GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL of ELECTRICAL and COMPUTER ENGINEERING

ECE 2025 Fall 2000
Lab #9: Three GUIs

Date: 31 Oct.–6 Nov. 2000

This is *the official* Lab #9 description.

The Warm-up section of each lab must be completed in Lab and the steps marked *Instructor Verification* must also be signed off **during the lab time**.

The lab report for this lab will be **informal.** Discuss your results from section 3.

The lab report will be **due during the week of 7–13 Nov. at the start of your lab**.

1 Introduction

This lab concentrates on the use of three MATLAB GUIs: two for convolution and one for frequency response.

1. **LTIdemo:** GUI for frequency response of discrete-time filter. An input sinusoid is transformed to an output sinusoid.
2. **convdemo:** GUI for discrete-time convolution. This is exactly the same as the MATLAB functions `conv()` and `firfilt()` used to implement FIR filters.
3. **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 sinusoidal response of LTI systems in two closely related cases: (1) an infinitely long sinusoid that extends over the range $0 \leq n < \infty$, and (2) a one-sided sinusoid that has a starting point (e.g., at $n = n_0$). In the second case, the signal is zero for $n < n_0$ and the filter's output will exhibit a start-up *transient* at the beginning of the output before settling into a *steady-state* behavior that is sinusoidal.

All of these demos can be downloaded from Jordan Rosenthal's web page:

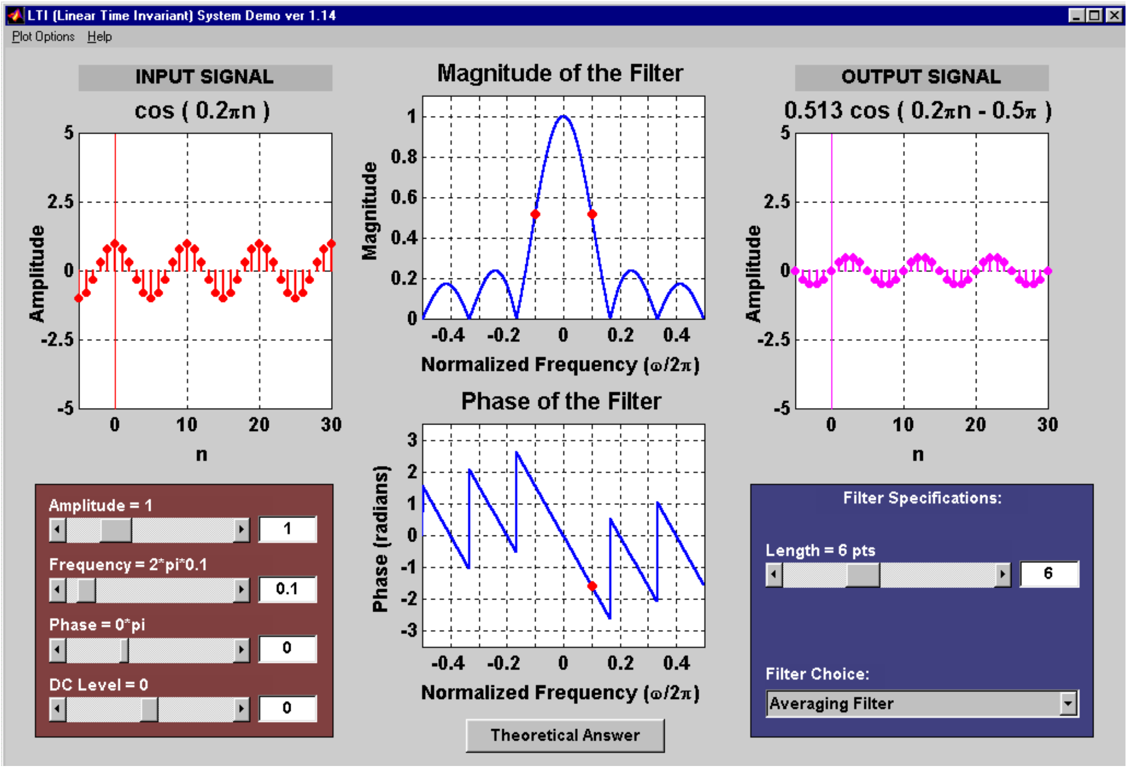
`http://users.ece.gatech.edu/jr/MatlabGUIs/matlabguis.html`

2 Warm-up: Run the GUIs

Three GUIs have been introduced during lectures over the past few weeks. The first objective of this lab is to demonstrate usage of each of these. You must download the ZIP files for each and install them. Each one installs as a directory containing a number of files. It is best to run the GUIs from their home directories. If you put both of the convolution GUIs on the `matlabpath` there will be conflicts with file names, so you should not add these directories to the `matlabpath`.

2.1 LTI Demo

In this demo, you can select an input signal that is a sinusoid, and see the change created by the frequency response. This demo reinforces the concept that “sinusoid in gives sinusoid out.” Figure 1 shows the interface for the LTI demo.



cos (0.2πn)

0.513 cos (0.2πn - 0.5π)

Amplitude = 1   1

Frequency = 2*pi*0.1   0.1

Phase = 0*pi   0

DC Level = 0   0

Length = 6 pts   6

Filter Choice: Averaging Filter

Figure 1: LTI demo interface.

In the warm-up, you should perform the following steps with the LTI demo:

(a) Set the input to $x[n] = \cos(0.3\pi n)$

(b) Set the filter to a five-point averager

(c) Estimate the amplitude of the output signal

(d) Use the GUI to display the exact formula for the output signal.

Instructor Verification (separate page)

2.2 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 2 shows the interface for the `convdemo` GUI.

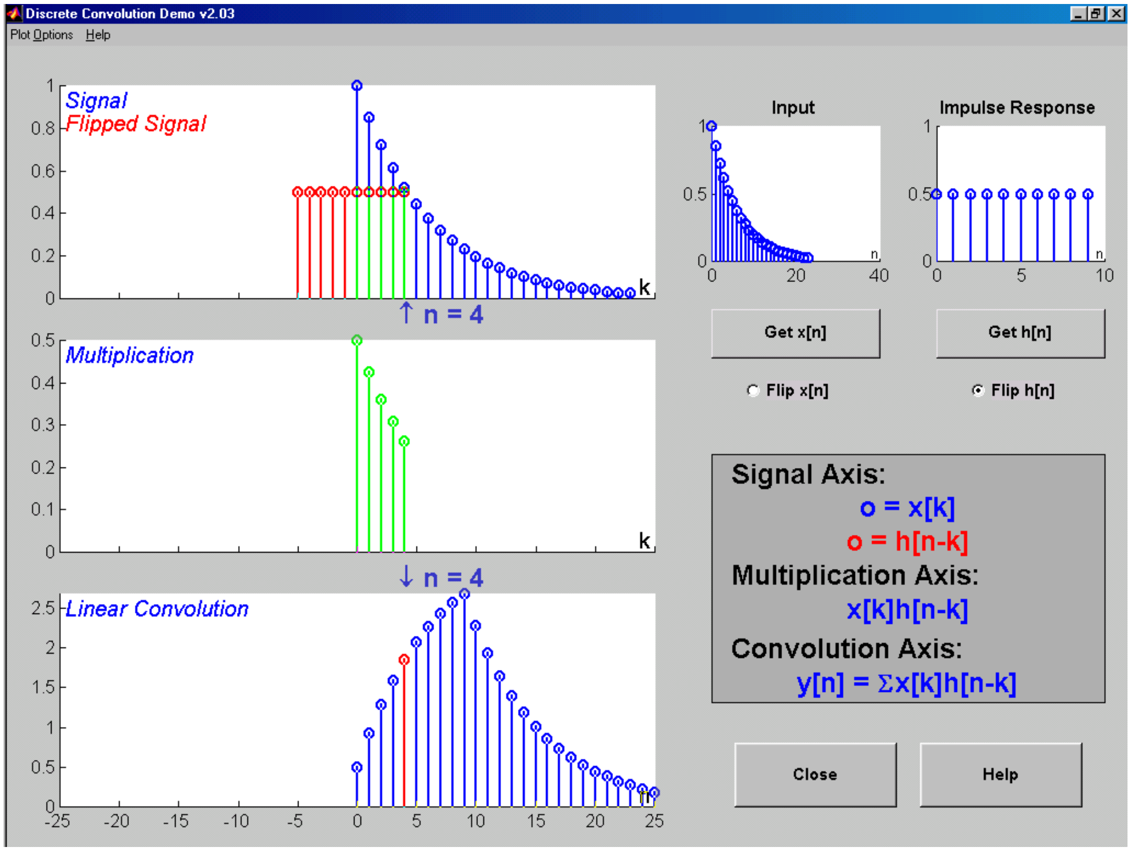


Figure 2: Interface for discrete-time convolution GUI.

In the warm-up, you should perform the following steps with the `convdemo` GUI.

(a) Set the input to a finite-length exponential: $x[n] = 0.8^n\,(u[n] - u[n-10])$.

(b) Set the filter to a five-point averager.

(c) Use the GUI to produce the output signal.

(d) 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."

Note: the discrete-time and continuous-time convolution demos are quite similar. In some versions, they share the same MATLAB code, so you might encounter weird behavior when using them in succession, such as a continuous-time signal popping up in the discrete-time demo. If this happens, issue a clear command to MATLAB. For example, `clear all` or `clear classes`. In addition, do not add both convolution demo directories to the `matlabpath`, because they use similar filenames and MATLAB will become confused when plotting the signals inside the GUIs.

2.3 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. Figure 3 shows the interface for the `cconvdemo` GUI.

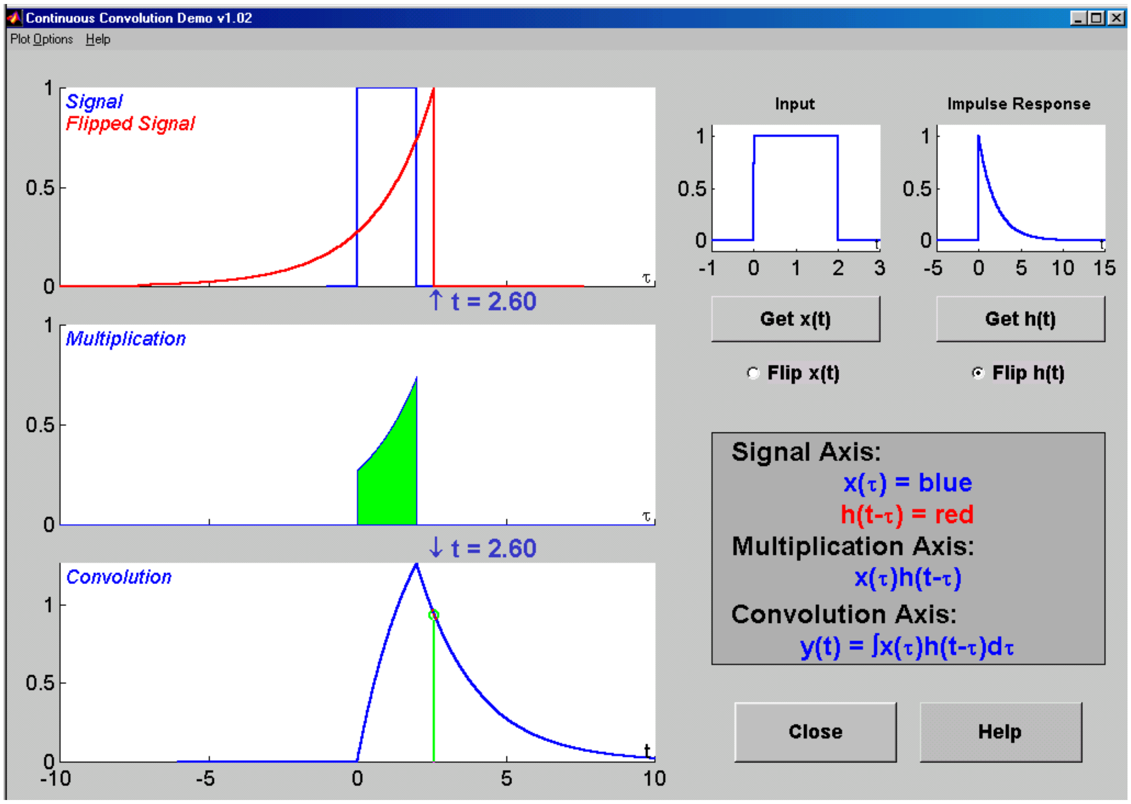


Figure 3: Interface for continuous-time convolution GUI.

In the warm-up, you should perform the following steps with the `cconvdemo` GUI.

(a) Set the input to a shifted pulse $x(t) = u(t-1) - u(t-4)$

(b) Set the filter to an exponential $h(t) = e^{-0.3t}[u(t) - u(t-10)]$.

(c) Use the GUI to produce the output signal. Use the *sliding hand tool* to grab the time marker and move it to produce the flip-and-slide effect of convolution.

Instructor Verification (separate page)

3 Lab Exercises

In each of the following exercises, you should make a screen shot of the final picture produced by the GUI to validate that you were able to do the implementation. In all cases, you will have to do some mathematical calculations to verify that the MATLAB GUI result is correct.

3.1 Sinusoidal Response

Use the LTIdemo to do the following problem:

(a) Find the output of a digital filter whose impulse response[1] is

$$h[n] = u[n-3] - u[n-10]$$

when the input is

$$x[n] = 1.5\cos(0.3\pi(n-2))$$

(b) Repeat for the same impulse response, but the input is

$$x[n] = 0.5\cos(2\pi n/7 - \pi/3) + 0.3$$

No screen shot of the GUI is needed for this case; just record your observation of the output signal.

(c) For both cases above, use the analytical form of the frequency response $H(e^{j\hat{\omega}})$ to derive the true output using mathematics.

3.2 Discrete-Time Convolution

In this section, you will be able to compare your results to those obtained in the previous section. Use the discrete-time convolution GUI, `convdemo`, to do the following:

(a) Find the output of a digital filter whose impulse response is

$$h[n] = u[n-3] - u[n-10]$$

when the input is

$$x[n] = 1.5\cos(0.3\pi(n-2))u[n-2]$$

Note: the GUI requires that $x[n]$ have finite length, but the definition of $x[n]$ above is infinite (one-sided). To fake out the GUI, make the length of $x[n]$ equal to 100. Then IGNORE the ending part of the signal, i.e., ignore all output points for $n \geq 50$. The plot of the GUI will only show a small region of about 30 points, so it will be easy to ignore the ending part of the signal.

(b) The output signal from part (a) can be divided into two regions: a *transient* region and a *steady-state* region. The steady-state region is where the output is an exact sinusoid. Determine the starting index for this region. Then use mathematics to find the exact formula for the output signal in the steady-state region.

(c) Relate the output signal in part (a) above to the corresponding output from the previous section, Section 3.1. Determine where the corresponding outputs are the same and where they are different. Identify the portion of the output signal that would be called the *transient*, or start-up region.

[1] Notice that LTIdemo allows user input of the filter coefficients.

(d) Repeat the convolution for the same impulse response, but the input is

$$x[n] = \cos(2\pi n/7 - \pi/3)u[n]$$

Once again, make the length of $x[n]$ equal to 100 points. No screen shot of the GUI is needed for this case; just record your observation.

(e) Make the same explanations and comparisons as called for in parts (b) and (c). In particular, identify the *transient* region in the output signal.

3.3 Continuous-Time Convolution

In this section, use the continuous-time convolution GUI, `cconvdemo`, to do the following:

(a) Find the output of an analog filter whose impulse response is

$$h(t) = u(t-3) - u(t-8)$$

when the input is

$$x(t) = e^{-2t}[u(t) - u(t-3)]$$

(b) Use the convolution integral to get the exact form of the output signal and verify that the GUI is correct. Also verify that the duration of the output signal is correct.

(c) Repeat for the same impulse response, but the input is

$$x(t) = \sin(4\pi t/5)u(t)$$

Note: the GUI requires that $x(t)$ have finite length, but the definition above is infinite (and one-sided). To fake out the GUI, make the length of $x(t)$ equal to 100. Then IGNORE the ending part of the signal, i.e., ignore all output points for $t \geq 50$. The plot of the GUI will only show a small region of about 20 seconds, so it will be easy to ignore the ending part of the signal.

Include a screen shot of this case.

(d) For the output from part (c), determine the analytical form of the true output using the convolution integral.

(e) Describe the output signal in part (c) as having two regions: steady-state and transient. Determine the duration of the transient region.

Lab #9
ECE-2025
Fall-2000
INSTRUCTOR VERIFICATION PAGE

Staple this page to the end of your Lab Report.

Name: ____ Date of Lab: ___________

Lab #8: Demonstrate the octave filtering system.

Verified:______________________ Date/Time:___________

Part 2.1: Demonstrate that you can run the LTI demo. Explain how the output signal is calculated by pointing to the relevant plots shown in the interface window.

Verified:______________________ Date/Time:___________

Part 2.3: Demonstrate that you can run the continuous-time convolution demo. Explain why the output signal is correct, and explain how the duration of the output signal is related to the durations of the impulse response and the input signal.

Verified:______________________ Date/Time:___________