EE3TP4: Signals and Systems
Lab 2: Impulse Response and Convolution

Tim Davidson
Ext. 27352
davidson@mcmaster.ca

Objective

To calculate and interpret convolution sums and the pulse response of continuous-time linear time-invariant systems.

Report

A report for this lab is to be handed in between 2:00pm and 3:30pm on Friday 26th October 2001 in CRL-203. In the report you should answer all questions, derive the requested mathematical expressions, interpret them, and demonstrate how they describe the (computer) experiments.

1. Numerical Computation of the Convolution Sum

An important property of a linear time-invariant system is that its output is determined by the input and its impulse response (in the absence of any initial conditions). If $h[n]$ is the response of a discrete-time linear time-invariant system to a unit impulse $\delta[n]$, then

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]\, h[n-k]$$

is the system's response to an input $x[n]$. In this section we will write a short Matlab function file which computes $y[n]$ using the graphical method suggested in lectures and the text. That is, for each value of n we will compute $w_n[k] = x[k]\, h[n-k]$ for all (necessary) values of k, and then compute $y[n] = \sum_k w_n[k]$.

(a) Download the file `my_conv_skel.m` from the course web site and save it as `my_conv.m` in your working directory. The file's URL is `http://www.ece.mcmaster.ca/~davidson/EE3TP4/labs/my_conv_skel.m`.

(b) This file contains a skeleton of a Matlab function which will perform convolution using the graphical method. Follow the specifications and instructions in that file and complete the program.

(c) Analytically compute the responses $y_1[n]$ and $y_2[n]$ of a system with impulse response

$$h_1[n] = \begin{cases} 1 & 0 \leq n \leq 5 \\ 0 & \text{otherwise} \end{cases}$$

to inputs

$$x_1[n] = \begin{cases} 1 & 0 \leq n \leq 5 \\ 0 & \text{otherwise} \end{cases} \qquad \text{and} \qquad x_2[n] = \begin{cases} n & 0 \leq n \leq 5 \\ 0 & \text{otherwise} \end{cases}$$

respectively. Provide a formula for both $y_1[n]$ and $y_2[n]$, and a sketch.

(d) Use your completed program to compute $y_1[n]$ and $y_2[n]$ numerically. Plot each output sequence against n and verify that your analytic and numerical results agree.

(e) MATLAB has an internal function `conv` which computes the convolution of two vectors. Type `help conv` for information on how to use it. Note that the `conv` function uses a different algorithm from your program, and does not produce output times. Use `conv` to verify your results from parts (c) and (d).

(f) Analytically compute

$$y_3[n] = \sum_k x_3[k]h_1[n-k], \qquad y_4[n] = \sum_k x_3[k]h_2[n-k],$$
$$y_5[n] = \sum_k x_3[k]h_3[n-k], \qquad y_6[n] = \sum_k x_4[k]h_4[n-k],$$

where

$$x_3[n] = -x_2[n-3], \qquad h_2[n] = h_1[n-4],$$
$$h_3[n] = h_1[n+2],$$
$$x_4[n] = \delta[n] + \delta[n-2], \qquad h_4[n] = 2\delta[n+2] - 2\delta[n-1].$$

(g) Use your own function `my_conv` and MATLAB's internal `conv` function to numerically compute $y_3[n]$, $y_4[n]$, $y_5[n]$ and $y_6[n]$. Verify that both numerical methods yield the same result as the analytic method.

(h) When performing the analytical convolutions above, it is usually helpful to draw pictures of $x[k]$, $h[n-k]$, $w_n[k]$ and $y[n]$. I have written a MATLAB function which will help you do this. (The function is my first attempt to produce a MATLAB equivalent of what is available as a Java script on the 'Joy of Convolution' web site at Johns Hopkins University. That site is linked from the course web site.) Download the file `TND_conv.m` from the course web site and save it in your working directory. The file's URL is `http://www.ece.mcmaster.ca/~davidson/EE3TP4/labs/TND_conv.m`.

(i) Type `help TND_conv` to see the instructions for using this function. (You may need to type `more on` to avoid the instructions scrolling off the screen.)

(j) Repeat the numerical computations in parts (d) and (g) using `TND_conv`, with `plots=1` and `e_zeros=3`. Press `<Return>` to update the graphs and compute $y[n]$ for the next value of n. Be aware that the index on the horizontal axis of the first three plots is k, whereas that of the last plot is n. This becomes clear if you expand the figure window to full-screen size. (*Warning:* It will be quite difficult to steal parts of this code to complete `my_conv.m` in part (a). This is because the indexing is quite different due to the extra zeros which have been added in order to produce nice plots.)

2. Pulse Response of a Continuous-Time Linear Time-Invariant System

If $h(t)$ is the response of a continuous-time linear time-invariant system to a unit impulse $\delta(t)$, then

$$y(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)\,d\tau$$

is the system's response to an input $x(t)$. The impulse response can be closely approximated by the response of a system to a short pulse of unit area. By 'short' we mean that the pulse is much shorter than the time constant(s) of the system.

(a) Consider the system described by the first-order differential equation:

$$\frac{dy(t)}{dt} + 3y(t) = x(t), \tag{1}$$

where $x(t)$ is the input and $y(t)$ is the output. For this system:

i. Use techniques from your second-year courses to show that the 'step response', that is the response to $u(t)$, is $s(t) = 1/3 - e^{-3t}/3$ for $t \geq 0$ and 0 for $t < 0$.

ii. Since the system is linear and "nice", differentiating the input results in an output which is the derivative of $s(t)$. Use the fact that $\frac{du(t)}{dt} = \delta(t)$ to show that the impulse response is $h(t) = e^{-3t}$ for $t \geq 0$ and 0 for $t < 0$.

iii. Plot both $h(t)$ and $s(t)$ as a function of t. Use a range of t from -1 to 4 seconds at intervals of 0.05 seconds.

(b) Obtain the file `lsim_lab2.m` from the course web site and save it in your working directory. The file's URL is `http://www.ece.mcmaster.ca/~davidson/EE3TP4/labs/lsim_lab2.m`. The MATLAB command `y=lsim_lab2(x,t)` will simulate the output of the system in (1) to an input $x(t)$. The vector `t` contains the (uniformly spaced) sampling times, the vector `x` contains the samples of $x(t)$ at those times, and the vector `y` contains the samples of the output at those times.

(c) Using `lsim_lab2` simulate the responses of the system to the input $x(t) = p_\epsilon(t)$, where

$$p_\epsilon(t) = \begin{cases} 1/\epsilon & -\epsilon/2 < t \leq \epsilon/2, \\ 0 & \text{otherwise.} \end{cases}$$

Choose $\epsilon = 0.4, 0.2$ and 0.1, and choose `t=[-1:0.05:4]`. Save the outputs in vectors `d1`, `d2` and `d3`. Note that you will need to construct an input vector `x` whose ith element is $p_\epsilon(-1 + 0.05(i-1))$.

(d) On separate figures plot `d1`, `d2` and `d3` against `t`. On each graph, include a plot of your analytic calculation of $h(t)$ from part (a)i. (Type `help hold` for information on how to plot multiple curves on one graph.) Comment on the quality of the approximations. How could you make the approximation more accurate? Try it, and plot the results.

(e) Why is it necessary that $p_\epsilon(t)$ has unit area? Try plotting the response to

$$q_\epsilon(t) = \begin{cases} 1 & -\epsilon/2 < t \leq \epsilon/2, \\ 0 & \text{otherwise.} \end{cases}$$

Does the response approximate $h(t)$? Why or why not? Discuss the consequences.