Introduction to LabVIEW for Control Design & Simulation

Ricardo Dunia (NI), Eric Dean (NI), and Dr. Thomas Edgar (UT)

Reference Text : <u>Process Dynamics and Control</u> 2nd edition, by Seborg, Edgar, Mellichamp, Wiley 2004

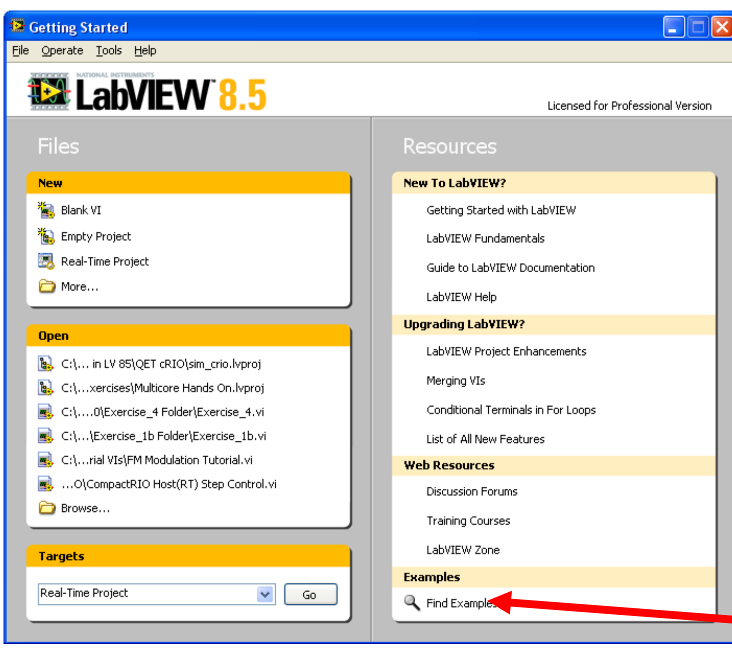
LabVIEW, which stands for *Laboratory Virtual Instrumentation Engineering Workbench*, is a graphical computing environment for instrumentation, system design, and signal processing.

The *Control Design and Simulation (CDSim)* module for LabVIEW can be used to simulate dynamic systems. To facilitate model definition, CDSim adds functions to the LabVIEW environment that resemble those found in SIMULINK. There is also the ability to use m-file syntax directly in LabVIEW through the new MathScript node.

The purpose of this tutorial is to introduce you to LabVIEW and give you experience simulating dynamic systems. . In the first section, you will build a model of the open-loop system for the second order plus time delay process $G(s)=\dfrac{2e^{-s}}{(10s+1)(5s+1)}$ and determine the unit set-point and unit disturbance responses. In the second section, you will build a closed-loop model of the same process. After the closed-loop model is constructed, you should simulate the unit disturbance response and the unit set-point response for two different PID controller tuning methods, ITAE (set-point) and ITAE (disturbance), in Table 12.3 (SEM).

Log onto a PC computer. Click Start->National Instruments->LabVIEW . Open LabVIEW. (Note : as of LabVIEW 8.6, the Control Design & Simulation module is also supported for Mac)

Start a new program (VI)



Browse example VIs

Figure1. Initial LabVIEW Screen

To start a new program (called *VI* for *V*irtual *I*nstrument), click “Blank VI”

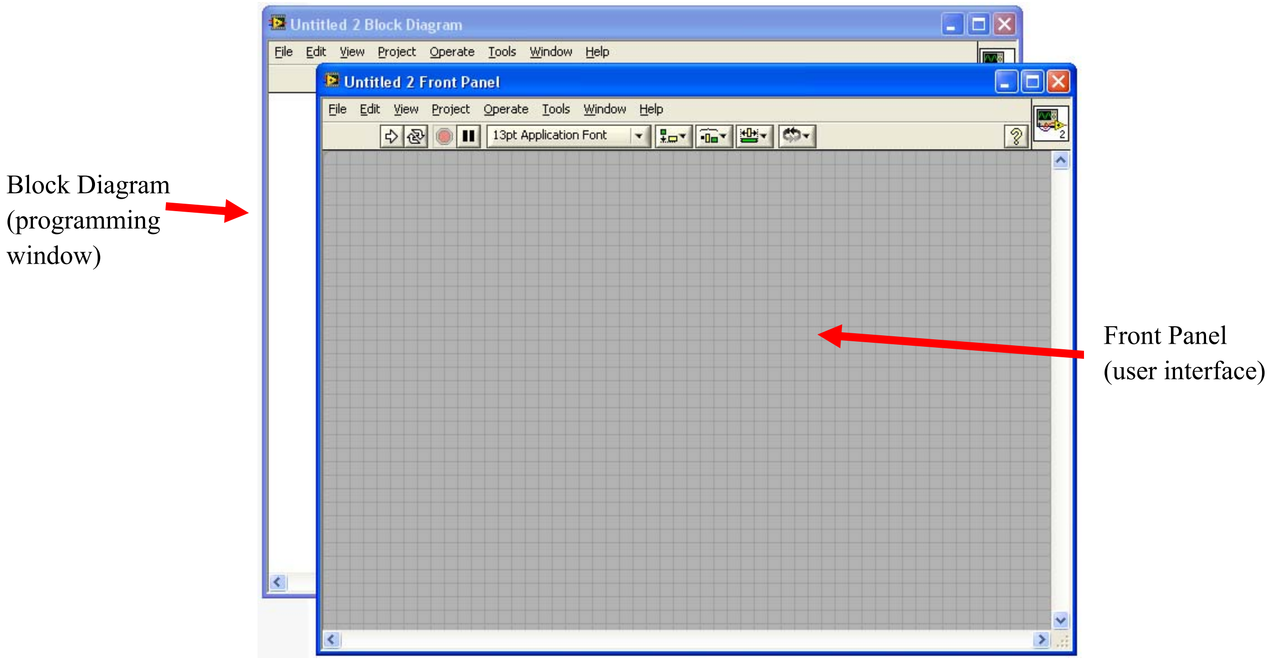


Figure 2. LabVIEW new VI

Click in the block diagram to view the area where graphical programs are written. Right-click inside the block diagram to view the palette of functions used in creating programs. Select the Control Design & Simulation->Simulation palette to view the library of simulation functions.

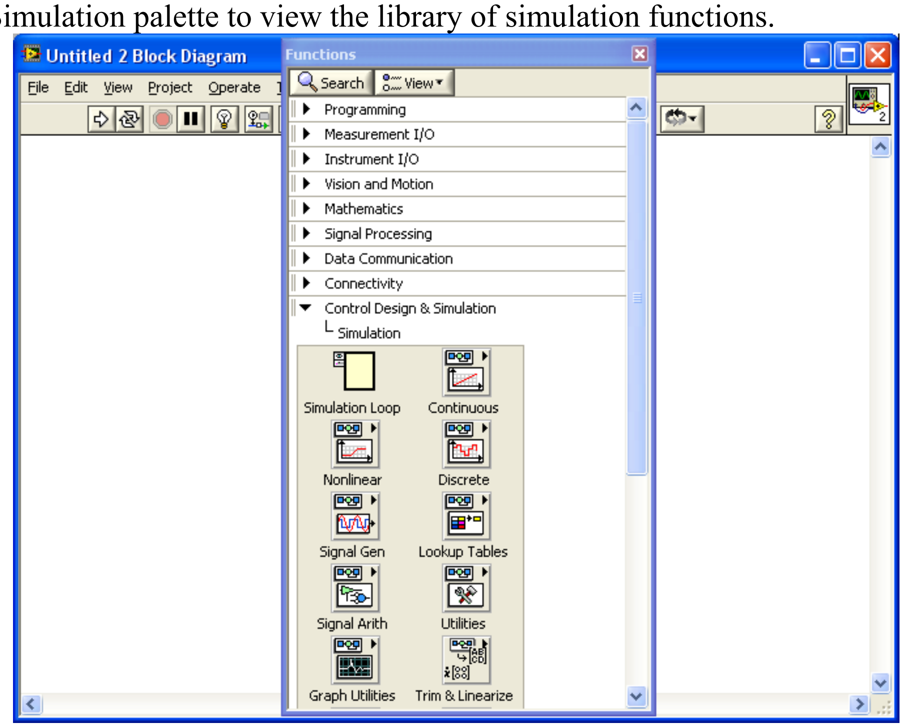


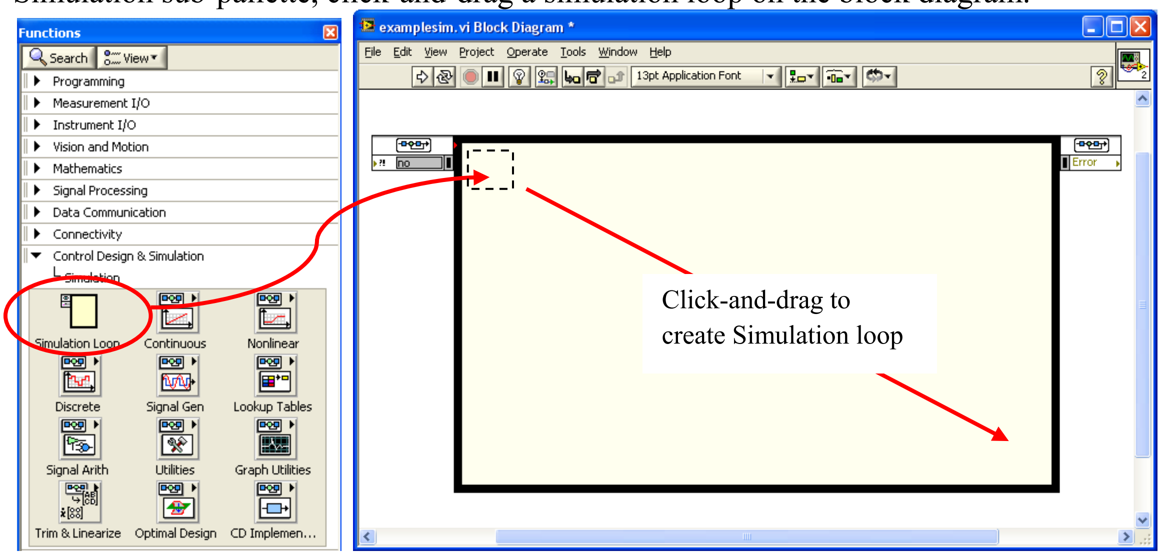
Figure 3. Simulation functions sub-pallette
In Control Design & Simulation pallette

In the next section, you will build a model of the open-loop system for the process mentioned earlier and determine the unit set-point and unit disturbance responses.   The following steps will guide you through the discussed tasks.

Construction of an Open-Loop Block Diagram (Chapters 4, 5, 6, and 7)

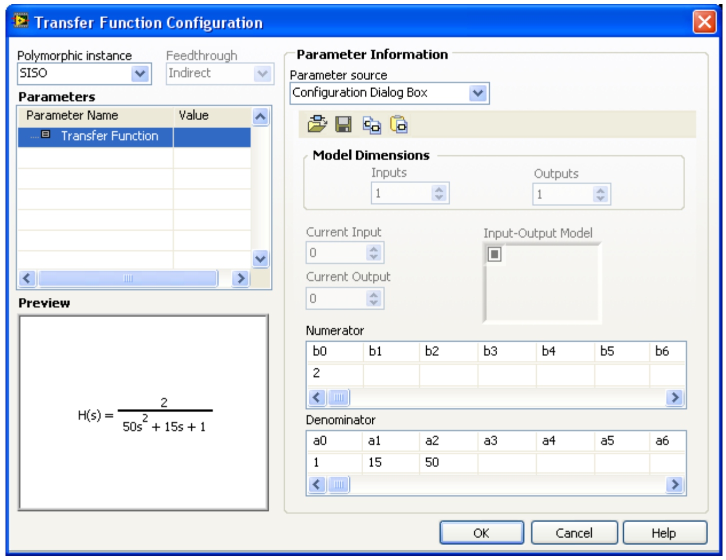
1. Open a new VI by selecting **File->NewVI.** The new window will be titled *untitled*. You will build your closed-loop model in the block diagram. Save the empty model by choosing **File->Save** . Name the model, *examplesim.* From this point on, the model will be referred to as *examplesim.*

2. Click on the block diagram, then right-click to bring up the functions palette. From the Simulation sub-pallette, click-and-drag a simulation loop on the block diagram.



3. Place the Transfer Function and Transport Delay blocks from the “Continuous” pallette, respectively, to *Examplesim*. Connect the output of the Transfer Function block to the input of the Transport Delay block. Click on the “*Transfer Function*” label and rename to “*Process TF*”. This block represents the process. Note that in this problem, the process is $G(s) = G_vG_pG_m$, not G_p.

Open the dialog box of Process TF by double clicking on it. Specify Numerator as [2] and Denominator as [1 15 50]. This indicates the transfer function $2/(50s^2 + 15s + 1)$. The Transfer Function block allows specification of vectors for the numerator and denominator from either a configuration dialog box, or a terminal from the block diagram. The vector elements are treated as the coefficients of *ascending*



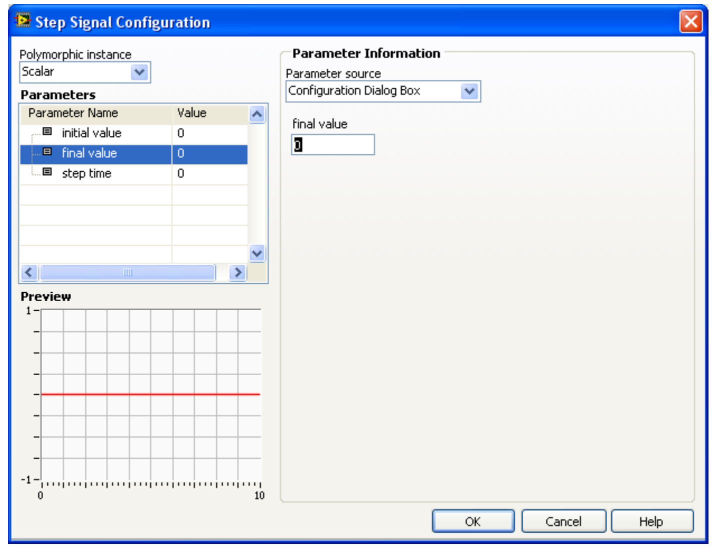
powers of s in the polynomials representing the numerator and denominator of the transfer function. To see the denominator polynomial of s completely displayed in the block's icon, you may have to resize the block's icon.
Double-click on the Transport Delay and set Time delay to 1. Note that the Transport Delay block can be used to represent other types, such as measurement delay.

4. Copy the Process TF and Transport Delay blocks and place the copies slightly above the originals. The copies will automatically change names to *"Process TF1"* and *"Transfer Delay1"*. To quickly copy the original blocks, select both of them, hold the CTRL key and drag using the left mouse button. Rename the Process TF1 block *"Disturbance TF"*. In this example G_p and G_d are the same, so the Numerator and Denominator parameters in the dialog box of Disturbance TF are not changed (See Figure 4).

5. Place a copy of the Summation block, located in the "Signal Arithmetic" pallette, to the right of the Transport Delay block. Right-click on the summation block and select Visible Items->Label to see the label "Summation". Connect the output from each Transport Delay block to the input of the Summation block. The number of inputs and their polarity can be modified from the dialog box. Later in the tutorial you will be required to do this.

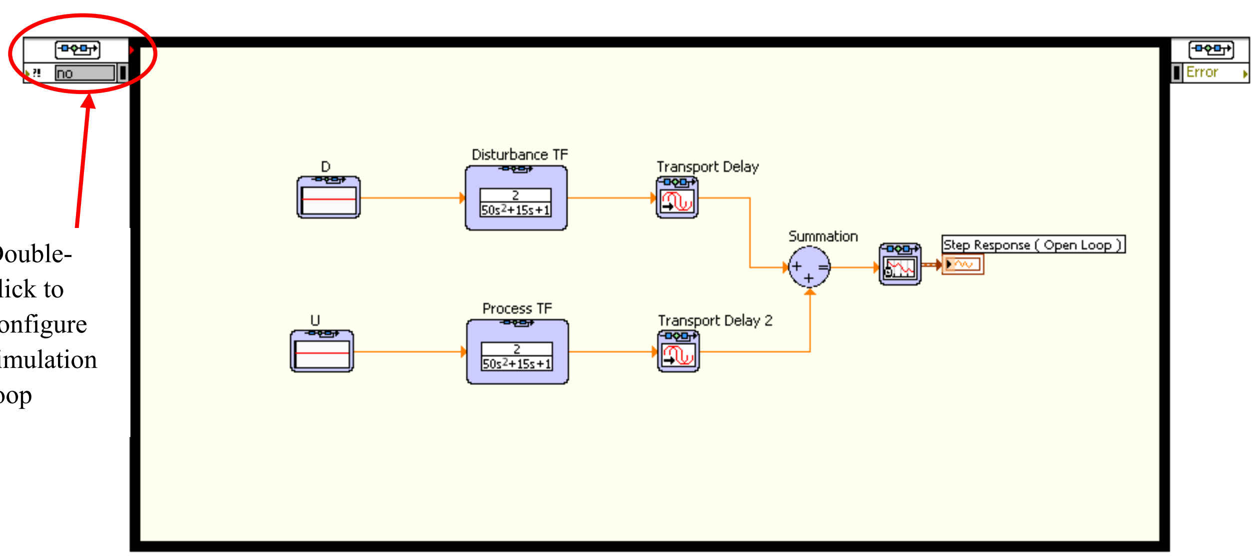
6. Place a SimTime Waveform graph, from the "Graph Utilities" palette, to the right of the Summation block. Connect the output of the Summation block to the input of the SimTime Waveform block.

7. Place a Step Signal block, from the "Signal Generation" palette, to the left of Disturbance TF and connect it to the input of Disturbance TF. The Step Signal block generates a step function. The initial value, final value and step time (time at which the step occurs) of the function can be specified. For now, double-click to open its dialog box and set the initial value, and final value, and step time to zero, i.e., disabling the block. Rename the block "D".



8. Place a copy of D to the far left of Process TF and rename the new block "*U*". Connect U to the input of Process TF. Double-click on U and set Step time to 0, Initial value to 0 and Final value 1. U will generate a unit step function in the manipulated variable at time zero. The model developed to this point is a model of the open-loop system. It should look similar to the model below.

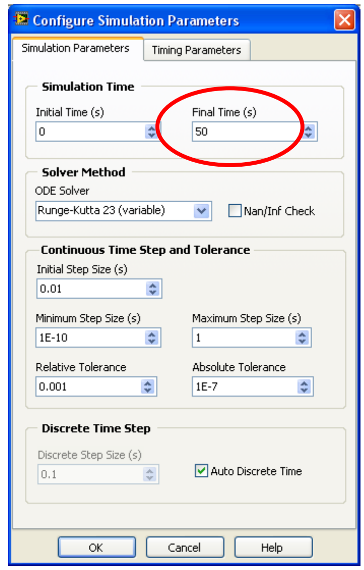
9. Now we are ready to simulate the open-loop response of the system. To select the integration technique and parameters to be used during simulations, double-click on the left terminal of the Simulation loop.



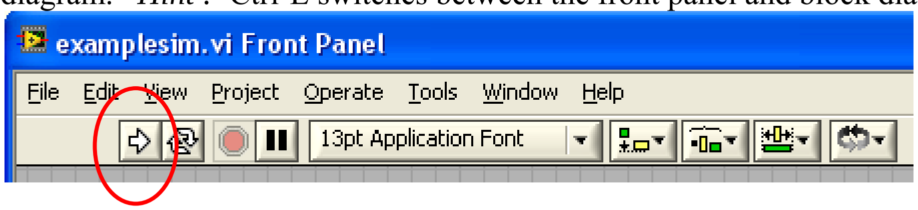
Double-click to configure simulation loop

Figure 4. Open-Loop Block Diagram

A dialog box is opened showing all the simulation parameters that can be modified. Set the Final Time to 50 and the Max Step Size to 1. Note that the LabVIEW Simulation loop includes an ODE solver. The maximum step size determines the largest step LabVIEW uses in numerically integrating the ODE . Since this system is easy to numerically integrate, a max step size of 1 will result in a smooth curve. Larger step sizes will produce more jagged curves.



Run the simulation by clicking the ***Run*** arrow on either the front panel or the block diagram. *Hint :* Ctrl-E switches between the front panel and block diagram.



10. The response will be automatically plotted. Double-click on the title to change the name of the plot. You can also right-click on the plot to view axis settings, autoscaling, and other plot parameters.

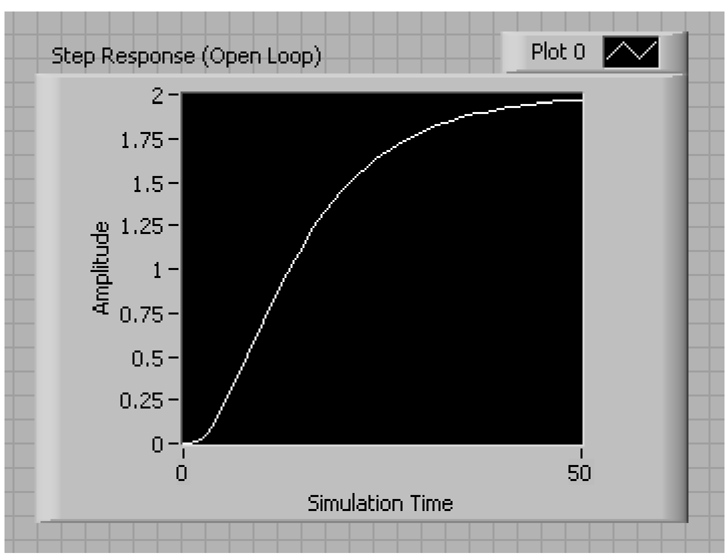


Figure 5. Unit step response of open-loop system

11. Now simulate the open-loop unit disturbance response. Double-click on U and set final value to 0. This eliminates the unit step in the manipulated variable. Double-click on D and set step time to 0 and final value to 1. This creates a unit disturbance step. Again, hit the ***Run*** arrow to begin the simulation. The front panel will show the response. Double-click on the graph title to replace `Step Response (Open-Loop)` with `Disturbance Response (Open-Loop)`. Figure 6 shows the resulting plot. Notice that the open-loop set-point response and disturbance response are the same. Why is this expected?

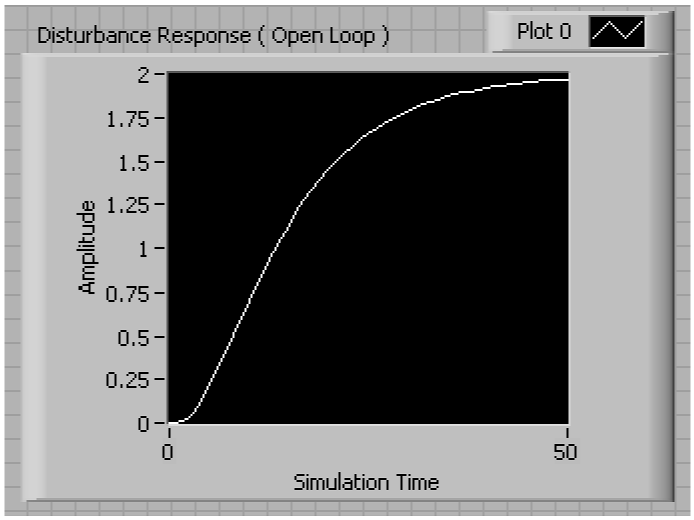


Figure 6. Unit disturbance response of open-loop system

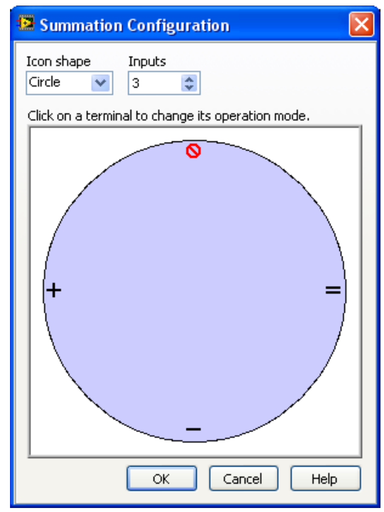
12. If you are not continuing to the next section, save the file *examplesim* so that you can use it in constructing a closed-loop block diagram.

Construction of Closed-Loop Block Diagram (Chapters 11 and 12)

1. Open the file *examplesim* if it is not already open.

2. Click on the connection between the U block and Process TF block and delete it. Rename the U block to "Y_{sp}". This block will be used to produce a step change in the set-point.

3. Place a copy of the Sum block to the right of Y_{sp}. It will automatically be given the label "Summation2". Open its dialog box and change the the lower input from + to -. The top input will have a + located to the right of it while the bottom input will have a – located above it. Therefore, the output of Summation2 will be the top input minus the bottom input.



Connect the output of Y_{sp} to the top input of Sum1. Also, connect the output from Sum to the bottom input of Sum1. This can be done by clicking on the bottom input of Sum1 and dragging the arrow to the output of Sum. The output of Sum1 is the error between the set-point, Y_{sp}, and the controlled variable, Y. Your model should look like Figure 7:

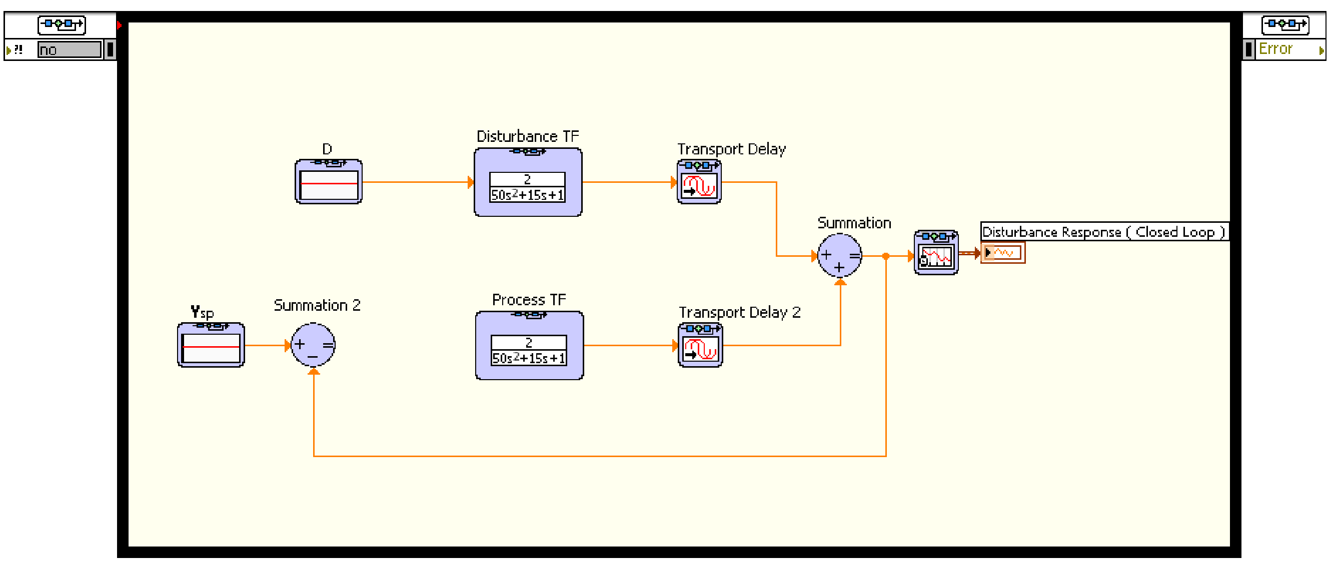
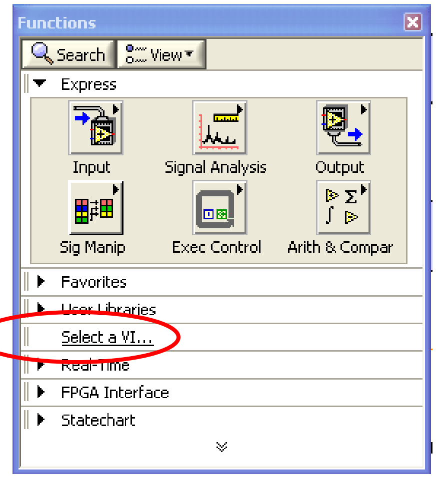


Figure 7. Partially Completed Closed Loop Diagram

4. We are now ready to add the PID feedback controller to the loop. There are several versions of the PID controller block available in LabView, however, for our purposes we have a special version of available for free at the University of Texas Process Controls class website:

   http://www.che.utexas.edu/course/che360/links.html

   Look for the link "PID Controller Block .vi for LabView Tutorial" and download and save the file to your preferred location.



5. Now, return to LabView and right-click in the block diagram to bring up the functions palette, and click *Select a VI*. This allows you to bring in any user-defined LabVIEW VI into your current program. Click on the path where PID Controller.vi was saved. Then select the "PID Controller" and drag it to the right of the newest sum block. Connect the output of the Summation2 to the input of PID controller and the output of PID controller to the input of Process TF. Double-click on the PID controller and enter the ITAE (disturbance) controller settings given in Table 12.3 of your textbook, as well as at the end of this document. Please note that PID controller settings are K_c, τ_I, and τ_D where P = K_c, I = $K_c/\ \tau_I$, and D = $K_c{*}\ \tau_D$, so numerical values of P, I, and D should reflect these definitions. The model you have developed represents the closed-loop system. Your model should now look similar to Figure 8:

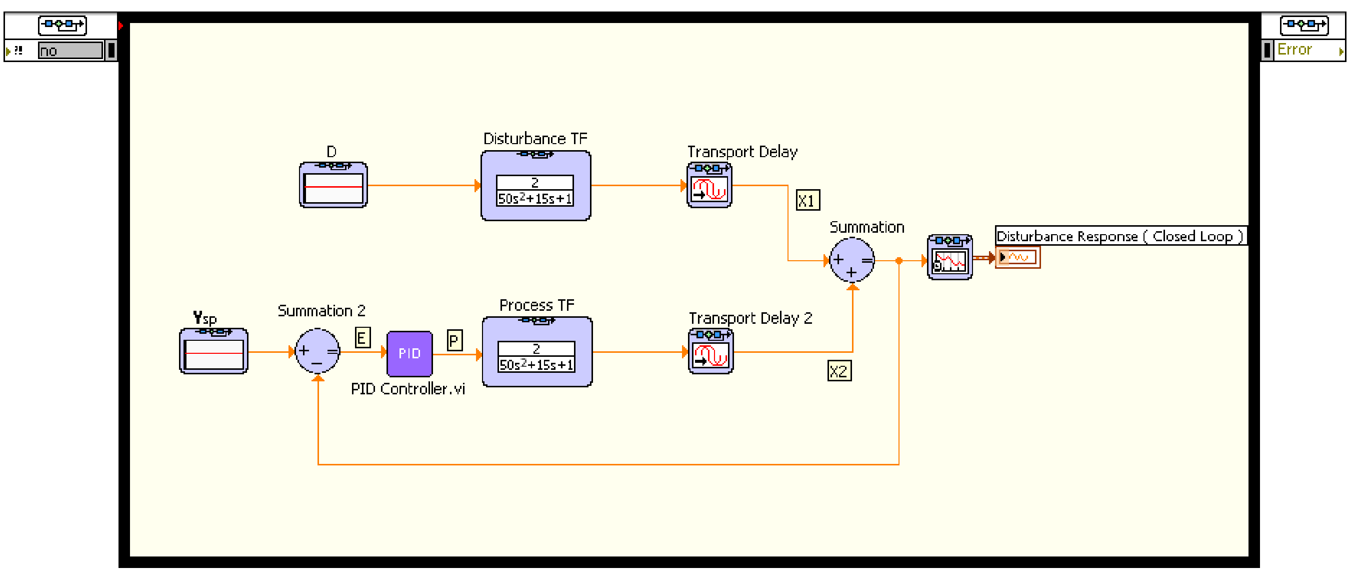
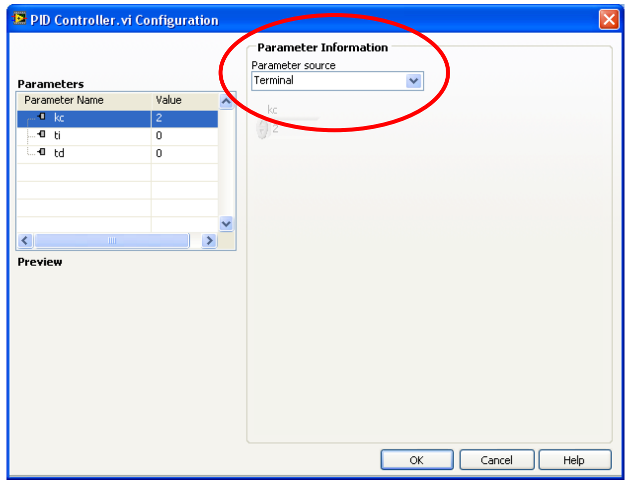


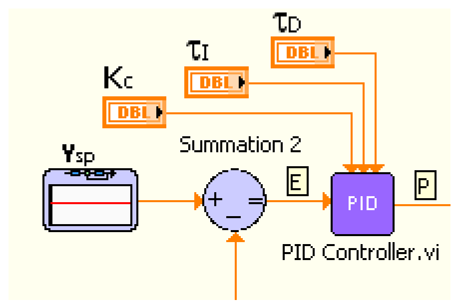
Figure 8. Closed-Loop Diagram

Note that some text has been added to the block diagram shown on the previous page. Simply by double-clicking on a point in *examplesim* and typing, text is added to the diagram at the point where you clicked. The E, P, X1, X2 text in the block diagram have no effect on its operation.

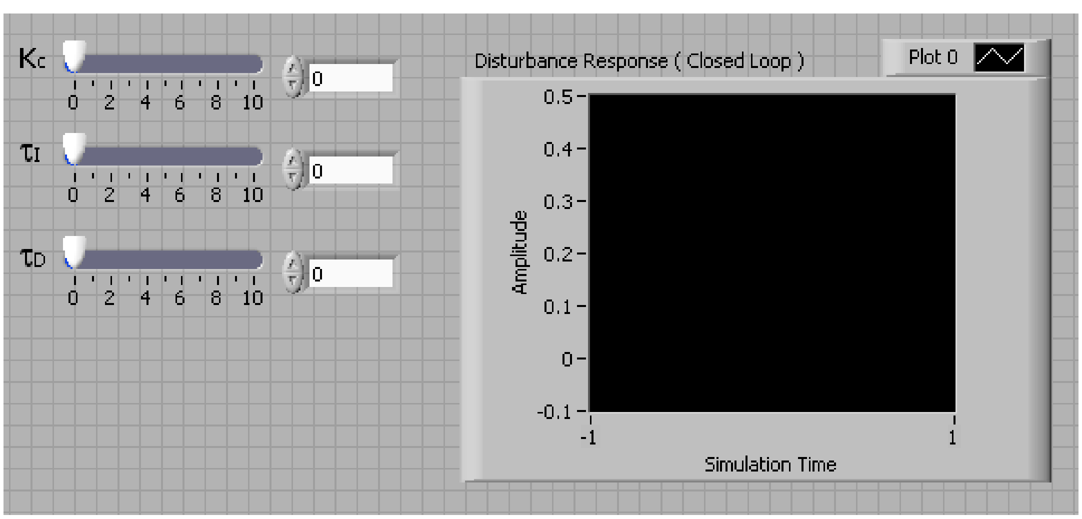
6. An important feature of LabVIEW is interactivity. We can use this capability to make the PID controller gains interactive from the front panel, rather than having to edit them on the block diagram. Double-click on the PID Controller.vi and change Parameter source from *Configuration Dialog Box* to *Terminal.*



7. Select Ctrl-H to bring up Help on the PID Controller.vi and see where the gain terminals are located. (they come in from the top)
Right-click on each terminal and select Create->Control to automatically wire a control to the terminal.
The inputs to the PID Controller.vi should look like this :



By default, LabVIEW creates a standard Numeric control, but this can easily be changed. Go to the front panel , right-click on the control, select Replace->Horizontal Pointer Slide. Then right-click on the new control and Visible Items->Digital Display. This way, gains may be entered either from the slide or typed in to the numeric control. Do this for each PID gain.
The front panel should look like this :



8. Now we are ready to simulate the closed-loop response of the system. We will start with the set-point response. Click on block D and set the Final value to 0 so that no step in the disturbance will occur. Create a step in the set-point by clicking on Y_{sp} and setting the final value to 1. **Run** the simulation. The resulting graph will be for the unit set-point response, because D (disturbance) has been disabled.

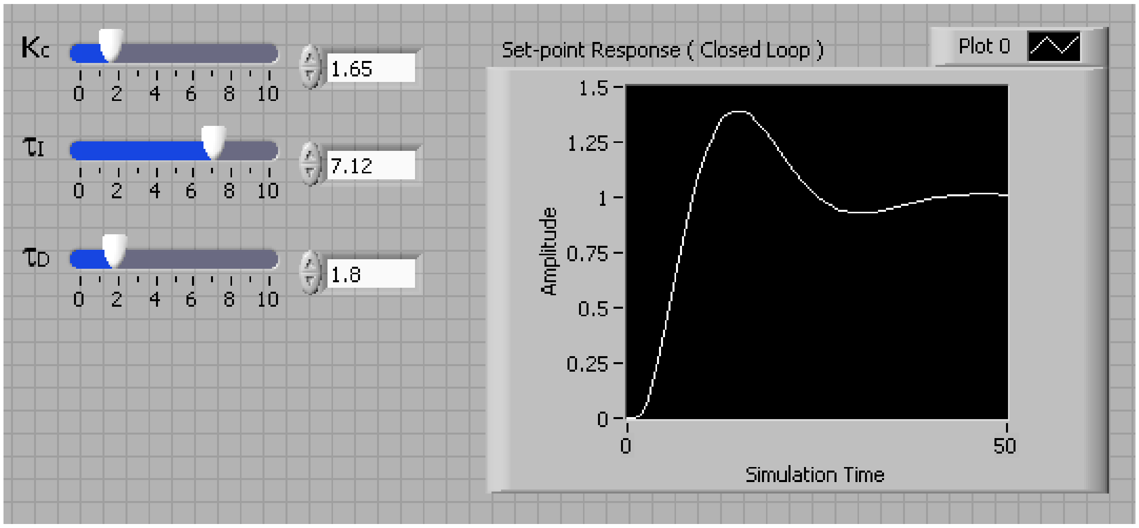


Figure 9. Unit set-point response for ITAE (disturbance) settings.

9. Now simulate the unit disturbance response. Double-click on Y_{sp} and set final value to 0. Double-click on D and set final value to 1. Again, **Run** the simulation. Label this plot the same way as the previous one except replace the title `Set-point Response` with `Disturbance Response`. Figure 10 shows the resulting plot.

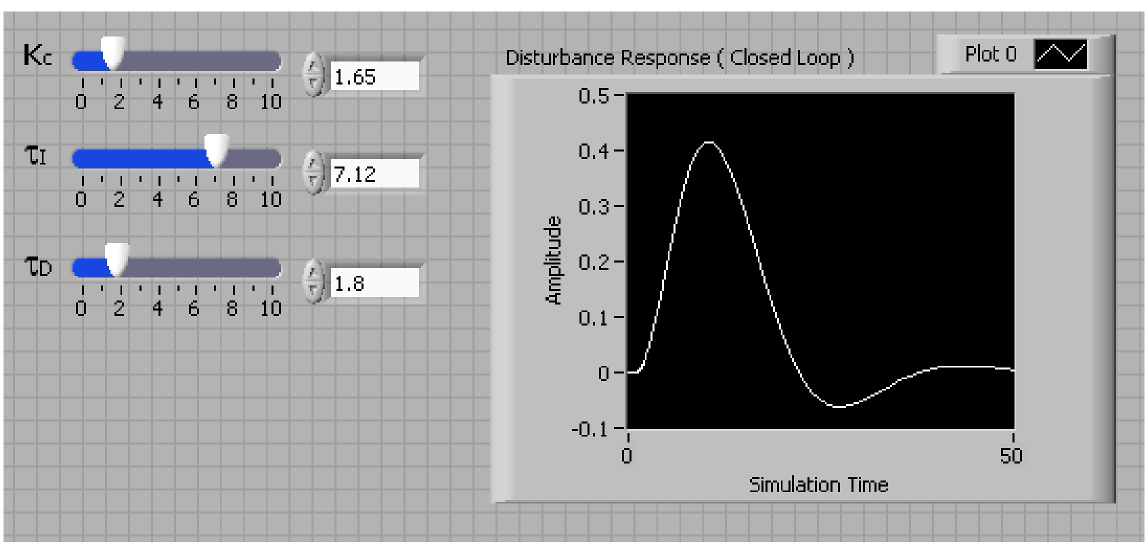
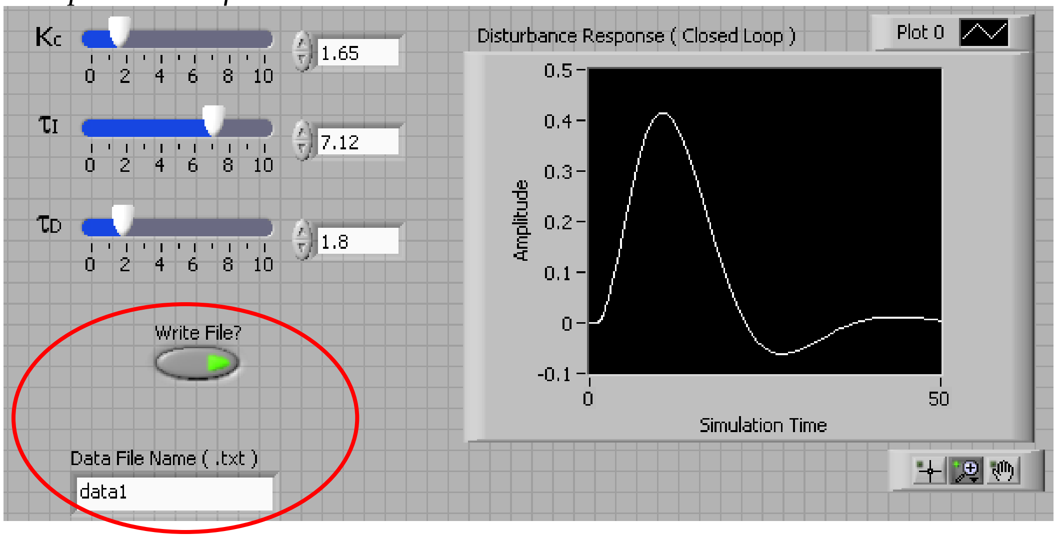
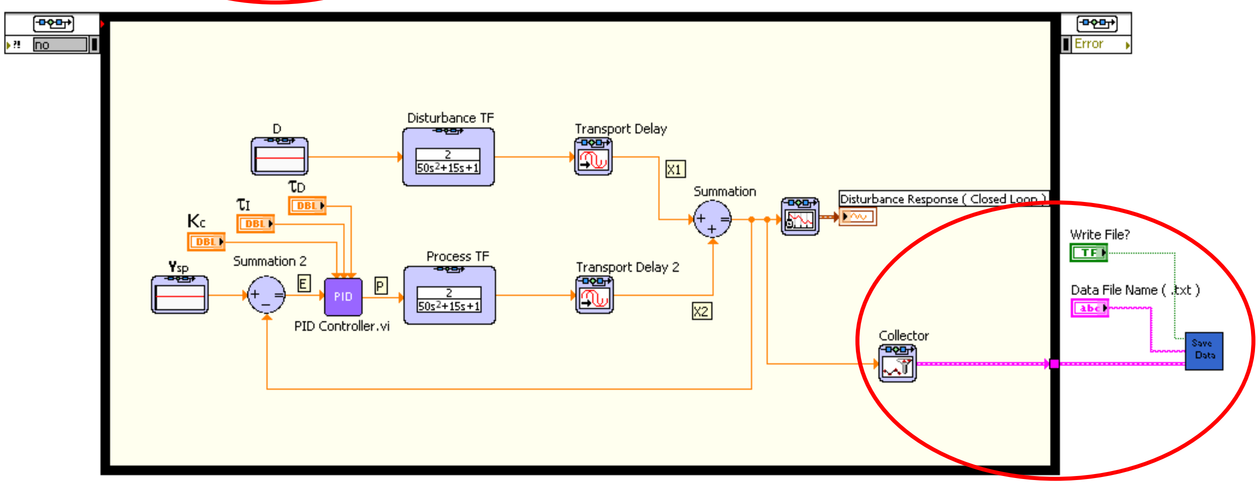


Figure 10. Unit disturbance response for ITAE (disturbance)

10. To save data for plotting, examine the changes that were made to the VI, saved as *examplesim with file.vi*





11. Different settings may compared to each other, using *Read and Plot Data.vi*

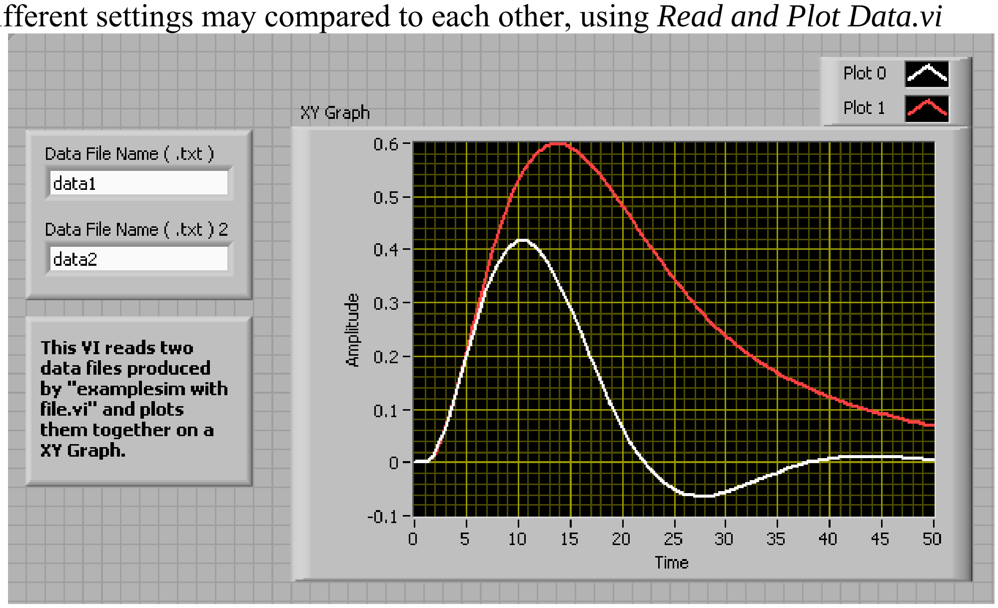
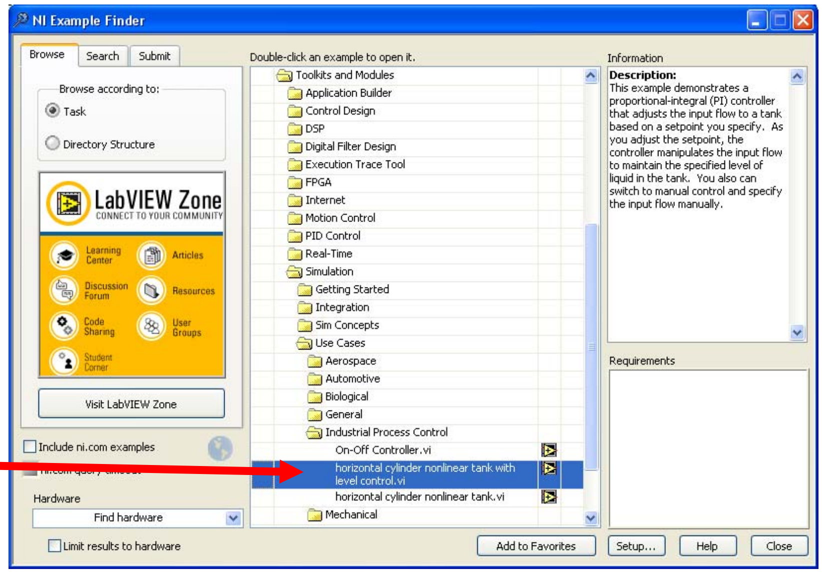


Figure 11. Comparison of PID controllers with ITAE settings from Table 12.3.

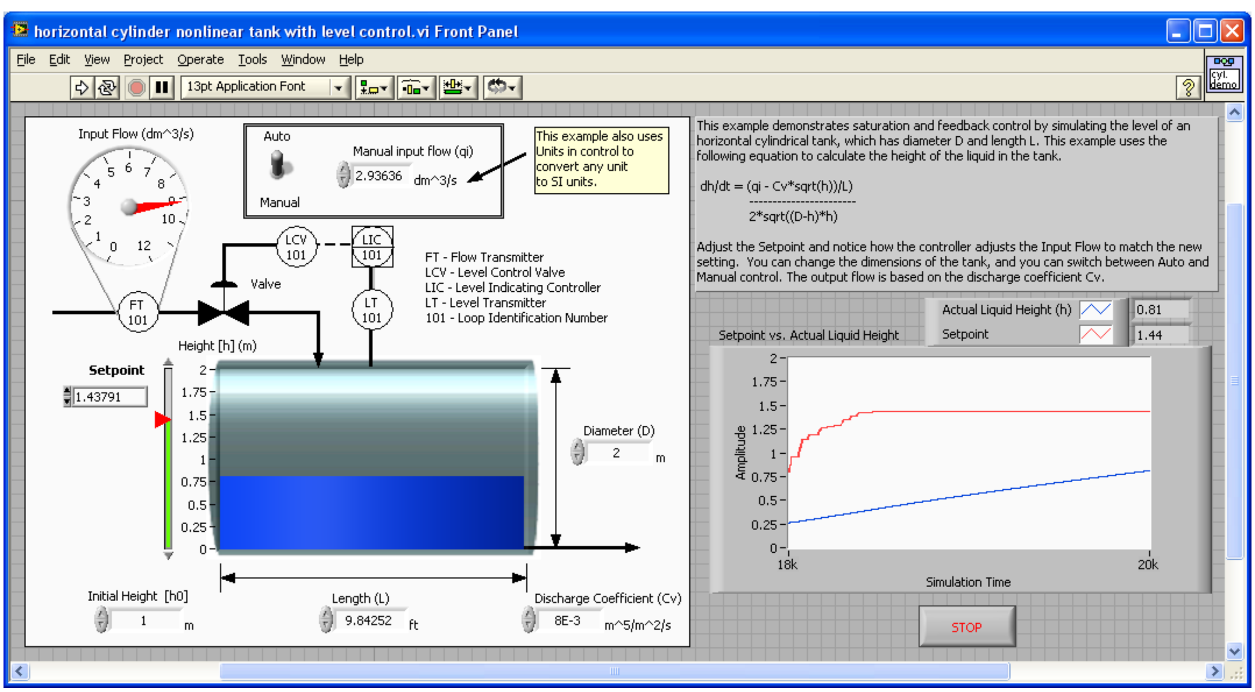
Additional Examples in LabVIEW :

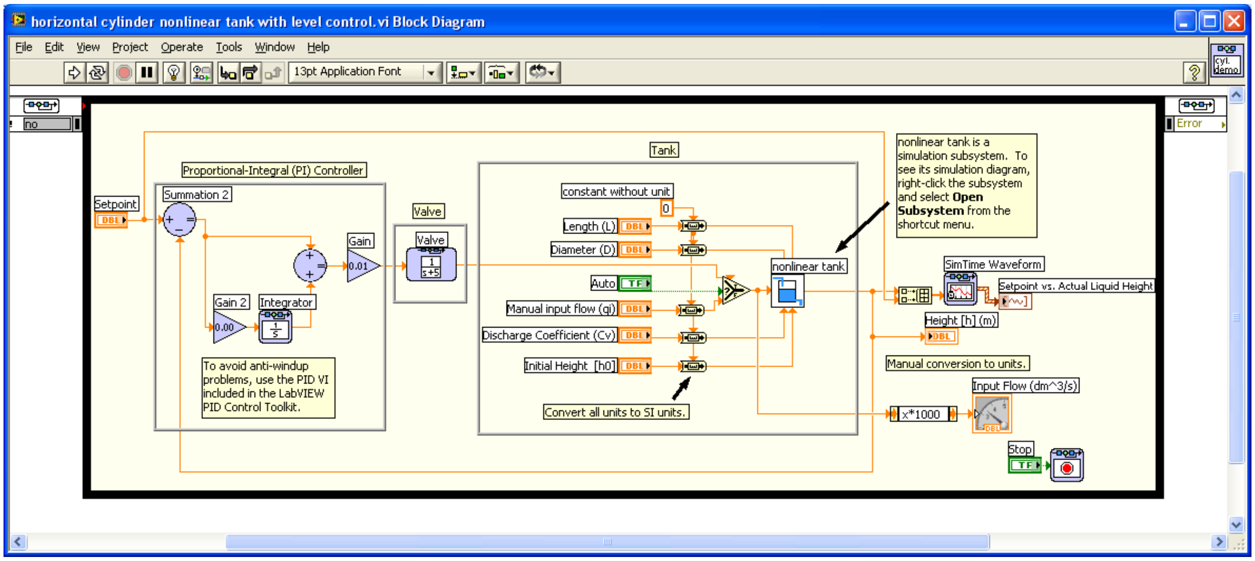
There are many helpful examples in LabVIEW that illustrate process control concepts. To find them, open Help->Find Examples and go to the following path : Toolkits and Modules-> Simulation->Use Cases-> Industrial Process Control.

Double click on the VI shown.



Description:
This example demonstrates a proportional-integral (PI) controller that adjusts the input flow to a tank based on a setpoint you specify. As you adjust the setpoint, the controller manipulates the input flow to maintain the specified level of liquid in the tank. You also can switch to manual control and specify the input flow manually.





Controls Tutorials for LabVIEW :

http://zone.ni.com/devzone/cda/tut/p/id/6368 based on Prof. Dawn Tilbury's tutorials from University of Michigan

http://cnx.org/content/col10401/latest/ NI LabVIEW training course on Rice's Connexions site

http://techteach.no/labview/ by Finn Haugen

Gains for Closed Loop exercise :

	Kc	τ_I	τ_D
ITAE(disturbance)	1.65	7.12	1.80
ITAE (set point)	1.07	16.25	1.55