

LabView Exercise 3, Due before Friday October 22nd at 3:00pm

For this exercise I would like you to reproduce the two Vis shown in the pictures below. Both Vis perform the same task using different methods to create an “if-else” condition and one has you use a SubVI. Again for this this exercise 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.

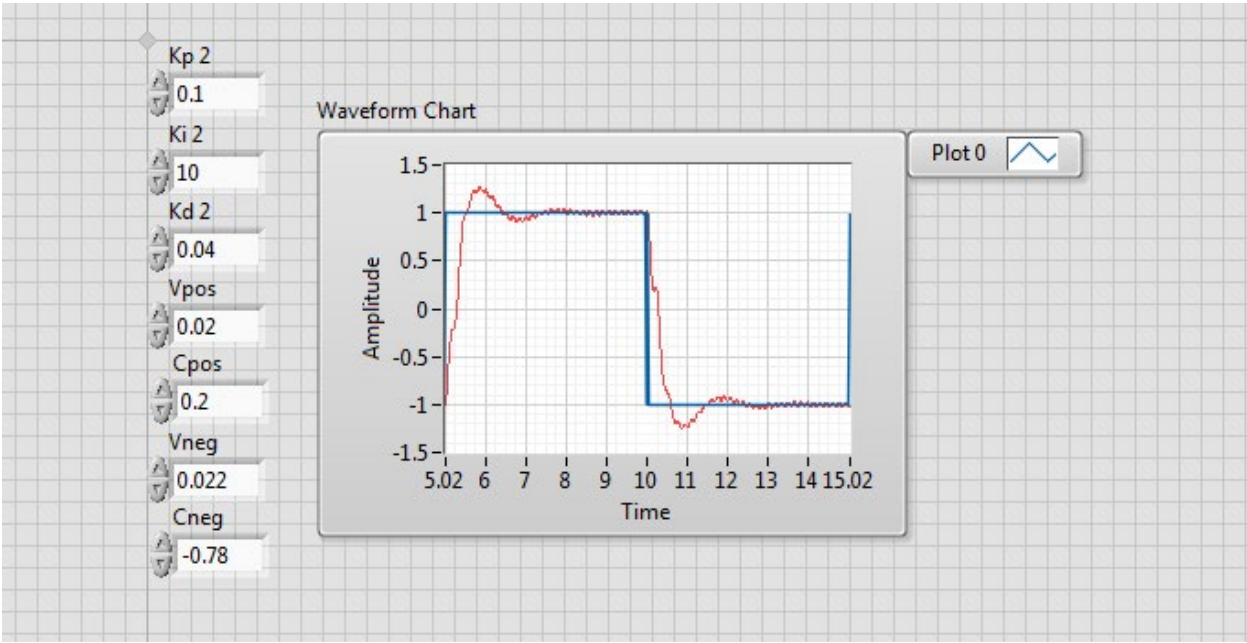
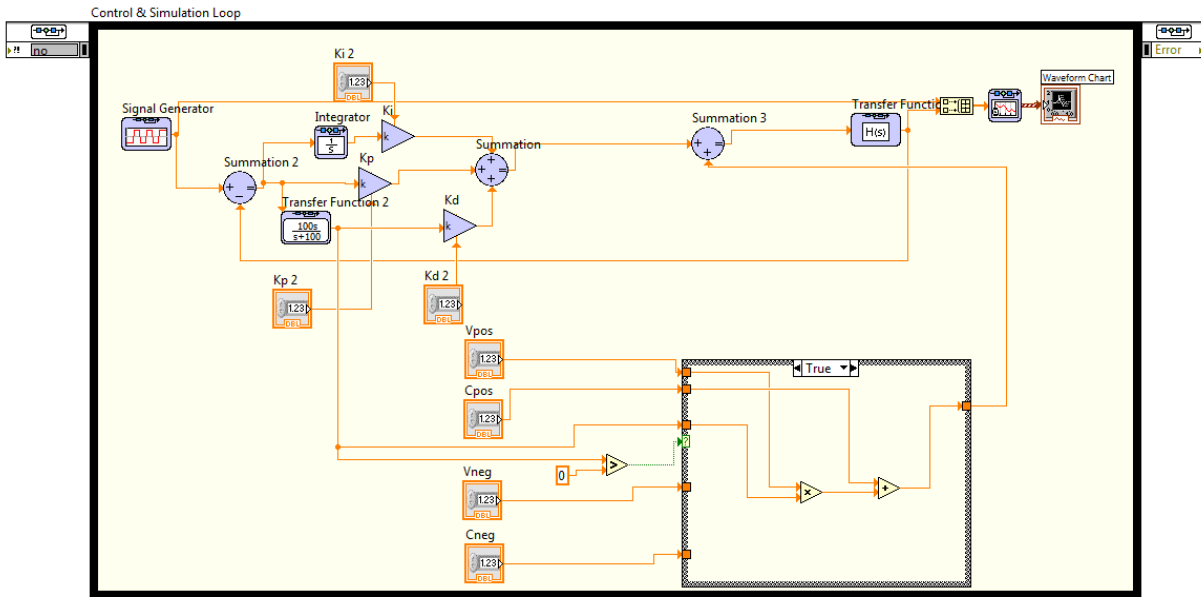
Reproduce the below VI.

Instructions/tips:

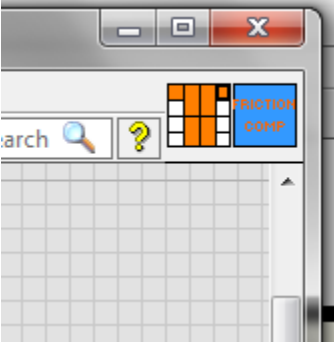
1. The entire program is inside a Simulation Loop. The default simulation settings should work fine for this exercise.
2. You will notice that I am using a Case Structure in this VI. The case structure is implementing the following if statement:

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

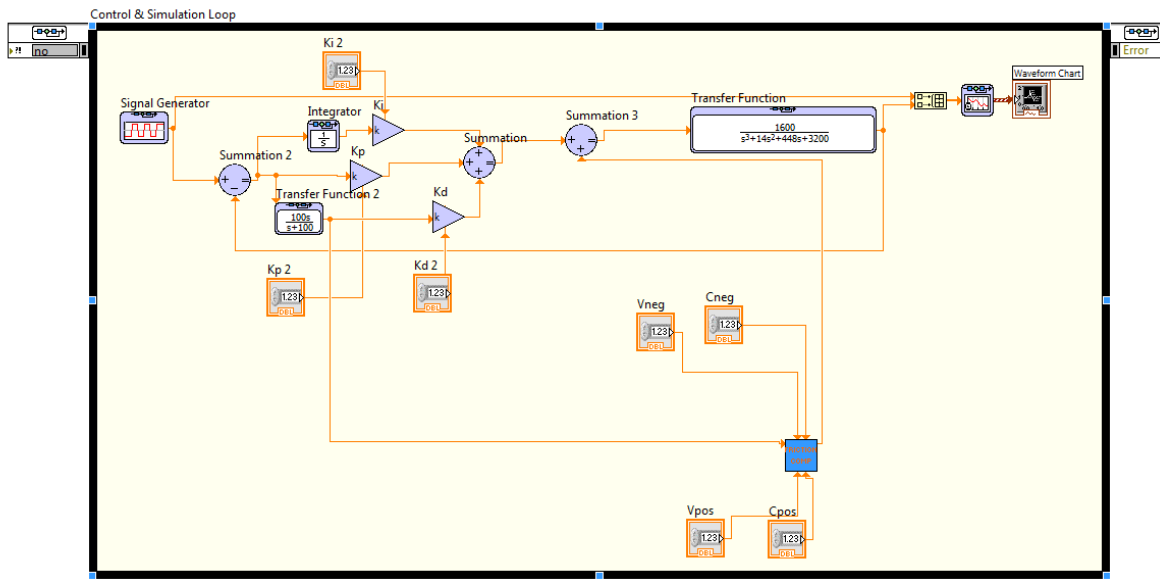
3. The signal generator is set to Square wave, Amplitude of 1 and frequency of .1 hertz
4. Gain Blocks are set to “terminal” mode so that the gain can be changed with a control.
5. We are implementing a PID controller. Gain values are given in the Control Items in the Front Panel window.
6. The Plant transfer function is $1600/(s^3 + 14*s^2 + 448*s + 3200)$.
7. $100*s/(s + 100)$ 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. More steps continue below.



11. Now as a final task for this VI 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=vr2qqv1F4kk>
12. 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 with SubVI



13. Now either create a new VI or you can simply copy the Simulation Loop and all of its contents and run two simulation loops in one VI.
14. 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.
15. 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."

