Lab 1: Continuous-time Signals

Outline for lab report

- Title page: Title, name, course name, section number
- Abstract: 300 words maximum
- Objectives: List in your own words
- Introduction and Motivation
- Background material: Includes any theoretical background
- Results and Discussion: All tables and plots should have a title. All plots should be clearly labelled with axes and title. Analytical work should be clearly written and results/tables/figures should be discussed wherever necessary.
- Conclusion
- Appendices: Put listings of MATLAB code and any other Appendices you see fit.

Guidelines for good coding practice and lab report submission

- MATLAB code should be properly commented.
- Code should start with a title, lab number, version number, date and author.
- All functions and script purpose should be explained along with input and outputs.

Introduction

A continuous-time signal takes on a value at every point in time, whereas a discrete-time signal is only defined at integer values of the time variable. Whereas, discrete-time signals can be easily stored and processed on a computer, it is impossible to store the values of a continuous-time signal for all points along a segment of the real line.

How then do we process continuous-time signals? In this lab, we will show that continuous-time signals may be processed by first approximating them by discrete-time signals using a process known as sampling. We will see that proper selection of the spacing between samples is crucial for an efficient and accurate approximation of a continuous-time signal. Excessively close spacing will lead to too much data, whereas excessively distant spacing will lead to a poor approximation of the continuous-time signal.

You can refer the Mathworks documentation to learn about signal generation and processing using MATLAB.

https://www.mathworks.com/help/signal/gs/impulse-step-and-ramp-functions.html?searchHighlight=ramp&s_tid=doc_srchtitle

Lab Assignments

1. This part shows an example of how to write a script to plot a unit step response. The script for a generic unit step function is attached.
   function y = ustep(t,ad)

```
%****************************************************************
% Lab1: Continuous-time Systems
% Problem 1: Script to generate and plot unit step response
% Inputs: time support and sampling steps of the signal
% Output: unit step response
% Reference: Signals and Systems with Matlab, Chaparro Luis F.
%*****************************************************************

clear all;
clf
Ts = 0.01;    % Sampling time
t = -5:Ts:5; % support of signal

% unit-step function with support [-5,5], delayed by 3
y = ustep(t,-3);
plot(t,y);
axis([-5 5 -1 5]);
title('Unit step response');
xlabel('time (seconds)');
ylabel('y(t)');
grid

%****************************************************************
% Function to generate unit step
% Input : time interval, signal advance/delay factor
% Output: Unit step response
%*****************************************************************

function y = ustep(t,ad)
% generation of unit step
% t: time
% ad : advance (positive), delay (negative)

N= length(t);
y = zeros(1,N);
for i = 1:N,
    if t(i)>= -ad,
        y(i) = 1;
    end
end
end
```

2. Write a MATLAB script for a generic ramp function r(t) and plot for y = ramp(t, 3, 0). The variable t is in the interval -5 < *t* < 5 and increments in steps of 0.01.
   function y = ramp(t,m,ad)
   % t = time support, m = ramp slope, ad = signal advance(positive)/delay(negative)

3. Using the ustep and ramp functions, plot the following continuous-time signals for the interval -5 < *t* < 5. The variable t should increment in steps of 0.01.
   - 3u(t-2)
   - 4r(t+3)

4. Write a script to generate a signal *y(t)* that uses the functions ramp and ustep,

$$y(t) = 3r(t+3) - 6r(t+1) + 3r(t) - 3u(t-3)$$

   Plot the signal *y(t)* in the interval -5 < *t* < 5. The variable t should increment in steps of 0.01. Calculate analytically y(1) and y(4) and verify that these values were obtained on your plot.

5. Consider the following script that uses the functions ramp and ustep to generate a signal *y(t).* Use the 'help' facility to understand any of the MATLAB commands.

```
clear all;
clf;
t = -5:0.01:5;
y1 = ramp(t,2,2.5);
y2 = ramp(t,-5,0);
y3 = ramp(t,3,-2);
y4 = ustep(t,-4);
y = y1 + y2 + y3 + y4;
plot(t,y,'k');
axis([-10 10 -3 5]);
grid
```

   a. Obtain analytically the formula for the signal *y(t).*
   b. Show that *y(t) = 0* for *t* < -5 and *t* > 5 both analytically and by a plot of y(t)..
   c. Write a function to compute and plot the even and odd components of *y(t).*

   HINT: For even/odd decomposition, create a function:

```
function [ye,yo] = evenodd(y)
yr = fliplr(y);
ye = 0.5 * (y + yr);
yo = 0.5 * (y - yr);
```

6. An acoustic signal x(t) has a duration of 3.3 minutes and a radio station would like to use the signal for a 3-minute segment. Explain how this can be done. You do not have to program it up.

7. Consider the signal x(t) = cos(πt) + cos(2πt/3). Show that x(t) is periodic, find the period of x(t) and plot x(t) in the interval -10 < t < 10 to verify your claim.