ETI 2506: TELECOMMUNICATION SYSTEMS LABS

Laboratory Exercise 1: Sampling and Quantization

PART 1: MATLAB INTRODUCTION

INTRODUCTION

During the laboratory sessions, we will use a high performance numerical computing environment named MATLAB (abbreviation for matrix laboratory). The MATLAB software is widely used in scientific research and industrial R&D.

The objective of this first part of the lab is to get you familiar with the MATLAB environment and its basic functionalities. For that purpose, please open the following web page.

https://www.mathworks.com/support/learn-with-matlab-tutorials.html

and study the items below. You must read the following sections of the MATLAB and Simulink Tutorials

Build a Foundation with Interactive Courses

- Quickly learn the essentials of MATLAB®

- Learn to create, edit, and troubleshoot Simulink® models.

Get Started with Introductory Videos

Getting Started with MATLAB

Getting Started with Simulink, Part 1: Building and Simulating a Simple Simulink Model

PART 2: SIGNAL GENERATION, SAMPLING AND QUANTIZATION

Exercise 2.1: Basic digital signals Generation

(a) Write a MATLAB program to generate and display (using the **stem** function) the signals defined in **Table 1**. The MATLAB code of the first signal (dirac) is given in the report template as an example.

(b) Write a MATLAB function **[x, t] = sin_NU(f0, fs, T)** to generate a sine signal. The output parameters x and t are the signal and time vectors, respectively. The input parameters are **f0** (signal frequency in Hz), **fs** (sampling frequency in Hz), **T** (signal duration in sec.).

(c) Test your **sin_NU** function with the input parameter values below and display the result using the **plot** function.

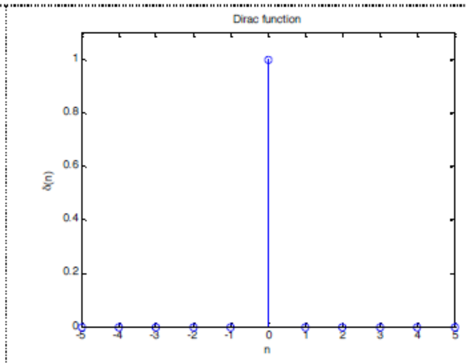
$$\{\ f_0 = 10, \quad f_s = 1000, \quad T = 0.5\ \}$$

Table 1: List of basic digital signals to generate.

Function	Equation	Parameters
Dirac (unit impulse)	$\partial(n) = \begin{cases} 1 & \text{if} \quad n = 0 \\ 0 & \text{otherwise} \end{cases}$	$n = -N/2, \ldots, N/2$ $N = 10$
Unit step (Heaviside step)	$u(n) = \begin{cases} 1 & \text{if} \quad n \geq 0 \\ 0 & \text{otherwise} \end{cases}$	$n = -N/2, \ldots, N/2$ $N = 10$
Sign	$x(n) = 2u(n) - 1$	$n = -N/2, \ldots, N/2$ $N = 10$
Rectangle	$\text{rect}_M(n) = u(n+M) - u(n-M)$	$n = -N/2, \ldots, N/2$ $N = 10, \quad M = 2$
Sine	$x(n) = x_0 \sin(2\pi f_0 n T_s)$	$n = 0, \ldots, L-1$ $L = 20,$ $f_0 = 100\,\text{Hz}, \quad 1/T_s = f_s = 1\text{kHz}$
Sine cardinal	$x(n) = \begin{cases} 1 & \text{for} \quad n = 0 \\ \dfrac{\sin(\pi n T_s)}{\pi n T_s} & \text{otherwise} \end{cases}$	$n = -L, \ldots, L$ $L = 50$ $T_s = 0.1\,\text{sec}$

As an example below is the code to generate the Dirac/Unit Impulse Signal as well as the displayed Dirac Function.

```
% Dirac
N = 10;            % number of samples
n = -N/2:N/2;      % vector
d = [zeros(1,N/2) 1 zeros(1,N/2)];
figure;            % display
stem(n,d);
xlabel('n');
ylabel('\delta(n)');
title('Dirac function');
axis([-N/2 N/2 0 1.1]);
```



Repeat the exercise for the other five signals in table 1.

Exercise 2.2: Audio aliasing

To illustrate the aliasing phenomenon, let's perform two simple experiments allowing us to "hear" it. Using the **sin_NU** function of Exercise 2.1:

(a) Generate two 1 kHz sine signals (2 seconds duration), first signal at 20 kHz sample frequency and second signal at 1.5 kHz sample frequency;

(b) On the same graph, use the **plot** function to display the two signals versus t in the range below:

$0 \leq t \leq 5$ msec.

(c) Listen to the two signals one after another using the function **soundsc(x, fs)**; and

(d) Give your interpretation of this listening.

The code below allows you generate the two signals for part a as the example. Proceed to answer parts b to d of exercise 2.2

```
T = 2; %parameters
f0 = 1000;
fs1 = 20000;
fs2 = 1500;
[x1, t1] = sin_NU(fs1,f0,T);
[x2, t2] = sin_NU(fs2,f0,T);
figure;
plot(t1,x1,t2,x2,'LineWidth',3.0),
axis([0, 0.005, -1.1, 1.1])
legend('High Frequency','Low Frequency')
xlabel('Time')
ylabel('Signals')
title('Audio aliasing');
%%%
soundsc(x1,fs1)
%%%
soundsc(x2,fs2)
```

Exercise 2.3: Quantization

Quantization is done by replacing each value of an analog signal x(t) by the value of the nearest quantization level. To exemplify this operation, let's simulate a unipolar ADC (Analog to Digital Converter) having the technical specifications: R = 10 Volts (full-scale range) and B = 3 (number of bits).

(a) Write a MATLAB function **y = adc_NU(x, R, B)** where x and y are vectors containing the input signal and the quantized signal, respectively;

(b) Test your function with an input ramp signal ranging from -5 to 15 Volts (1 volt per step).

(c) On the same graph, use the **plot** and **stem** functions to display the input signal and quantized signal, respectively.

The code below allows you answer question 2.3 a above. Work out the part b and c.

```
function y = adc_NU(x, R, B)
level = [0:R/(2^B):R-R/(2^B)];
temp = [-Inf,(level(2:end)-R/(2^(B+1))),Inf];
y = zeros(1,length(x));
for i = 1:length(level)
    y = y + (x >= temp(i)).*(x < temp(i+1)).*level(i);
end
```

Exercise:

1. Write a lab report for the above exercises and include your results and conclusions for the same.