



ECE 3780 Laboratory 2

Convolutions

You are given a series of Matlab files that can run convolutions. Open Matlab and open and run the file cconvdemo.m

You can watch the posted video Lab2Part1 for more instructions.

A second video (Lab2Part2) shows the convolution of $x(t) = [u(t)-u(t-2)]$ with $h(t) = \exp(-0.5t)[u(t)-u(t-5)]$ solving the integral $c(t) = \int_{-\infty}^{\infty} h(\tau)x(t-\tau)d\tau$

Problem 1

With the use of the program (Matlab files), solve $c(t) = \int_{-\infty}^{\infty} h(\tau)x(t-\tau)d\tau$ for the same two signals. Obtain the mathematical expressions as the example in the video.

Problem 2

With the use of the program, solve $c(t) = \int_{-\infty}^{\infty} h(\tau)x(t-\tau)d\tau$ for $x(t) = [u(t-2)-u(t-4)]$ with $h(t) = [u(t+5)-u(t+1)]$. Obtain the mathematical expressions as the example in the video.

Problem 3

With the use of the program, solve $c(t) = \int_{-\infty}^{\infty} h(\tau)x(t-\tau)d\tau$ for

- (a) $x(t) = [u(t-2)-u(t-4)]$ with $h(t) = \delta(t+1)$
- (b) $x(t) = [u(t-2)-u(t-4)]$ with $h(t) = \delta(t-2)$

Problem 4

Signals with echoes.

First part

- ✓ Create in Matlab a vector of 100 samples with value zero. Then change the value of the first 10 samples to 1.
- 1. This will be a rectangular pulse and your signal of interest. $\underline{x(t)}$
 - ✓ Change the values between 50 to 59 to 0.5. This will be your echo of the rectangular pulse delayed by 50 or 49 samples - depending on your Matlab implementation, with an attenuation of 0.5. Plot both vectors and verify your delay and amplitude distortion.
 - Create a filter as:

$$\bullet \quad h(t) = \sum_{k=0}^N -\alpha^k \delta(t - kt_d) \quad ! \text{ the filter} \quad (1)$$

where td is the delay and α is the attenuation value. To perform convolutions between the original signal and the filter use the `conv` command. Plot the first 100 samples of the convolution. Can you eliminate the echo?

- Convolution is the operation that determines the output of a Linear Time Invariant system given any input with the system impulse response. This part emphasizes the importance of these concepts (linearity, convolution, impulse response). This part of the lab has been implemented in the discrete time. Designing digital filters to eliminate signal distortions such as echoing requires knowledge of the Fourier Transform as well as the Z transform.
- Do you have better results as you increase N . Explain.
- You were given the values of td and α . In a more real scenario, you may not know this. Use your filter with different values of td and α and see the results.
- Having seen the results above, would you say that computing the energy of the output of your filter $y(n) = x(n) * h(n)$ given by: $E_y = \sum_{n=0}^N y(n)^2$ where N is the length of the signal for different values of td and α in your filter can be a way to determine these parameters?
- Would you find the minimum or maximum energy at the correct values?
- Which parameter (delay or attenuation) is more accurate? Plot the energies for these different values using the `mesh` command. You will have a 2D matrix at that time with the x axis indicating the delay values and y axis indicating the attenuation values (or vice versa depending on your implementation). Try to label the axis of the mesh with the different values of the attenuation and delays. Use the mouse to move the mesh plot and find the position of the minimum and maximum values of energy.

Problem 5

In order to determine the delay value on an echo signal, you can compute the auto-correlation. You have two options for doing this, use the command `xcorr` or use the `conv` command. If you use the convolution command, what do you need to do in order to have the same result as using `xcorr`?

Start with a simple signal, lets say

`x = [0 0 0 2 3 1 0 0 0 0 0 1 1.5 0.5 0 0 0 0];`

Plot it using the `stem` command. These samples contain an original signal and its repetition separated by 8 samples (that is $N = 8$). How does the cross-correlation looks like? How many peaks do you see and why? What is the distance between these peaks?

Here is another example that can help you understand convolutions:

