Study of signal frequency, spectrum, bandwidth, bit rate, quantization using MATLAB

SUDIPTA KUMAR DAS

Department of Computer Science and Engineering

AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

October 2, 2022

CONTENTS

LIST OF FIGURES

1 ABSTRACT

This experiment specially helps to understand the use of MATLAB for solving communication engineering problem. By the help of this experiment we can understand how the data converts from digital to analog and the convert again from analog to digital. We can also understand the sampling theorem and the effect of sampling rate on the signal. And how the receiver device get back the original signal by the help of sampling and we can see it visually by graph. This experiment also helps to develop the basic knowledge of MATLAB environment, commands and syntax.

2 INTRODUCTION

The proprietary multi-paradigm programming language and numerical computing environment known as MATLAB were created by MathWorks. MATLAB is an acronym for ”MATrix LABoratory.” Matrix manipulation, function and data visualization, algorithm implementation, user interface building, and connecting with other programming languages are all possible with MATLAB.

We can use MATLAB for

- Analyze data
- Develop algorithms
- Create models and applications

MATLAB lets you take your ideas from research to production by deploying to enterprise applications and embedded devices, as well as integrating with Simulink® and ”Model-Based Design”.

An array that doesn’t need to be dimensioned is the fundamental data element in the interactive system known as Matlab. This saves the user time compared to writing programs in a variety of technical computer tasks, particularly those involving matrix and vector operations. scalar, inert programming languages like C or Fortran.

Toolboxes are a kind of application-specific solutions offered by Matlab. Being extremelyIt’s crucial for the majority of Matlab users that toolboxes enable learning and application of specializedtechnology. The so-called M toolboxes are extensive collections of Matlab routines.files that enhance the Matlab environment to address certain issue types.

3 PERFORMANCE TASK

Calculation :

Here, My ID is $20 - 43658 - 2$
The main equation is $x = Acos(2\pi ft)$
so we can write $x_1(t) = A_1cos(2\pi f_1t)$ and,
we can also write $x_2(t) = A_2cos(2\pi f_2t)$
Now,
$f_1 = CX10 = 4X10 = 40$
$f_2 = FX10 = 5X10 = 50$
$A_1 = GD = 8X3 = 24$
$A_2 = AXF = 2X5 = 10$
Sample-per-period = 20
f_s = FX(Sample-per-period) = $50X20 = 1000$
Here, there is several frequencies, in that case, we should consider the highest frequency so that we can get the maximum amount of sampling to get better wave signal.
$X_3 = X_1 + X_2$

Code :

```
%{
My ID = 20-43658-2
        AB-CDEFG-H
%}

%{ Setting up variables %}
f1 = 4*10
f2 = 5*10
A1 = 8*3
A2 = 2*5
fs = 50*20
t  = (0:(1/fs):1)

%{ Calculating x1 x2 & x3 %}
```

```
x1 = A1*cos(2*pi*f1*t)
x2 = A2*cos(2*pi*f2*t)
x3 = x1 + x2

%{ Plotting x1 x2 & x3 %}
subplot(4,1,1)
plot(t,x1,'g')
xlabel('Time')
ylabel('Amplitude')
title('X1')

subplot(4,1,2)
plot(t,x2)
xlabel('Time')
ylabel('Amplitude')
title('X2')

subplot(4,1,3)
plot(t,x3,'r')
xlabel('Time')
ylabel('Amplitude')
title('X3')

%{ Plotting Frequency Domain Graph of x3 %}
length = length(x3)
length_in_power = 2^nextpow2(length)

fx3 = fft(x3,length_in_power)
ffx3 = fx3(1:length_in_power/2)
xfft = fs*(0:length_in_power/2-1)/length_in_power

subplot(4,1,4)
plot(xfft,abs(ffx3),'k')
xlabel('Amplitude')
ylabel('Frequency')
title('Frequency Domain Graph of X3')
```

Figure 3.1. Performance Task

4 DISCUSSION

As it was an important experiment on the basic of matlab so we have taken the values in different variables. We breakdown in an atomic level and then we have to calculate the values of the variables for more clarification. We have calculate both calculator and the MATLAB so the possibility of error is zero.

5 CONCLUSION

As the name suggests MATLAB is a matrix laboratory. It is a high-level language and interactive environment that enables us to perform computationally intensive tasks faster than with traditional programming languages such as C, C++, and Fortran. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, C#, Java, Fortran and Python.