Labview VI Example  
Virtual Filters  
Written by: Dan Lankow 2014

1. ABSTRACT
For this lab, you will be introduced to Labview. You will be implementing a Low Pass, High Pass, and Band Pass filter in Labview in order to gain an understanding of Labview’s operation and functionality. No previous experience with Labview will be necessary, and the analytical treatment of these filters is provided. You will also be asked to implement indicators which display the 3 dB breakpoint.

2. INTRODUCTION
Labview is a software programming tool which allows for the design and implementation of virtual instrumentation. The program is categorized as a Graphical User Interface (GUI). The presentation of the code in Labview is a visual block diagram which describes the data flow within the VI. Labview can take an input, in this case the input will be virtual, however with proper interfaces you could sample a real signal, and manipulate in accordance with the design of a Virtual Instrument, VI, and output the manipulated data. For example, it can take data from a sensor, transducer, and plot the voltages it samples, (i.e. it can function as an analog to digital converter as well). It can manipulate that data in a great many ways due to the sheer quantity, and versatility of the various built in functions.

In this application, we will use Labview to simulate an input signal, and simultaneously monitor the action of low pass, high pass, and band pass filters.

A note on Labview layout. Labview consists of two complementary components:
i) The Front Panel, which displays your controls, and indicators.  
   It functions as the user interface for the VI.
ii) the Block Diagram which displays all of your functions and express VIs.
   This is where all of the actual work takes place.
The functions, such as mathematical operations, and express VIs, like built analysis tools and signal simulation, are located on menus, called palettes. These are generally accessible by right clicking in the Block Diagram or Front Panel, which brings up the palette menu. In addition, placing functions and express VIs in the Block Diagram will automatically place corresponding elements into the Front Panel. Among the many useful features in Labview, there are three of particular interest at this time.

* The first is the Context Help option, Ctrl+h, which brings up a small help window which will give you information about any element you mouse over.
* The second is Ctrl+click and drag, which copies whatever element you have executed this command on. This makes large VIs much easier to create larger, more complicated VIs because any repeated elements can simply be copied directly, instead of having to open the appropriate palette and selecting it over and over again.
* Finally, Ctrl+e will swap back and forth between the Front Panel to the Block Diagram.
For this Labview example, we will need to have some familiarity with the Simulate Signal express VI, math operations, and Waveform Graphs.

1. To start, we must first simulate an input signal. This will be done by placing a Simulate Signal express VI into your Block Diagram. This is found under the Signal Analysis palette.

![Image of Signal Analysis Palette](image)

Signal Analysis Palette, the bottom left icon is the Simulate Signal Express VI.

![Image of Simulate Signal Express VI and Waveform Graph](image)

Simulate Signal Express VI will appear as the left part of this image in your Block Diagram, and wired to a Waveform Graph, the output may look something like the right image.

Notice, this express VI did not place any associated elements into the Front Panel. This is because this Express VI does not have any front panel controls directly associated with it. It does, however, require some inputs in order to run.

2. To add these we will do the following:

* Simply right click on the appropriate portion of the express VI, Frequency, and Amplitude (Press Ctrl+h, and mouse over the arrows on the left side of the Express VI to identify these inputs).

* Open the Create menu, then select Control. Doing this will automatically “wire” the elements together.

* Additionally, you can select the control you want from the associated palette. In this case we must connect these elements to each other.

* To do so, we wire them together. Place the mouse cursor over the input or output you are looking to wire to or from. This will cause the cursor to change to a wire spool.

* Click and drag to the input, or output you are trying to connect. Also, notice that controls were
placed on your Front Panel.

3. Now that we are able to drive a simulated signal, we want to see it.
   * Right click on the the appropriate output, again CTRL+h is your best friend, it will be the lower right arrow.
   * Go to the Create menu, and this time choose Graph Indicator. This will insert a Waveform Graph. Notice, this placed an object on your Front Panel. This is where Labview will display the simulated signal.
   * Name this appropriately, such as Input. You can alter the properties, such as auto-scaling, plot legends, and so on by either right clicking the Waveform Graph in the Block Diagram, or in the Front Panel and selecting Properties. You will need to do this to allow for a large frequency range, and to increase the sample rate. A sample rate of 1000000 allows for very high frequencies to be used.

4. In order to simulate a filter, we will need to simulate a resistor, and capacitor. Labview is not a circuit simulator; we will instead place a Control.
   * Navigate to the Front Panel, right clicking, Selecting Num Ctrl, and selecting the type of control you want. Notice this time, a control has been placed on the Block Diagram.
   * Rename these appropriately, something like Resistance, and Capacitance for instance, so you can easily keep track of them.
   * You will need to change the properties of these controls so their ranges are appropriate, such as $1000-1000000$ for resistance, and $10^{-12}$ to $10^{-6}$ for capacitance.
   * It would also be nice to see what value is currently stored. By right licking in the Front Panel, and creating an indicator, we add an indicator to our Block Diagram.
   * We will then wire this up, as described previously, to the appropriate control.

5. It is time to build your low pass, high pass, and band pass filters. By using the equations 1, and 2, found in your previous lab. We will implement a data flow which will execute the mathematical operations to act on the signal as a filter would.
* We will need to open our arithmetic palette. Here you will find the components you need, Multiply, Square, Add, Divide, and Square Root.

* Since $2\pi$ is a constant, we will add this by simply right clicking the input we want it to be associated with, in this case, it will be the input to a Multiply operation, selecting Create, then Constant. We can then enter the constant, typing in the numerical value 6.28 will do the trick. Since this is a common constant, it is also located as a default constant on the Arithmetic Palette under Constants.

* Set up the operations so that your R, and C values flow correctly through the math operations, all the way up until it needs to be coupled with the signal. Feel free to expand this out, and factor accordingly, you may find that the number of operations needed is lower than the provided example uses. Recall that the band pass filter is simply the product of the two filters.

Low Pass Filters:

\[
V_{Out} = V_{in} \cdot \frac{X_C}{\sqrt{R^2 + X_C^2}} \quad \text{for } X_C = \frac{1}{2\pi fC} \quad (1)
\]

High Pass Filters:

\[
V_{Out} = V_{in} \cdot \frac{R}{\sqrt{R^2 + X_C^2}} \quad \text{for } X_C = \frac{1}{2\pi fC} \quad (2)
\]

6. Now that the math is done, we need to finish the data flow. This is a good time to point out that the wires between all of the math operations so far are orange, representing numeric data. The wire from your Simulate Signal Express VI is blue with a dashed line in it. This represents dynamic data. Mixing these in an arithmetic operation will not be an issue, as the result is still dynamic data. However, in more complicated VIs, it is important to know what data type is being handed off to what kind of operation to avoid data conflicts. At this stage we cannot see our filters, and we have not wired our signal to the arithmetic operations. We must add Waveform Graphs for each filter we have added. This is very straightforward.

* Again, Ctrl+click and drag in the first Waveform Graph you created to duplicate it.

* We will wire our simulated signal into the appropriate operations, and their outputs to Waveform Graphs.

We now have a VI which simulates a low pass, high pass, and band pass filter. Your goal now is to modify the Front Panel and Block Diagram to include the 3 dB break point. The calculation for this is available in your previous lab, I have included it here for your convenience. Similar to the previous portions of this example, you will implement the arithmetic steps necessary to calculate the 3 dB break point, and add an indicator. Images of the Block Diagram, and Front Panel of this project follow.

\[
f_C = \frac{1}{2\pi RC} \quad (3)
\]

High and Low Pass: