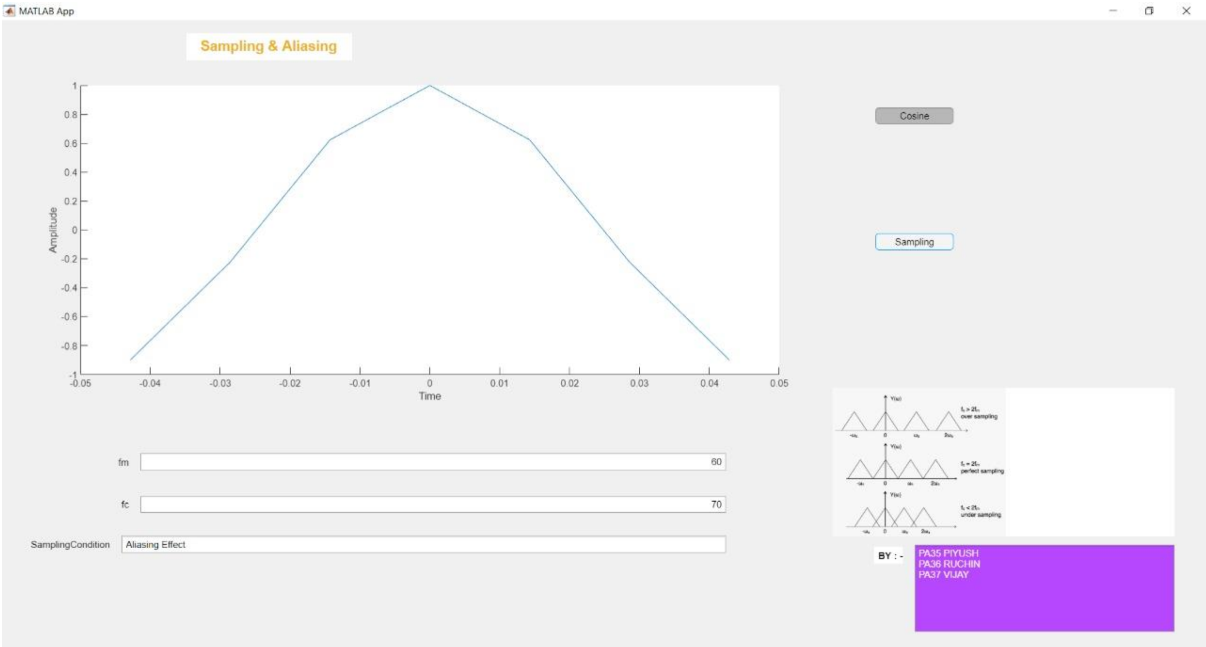


Case II:

$F_c = 2F_m$: Perfect Sampling

Case III:

$F_c < 2F_m$: Down Sampling (Aliasing Effect)



Specifications:

Cosine button: - Which will show the cos wave on axis.

Sampling button: - Which will show the different sampling conditions on axis.

Fc (Edit field numeric): - Which will take input of carrier frequency.

Fm (Edit field numeric): - Which will take input of message signal.

Axis: - Which will show output of our program.

Sampling Condition (Edit field Text): - Which will show the different sampling condition (Over Sampling, Perfect Sampling and Aliasing Effect).

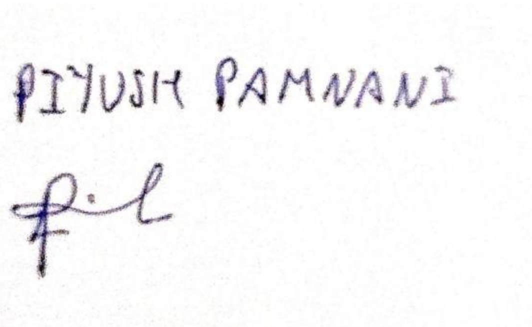
References:

References of the papers, books, pdfs, sites or any other material you referred before finalizing the topic should be given in the IEEE format.

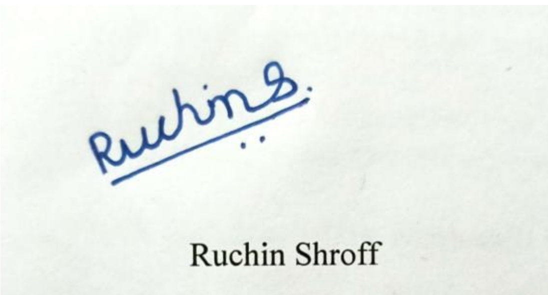
https://web.cs.wpi.edu/~matt/courses/cs563/talks/antialiasing/introduction.html
https://www.sciencedirect.com/topics/computer-science/sampling-theorem
https://www.tutorialspoint.com/digital_communication/digital_communication_sampling.htm

Signature of Students:

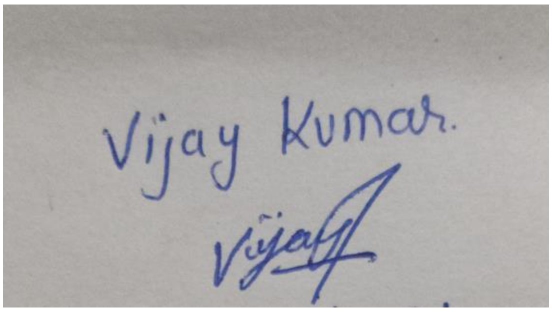
a) Piyush Pamnani:
   Roll No:35



b) Ruchin Shroff:
   Roll No:36



c) Vijay Choudhary:
   Roll No:37


Vijay Kumar.