

---

## DSP First

# Mini-Project 02: Discrete Convolution GUI

---

**Pre-Lab and Warm-Up:** You should read at least the Pre-Lab and Warm-up sections of this mini project and go over all exercises in the Pre-Lab section before coming to class.

**Verification:** The Warm-up section of the mini project must be completed **during class** and the steps marked *Instructor Verification* must also be signed off **during class**. The instructor must verify the appropriate steps by signing on the **Instructor Verification** line. When you have completed a step that requires verification, simply demonstrate the step to the instructor.

**Memo:** Turn in a one page (or so) memo on Section 4 with graphs and explanations. You are asked to **label** the axes of your plots and include a title for every plot. In order to keep track of plots, include your plot *inlined* within your memo. If you are unsure about what is expected, ask.

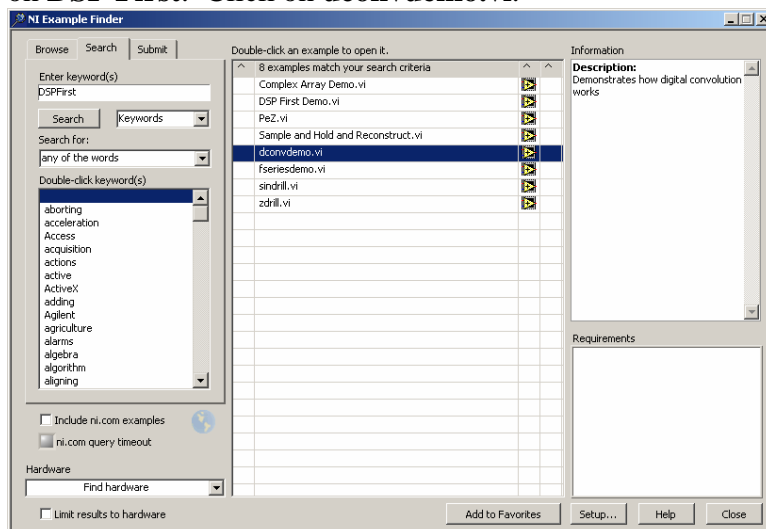
---

## 1 Introduction

This mini project concentrates on the use of **dconvdemo** a GUI for discrete-time convolution. This demo illustrates an important point about the behavior of a linear, time-invariant (LTI) system. It also provides a convenient way to visualize the output of a LTI system.

## 2 Pre-Lab: Run the GUIs

The first objective of this mini project is to demonstrate usage of the convolution GUI. First, load the demo by going to the CD-ROM and clicking on **Getting Started**. At the bottom of this page are instructions on how to load the **dspfirst** LABVIEW toolbox. Install the toolbox. In LABVIEW click **Help**»**Find Examples**. Click on the **Search** tab and enter **DSP**. Double-click on **DSP First**. Click on **dconvdemo.vi**.



## 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.

In the pre-Lab, you should perform the following steps with the `dconvdemo` GUI.

- Set the input to a finite-length pulse:  $x[n] = 2 \{ u[n] - u[n-10] \}$ .
- Set the filter's impulse response to obtain a five-point averager.
- Use the GUI to produce the output signal. Slide the control above the **Convolution** plot. By moving the pointer horizontally you can observe the sliding window action of convolution.
- 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.
- 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.
- 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.

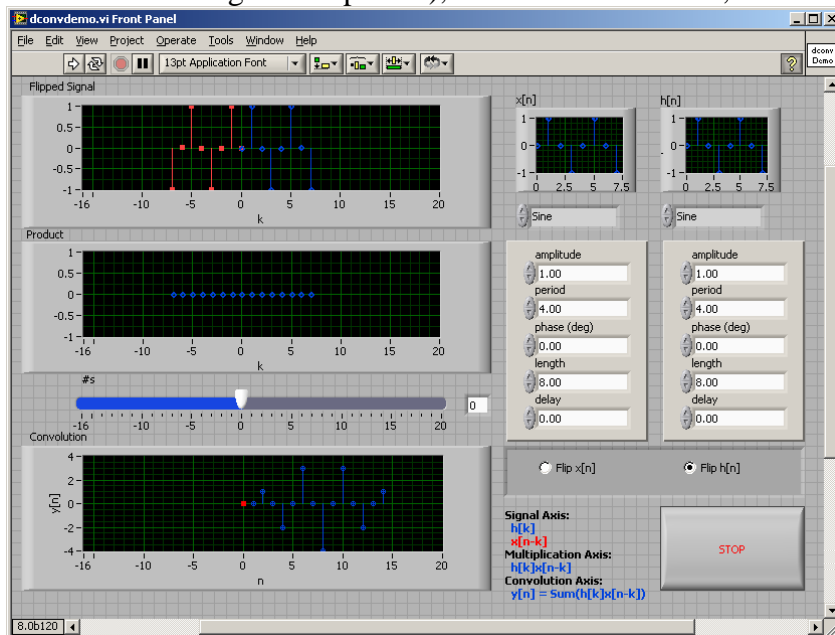


Figure 1 Interface for discrete-time convolution GUI.

## 3 Warm-up: Run the GUIs

The objective of the warm-up in this mini project is to use the convolution GUI to solve problems.

### 3.1 Discrete Convolution GUI

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

- Set the input to a finite-length sinusoid:  

$$x[n] = 2 \cos(2\pi(n-2)/5) \{ u[n-2] - u[n-12] \}$$

- b) Set the filter's impulse response to obtain a 5-point averager.
- c) Use the GUI to produce the output signal.
- d) Explain why the output has five different regions and why the output is zero in three of the five.

**Instructor Verification** (separate page)

## 4 Mini Project 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 LABVIEW GUI result is correct.

### 4.1 Discrete-Time Convolution

Use the discrete-time convolution GUI, `dconvdemo`, to do the following:

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

$$h[n] = u[n+3] - u[n-1]$$

when the input is

$$x[n] = 2 \cos(2\pi(n-3)/4) \{u[n-1] - u[n-11]\}$$

Note: You will have to adjust both the Length and Delay to get the  $u[n]$ 's right, then adjust the Phase to get the cos right.

- b) Use the GUI to determine the length of the output signal and notice that you can see five regions. Label the five regions. Note: the regions of partial overlap would be called *transient regions* while the region of complete overlap would be the *steady-state region*.
- c) Use **numerical convolution** to get the exact values of the output signal for each of the five regions. Thus, you will verify that the GUI is correct. Also verify that the duration of the output signal is correct.

**Mini Project 2**  
**INSTRUCTOR VERIFICATION PAGE**

*For each verification, be prepared to explain your answer and respond to other related questions that the professor asks. Turn this page in at the end of your memo.*

Name: \_\_\_\_\_ Date of Lab: \_\_\_\_\_

Part 4.1: Demonstrate that you can run the discrete-time convolution demo. Explain why the output is zero in three of the five regions identified for the output signal.

Verified: \_\_\_\_\_ Date/Time: \_\_\_\_\_