Discrete-Time Signals and Systems
Introduction to LabVIEW

Background and Warm-Up: This miniproject consists of a Background, a Warm-Up, and an Assignment. The Background contains instructions for installing LabVIEW. The Warm-Up contains exercises to teach you the rudiments of using LabVIEW to create "virtual instruments" (LabVIEW programs). There is an instructor verification at the end of the Warm-Up to allow the instructor to verify that your installation of LabVIEW is working correctly. The Assignment asks you to create a memo to be handed in. You should read this miniproject assignment, Install LabVIEW, and complete the Warm-Up and the Assignment before the due date as shown in the course calendar.

Verification: When you have completed a step that requires verification, simply demonstrate that step to your instructor. Your instructor must verify the step and sign the *Instructor Verification* form included as the last page of these instructions. A small amount of class time will be set aside for checking off the instructor verification of the Warm-Up (Section 2) exercise. You can also get the verification signed by bringing your laptop to your instructor's office. The signed *Instructor Verification* page must be included as the last page of the memo to be handed in.

Miniproject Memo: It is only necessary to turn in a memo on the Assignment (Section 3) part of the project. The memo must include the required graphs and explanations, the theoretical calculations and results, and your LabVIEW code (print the block diagram and the front panel). You are asked to label the axes of your plots and include a title for every plot. In order help the reader keep track of the plots, include your plots *in-line* within your report. Use figure numbers and captions, and follow the ECE Writing Guidelines. If you are unsure about what is expected, ask.

> The most common reason that students lose points on the initial memo is for formatting errors. You can save yourself some points, and your instructor some time, if you simply check to make sure that you put your initials on the memo by your name, label your figures "Figure 1" etc., and use captions on the figures to explain what they are supposed to show. Also, put your figures within the body of the memo, not all at the end, and put each figure immediately after the first place that it is referenced (you did mention the figures in your write up didn't you?).
>
> After formatting errors, arithmetic errors are common, like switching units (radians, degrees, seconds), having the wrong sign for the phase angle (watch out for this one!), and leaving huge numbers of radians unsimplified, like 2000.3 π instead of just 0.3 π.

1 Background

In this first week, the miniproject handout will give step by step instructions. Later, you will work more independently. Make sure that you read through the information in Sections 1-4 below and do everything in Section 1 prior to beginning the Warm-Up. You may want some headphones so you can listen to some signals.

1.1 Overview

LabVIEW will be used extensively in all the miniprojects. The primary goal of this miniproject is to familiarize you with using LabVIEW. Here are three specific goals for this miniproject:

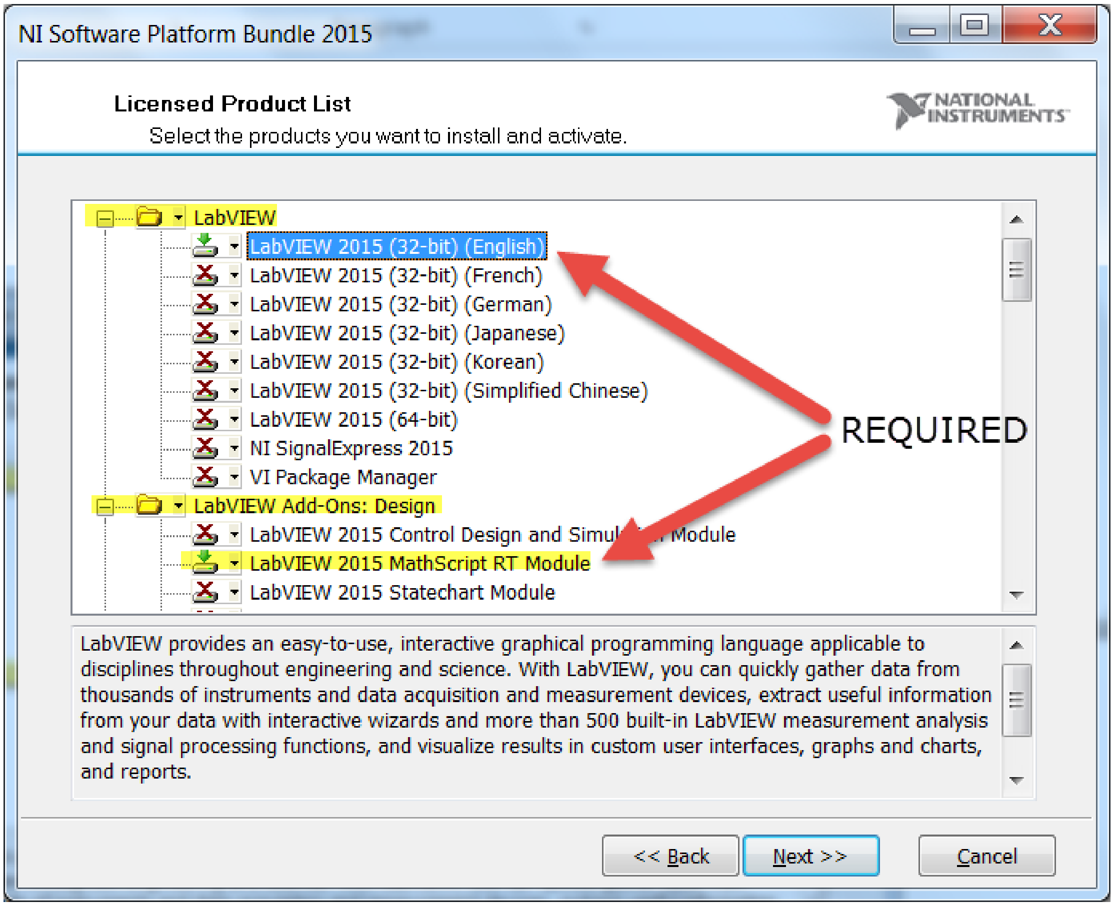
- Learn basic LabVIEW commands and syntax, including the *help* system.
- Learn to write and edit your own VIs (virtual instruments) in LabVIEW, and run them.
- Learn a little about advanced programming techniques for LabVIEW[1].

1.2 Install LabVIEW

Here's how to get LabVIEW 2015 installed. *Warning*: allow at least 45 minutes to install LabVIEW.

- Note that all the packages described below are needed to complete the miniprojects this quarter; be sure to carefully follow all of these instructions.
- Log on to your laptop as `localmgr`; connect to the network via a cable, if possible.
- In the Windows Start menu, in the "Search programs and files" box, type \\software. Work your way down to "**Course Software » National Instruments Spring 2014**". Open the file "Readme.txt" and write down the LabVIEW serial number. Then back up one level and go to "**NI LabView Fall 2015**".
- Run **`setup.exe`**. You may be prompted to first install the Microsoft .NET Framework and reboot. Do so, and then repeat the previous steps to run **`setup.exe`** again.
- When the installer asks for a serial number, use the one you copied from the Readme file. Type the serial number in the box, and click to add it to the list. Note that you are given the option to install LabVIEW without a serial number. Doing so will install a 30-day trial version that will evaporate midway through the quarter.
- Select the default answers until you get to the **Licensed Product List** shown below. Initially nothing is selected for installation, not even LabVIEW! Open the LabVIEW folder and select "LabVIEW 2015 (English)". You must also select the "LabVIEW 2015 MathScript RT Module". You may select additional components if you wish.

[1] Read Section I (pages 1-19) of the *Introduction to LabVIEW – Six Hour Course* (in the class folder) for more details.



- Continue selecting the defaults until the installation is complete.
- The licensed version of LabVIEW has to be "activated". This should take place automatically toward the end of the installation process, and you should get a message telling you that the activation has happened.
- Now, with LabVIEW not running, point your web browser to **www.rose-hulman.edu/dspfirst**.
- Click on **Getting Started**. (Near the top of the left column.)
- Scroll down to "LabView" and follow the instructions for installing the dspfirst package.
- Note that actual dragging (see instructions) may work better than "copy and paste" in this instance.
- You may need to choose "Move and Replace" a few times to replace files that are being overwritten. This is expected.
- Finally, start LabVIEW and follow along with the two-minute video http://youtu.be/M6gDpso6ysI to configure the "look and feel" of LabVIEW to match that of the instructor's setup and other tutorial videos.

1.3 Getting Started - Open and Run a Virtual Instrument

In this part of the miniproject you will examine the **DSP First Demo** VI and run it. Change the frequencies and types of the input signals and notice how the display on the graph changes. Here is how to do it:

a) Select **Start»All Programs » National Instruments » LabVIEW 2015** to launch LabVIEW. The **LabVIEW** dialog box appears.

b) Select **Help » Find Examples.** The dialog box that appears lists and links to all available LabVIEW example VIs.

c) On the Browse tab, click the Directory Structure radio button.

d) Double-click the **dspfirst** folder to open it. Then double-click **DSP First Demo.vi.** This will open the DSP First Demo VI Front Panel. A panel may pop up behind the Example Finder that you need to click. If this takes a while, look for that panel.

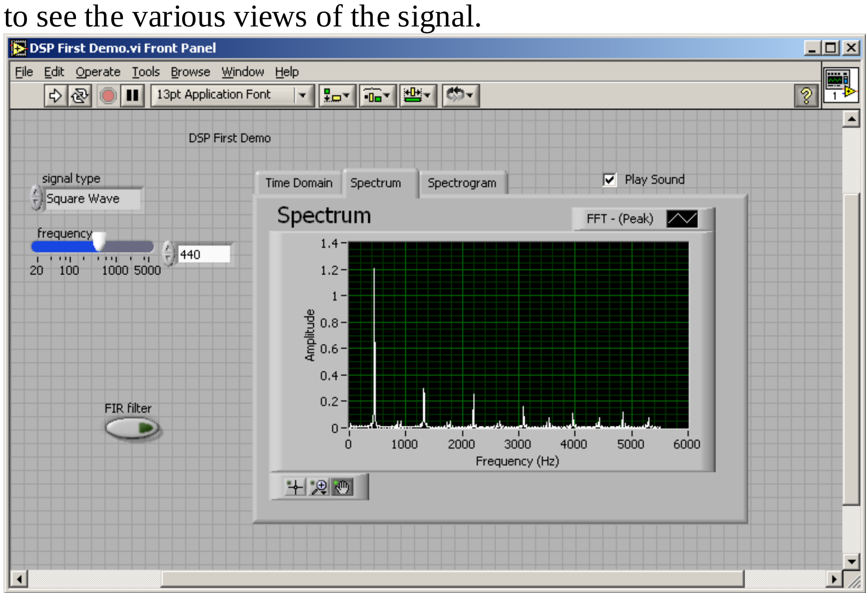
Note: This is how you find all the DSP First demos.

e) Note: if you can't see the words "play sound" on the Triple Display dialog (see figure below) on your screen, you might need to reset the Application Font size. To do so, go to **Tools»Options,** Select Environment, then scroll down to Fonts. Uncheck "Use default font" and then use the "Font Style" button to change the size to 12. Close LabVIEW completely (even the start screen), it is ok to hit "Don't Save," and then re-open the example. The fonts in the tabs should be much smaller now.

Front Panel

a) Check the box next to **Play Sound**.

b) Click the **Run** button () on the toolbar, shown at left, to run this VI once. You should hear a short tone. This VI determines the result of filtering a generated signal. This example also displays the magnitude spectrum for the generated signal. The resulting signals are displayed in the graphs on the front panel, as shown in the following figure. Try clicking the **Time Domain**, **Spectrum**, and **Spectrogram** tabs to see the various views of the signal.



c) Try changing the **signal type** and **frequency**. You will have to click the **Run** button after each change to see/hear the result.

Block Diagram

a) Select **Window»Show Block Diagram** or press the <Ctrl-E> keys to display the block diagram for the Signal Generation and Processing VI.

This block diagram contains several of the basic block diagram elements, including subVIs, functions, and structures, that you will learn about later in this course.

b) Select **Window»Show Front Panel** or press the <Ctrl-E> keys to return to the Front Panel.

c) Close the VI and do not save changes.

2 Warmup: Building VIs – Display Complex Numbers in Polar and Rectangular Forms

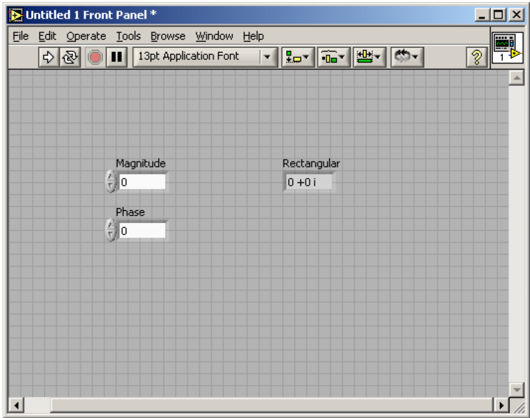
The previous exercise gave you experience with running a VI. The real fun starts when you build your own VIs. The next exercise will walk you through building a VI and modifying it to do interesting things. Complete the following steps to create a VI that takes a complex number in polar form and displays it in both polar and rectangular form.

In wiring illustrations, the arrow at the end of this mouse icon shows where to click and the number on the arrow indicates how many times to click.

Front Panel

1. Select **File»New VI** to open a new front panel.
2. Create a numeric digital control. You will use this control to enter the value for the sinusoid magnitude.
    a. Right-click an open area on the front panel to display the **Controls** palette. Select the `Numeric Control` on the **Controls»Modern»Numeric** palette.
    b. Move the control to the front panel and click to place the control.
    c. Type `Magnitude` inside the label and click outside the label. If you do not type the name immediately, LabVIEW uses a default label. You can edit a label at any time by double clicking on it.
3. Create a second number control.
    a. Hold the Ctrl key down and click and drag the first control. You now have a second control.
    b. Label the second control `Phase`.
4. Create a numeric digital indicator. You will use this indicator to display the complex value.
    a. Select the `Numeric Indicator` on the **Controls» Modern»Numeric** palette.
    b. Move the indicator to the front panel and click to place the indicator.
    c. Type `Rectangular` inside the label and click outside the label.

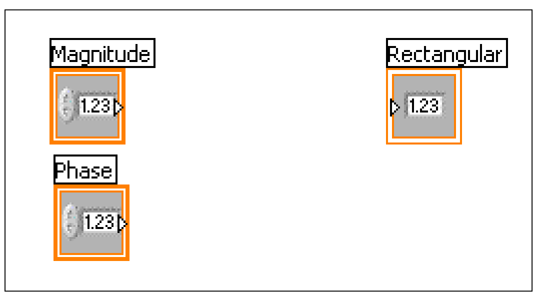
d. Make this indicator complex by right-clicking on the control and selecting **Representation**. Click on the **CDB** block which is in the middle of the bottom row.



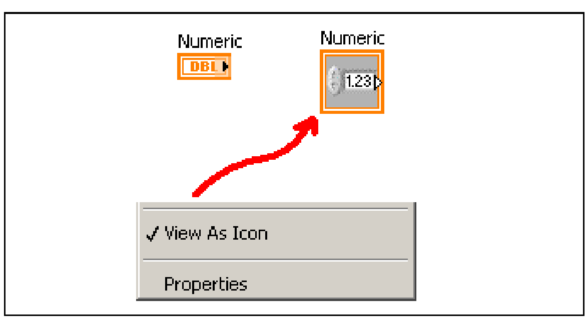
LabVIEW creates corresponding control and indicator terminals on the block diagram. The terminals represent the data type of the control or indicator. For example, a DBL terminal represents a double-precision floating-point numeric control or indicator. A CDB terminal represents a complex double-precision floating-point numeric control or indicator.

Block Diagram

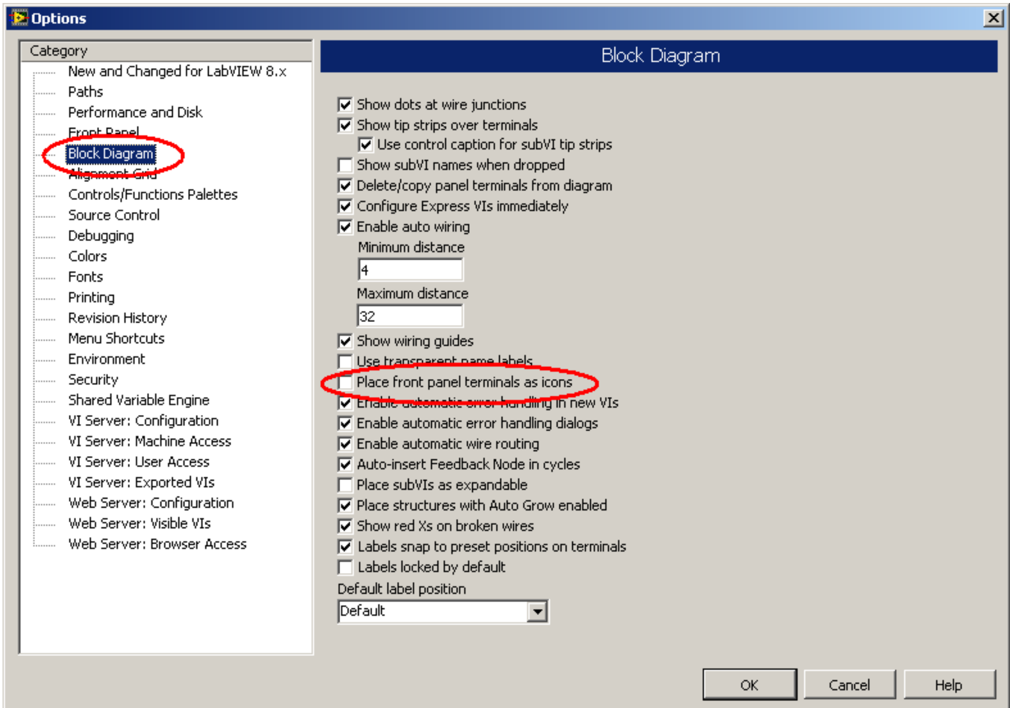
5. Display the block diagram by clicking it or by selecting **Window»Show Block Diagram**.



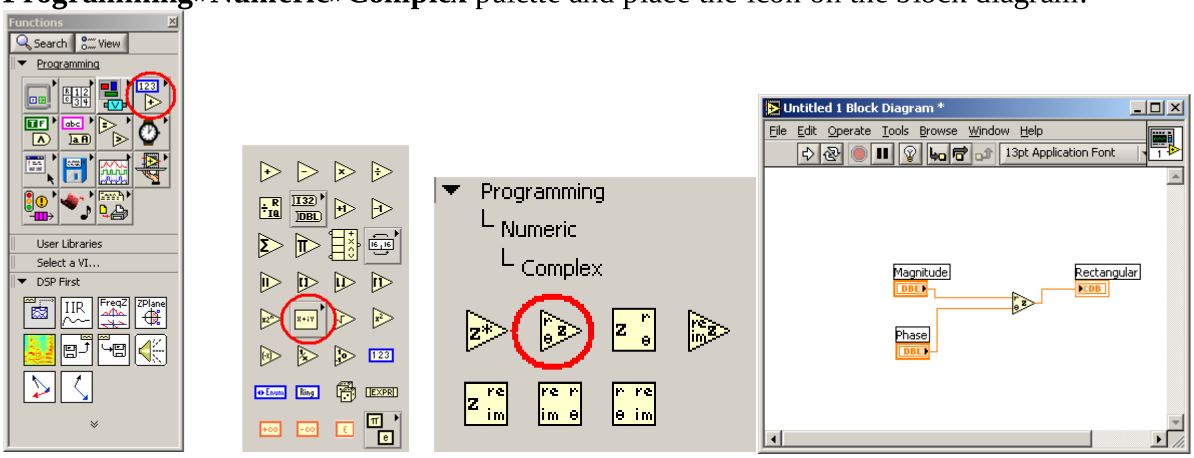
Note: Block Diagram terminals can be viewed as icons or as terminals. To change the way LabVIEW displays these objects right click on a terminal and select **View As Icon**.



You can make the terminal view the default by selecting **Tools»Options…** then click on **Block Diagram** on the left and *un*select “Place front panel terminals as icons”.

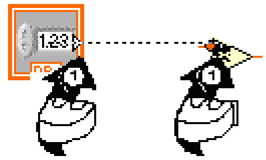


6. You can display the available complex blocks by right-clicking an open area on the block diagram to display the **Functions** palette. Select the `Polar to Complex` function on the **Programming»Numeric»Complex** palette and place the icon on the block diagram.



These are the blocks for manipulating complex numbers. You should be able to guess what they do. If not, display the **Context Help** window (**Help»Show Context Help** or <Ctrl-H>) and point to each block.

7. Use the Wiring tool to wire the icons on the block diagram as shown above.
   - To wire from one terminal to another, hover your cursor near the first terminal. The cursor should turn to a spool of wire. When it does, click the first terminal, move the tool to the second terminal, and click the second terminal, as shown in the following illustration. You can start wiring at either terminal.



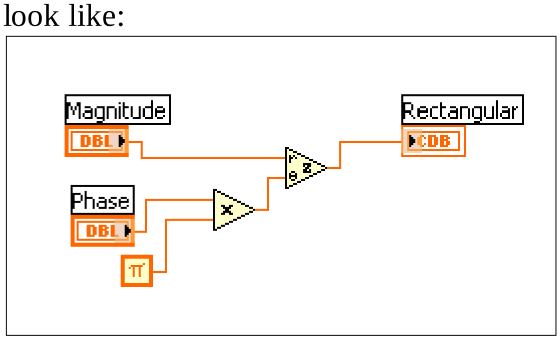
- You can bend a wire by clicking to tack the wire down and moving the cursor in a perpendicular direction. Press the spacebar to toggle the wire direction.
- When you move the Wiring tool over a terminal, the terminal area blinks, indicating that clicking will connect the wire to that terminal and a tip strip appears, listing the name of the terminal.
- To cancel a wire you started, press the <Esc> key, right-click, or click the source terminal.

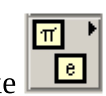
8. Display the front panel by clicking it, by selecting **Window»Show Panel**, or by typing <Ctrl-E>.
9. Save the VI because you will use this VI later in the course.
   a. Select **File»Save**.
   b. Navigate to your class directory.

Note Save all the VIs you edit in this course in your class directory.

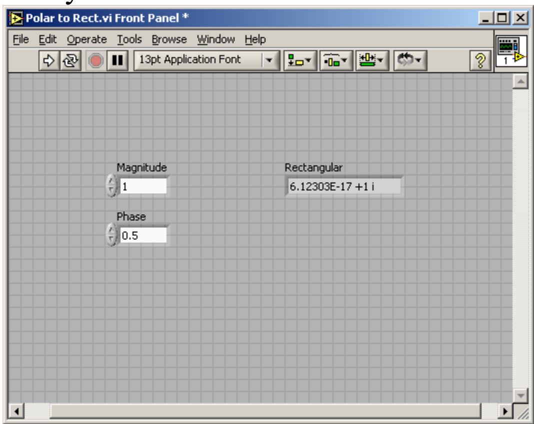
   c. Type `Polar to Rect` in the dialog box.
   d. Click the **Save** button.

10. Enter numbers in the digital controls and run the VI. The result is too big for the box, so stretch the Rectangular box so you can see both the real and imaginary parts.
    a. Double-click each digital control and type in a new number.
    b. Click the **Run** button to run the VI.
    c. Try several different numbers and run the VI again.
11. It would be nice to enter the phase in fractions of π. Modify your block diagram to look like:



Hint: You will find the π block in the palette that looks like .

12. Test your VI. Here's what mine looks like:

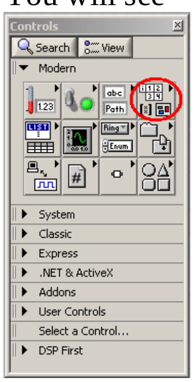


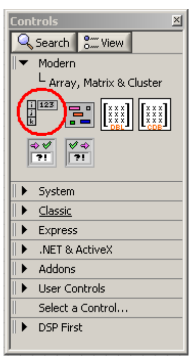
13. Select **File»Close** to close the VI.

2.1 Arrays of Complex Numbers

Throughout this class we will be doing a lot of work with arrays, so let's see how LabVIEW handles arrays of complex numbers.

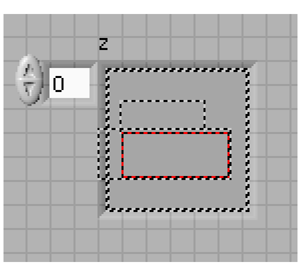
1. Select **File»New VI** to open a new front panel.
2. Go to the front panel and create an array control by right-clicking on the background. You will see

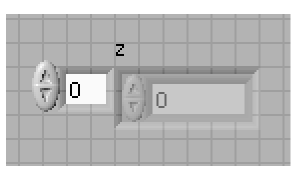


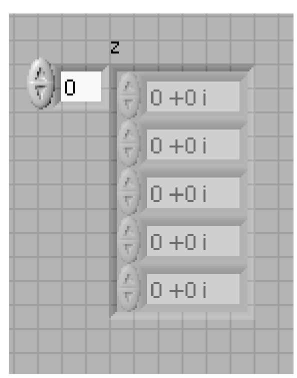


Select **Array, Matrix & Cluster**. You will see the second palette above, select **Array** and place it on the front panel. Change the name to `z`. Note: In LabVIEW an array is a one dimensional list of elements, while a matrix is a two dimensional list.

3. Now you need to specify what the array will hold. Do this by selecting a Numeric Control (like you did before) and dragging it *into* the array.

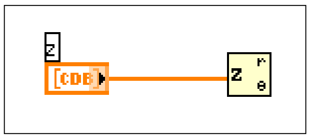




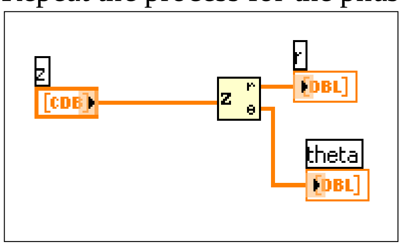


Release the control and it will look like the middle picture above.

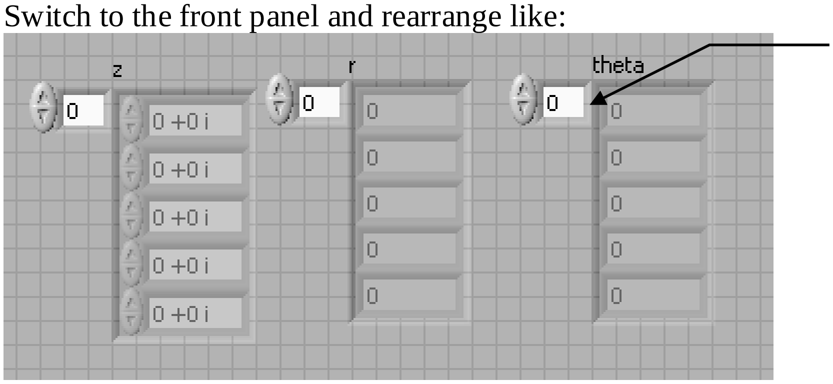
4. Change the data type to Complex Double (CDB) by right-clicking on the control and selecting **Representation**. Select **CDB** from the bottom middle. You just made this control handle complex double values.
5. Click on the bottom of the control and drag it down a bit. You will now see something like the right-most picture above. This lets you see and input several values at once.
6. Switch to the **block diagram**. <Ctrl-E> is a fast way to do this.
7. Your goal is to display the real, imaginary, absolute value, and phase of `z` using these Complex blocks. I'll start you with the first one and you should be able to figure out the rest. Place the **Complex to Polar** block (middle one) on your block diagram and wire as shown.



8. Right-click on the **r** terminal of the **Complex to Polar** block and select **Create»Indicator**. This is a quick way to making a new indicator.
9. The indicator is labeled **r** by default. You may wish to change it to **magnitude**.
10. Repeat the process for the phase output as shown below.



11. Switch to the front panel and rearrange like:



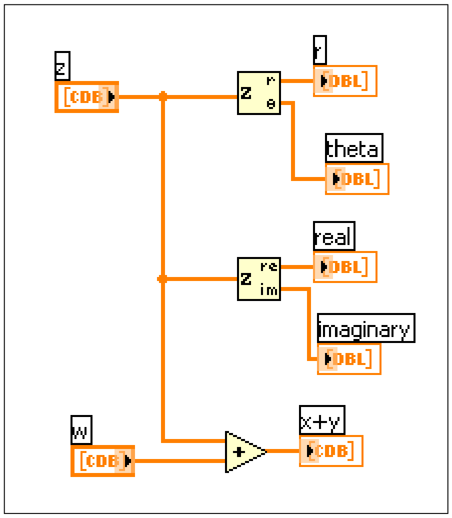
I find that clicking in between the textboxes is a good place to "grab" the entire control for repositioning. However, I've been told that clicking near the top is supposed to be the "standard" place to click for all objects.

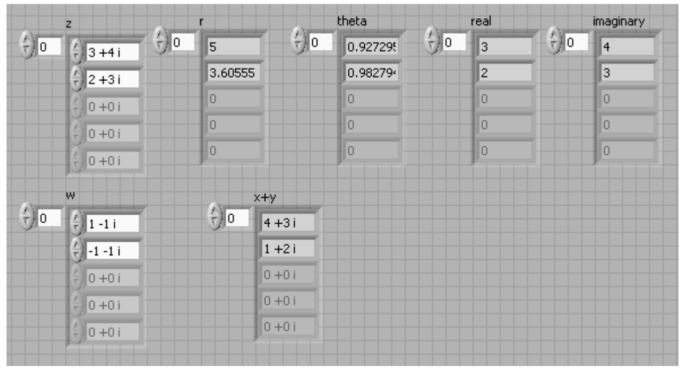
12. Enter a complex number in `z` (try 3+j4, which has an easy magnitude)[2] and click **run**. Does the correct answer appear? Add other values to `z` and try again.
13. Place the **Complex to Re/Im** block on the block diagram and attach indicators. Click **run** and verify that they are working.
14. Go to the block diagram view and add another complex variable input by holding the Ctrl key down and clicking and dragging the `z` block. Label this new block `w`.

---

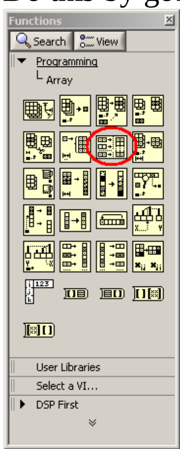
[2] You'll have to type `3+4i`.

15. Get an **add** block[3] and add z to w. Attach an indicator. Verify that it works. Note the default label for the **add** result is ‘x+y’. You may wish to change it to ‘z+w’.





16. You can combine the arrays z and w into one array.
    a. Do this by going to **Programming»Array** and selecting **Build Array**.



    b. Click on the bottom of the **Build Array** icon and stretch it until it has two inputs.
    c. Right-click on **Build Array** and select **Concatenate Inputs.**
    d. Attach z and w to the inputs.
    e. Create an indicator for the output. Hint: Right-click, create indicator.
    f. Run the VI. Does it work as expected?
    g. This also works for combining scalars into an array.
17. Save this as `Complex Arrays`.

You have seen that it is rather easy to build up entire arrays of complex numbers and manipulate them. Next you’ll see how to build some signals.

[3] Reminder: right-click on the block diagram to get a palette of blocks. Click on **Numeric** and select the **Add** block. If you aren’t sure where a block is, right-click on the background and then click on **Search**. Type the name of the block for which you are looking.

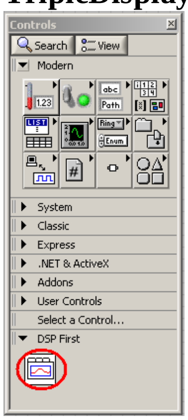
2.2 Generating Signals

In this section we'll see that it's easy to generate waveforms in LabVIEW. A waveform is a datatype that contains an array of sample values and information about the sampling rate.

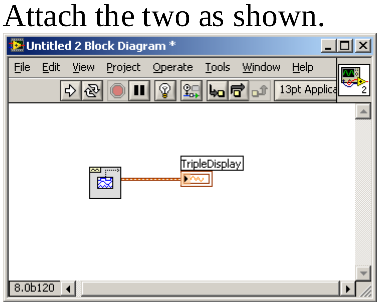
1. Start with a new VI by clicking **File:New VI**.
2. Select the **block diagram** <Ctrl-E>.
3. Right-click to open the Functions palette and then click the pushpin in the upper left to keep the palette open. Click on **DSP First**. You will see:

   These are the VIs for this class.
4. Select the **DSP First Function Generator** (upper left) and place it on the block diagram.
5. Go to the Front Panel <Ctrl-E>, right-click, choose DSPFirst, then select and place the **TripleDisplay**.

   This will place a VI on the block diagram. The **TripleDisplay** block has the connector on the right side rather than the left side, so right-click on it and select **Change to Indicator**.
6. Attach the two as shown.

7. Click the run button.

8. Use the context help[4] to learn what the Amplitude, Frequency, Phase, and Sampling Info inputs on the **DSP First Function Generator** default to. What are the units on the Phase?
9. Attach controls to Amplitude, Frequency, and Phase inputs. Hint: right-click on the input and select **Create Control**.
10. Try different values.

The **DSP First Function Generator** generates cosines by default. You can generate other waveforms with it by attaching a control to the **signal type** connector. There are many other sinusoidal generators in LabVIEW, however they generate sines. Since electrical engineers use cosines as our "zero-phase" sinusoids, stick with our generator.

2.3 LabVIEW Sound

The exercises in the section involve sound signals, so you should consider using headphones for listening.

1. Generate a tone (i.e. a sinusoid) in LabVIEW and listen to it. The previous section showed how to generate a sinusoid. For this part, the frequency of your sinusoid tone should be 2000 Hz and its duration should be 1 second. Use a sampling rate equal to 11025 samples/sec. To set the sampling rate and duration, right-click on the **sampling info** terminal on the **DSP First Function Generator** (bottom left) and select **Create»Constant**. The top number in the box is the sampling rate and the bottom is the number of samples. Change these to match above.
2. Listen to the tone by clicking the **Play Sound** check box on the **TripleDisplay**. Click run. Does the tone sound correct?
3. The Time Domain display is not very informative at this scale. See if you can zoom in to see something more obviously sinusoidal. Click the magnifying glass at the bottom of the graph for zoom options.
4. Try different amplitudes and frequencies. To make this easier, change the input constants to controls. Hint: right-click the constant and select **Create»Control**. Delete the constant and attach the control. Can the frequency be too high or too low?

Instructor Verification (separate page)

3 Exercise: Manipulating Sinusoids with LabVIEW

Now you're on your own. **Include a short summary of this Section in your memo, including plots and theoretical calculations.** Build a LabVIEW VI to do steps (a) through (d) below. Include a printout of both the front panel and the block diagram with your report.

(a) Set the **sampling info** so that the **Function Generator** will produce approximately two cycles of the 4000 Hz sinusoids defined in the next part, part (b). Set the sampling rate so that you have at least 25 samples per period of the sinusoidal wave.

(b) Generate two 4000 Hz sinusoids with arbitrary amplitude and time-shift,

$$x_1(t) = A_1 \cos\left[2\pi 4000\left(t - t_{m_1}\right)\right] \quad x_2(t) = A_2 \cos\left[2\pi 4000\left(t - t_{m_2}\right)\right].$$

Select the values of the amplitudes and time-shifts as follows: Set A_1 equal to your age and set $A_2 = 0.8A_1$. For the time-shifts, set $t_{m_1} = -T/M$ and $t_{m_2} = T/D$ where D

[4] If the small context help window isn't appearing, type <Ctrl-H> to make it appear.

and M are the day and month of your birthday and T is the period. Make a plot of each signal over the range of $0 \le t \le 2T$. For your final printed output in part (d) below, place three **TripleDisplay**s one above the other on the front panel. Be sure you check to see that you calculate your phase values in the correct units for your LabVIEW code (look at the inputs to the function generator block).

(c) Create a third sinusoid as the sum: $x_3(t) = x_1(t) + x_2(t)$. In LabVIEW this amounts to summing the vectors that hold the values of each sinusoid. Make a plot of x_3 over the same range of time as used in the plots of part (b).

(d) For each the three plots, put a title on the plot and include your name in one of the titles. Then for each plot click the **Export Simplified Image** tab. This will put the plot on your clipboard and then you can paste the plot into your word processor.

3.1 Theoretical Calculations

Remember that the phase of a sinusoid can be calculated after measuring the time location of a positive peak,[5] if we know the frequency. Specifically, for a sinusoid $x(t)$ of frequency f_0 with amplitude A and time delay t_m, we have

$$\begin{aligned} x(t) &= A\cos\left[2\pi f_0 (t - t_m)\right] \\ &= A\cos(2\pi f_0 t - 2\pi f_0 t_m) \\ &= A\cos(2\pi f_0 t + \phi). \end{aligned}$$

We see that the phase angle ϕ is given by $\phi = -2\pi f_0 t_m$.

(a) Make measurements of the "time-location of a positive peak" and the amplitude of each of your sinusoids by reading the plots of $x_1(t)$ and $x_2(t)$, and write the values you obtain for A_1, t_{m_1}, A_2, and t_{m_2} directly on the respective plots. Then calculate (by hand) the phases of the two signals $x_1(t)$ and $x_2(t)$. Write the calculated phases ϕ_1 and ϕ_2 directly on the plots.

Note: When doing your computations, express phase angles in degrees, not radians.

(b) Measure the amplitude A_3 and time-shift t_{m_3} directly from the plot of $x_3(t)$ and then calculate the phase ϕ_3 by hand. Write these values directly on the plot. Show by annotating the plot how the amplitude and time-shift were measured, and how the phase was calculated.

(c) We can add two sinusoids using phasors by a little complex-number arithmetic:

$$A_3 e^{j\phi_3} = A_1 e^{j\phi_1} + A_2 e^{j\phi_2}.$$

Carry out an addition of the phasors for $x_1(t)$ and $x_2(t)$ to determine the phasor for $x_3(t)$. Use the values of A_1, ϕ_1, A_2, and ϕ_2 that you obtained in step (a) in your phasor calculations.

[5] Usually we say "time-delay" or "time-shift" instead of the "time location of a positive peak."

(d) Prepare a table comparing the values of A_3 and ϕ_3 you obtained in step (b) with the values you obtained in step (c). Show the amplitude error as a percentage and the phase angle error as a difference in degrees.

3.2 Memo

Please refer back to Miniproject Memo on page 1 of these instructions for information on the format of your memo. When you write the memo, be sure to make it painfully clear how you found the time delay values you used to determine the phases of $x_1(t)$ and $x_2(t)$. Also make it clear how the amplitude and phase measurements you made from the graphs of $x_1(t)$ and $x_2(t)$ were used to perform a phasor (complex addition) calculation to correctly predict the measurements you made of $x_3(t)$. Note that you can check the sign of each phase angle you report by plotting the equation $A\cos(2\pi f_0 t+\phi)$ and verifying that the result matches the original plot.

4 LabVIEW things to remember for future miniprojects:

- Where to find DSP First demos
- How to turn on and use Context Help
- Using right-click to find palettes
- How to search for VIs (blocks)
- Using right-click to attach constants, controls, and indicators
- Where to find the complex operators
- How to create array controls
- How to declare the ‘type’ of an array
- How to change the data representation of a numeric input
- How to generate a cosine waveform
- How to change the sampling rate and number of samples generated
- How to use a **TripleDisplay**.
- How to copy a plot to be pasted into a word processor.
- … What else?

Introduction to Lab VIEW
INSTRUCTOR VERIFICATION SHEET
Attach this as the last page of your memo.

Name: ______________________________ Date of Miniproject: __________________

Part 2.2 Generating Signals
Part 2.3 LabVIEW Sound

Verified:__________________________ Date/Time:_____________________