

2 Lab 2: LabVIEW and Control System Building Blocks

2.1 Introduction

Controllers are built from mechanical or electrical building blocks. Most controllers are implemented in a program using sensors to understand the behavior of the system and actuators to command the system. In this lab, you will implement digital controllers on the computer using LabVIEW.

2.2 Pre-Lab

1. What is the Laplace transform of:

- An ideal integrator
- An ideal differentiator

2. Figure 1 shows a plot of $y(t)$. Sketch the derivative of $y(t)$.

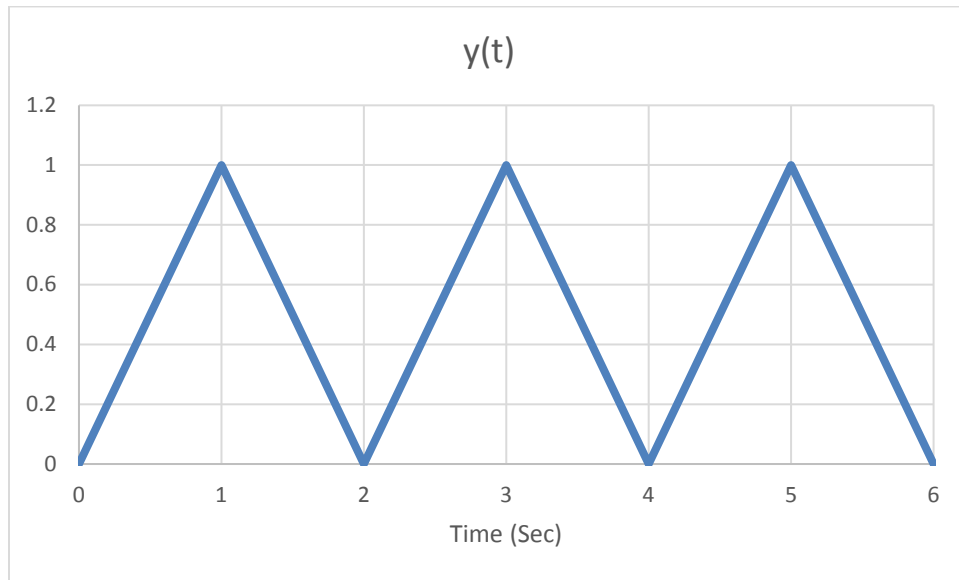


Figure 1 - Plot of $y(t)$

3. Figure 2 shows a plot of $x(t)$. Sketch the integral of $x(t)$.

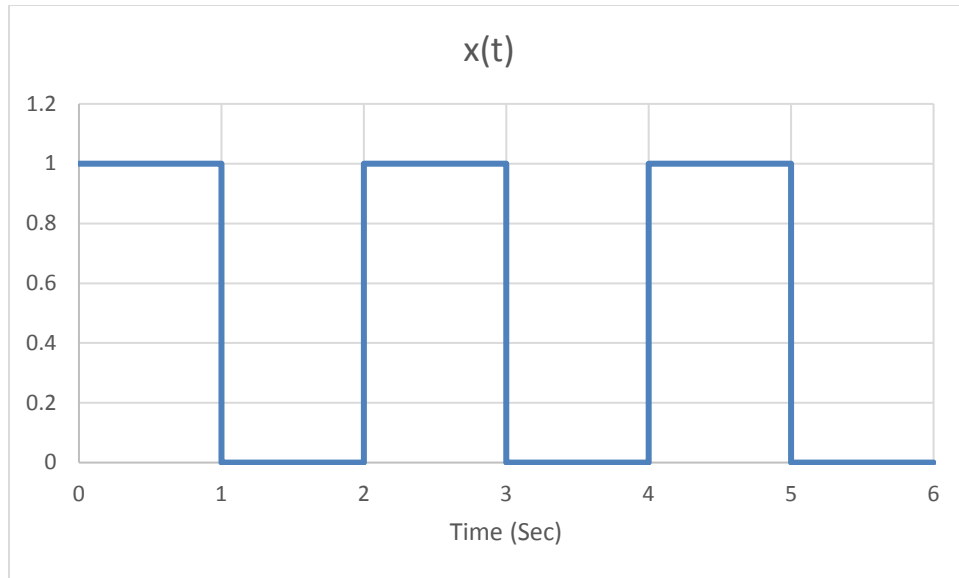


Figure 2 - Plot of x(t)

4. Review tutorials in the "Core LabVIEW Concepts" section of the LabVIEW online tutorials (<http://www.ni.com/getting-started/labview-basics/>)

2.3 Objectives

By the end of this laboratory experiment, you will be able to:

- Build and execute a program in LabVIEW
- Create and plot the control systems building blocks (summer, differentiator, integrator) in LabVIEW

2.4 References

Beyond the exercises in this lab, there are LabVIEW tutorials that you might find helpful: <http://www.ni.com/getting-started/labview-basics/>

Control and Simulation tutorials for LabVIEW: <http://www.ni.com/tutorial/11653/en/>

2.5 LabVIEW

In this section of the lab, you will be introduced to LabVIEW.

Open up LabVIEW 2016 (32-bit) from the desktop icon. If you see the *Set Up and Explore* screen, click Close. On the LabVIEW screen click **Create Project**, see Figure 3. Then select the **Blank VI** template and click Finish. Two windows will open, one that is gray with grid lines (Front Panel), and another blank, white window (Block Diagram). You will be using both of these windows to create programs and simulations in

LabVIEW. The *Front Panel* is the interface where the user sets parameters and views data with indicators and graphs. The *Block Diagram* includes the code and signal routing which controls elements of the Front Panel. Think of the two working together as a control panel - the Front Panel includes the switches, LED's, and gauges that you read on the outside, and the Block Diagram being the circuitry on the inside which connects everything together.

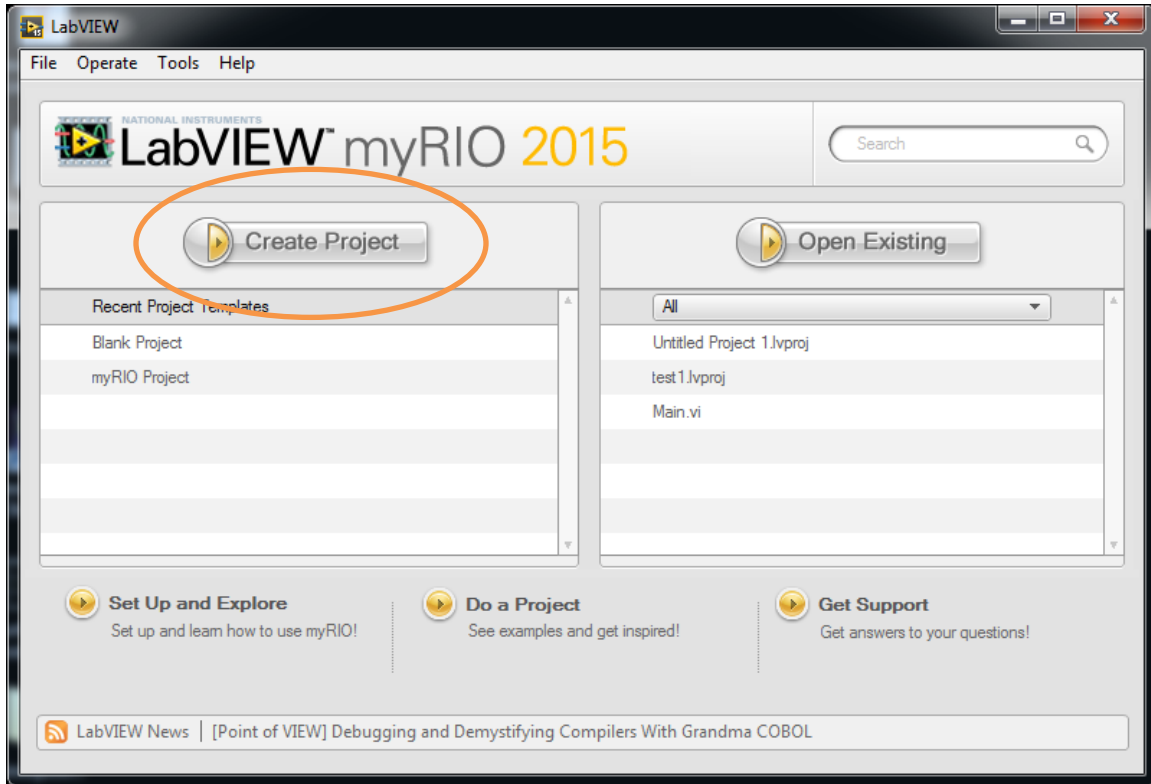


Figure 3 - LabVIEW Screen with Create Project button circled.

2.5.1 Summer

A **Summation** block in the Block Diagram computes the sum or difference of the inputs. You will be building the simulation shown in Figure 6 and Figure 7.

- Right click in the Block Diagram window to pull up a list of available functions and blocks. Then click the double arrow at the bottom expand the menu. Select **Control & Simulation** from the menu and click the pushpin icon on the menu that pops out. This will "pin" the menu to your desktop so you can access it more easily. Once it is pinned, it should look like the window in Figure 4.

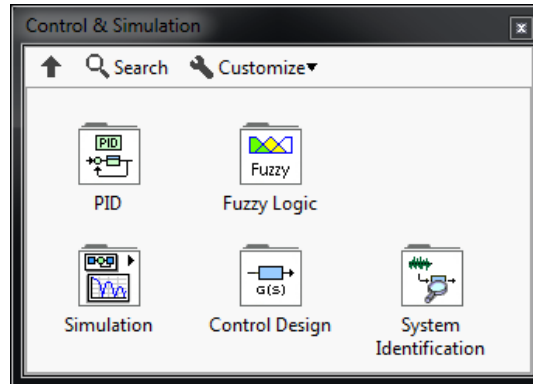


Figure 4 - Pinned Control & Simulation menu

- In the Control & Simulation window, select **Simulation** to open the Simulation window. In the Simulation window, select **Control & Simulation Loop**. In the Block Diagram window, click in the upper left corner and drag the cursor to create the Control and Simulation Loop. It should look like an empty version of the tan box with dark edges in Figure 7.
- In the Block Diagram window, right click on the small block connected to the upper left of the Control & Simulation Loop and select Properties. Change the following parameters
 - ODE Solver to Runge-Kutta 4.
 - Step Size (s) to 0.01
 - Final Time (s) to 5

Check to make sure the rest of the parameters match Figure 5. Then click OK.

- Now we are going to add the components inside the Control and Simulation Loop to match Figure 7. In the Simulation window, select **Graph Utilities**. Then click and drag a **SimTime Waveform** into the Block Diagram window and inside the Control and Simulation Loop. Once the block appear in your Block Diagram, you should see a corresponding "Waveform Chart" in the Front Panel. You may resize your Waveform Chart in the Front Panel using the double-ended arrow mouse cursor.
- From the Graph Utilities Window, click the Up Arrow button to get back to the Simulation Window. Click **Signal Arithmetic**. Click and drag a **Summation** block into the Control & Simulation Loop. Double click on the Summation block to change the inputs on the block. In the next window, click on the symbol of the port you want to change until it is the operation you want. It cycles through +, -, =, and no input.
- Now, go to the Front Panel. Right click in an empty area of the window. You will see a list of available functions appear, similar to the list you saw in the Block Diagram. Select **Numeric** and then **Horizontal Pointer Slide**. Repeat this step so you have two different sliders like Figure 6.

- Go back to the Block Diagram and you will have two new blocks corresponding to each of the sliders added in the previous step. Connect each of the sliders to the inputs of the Summation block, and connect the output of the Summation block to the Simtime Waveform block. Your block diagram should match Figure 7. To draw wires between blocks, hover over the edge of the output of the first block until your cursor changes to a spool of wire, then click on the edge of first block and the port on the input of the block you want to connect it to. Once everything is routed correctly, the Run arrow will appear in the left side of the tool bar. If this arrow appears broken and says List Errors, there is a block in your diagram that is not connected.
- To run the simulation, click the first button (arrow) on the tool bar. The result of the simulation will display in the Waveform Chart in the Front Panel window.
- Move the two sliders to new values and run the simulation again. Record your data in the data table. Repeat four more times.

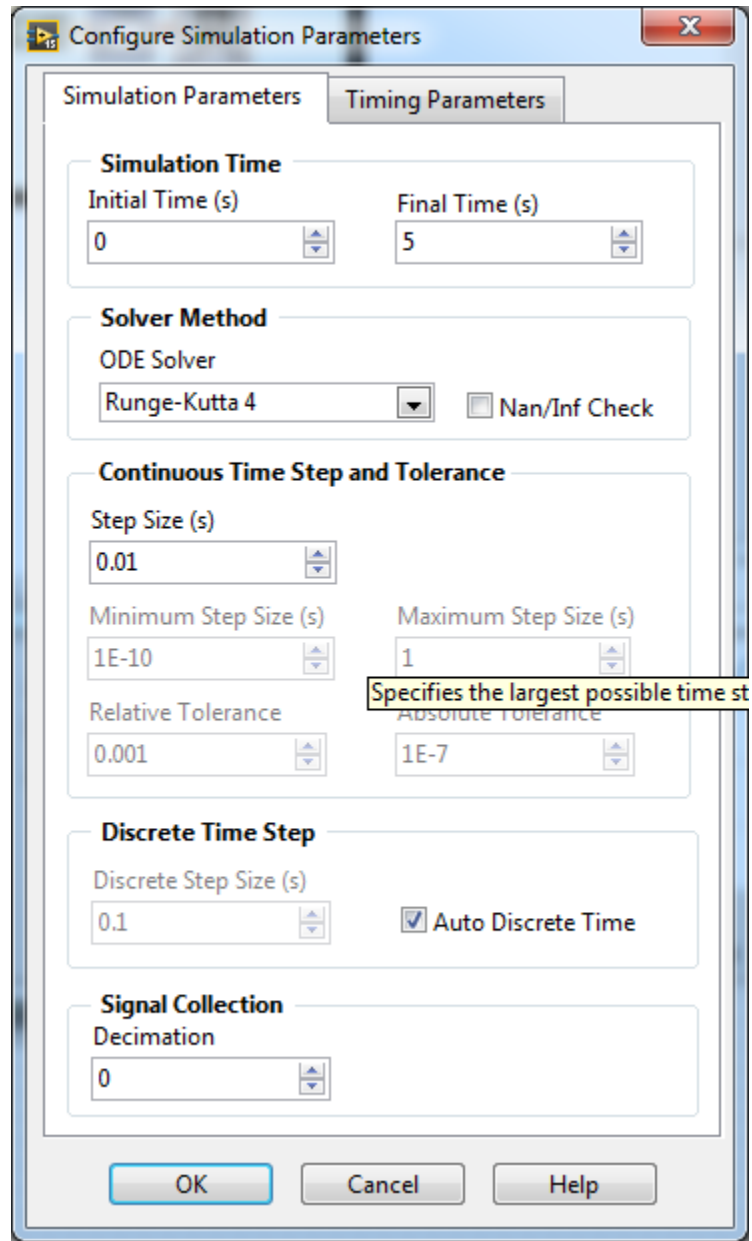


Figure 5 - Parameters window for Control & Simulation loop

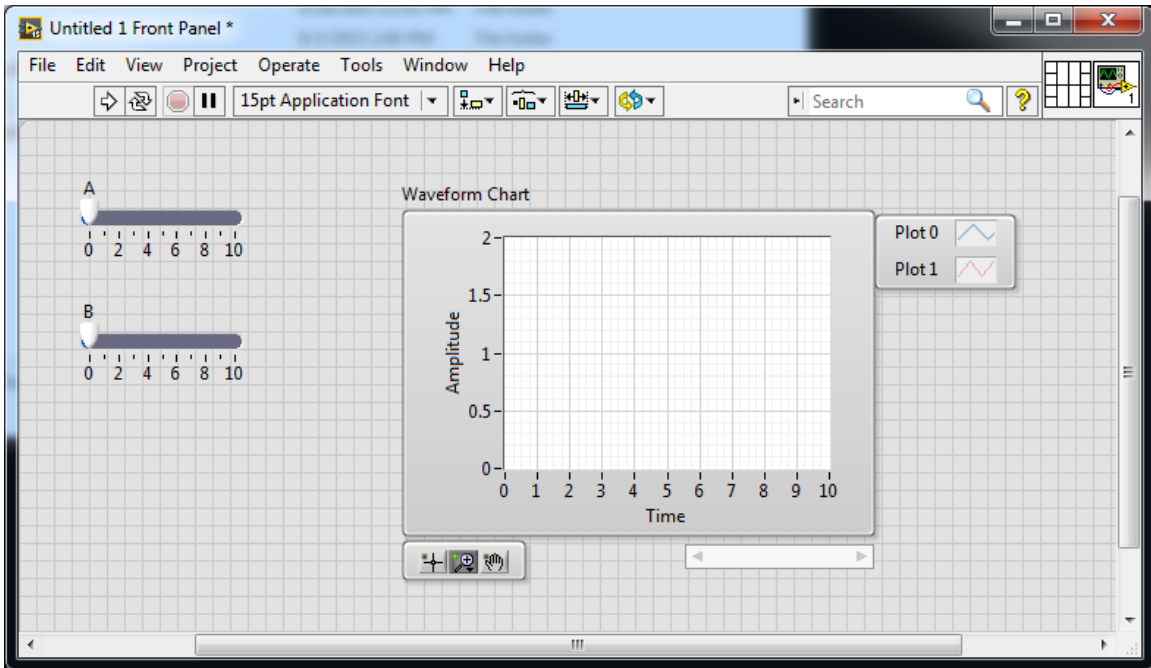


Figure 6 - Front Panel window with the complete summation simulation

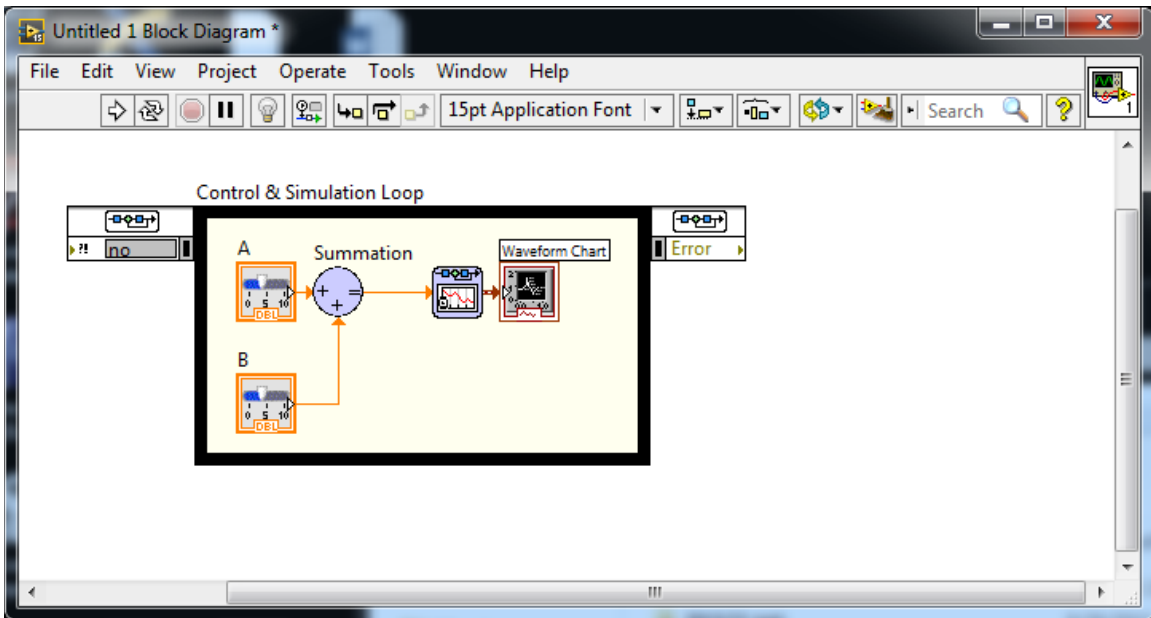


Figure 7 - Block Diagram Window with the complete summation simulation

2.5.2 Differentiator

Next, you will build a simulation of a differentiator. The complete simulation will match the Front Panel and Block Diagram in Figure 8 and Figure 9.

- Starting in the Block Diagram from the Summation simulation, remove the slider inputs and the summation block.
- The input signal to the differentiator will be a Triangle Wave. LabVIEW does not have a Triangle wave generator. However, a triangle wave can be created by multiplying a sawtooth and square wave. Find the "Signal Generator" block in the Simulation menu of Control & Simulation. Drag two Signal Generator blocks into the Control & Simulation loop.
- Double click on one Signal Generator to configure it with the following parameters:
 - Signal type - **Sawtooth**
 - Frequency - **0.5 Hz**
 - Amplitude - **1**
- Configure the parameters of the other Signal Generator to the following:
 - Signal type - **Square**
 - Frequency - **0.5 Hz**
 - Amplitude - **1**
 - Phase - **180 degrees**
- Drag a **Multiplication** block in to the Control & Simulation loop from the Signal Arithmetic menu.
- Right click in the Block Diagram window to pull up the menu. Select **Array** then drag a **Build Array** in to the Control & Simulation loop. To increase the size of the Build Array, hover over the center of the bottom edge until you get a double-sided arrow pointing up and down. Extend the array block until it has two rows like Figure 9.
- From the Simulation menu in Control & Simulation, select **Continuous Linear Systems**. Drag the **Derivative** block in to the Control & Simulation loop.
- Connect the blocks as shown in Figure 9.
- Run the simulation and record the results in your data table. The first triangle wave input should look like the one in Figure 8.
- If you would like to zoom in on the graph to get a better reading, right click on the Waveform Chart in the Front Panel window and go to **Visible Items**. Select **Graph Palette**.

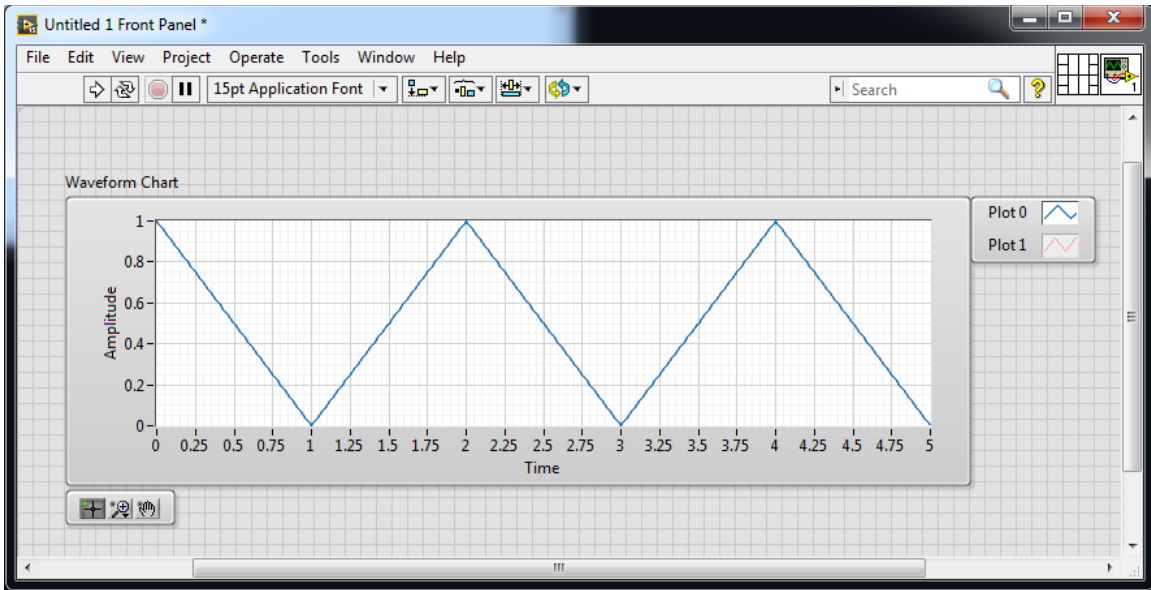


Figure 8 - Front Panel window with the complete differentiator simulation

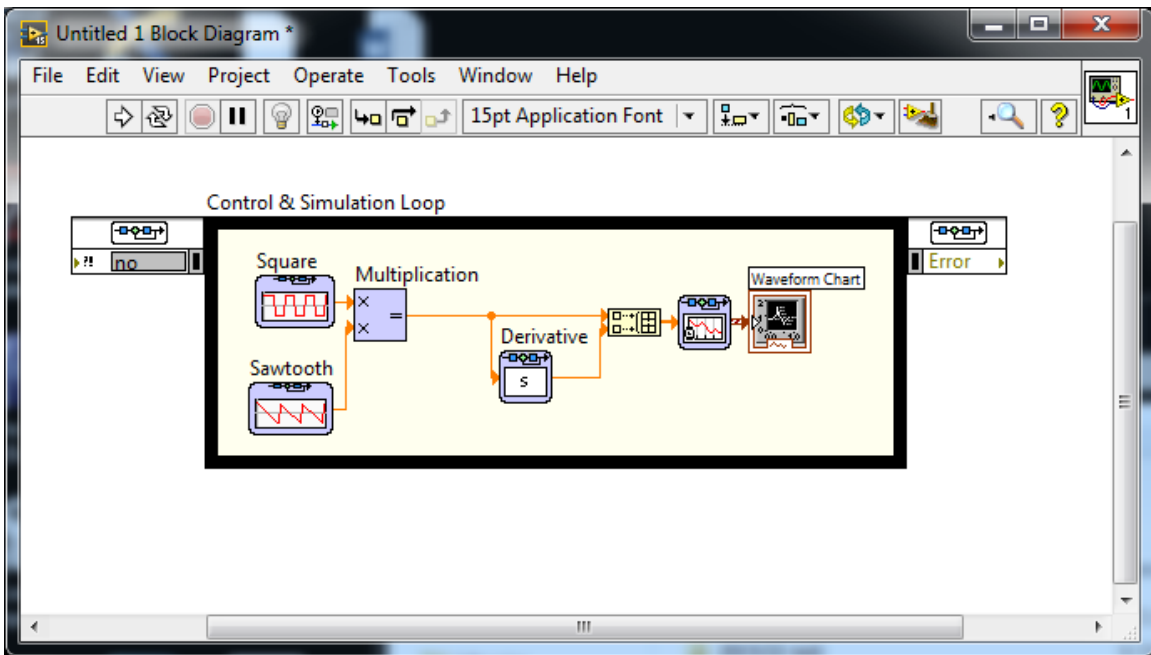


Figure 9 - Block Diagram window with the complete differentiator simulation

- Change the frequency of the square wave and sawtooth wave to match the values in the table below and record the output in your data table

Frequency (Hz) Square & sawtooth waves	Amplitude Sawtooth wave
1	1
2	2
3	3
4	4

- Since we built a differentiator, how would you expect the output amplitude to relate to the input slope? Do you notice any discrepancy between the output amplitude and the input slope for a certain input?
- Remove the sawtooth wave and the multiplication block. Press **Ctrl+B** to clear any broken wires. Then, connect the square wave directly to the Derivative block, and to the open input of the Build Array block. What happens to the output? Can you explain this result?

2.5.3 Integrator

Finally, you will build a simulation of an integrator. The final simulation will look like Figure 10.

- Start with the differentiator block diagram. In your differentiator simulation block diagram, remove the Derivative block and replace it with an Integrator block from the Continuous Linear Systems menu.
- Change the frequency of the square wave to 1 Hz.
- Run the simulation and record your data.
- Repeat the exercise with 4 different amplitudes and record your data.
- Since we built an integrator, how would you expect the output slope to relate to the input amplitude? Do you notice any discrepancy between the output slope and the input amplitude for a certain input?

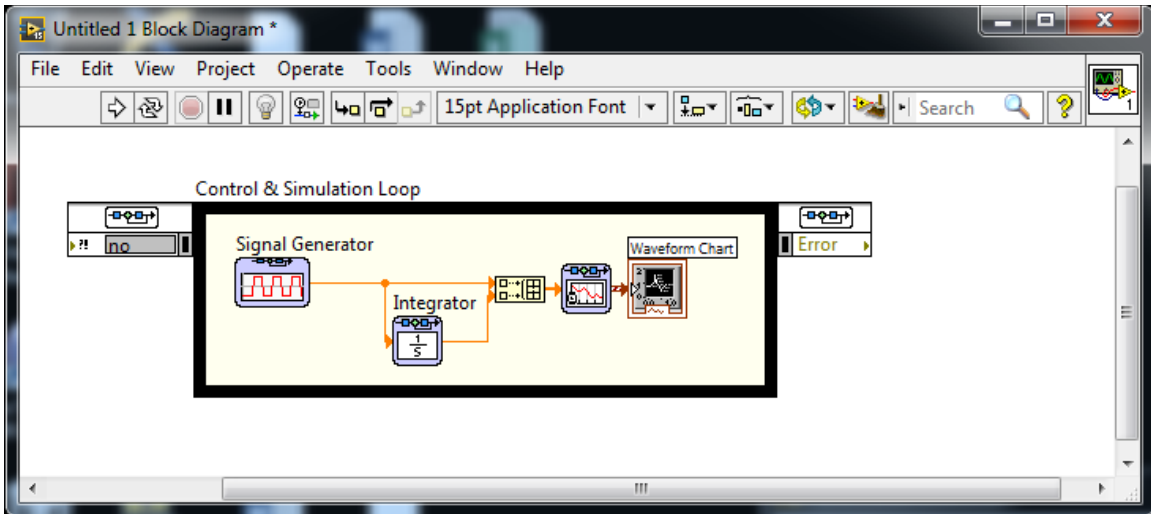


Figure 10 - Block Diagram window with the complete integration simulation

2.6 Post Lab

Include the answers to the following questions in your lab report.

1. Do you think we simulated an ideal differentiator?
2. What happened when the input to the differentiator was a square wave? Explain why you obtained this result.
3. How are the transfer functions of an integrator and differentiator related to one another?

2.7 Data sheet

Summing Junction		
Input #1	Input #2	Output

Integrator (Square Wave Input)	
Input Amplitude	Output Slope

Differentiator (Triangular Wave Input)			
Measured Input Frequency	Measured Input Peak-to-Peak	Input Slope	Output Amplitude