LAB 2: System Properties and Convolution

- **Report Due:** Sunday, October 21, 2018, 11:59pm.

Objective

In this lab assignment you will learn how to use M-files in MATLAB to exercise **convolution** and investigate system properties.

Convolution

In this course we introduce and work with continuous-time convolution operation. However, there is one small problem: computers are digital and thus time and signals cannot be represented *continuously*; we can only deal with samples of a signal at discrete points in time. For our purposes here, we can use a very large number of these time points to make our signal *appear* continuous and thus provide an accurate representation of the underlying continuous-time signal. Specifically, we will approximate the convolution integral with summation. Section 2.4 in the textbook explains the steps in developing the convolution integral and arrives at the equation:

$$y(t) = \lim_{\Delta\tau \to 0} \sum_{n} x(n\Delta\tau)h(t - n\Delta\tau)\Delta\tau; \tag{1}$$

$$= \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau. \tag{2}$$

We can approximate the continuous-time convolution integral given in Equation (2), in MATLAB by implementing the summation in Equation (1) (without the limit, of course). To get started, the sample code given in Listing 1 creates signals $x(t)$ and $h(t)$, generates $y(t) = x(t) * h(t)$ and then plots all three signals. You can save this code into a text file and give it a unique name and an ".m" extension. For example, save the code as **`lab2conv.m`**. You can now run this M-file in MATLAB by entering the command **`lab2conv`** at the MATLAB prompt.

Preparation

Read *Lathi, Chapter 2*, sections 2.1-2.5. You *need* to understand the concept of convolution and master the skills of how to solve the convolution integral analytically before you can productively work on this lab assignment.

Lab Assignment

A. Impulse Response

Problem A.1 [0.5 Marks] Complete *Lathi, Section 2.7-1 Script Files*, page 213. Use MATLAB command **`poly`** to generate the characteristic polynomial from the characteristic values specified by **`lambda`**.

Problem A.2 [2 Marks] Plot the impulse response of the system in Problem A.1 for **`t = [0:0.0005:0.1]`**.

Problem A.3 [0.5 Marks] Complete *Lathi, Section 2.7-2 Function M-Files*, page 214.

B. Convolution

Problem B.1 [3 Marks] *Lathi, Section 2.7-4 Graphical Understanding of Convolution*, page 217. Plot $y(t)$ at step $t = 2.25$ as shown in Figure 2.28 on page 219. Use the MATLAB command `pause` instead of `drawnow` to observe the steps of the convolution operation slowly.

Problem B.2 [4 Marks] Perform the convolution of the signal $x(t)$ in **Figure P2.4-28 (a) (page 229)** with $h(t)$ in **Figure P2.4-30 (page 230)**. Plot all signals and results.

Problem B.3 [5 Marks] Perform the convolution of the signal $x_1(t)$ and $x_2(t)$ in **Figure P2.4-27(a)**, **(b)** and **(h)**. Plot all signals and results.

C. System Behavior and Stability

Problem C.1 [4 Marks] Consider the LTI systems S1, S2, S3 and S4 represented by their respective unit impulse response functions given as follows:

$$h_1(t) = e^{\frac{t}{5}} u(t); \tag{3}$$

$$h_2(t) = 4e^{-\frac{t}{5}} u(t); \tag{4}$$

$$h_3(t) = 4e^{-t} u(t); \tag{5}$$

$$h_4(t) = 4\big(e^{-\frac{t}{5}} - e^{-t}\big) u(t); \tag{6}$$

Plot each unit impulse response function for `t = [-1:0.001:5]`.

Problem C.2 [1 Mark] Determine the characteristic values (eigenvalues) of systems S1–S4.

Problem C.3 [5 Marks] Truncate the impulse response functions $h_1(t), \ldots, h_4(t)$ such that they are nonzero only for $0 \leq t \leq 20$. Determine the convolution of the truncated impulse response functions with the input signal $x(t) = \big[u(t) - u(t-3)\big] \sin 5t$ using the M-file in **Problem B.1** with the following changes `tau = [0:dtau:20]` and `tvec = [0:0.1:20]`. Plot the output of each system. State and explain your observations. Is there any relationship between the outputs of systems S2, S3, and S4? Explain.

D. Discussion

Problem D.1 [2 Marks] Calculate the results of **Problems B.1, B.2** and **B.3** above by hand and compare to those obtained with your MATLAB code.

Problem D.2 [1 Mark] What can you say about the width/duration of the signal resulting from the convolution of two signals?

```matlab
% sample code lab2conv.m
% create equally spaced time intervals
t1      = -10;
t2      = 10;
N       = 2000 ;
Delta_t = (t2-t1)/N;
t       = t1:Delta_t:t2;

% create x(t)
x = zeros(size(t));
x(find(t>=-1 & t<=1)) = 1;

% create h(t)
% NOTE: The for loop with the nested if statement can be
%       replaced by the compact code:
% h = t; h( [t<-1]|[t>1] ) = 0;
for i=1:length(t)
    if (t(i) < -1 || t(i) > 1)
       h(i)=0;
    else
       h(i) = t(i);
    end
end

% need to multiply by Delta_t as in the sum in Eq. 1a
x1 = x*Delta_t;

% perform convolution
y = conv(x1,h);

% plot results
subplot (3,1,1); plot(t,x); axis([t1 t2 -0.1 1.1])
% add title and axis label
title ('x(t)'); xlabel('t');

subplot (3,1,2); plot(t,h);
title ('h(t)'); xlabel('t');

% to plot arrays (variables) must be equal in size
% In our case y is longer than t, (use size() to see),
% so we only take the middle 2001 values;
% Play with this to see what happens with
% a different range
subplot (3,1,3); plot(t,y(1000:3000));
title ('y(t)'); xlabel('t');
```

Listing 1: Sample convolution code.