**Department of Electrical Engineering**

|  |  |
| --- | --- |
| **Faculty Member:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Dated: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |
|  |  |
| **Course/Section:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Semester: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |
|  |  |

**EE-232 Signals and Systems**

**Lab #6 Convolution**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Reg. no.** | **Report Marks / 10** | **Viva Marks / 5** | **Total/15** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Lab6: Convolution**

**Objectives**

The goal of this exericse is to gain familiarity with the convolution fuction provided in MATLAB and to write a function that would perform the convolution operation. The convolution examples considered in this lab will relate only to the case of discrete time signals.

* How to use the convolution operator in MATLAB
* How to write a function that will perform convolution operation
* Applications of Convolution Operator
* How to use spfirst convolution GUIs

**Lab Instructions**

* This lab activity comprises of three parts: Pre-lab, Lab Exercises, and Post-Lab Viva session.
* The students should perform and demonstrate each lab task separately for step-wise evaluation (please ensure that course instructor/lab engineer has signed each step after ascertaining its functional verification)
* Only those tasks that completed during the allocated lab time will be credited to the students. Students are however encouraged to practice on their own in spare time for enhancing their skills.

**Lab Report Instructions**

All questions should be answered precisely to get maximum credit. Lab report must ensure following items:

* Lab objectives
* MATLAB codes
* Results (graphs/tables) duly commented and discussed
* Conclusion

# Introduction to Systems

## Pre-Lab

The output y[n] is given by the convolution of the input x[n] with the system impulse response h[n]. This operation can be written as the convolution sum:

Again, the shorthand notation is y[n] = x[n] \* h[n]

Although it is not difficult to write a program to evaluate the convolution integral/sum explicitly, an alternate method called graphical convolution can provide valuable insight into the convolution process.

Using the computer-aided graphical approach, it is often possible to visualize the result of a convolution by inspecting the input sequences. The method involves properly aligning the two sequences on a common time axis, multiplying them, and then adding sequences values. The following exercises introduce convolution and the associated concepts of time-invariant system and impulse response with computer simulation and real-time measurement.

1. Calculate the result of convolving x[n]= {1 2 1 2} with h[n]= {0 1 2 3 4 1 1}.
2. Calculate the result of convolving x[n]= {1 2 1 2} with h[n]= {0 1 2 3 4 1 1}.
3. Calculate the result of convolving x[n]= {-1 0 1} with h[n]= {1 2 1}.

### Lab Exercise: Convolution of sequences

In this section, we will apply Equation (3) to calculate the convolution of two sequences and to examine one of the important properties of convolution.

Given the following two sequences:

x[n]= {1,2,1,2} h[n]= {0,1,2,3,4,1,1} Where \_ indicates the zero position.

**a)** Write a Matlab function ‘***my\_conv***’ that will convolve the signal x[n] with the system impulse response h[n] and produce the output y[n]. Plot the output y[n] on a graph with correct axis.

**b)** Compare your result with built in function of Matlab ‘***conv***’.

**c)** If x[n] starts from -1 and h[n] starts from -2 then what will be the result of convolution using ‘***my\_conv***’ and ‘***conv***’? Is the result of ‘***my\_conv***’ similar to the result you get on paper? If not, how will you get the correct result with respect to position of signal values. (Note: You have to make time vector to obtain correct plotting on Matlab).

**d)** Convolution of x[n]={-1 0 1} with h[n]={1 2 1} results in y[n] ={-1 -2 0 2 1}. Verify this by using the function **‘*my\_conv*’.**

# Convolution GUIs

## Pre-Lab

This lab concentrates on the use of two MATLAB GUIs for convolution.

* **dconvdemo:** GUI for discrete-time convolution. This is exactly the same as the MATLAB functions conv() and firfilt() used to implement FIR ﬁlters.
* **cconvdemo:** GUI for continuous-time convolution.

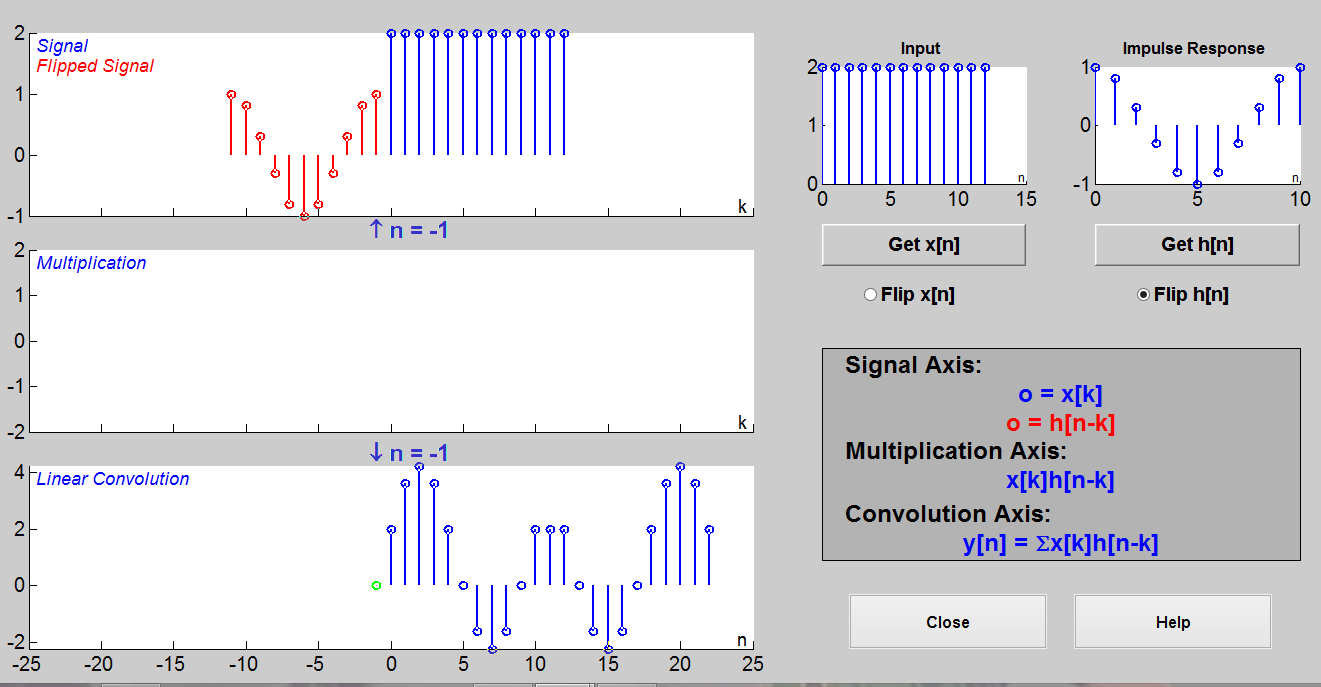
Each one of these demos illustrates an important point about the behavior of a linear, time-invariant (LTI) system. They also provide a convenient way to visualize the output of a LTI system. Both of these are on the SP First toolbox .You already know the procedure of downloading and adding the spfirst toolbox to MATLAB.

## Lab exercises

The objective of this lab part is to demonstrate usage of two convolution GUIs.

**2.2.1 Discrete-Time Convolution Demo**

In this demo, you can select an input signal x[n], as well as the impulse response of the ﬁlter h[n]. Then the demo shows the “ﬂipping and shifting” used when a convolution is computed. This corresponds to the sliding window of the FIR ﬁlter. Figure 1 shows the interface for the dconvdemo GUI.



**Figure 1: Interface for discrete-time convolution GUI.**

In this Lab exercise, you should perform the following steps with the dconvdemo GUI.

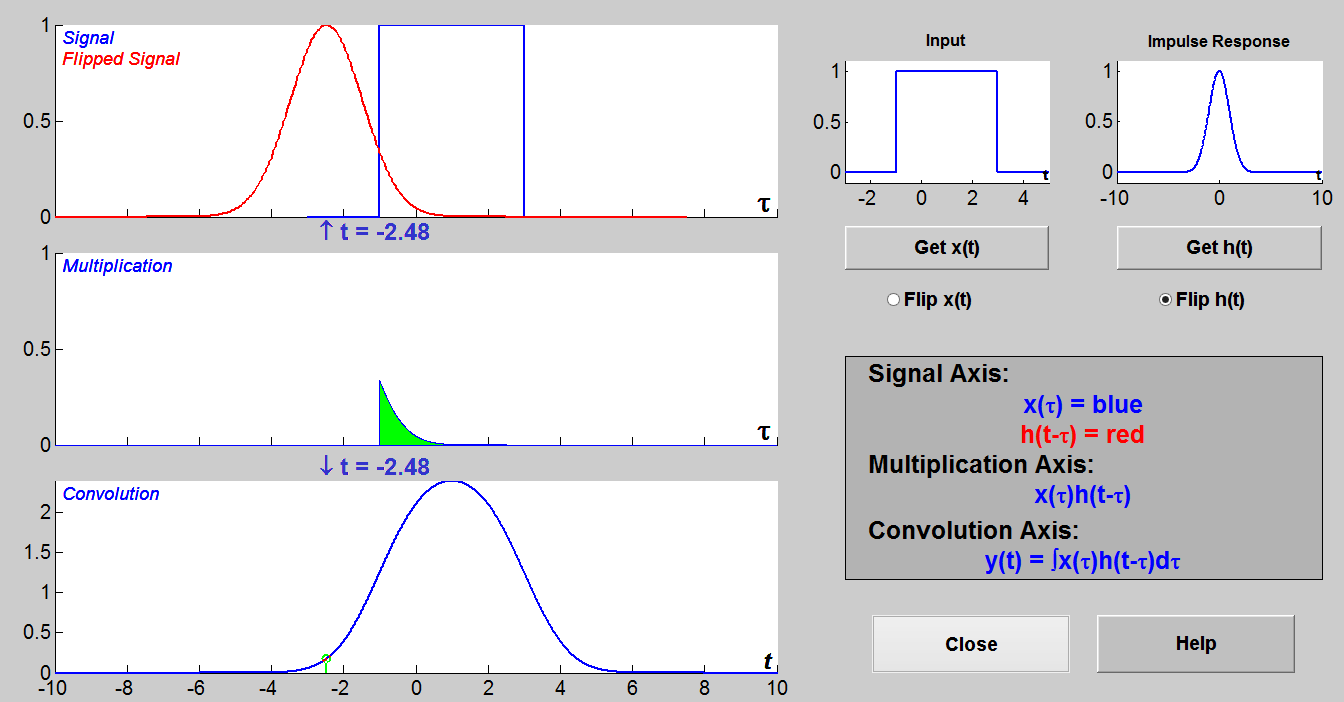
1. Set the input to a ﬁnite-length pulse: x[n] = 2 {u[n] - u[n - 12]}.
2. Set the ﬁlter’s impulse response as a cosine function with equation h[n]= cos (0.2\*pi\*n \* w[n]). Where w[n]= u[n]-u[n-11];
3. Use the GUI to produce the output signal that should be same as figure 1.
4. When you move the mouse pointer over the index “n” below the signal plot and do a click-hold, you will get a hand tool that allows you to move the “n”-pointer. By moving the pointer horizontally you can observe the sliding window action of convolution. You can even move the index beyond the limits of the window and the plot will scroll over to align with “n.”
5. Set the ﬁlter’s impulse response to a length-10 averager, i.e., h[n] = 0.1{u[n] - u[n - 10]}. Use the GUI to produce the output signal.
6. Set the ﬁlter’s impulse response to a shifted impulse, i.e., h[n] = delta[n - 3]. Use the GUI to produce the output signal.
7. **Compare the outputs from parts (c), (e) and (f). Notice the different shapes (triangle, rectangle or trapezoid), the maximum values, and the different lengths of the outputs.**

**2.1.2 Continuous-Time Convolution Demo**

In this demo, you can select an input signal x(t), as well as the impulse response of an ANALOG ﬁlter h(t). Then the demo shows the “ﬂipping and shifting” used when a convolution integral is performed.

In the Lab exercise, you should perform the following steps with the cconvdemo GUI.

1. Set the input to a pulse x(t) = u(t+1) - u(t - 3).
2. Set the ﬁlter’s impulse response as h(t)=e-0.5t^2 [u(t-5)- u(t-5)]
3. Use the GUI to produce the output signal that should be same as figure 2. Use the sliding hand tool to grab the time marker and move it to produce the ﬂip-and-slide effect of convolution.
4. Set the ﬁlter’s impulse response to a 4-second pulse with amplitude 0.25, i.e., h(t) = 0.25{u(t) u(t- 4)}. Use the GUI to produce the output signal.
5. Set the ﬁlter’s impulse response to a shifted impulse, i.e., h(t) = delta(t - 3). Use the GUI to produce the output signal.
6. **Compare the outputs from parts (c), (d) and (e). Notice the different shapes (triangle, rectangle or trapezoid), the maximum values, and the different lengths of the outputs.**



**Figure 2: Interface for continuous-time convolution GUI.**