ME 461 LabVIEW #4

These are optional Exercises for LabVIEW if You are Interested in Learning More LABVIEW Programming.

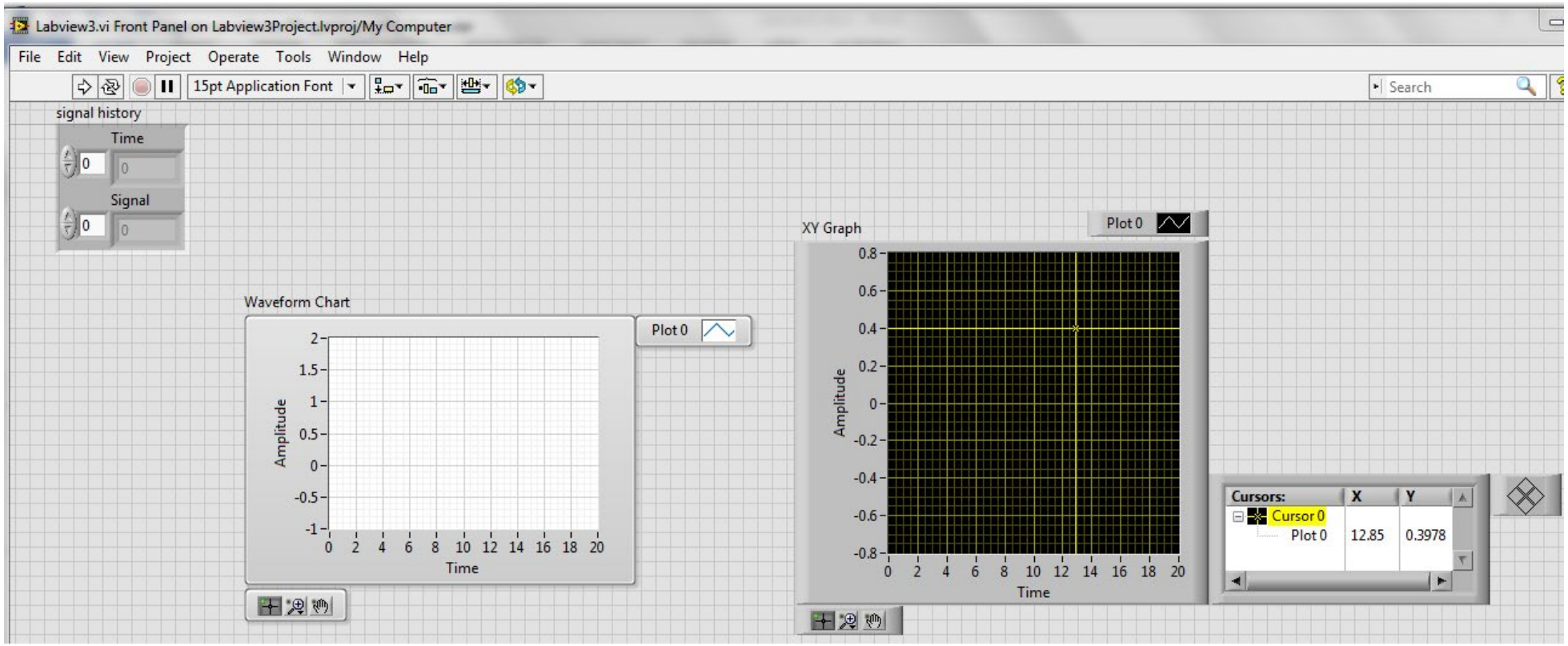
Exercise 1

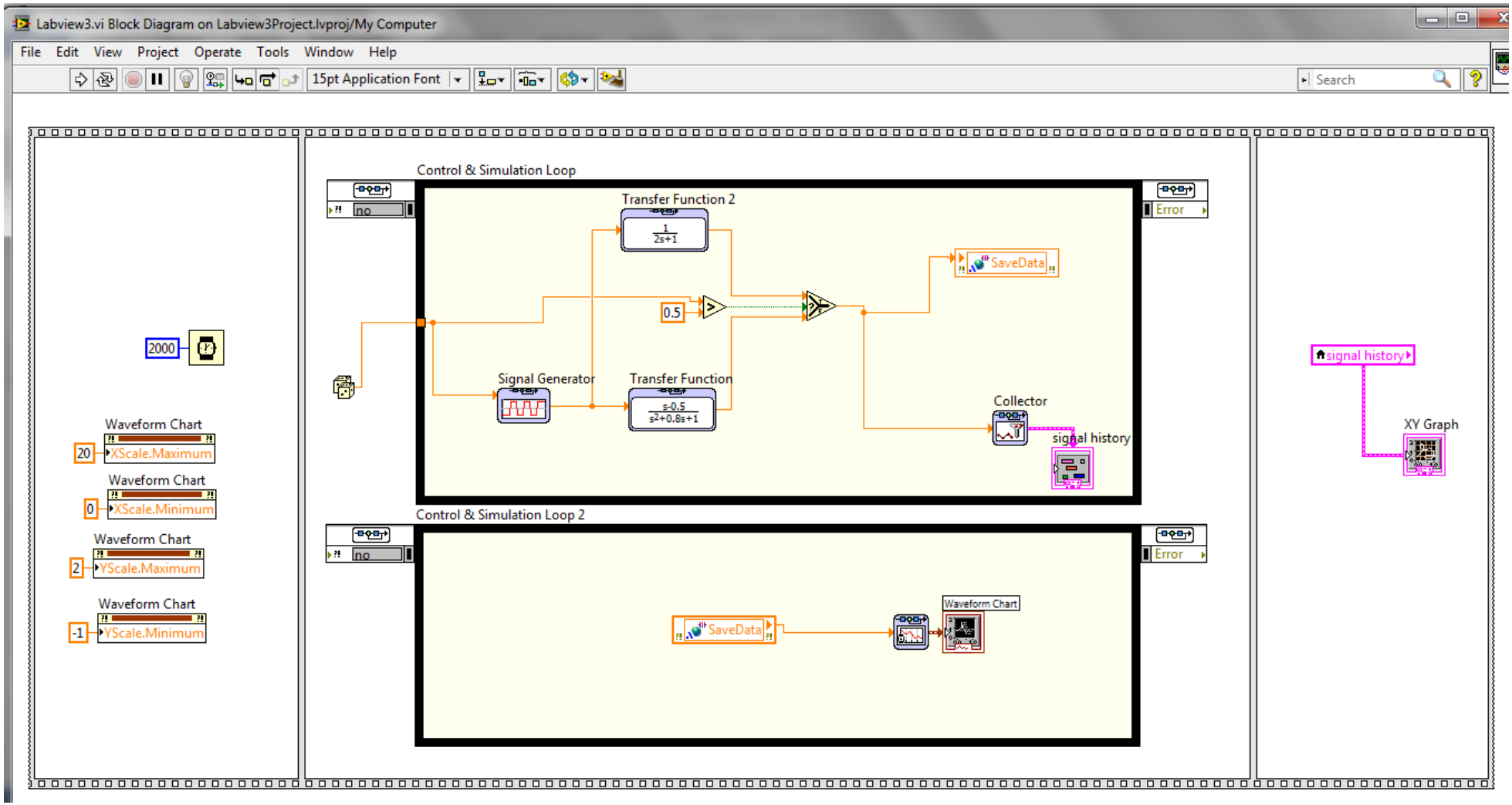
I would like you to go through the "Basics of Control Design and Simulation" tutorial. It can be found at http://coecsl.ece.illinois.edu/me461/labs/NI-Tutorial-BasicsofCDS.pdf . When you are done with this tutorial you will have saved three files: "My Controls Example VI.vi", "My Controls Example With Math Script VI.vi" and "My Control and Simulation Example VI.vi" Demo all three of these Vis working for the first part of your check off.

The last VI you build, "My Control and Simulation Example VI.vi", uses the Control and Simulation Loop. You should be familiar with MathWorks Simulink from previous classes like ME340 and ME360. You can think of the blocks you place inside the Control and Simulation Loop the same as blocks in a Simulink simulation.

Exercise 2

For this next exercise I would like you to reproduce the VI shown in the pictures below. Start out by creating a LabVIEW Project. Inside this project, add a single VI. The reason I want you to use a project is that I would like you to try out using a "shared variable" and these are created in a separate .lvlib file that will also become part of your project. I will give some pointers and instructions below, but they will not be necessarily in the exact order you need to perform the steps or completed steps. In other words, you may need to do a bit of your own hunting to find the necessary blocks and other parts of the program and use LabVIEW help to figure out how to use/wire certain blocks. Many of the blocks I used below come from the "Control Design & Simulation → Simulation" section of LabVIEW

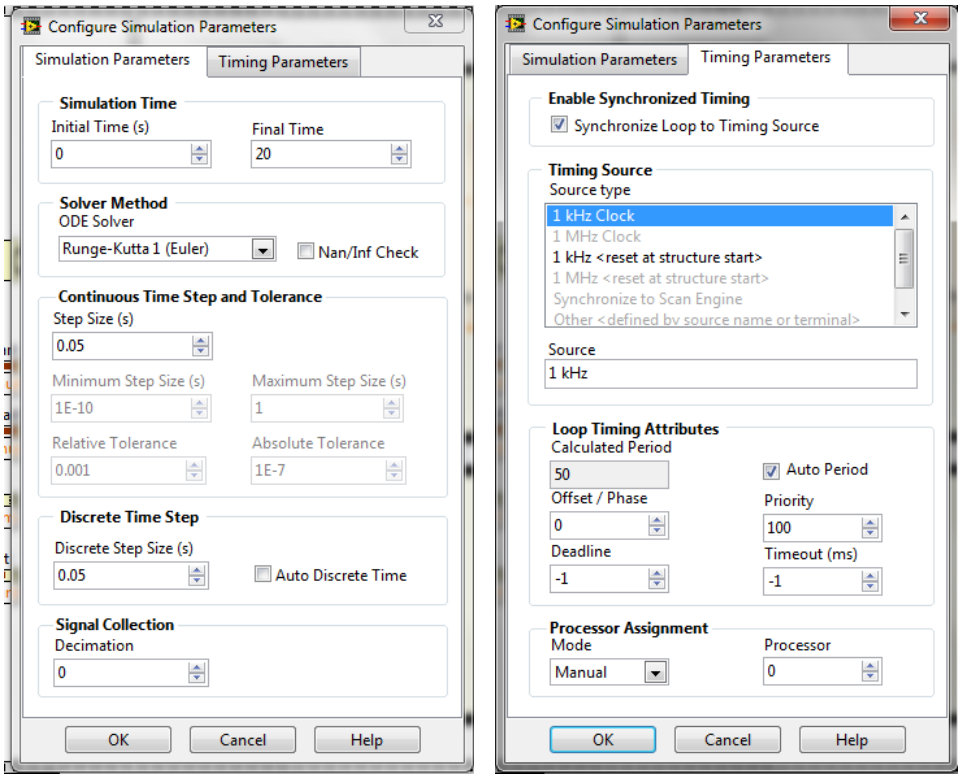

Labview3.vi Front Panel on Labview3Project.lvproj/My Computer
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
signal history
Time
0
0
Signal
0
0
Waveform Chart
Plot 0
Amplitude
Time
XY Graph
Plot 0
Amplitude
Time
Cursors:
X
Y
Cursor 0
Plot 0
12.85
0.3978


Labview3.vi Block Diagram on Labview3Project.lvproj/My Computer
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
Control & Simulation Loop
no
Error
Transfer Function 2
1
2s+1
SaveData
0.5
Signal Generator
Transfer Function
s-0.5
s²+0.8s+1
Collector
signal history
2000
Waveform Chart
20
XScale.Maximum
Waveform Chart
0
XScale.Minimum
Waveform Chart
2
YScale.Maximum
Waveform Chart
-1
YScale.Minimum
Control & Simulation Loop 2
no
Error
SaveData
Waveform Chart
signal history
XY Graph

Pointers/Instructions

1. Why am I using two Control and Simulation Loops? I am splitting the simulation into two parts. One to run the simulation and the second to plot the data. When you start using the simulation loop on the "MyRIO" development board you will find that to achieve consistent sample periods you will need to divide processing between the two processor cores of the MyRIO. So, to prepare you for this I'm showing you here how we are going to plot data. The simulation's "y" output is stored into a "Shared Variable" that has a Real-Time FIFO enabled. In the block diagram view below it is the orange "SaveData" blocks. We will set up one of the simulation loops to run on one processor core and the other simulation loop to run on another processor core.

2. After you add the two simulation loops, double click on their parameters tab in their top left corner which will bring up the Configure Simulation Parameters window. Set it up as in the pictures below. For the second Simulation Loop, set it up as the same except for the Processor Assignment. Set the processor as "1".



3. In your project window, right click on the "My Computer" under your project name and select New → Variable. Give the variable a name. Make the variable type "Single Process" and data type "Double". Click on the RT FIFO item and check the item "Enable RT FIFO". The shared variable is stored in a library .lvlib file. Save this library file with a name of your choosing.

4. To add the shared variables to your application, simply drag the shared variable from the project window into your block diagram window. Then in your Block Diagram, right click on the shared variable and select the appropriate "Access Mode."

5. Block Locations:

    a. Use the Flat Sequence to order when items of your program run. Programming → Structures.

    b. Transfer Function: Control Design & Simulation → Simulation → Continuous The transfer function inside the block can be whatever you choose.

    c. Signal Generator: Control & Simulation → Simulation → Signal Generation

    d. Notice that I set up the Signal Generator to have a terminal input. I want the Amplitude of the Signal Generator to change each run by using the random number generator. If you double click on the signal generator and select the amplitude parameter you can change its parameter source to "terminal."

    e. Collector: Control & Simulation → Simulation → Utilities. The collector is optimized for the simulation loop and is used to collect data for plotting later or even offline. To create the Collector's output variable that I called "signal history", hover your mouse just to the right of the Collector and right click. Select Create →Indicator.

    f. The Waveform Chart and its accompanying block to its left is called a "SimTime Waveform". Find it under Control & Simulation → Simulation → Graph Utilities.

    g. Select and Greater: Programming → Comparison. I am showing you the Select block so you can see how to choose between two signals and only route one of them to a desired block.

    h. Dice (Random Number): Programming → Numeric.

    i. Local Variable: Programming → Structures.

    j. To add the XY Graph to your program you must switch to the Front Panel window. From there right click and find the XY Graph in Modern → Graph.

    k. Property Nodes XScale.Minimum, XScale.Maximum, YScale.Minimum, YScale.Maximum. Property Nodes allow you to change properties of a front panel object (in this case a graph) during your program run. Here I am just setting the range of the X and Y scales. You will see there are a huge number of properties you can change on the fly if you wish. Since there are so many it is hard sometimes to find the correct property to change the item you are interested in. Using Help and trial and error

is the way to finally figure out which property to change. Create these property nodes by right clicking on the Waveform Chart block and selecting Create → Property Node → XScale → Range → Minimum. Do the same for X Max, Y Min, and Y Max. If you right click on the Property Node the top option allows you to change the property to a "Write". (Or "Read" if you need to switch back.)

6. In the Front Panel window, right click on both of the graph objectives and turn on their different "Visible Items". The "Graph Palette" allows you to zoom in and out the graph's data. The "Cursors Legend" is only available in the XY Graph and it creates a cross hair cursor. After you run your program, play around with both the zooming capability and the cursors.

To save your collected data to a file, to load into Excel or MATLAB, you simply right click on the plot and select Export. This allows you to export directly to an Excel file or you can have it copy the data to the clipboard and then you can pull up your favorite editor (I like Notepad++) and paste all the data into that file. Play around with this export feature and make sure you can pull data into Excel or your favorite editor.

Introduction for Ex 3,4,5

For this exercise, I would like you to first reproduce the VI shown in exercise 1. Then, in exercise 2 and 3, create a similar VI that performs the same function as the first using different methods. Exercise 1 utilizes an "if-else" condition, exercise 2 uses a SubVI, and exercise 3 implements the same function in the form of a Formula Node.

Again, for this this LabVIEW start out by creating a LabVIEW Project. Then inside this project, add a single VI. We are using a project this time so that the SubVI you will be creating will be saved in your project. I will give some pointers and instructions below, but you will need to do a bit of your own hunting to find the necessary blocks and other parts of the program and use LabVIEW help to figure out how to use/wire certain blocks. Many of the blocks I used below come from the "Control Design & Simulation → Simulation" section of LabVIEW.

Note: There are a lot of small blocks in the LabVIEW code I give you below for Exercise 1 and 2. I have found if you open the Word document, you can zoom into the code pictures a bit better than the PDF file.

Exercise 3

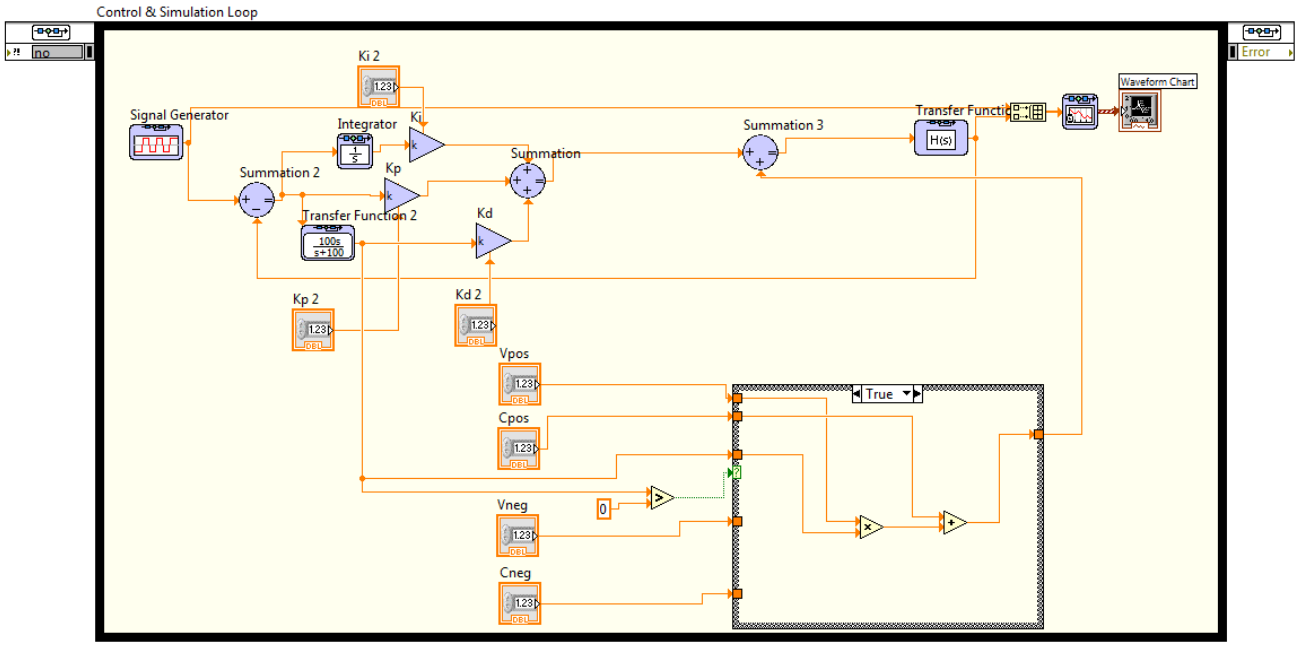
Reproduce the below VI.

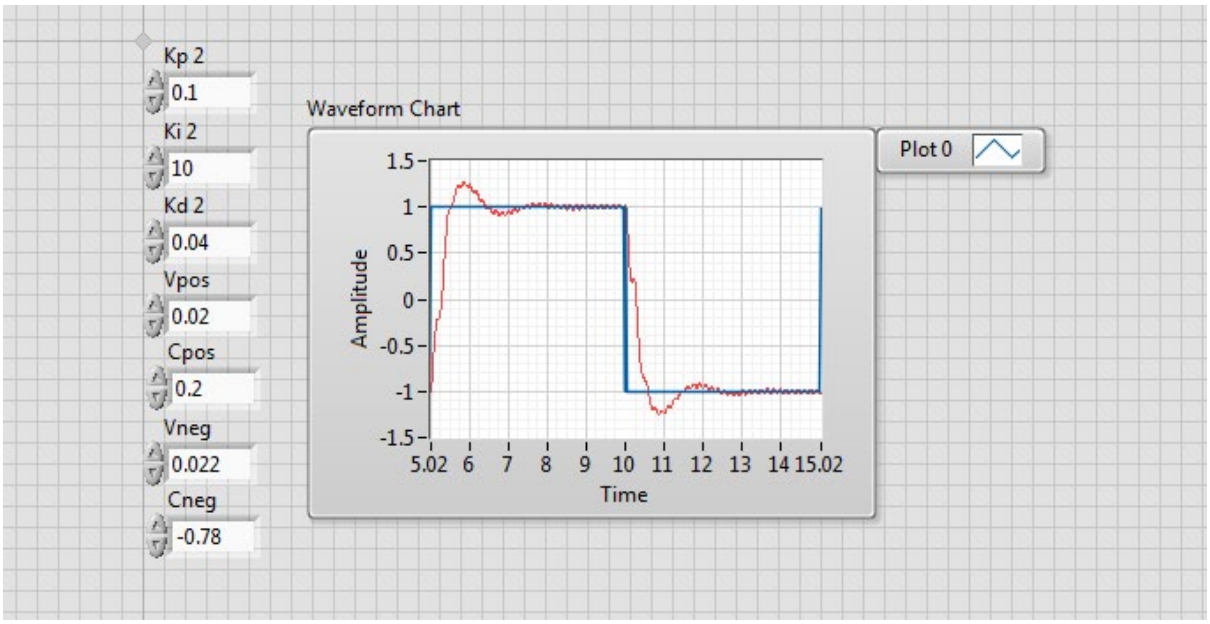
Instructions & tips:

1. The entire program is inside a Simulation Loop. The default simulation settings should work fine for this exercise.
2. You will notice that I am using a Case Structure in this VI. The case structure is implementing the following if statement without explicit code:

```
if (velocity > 0) {
        FricComp = Vpos*velocity + Cpos; // viscous friction*velocity + static friction
} else {
        FricComp = Vneg*velocity + Cneg;
}
```

3. The signal generator is set to square wave with an amplitude of 1 and frequency of 0.1 Hertz
4. Set gain blocks "terminal" mode so that the gain can be changed with a control.
5. We are implementing a PID controller. Gain values are given in the Control Items in the Front Panel window.
6. The plant transfer function is $\frac{1600}{s^3+14s^2+448s+3200}$.
7. $\frac{100s}{s+100}$ is an approximation of a derivative.
8. The graph is a SimTime Waveform block.
9. I am using a Build Array block in front of the SimTime Waveform to allow for plotting two signals on one graph.
10. Run the program with the given constant values below. You should see a similar output to the Waveform Chart shown.



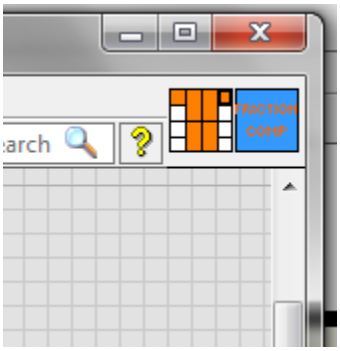


Exercise 4

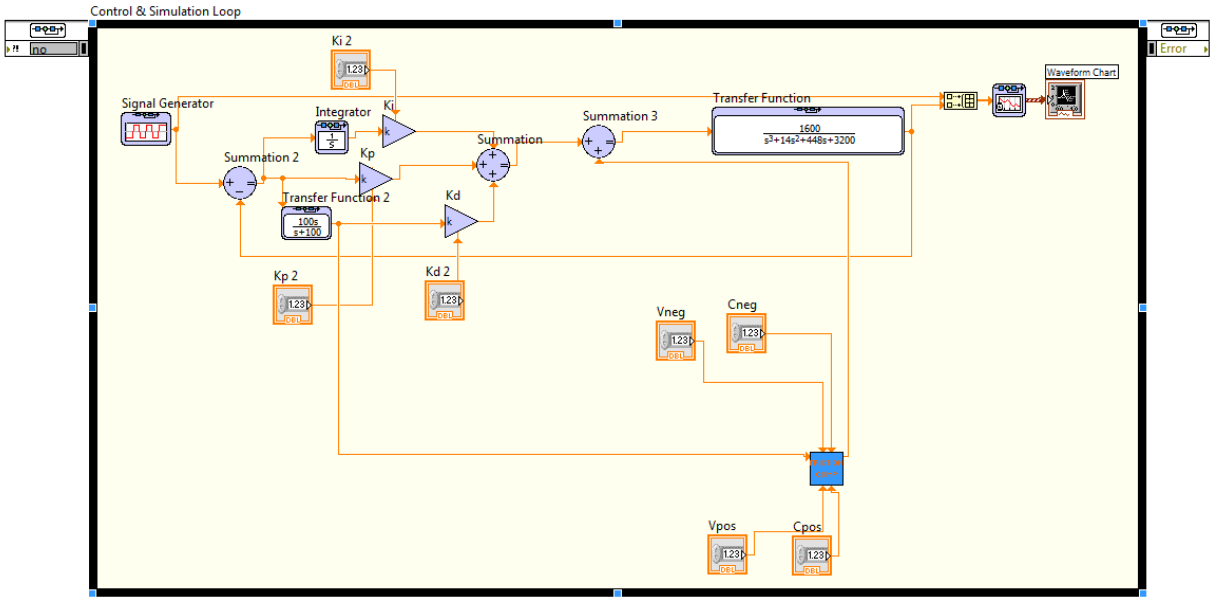
Now, change the Friction Compensation part of the VI into a SubVI. See the following tutorial to help you create this SubVI from the blocks already in your simulation:

https://www.youtube.com/watch?v=zr2qqv1F4kk

Make a simple icon for your VI and connect the inputs and outputs to the pattern shown in the picture below. There is also another picture below of the simulation with the SubVI.



Picture of Icon and Pattern to use



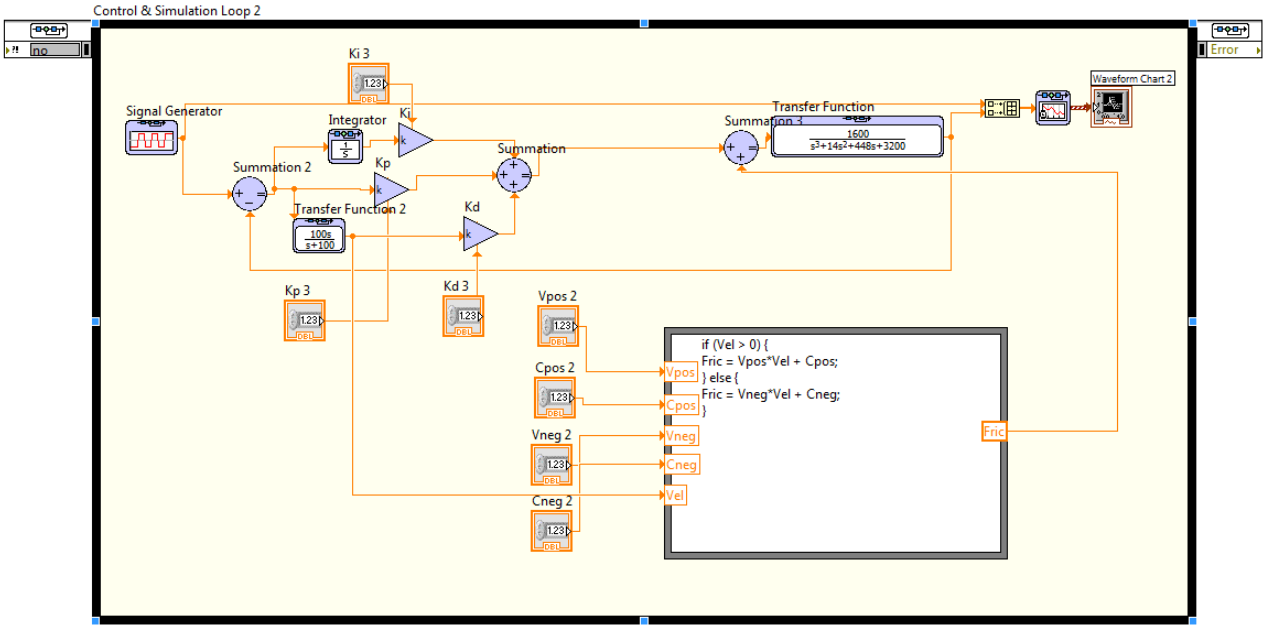
Picture with SubVI

Exercise 5

Now either create a new VI or copy the Simulation Loop and all its contents and run two simulation loops in one VI.

Here in this new VI or new simulation loop, everything is the same except I would like you to replace the SubVI with a Formula Node. A Formula Node allows you to add small amounts of C-code to your LabVIEW program.

To add a variable to a formula node right click on an edge of the formula node and select “Add Input.” To add an output for the formula, right click on an edge and select “Add Output.”


Control & Simulation Loop 2
no
Error
Ki 3
Signal Generator
Integrator
Ki
Summation
Summation 2
Kp
Transfer Function 2
Kd
Transfer Function
Summation 3
1600
s³+14s²+448s+3200
Waveform Chart 2
Kp 3
Kd 3
Vpos 2
Cpos 2
Vneg 2
Cneg 2
if (Vel > 0) {
Fric = Vpos*Vel + Cpos;
} else {
Fric = Vneg*Vel + Cneg;
}
Vpos
Cpos
Vneg
Cneg
Vel
Fric