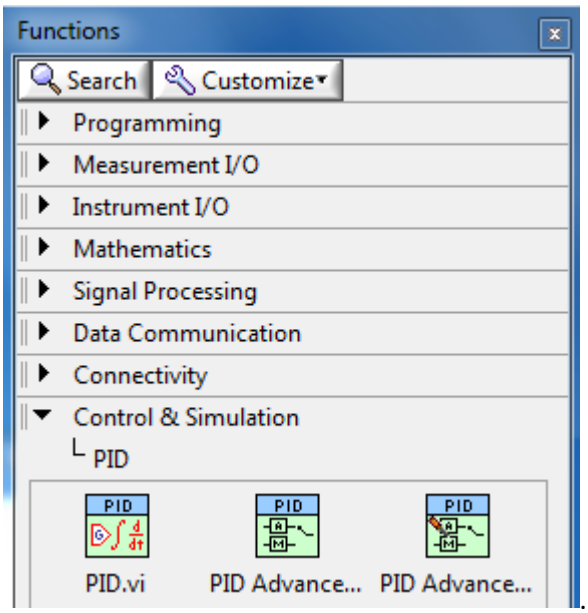
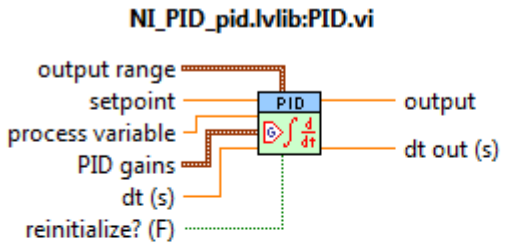
End of Semester Project
Part 2

Note: Due to limited lab equipment, you will probably have to work on this lab outside of normal lab times. This is a group lab (2 people per group)

Objective	To gain firsthand experience with PID Control
Background	See notes from classes on PID and Video links from class website
Procedure	In LabVIEW we will create a PID control program. First - use Sub VI's created for the analog input and output in the last lab by using "SELECT VI" from the functions menu and selecting the previously created VI's (saves you having to redefine the IO points). Now goto the Functions Menu | Control & Simulation | PID and select the first Block (PID.vi) from the menu and place it in your VI. 

Below is a help for the block:

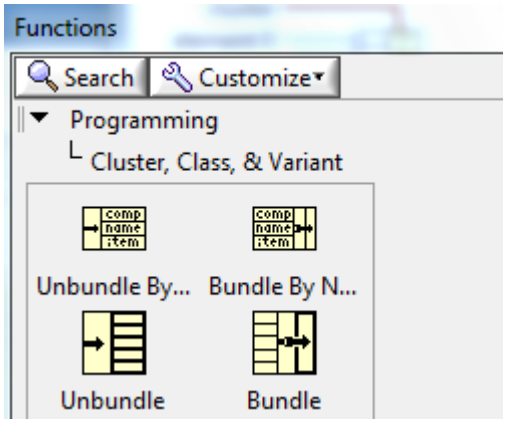


Implements a PID controller using a PID algorithm for simple PID applications or high speed control applications that require an efficient algorithm. The PID algorithm features control output range limiting with integrator anti-windup and bumpless controller output for PID gain changes. Use the DBL instance of this VI to implement a single control loop. Use the DBL Array instance to implement parallel multi-loop control.

You want your Process Variable (PV) to be your speed in RPM and your output to go to the data port of the analog write. Create Controls for Setpoint (SP), PID Gains and dt(s). Create Constant for Output Range and make Output High 10 and Output Low -10. You will want indicators for PV and Output. Lastly a constant for dt(s) of 1

Please create two graphs:

Graph one should have two lines, the first being your PV and the second should be the SP. To do this graph, place a waveform chart on the VI front panel. Right Click on the chart and select Properties. Go to the Appearance Tab and change Plots Shown to 2.Now go to the Block diagram, the waveform chart should be there. But to bring in two data points to the graph you need to use a “BUNDLE” block to merge the two signals. This can be found on the menu shown below:

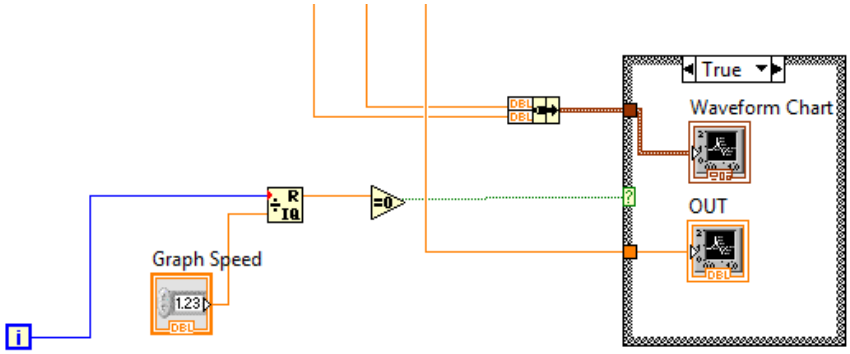


(Note – use Bundle, not Bundle by name, for this).

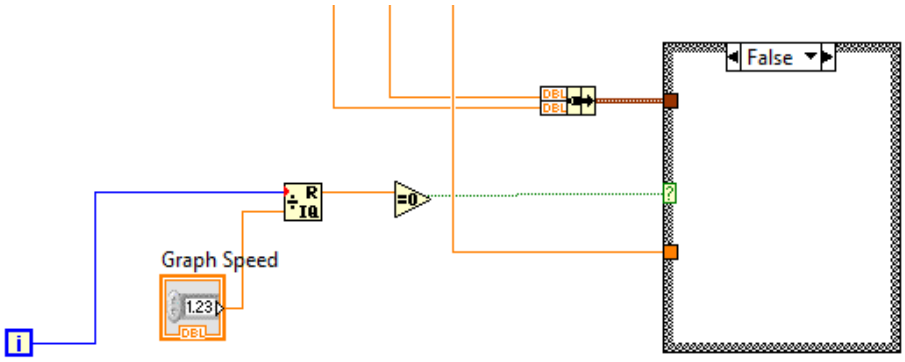
It should look as follows:

Create a 2nd chart that is just the output going to the analog write data.

It would be beneficial if you could adjust the time the graphs update and make it slower than the time it takes to execute the PID code. To do this use the following code (placing the two Waveform Charts created above in a Case Structure (found next to the While Loop in the Function menu under Structures)

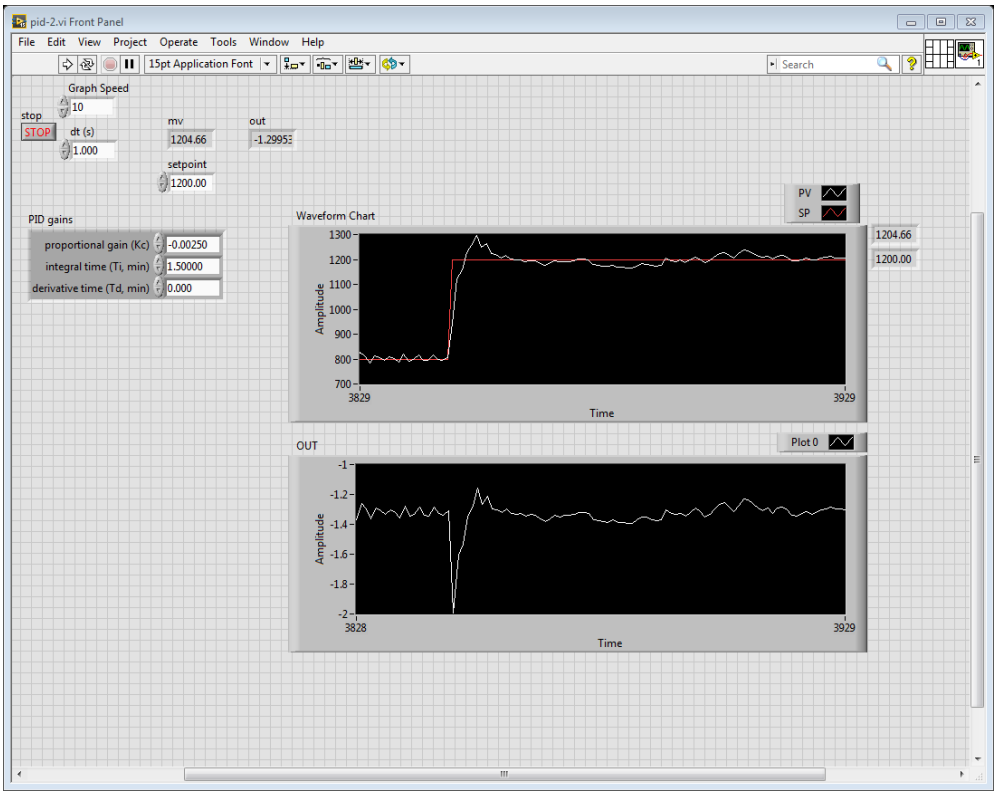


What it does is it takes the loop iteration count (ie the number of times the loop has executed) and divides it by the “Graph Speed” if the REMAINDER is zero it will then update the graph (TRUE Condition). The FALSE Condition is left blank

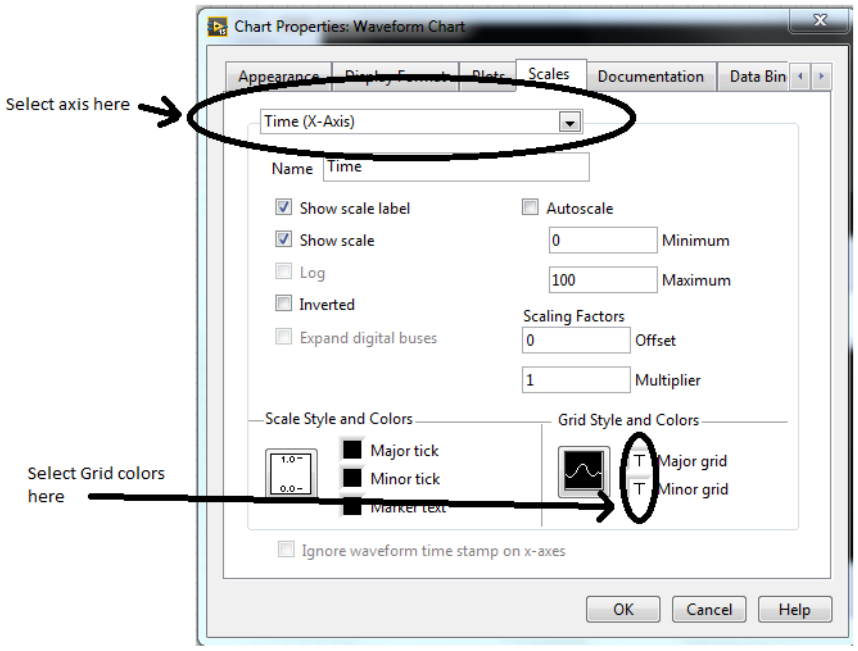


You will want to have everything in a While Loop. You will want to loop to have a 10ms wait timer within the loop.

The display should look something like the following:



It is helpful to add x and y grid lines to your graphs. To do that, right click on the graph and select PROPERTIES. Select the “Scales” tab. Select the axis you want to add a grid to and select the color (I would suggest a lighter shade of grey for the Major Grid and a darker shade of grey for the minor grid).



Now you will need to tune the loop (see next lab for procedure)