Lab 1: Intro to Simulink and Frequency Analysis

Objectives:

1) To learn how to build a basic model in Simulink.
2) To simulate signals and view their frequency domain representation.
3) To synthesize signals from their Fourier series coefficients.

Introduction:

A signal is ordinarily described as a function of time. This is what we visualize when we view the waveform on an oscilloscope. However, in communication systems it is also important that we know the frequency content of a signal. The spectral density of a signal characterizes the distribution of the signal's energy or power in the frequency domain. This concept is important when considering filtering in communication systems.

The mathematical tool which relates the frequency domain description of the signal to its time domain description is called the Fourier Transform. The Fourier Transform of a signal specifies the amplitudes and phases of the frequency content of the signal given its behavior in the time domain. The inverse Fourier transform can then recover the original time domain signal given its frequency domain description.

Simulink is a tool that allows us to visually create signal processing systems by connecting individual "blocks" that describe the actions of the system. After specifying the blocks that make the system do what we want, we can then pass any signal we want through it and view the output on a virtual "scope" in either the time or the frequency domain.

Preliminary:

Given a periodic signal $f(t)$ with period T. Its complex Fourier series coefficients are given by: $c_n = \frac{1}{T}\int_0^T f(t)e^{-j2\pi n\frac{t}{T}dt}$. Using Table 2.1 in the textbook,

1) Find the Fourier series coefficients of a square wave of odd symmetry of amplitude *A* and frequency *f* Hz (assume 50% duty cycle and zero DC).
2) Find the Fourier series coefficients of a triangular wave of odd symmetry of amplitude *A* and frequency *f* Hz (assume zero DC).
3) Sketch the amplitude and phase of the Fourier series coefficients obtained in 2 & 3.

Post-Lab Questions:

Q1. In the demo model built in this lab, we used a mux block to visualize two signals on a single scope. Is it possible to see two signals on the scope without using a mux block? If yes, explain how.

Q2. Does changing the max step size in the configuration parameters have any effect on the way the model runs?

Q3. In part A, does changing the simulation time have any effect on the model output?

Q4. In part B, what is the relation of the sample time to the signal frequency to get a reasonably good representation of the frequency domain components?

Q5. In part B, what effect does changing the FFT buffer size have on the output spectrum?

Q6. In part C, change the phase and magnitude of each harmonic. Which is more important: the phase of a harmonic or the magnitude?

Procedure:

Part A – Getting Familiar with Simulink

Launch Matlab and type '*simulink*' at the MATLAB command prompt to open Simulink. The Simulink library browser will open. It consists of various blocksets specific to different applications such as communications, signal processing, control systems, power systems etc. It is advisable to explore the various block libraries on your own to get a good feel for the capabilities of Simulink.

In order to familiarize yourself with Simulink, you will first build a simple system by following the steps given in this document. To open a new model, go to the *File* menu and select *New -> Model*. A blank model will open. This will serve as your canvas to build your system model.

In the Simulink library browser, go to the *Sources* option and select a *'Sine Wave'* source block. In order to include this block in your model, click and drag the block onto your model window. Similarly search for a *'Gain'* block (to amplify the signal) and a *'Scope'* block (to visualize the signal) in the various Simulink libraries. You should find the gain block under the *Commonly Used Blocks* or the *Math Operations* libraries. You should find the scope block in the *Commonly Used Blocks* or *Sinks* libraries. Include both in your model window.

For this model, we need two gain blocks. In order to duplicate a block, simply right click on the block and drag and place this duplicate block in your model window too. We would also like to have a *'Switch'* block to be able to select from multiple inputs and a multiplexer block to view multiple signals on the same scope. The multiplexer is found in the libraries as the *'Mux'* block. Search for these blocks in the basic libraries and include them in your model.

We also require a *'Clock'* block to use as a decision maker for switching between two signals using the switch. This block simply outputs the current simulation time. Look for it in the *Sources* library and include it in your model window.
(*Note:* This is not the clock used for synchronization of digital logic systems.)

Your model window should now have all of the blocks shown in Figure 1.

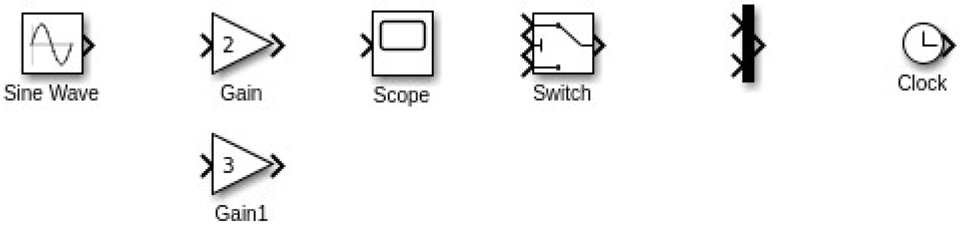


Figure 1 - Model window after creating blocks

In order to begin connecting the blocks, click on the sine wave block, then hold down the CTRL key and click on the first gain block. A connecting line will form between the two blocks. An alternate way to do this is to place your cursor near the output port of a block until the cursor changes into a cross-hair, then click and drag a line to the input port of the block to be connected. The same sine wave input is to be connected to the second gain block too. In order to draw a second branch, right click on the original branch and drag a line to the input port of the second gain block.

Connect the chosen blocks in the following manner.

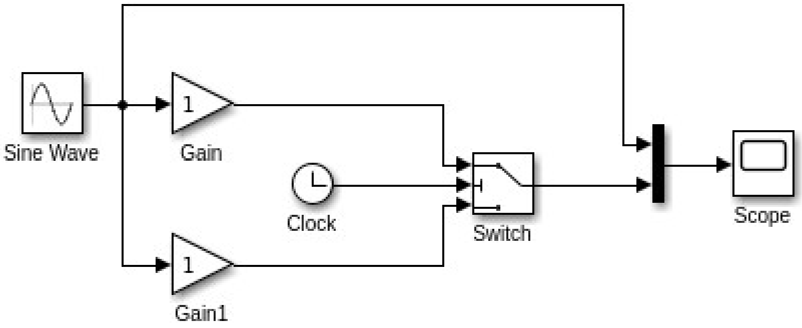
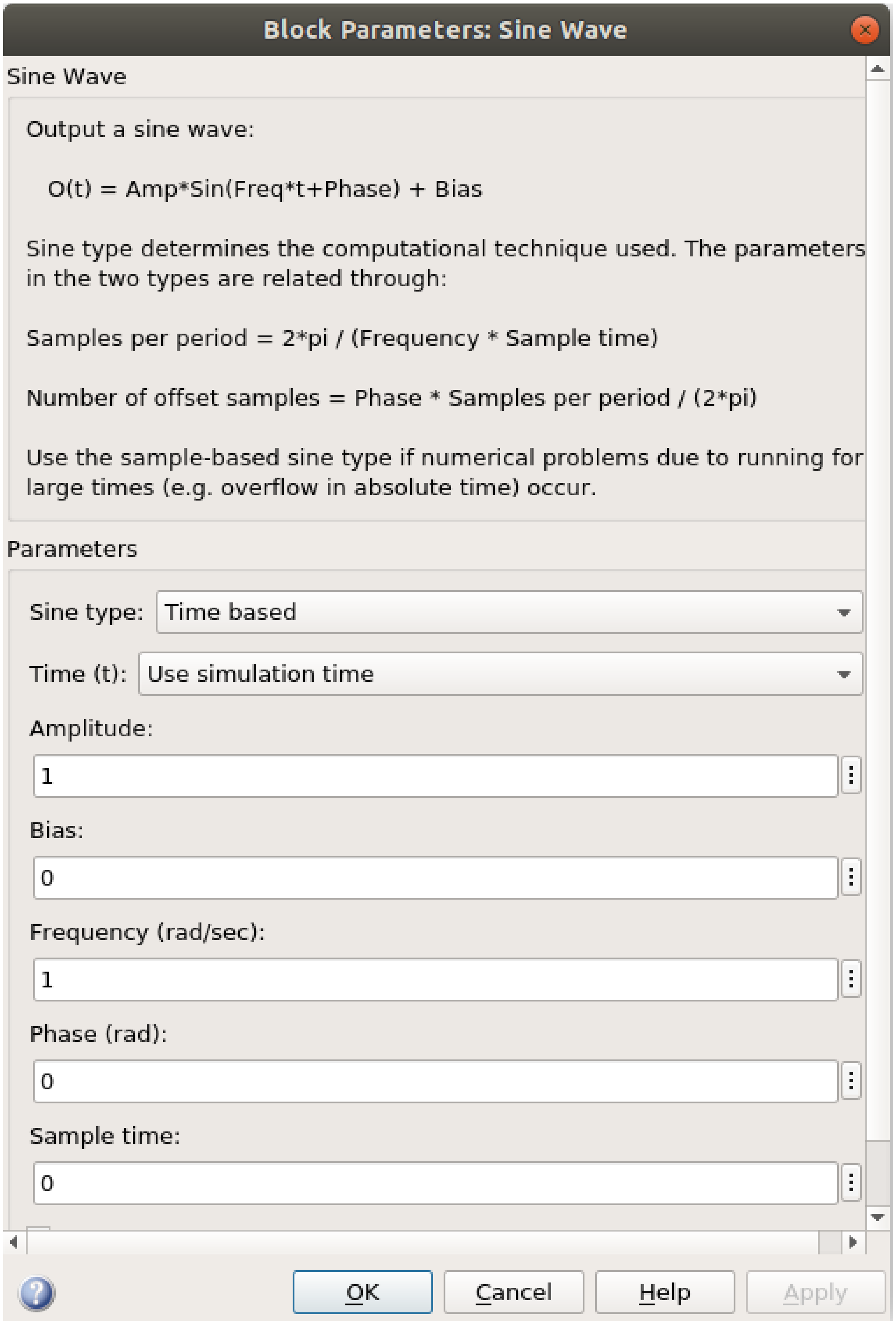


Figure 2 - Model window after connecting blocks

Next, we need to set the parameters for the individual blocks. Double clicking on the blocks opens the block parameters pane. First, double click on the sine wave source block. Set the block parameters as given in Figure 3, and click OK to apply the settings.



Block Parameters: Sine Wave

Sine Wave

Output a sine wave:

O(t) = Amp*Sin(Freq*t+Phase) + Bias

Sine type determines the computational technique used. The parameters in the two types are related through:

Samples per period = 2*pi / (Frequency * Sample time)

Number of offset samples = Phase * Samples per period / (2*pi)

Use the sample-based sine type if numerical problems due to running for large times (e.g. overflow in absolute time) occur.

Parameters

Sine type: Time based

Time (t): Use simulation time

Amplitude: 1

Bias: 0

Frequency (rad/sec): 1

Phase (rad): 0

Sample time: 0

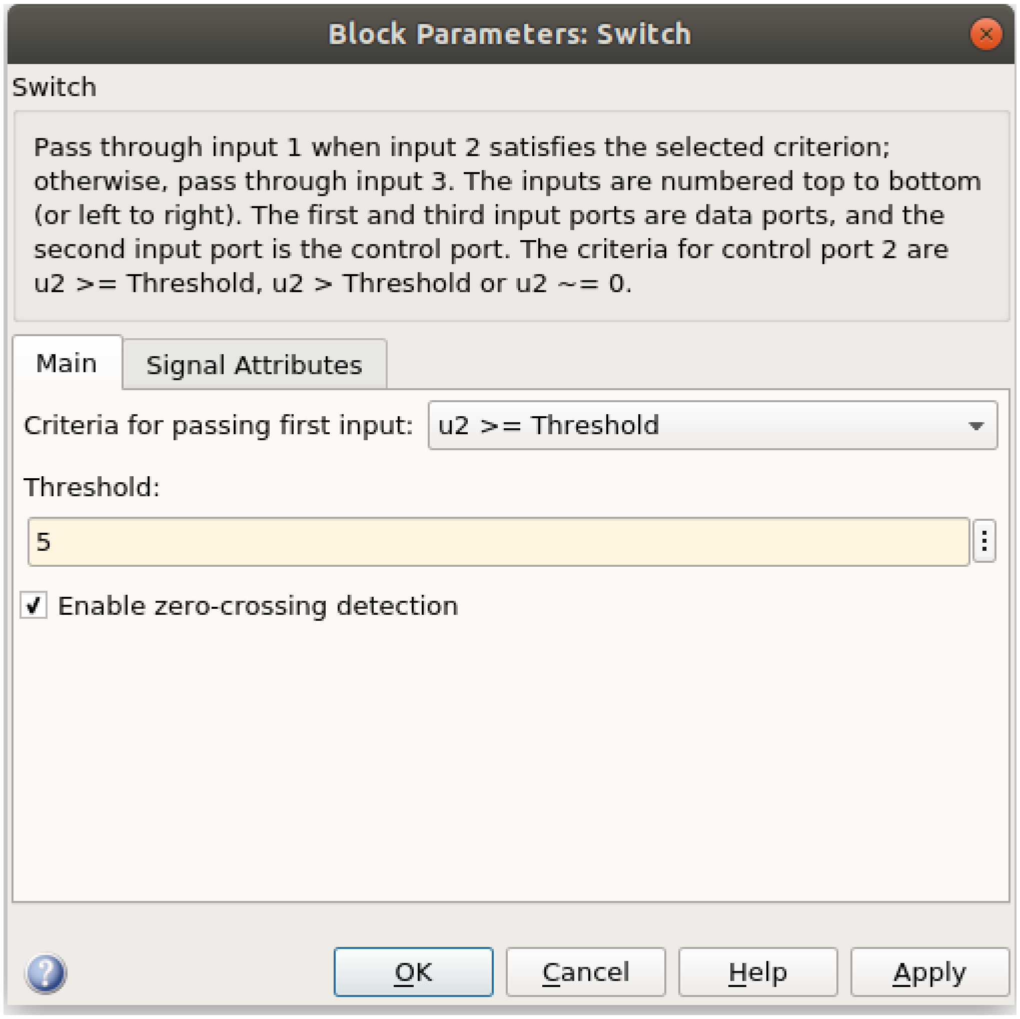
Figure 3 - Sine wave source parameters

Open the two gain block settings and set them up for gains of 0.5 and 3 respectively. The switch should have the parameters shown in Figure 4 in order to change from the second gain block output to the first after a simulation time of 5 seconds. The mux block will have two inputs by default. The clock and scope block settings need not be altered.

To add a title to the model, double-click at the location you want to type, then type "Building a Basic Model in Simulink". You may use the *Format* menu to change the font and font size. You may also right-click on the title box and select *'Show Drop Shadow'* to emphasize the title. The completed model is seen in Figure 5. Include a screenshot of the model in your lab report.

Set the simulation parameters by selecting *Model Configuration Parameters* from the *Modeling* menu. Under the Solver tab, set the solver to *'discrete'*, the stop time to 10.0, and the max step size to 0.03. Then click OK. Now the model is ready for simulation.

Go to Simulation > Start or press ▶to begin simulation. In order to visualize the results, double click on the scope block. The scope output should be as shown in Figure 6. Include a screenshot of this result in your lab report.



Block Parameters: Switch

Switch

Pass through input 1 when input 2 satisfies the selected criterion; otherwise, pass through input 3. The inputs are numbered top to bottom (or left to right). The first and third input ports are data ports, and the second input port is the control port. The criteria for control port 2 are u2 >= Threshold, u2 > Threshold or u2 ~= 0.

Criteria for passing first input: u2 >= Threshold

Threshold:

5

☑ Enable zero-crossing detection

Figure 4 - Switch block parameters

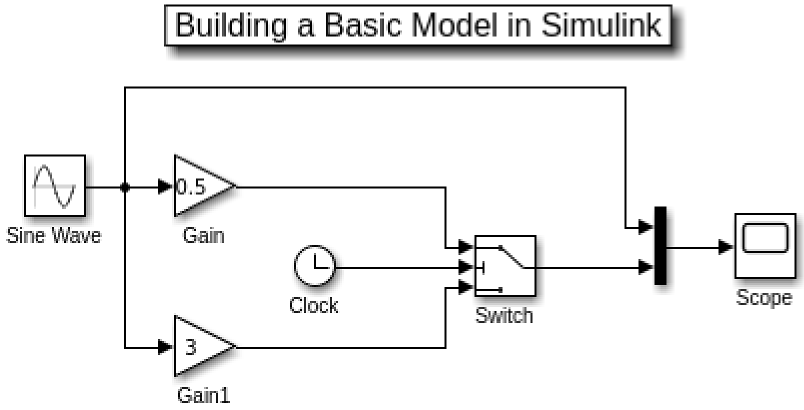


Figure 5 - Final model

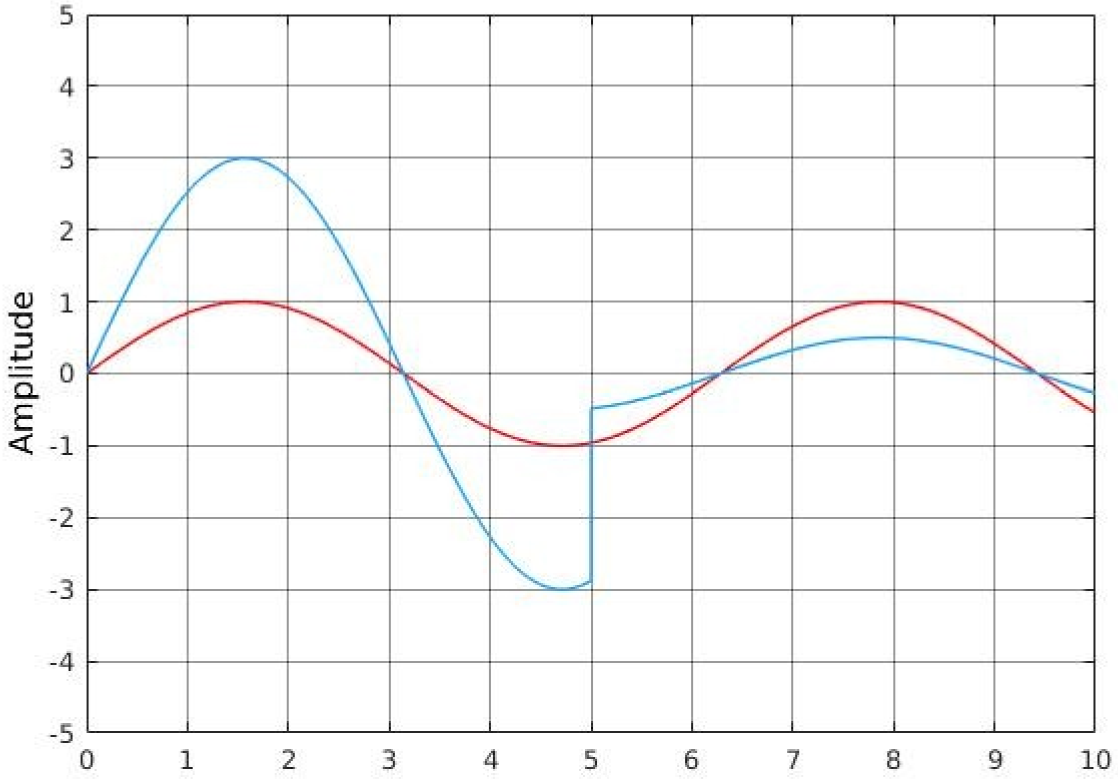


Figure 6 - Scope display

Change the frequency of the sine wave generator to 5*2*pi rad/sec (i.e. 5 Hz) and change the phase to pi/2 rad to turn it into a cosine wave. Simulate the model and observe the scope, then include a screenshot of the new scope output in your report. Similarly, observe the scope for the following variations of the original model and submit the outputs in your report:

1) Change the switch block to a *Product* block (remove the clock).
2) Change the clock to a cosine wave of frequency 4 rad/sec, amplitude 10, and bias 5.
3) Change two or more parameters of your choice in the original model.
4) Replace one of the gain blocks with any block of your choice.

Part B - Frequency Domain Analysis

In order to view the properties of a signal in the frequency domain, some different steps must be taken. Ordinarily, a signal source will generate a single value at each point in time, which then propagates through the entire system as Simulink processes the simulation. However, a proper analysis in the frequency domain requires several time-domain points to be stored up and processed at the same time, generally with an FFT block.

A handy block used to compute and display the frequency components of a signal is the *Spectrum Analyzer*, found in the Signal Processing Toolbox libraries. To start using it, create a new model and add a title of “Frequency Domain Analysis of a Signal”. Then find the spectrum analyzer block in the library and add it to your new model.

Upon opening the block settings, a number of options will need to be set. To display a raw magnitude spectrum (as opposed to the PSD), change the units to “dBW”, and set the display options to one-sided. In order to use the standard signal sources from the normal toolbox, it is necessary to check the *Buffer Input* box. When this is selected, the scope will take time-domain samples from a source and accumulate them in a buffer until it has enough to calculate the required FFT. Go to *Spectrum Settings* > *Main options,* change RBW (Hz) to Window length and set the buffer size to 512. In *window options,* set buffer overlap to 0, and in *Trace options*, set spectral averages to 1. Do not change any other settings from their default values.

A convenient signal source from the continuous-domain toolbox is the *Signal Generator* block. This block can generate a number of different periodic signals given only an amplitude and a frequency. Find this block and add it to your project, then open the settings. Leave the signal type at ‘sine’ and amplitude at 1, but change the frequency to 5 Hz.

It turns out that these two blocks cannot be directly connected together due to the way Simulink processes signals. The signal generator produces *continuous-time signals*, which are numerically computed using differential equations. However, the FFT scope requires a *discrete-time signal* to function, which is handled in an entirely different way by Simulink's model solvers. In order to convert the continuous signal into a discrete one, we require the *Zero-Order Hold* block. Find this block in the library, then place it between the signal generator and the spectrum scope and connect them together in a line. Your final model should resemble Figure 7.

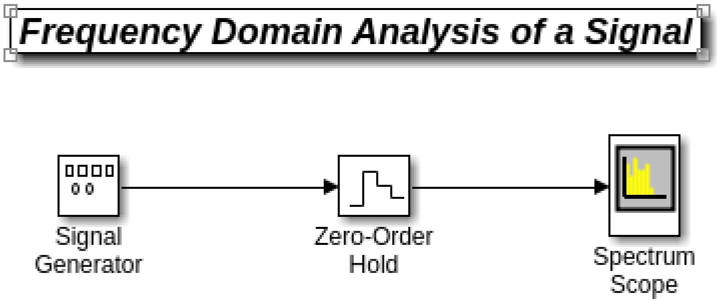


Figure 7 – Final model

There are still a few more things to do. By opening the zero-order hold settings, we can choose the sampling period for the discrete signal. Set it to 0.01 (f_s = 100 Hz). Finally, open the simulation configuration settings and use the same settings as the previous example, but make the max step size 0.001. We are now ready to run the simulation.

Upon finishing the simulation, the spectrum scope should pop up with the results of the FFT. For the sine wave, there should be a single peak at 5 Hz and nothing else. It may also be a good idea to connect a normal scope to the zero-order hold output to view the signal in the time-domain.
Now we are ready to look at a few more types of signals. Change the signal type in the generator to '*square*' and rerun the simulation without changing any other settings. There should be harmonics present this time; if they are not visible, go into the scope settings and lower the minimum Y-value. Compare the magnitudes of these harmonics to the results from the preliminary and note your observations in your lab report. Also, include a screenshot of your magnitude spectrum.

Next, change the generated signal to a sawtooth wave and repeat the above process, noting your observations on the harmonics and taking a screenshot of the spectrum. Finally, change the generator to output a random signal and record your observations on the spectrum produced.

Part C - Signal Synthesis from Fourier Series Coefficients

Given the Fourier coefficients calculated in the preliminary exercise, it is possible to reconstruct the square wave and sawtooth wave signals using only sine wave sources. To do this, open a new model, then create a sine wave source for the fundamental frequency of the square wave and one for each harmonic (use as many harmonics as you feel are necessary to reconstruct the signal accurately). Set the amplitudes, frequencies, and phases of each according to the Fourier coefficients you calculated previously.

To add all these sine waves together, the *Sum* block is required. Find this block in the libraries and add it to your project. Open the settings, and under the list of signs, add as many plus (or minus) signs as there are signals to add. This will expand the sum block to accept several inputs. Attach a scope to the input and observe how close the sum of sinusoids comes to representing the shape of the original signal, then include a screenshot of the scope output in your lab report. Repeat the above steps to recreate a sawtooth wave and report your observations.