

11-2018

ECPE 121 - Simulink Labs

Cherian Mathews

University of the Pacific, cmathews@pacific.edu

Follow this and additional works at: <https://scholarlycommons.pacific.edu/open-textbooks>

Part of the [Computer Sciences Commons](#), and the [Engineering Commons](#)

Recommended Citation

Mathews, Cherian, "ECPE 121 - Simulink Labs" (2018). *Pacific Open Texts*. 7.
<https://scholarlycommons.pacific.edu/open-textbooks/7>

This Book is brought to you for free and open access by the Open Educational Resources at Pacific at Scholarly Commons. It has been accepted for inclusion in Pacific Open Texts by an authorized administrator of Scholarly Commons. For more information, please contact mgibney@pacific.edu.

Simulink lab 1: Introduction to Simulink for Digital Signal Processing

Simulink allows users to simulate complex systems using block-diagrams. This provides an approach to system analysis, simulation, and design that uses block-diagrams in place of coding. In ECPE 121 we will use Simulink alongside Matlab to facilitate understanding of digital signal processing (DSP) concepts, and to analyze and design systems.

Our introduction to Simulink will begin by simulating a sinusoid $x(t) = \sin(2\pi Ft)$ of frequency $F = 1\text{Hz}$ ($\Omega = 2\pi$ rad/sec) and viewing it on an oscilloscope. We will then digitize the sinusoid at a sampling rate $F_s = 10$ samples/sec, corresponding to a sampling period $T = \frac{1}{10} = 0.1$ sec.

The resulting discrete-time sinusoid is $x(n) = \sin(2\pi FnT) = \sin(2\pi \times 1 \times n \times 0.1) = \sin(0.2\pi n)$. You can easily verify that the period of this sinusoid is $N = 10$ samples. The values of $x(n)$ over one period are as tabulated below.

n	0	1	2	3	4	5	6	7	8	9
$x(n)$	0	0.59	0.95	0.95	0.59	0	-0.59	-0.95	-0.95	-0.59

Table 1: Sample values of $x(n)$

Fig. 1 depicts the Simulink block diagram used to create and display $x(t)$ and $x(n)$. The Sine Wave block allows for generation of a sinusoidal signal of specified amplitude and frequency. The Zero-Order Hold block performs sampling of the continuous time sinusoid with a specified sampling period (in our case, $T = 0.1\text{s}$). The Scope block allows signals to be viewed.

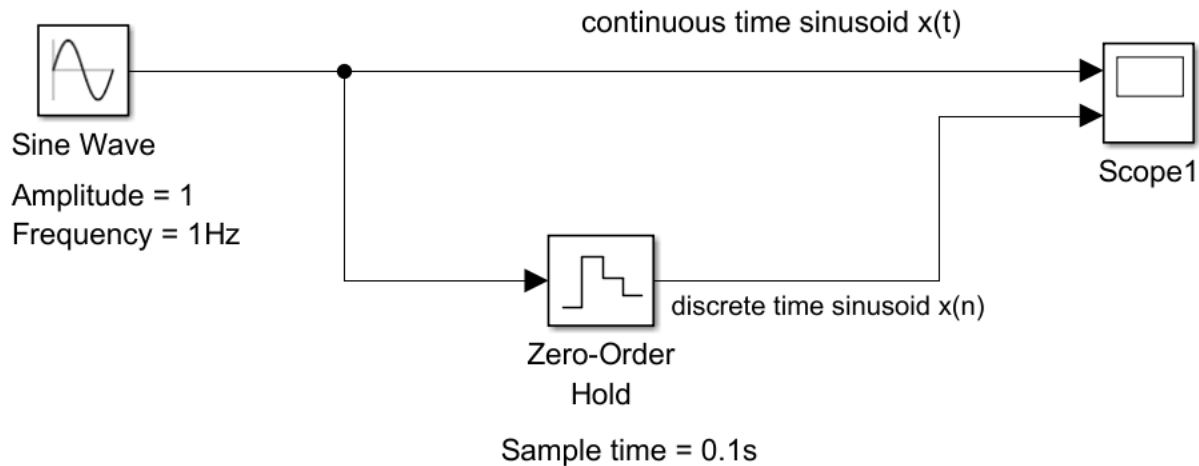


Fig. 1: Block diagram used to create and display $x(t)$ and $x(n)$

Fig. 2 shows the output of the Scope block. The yellow trace shows $x(t)$, which has a period of 1 sec as expected. The blue trace shows $x(n)$: as expected, $x(n)$ exhibits transitions only at

multiples of the sampling period $T = 0.1$ s. The sample values of $x(n)$ match the values of Table 1.

Watch the video posted under this assignment on Canvas to view how Simulink was used to perform the above simulation. Follow along, create the block diagram in Simulink, and run the simulation. Verify that you get the expected results. Note: The simulation configuration parameters used in the simulation are as depicted in Fig. 3. The shortcut CTRL+E can be used to open the Model Configuration Parameters window. A fixed-step solver is used for the simulation, with only discrete states. The step size for the simulation is set at 1/10000 sec; this ensures smooth display of the continuous-time sinusoid $x(t)$.

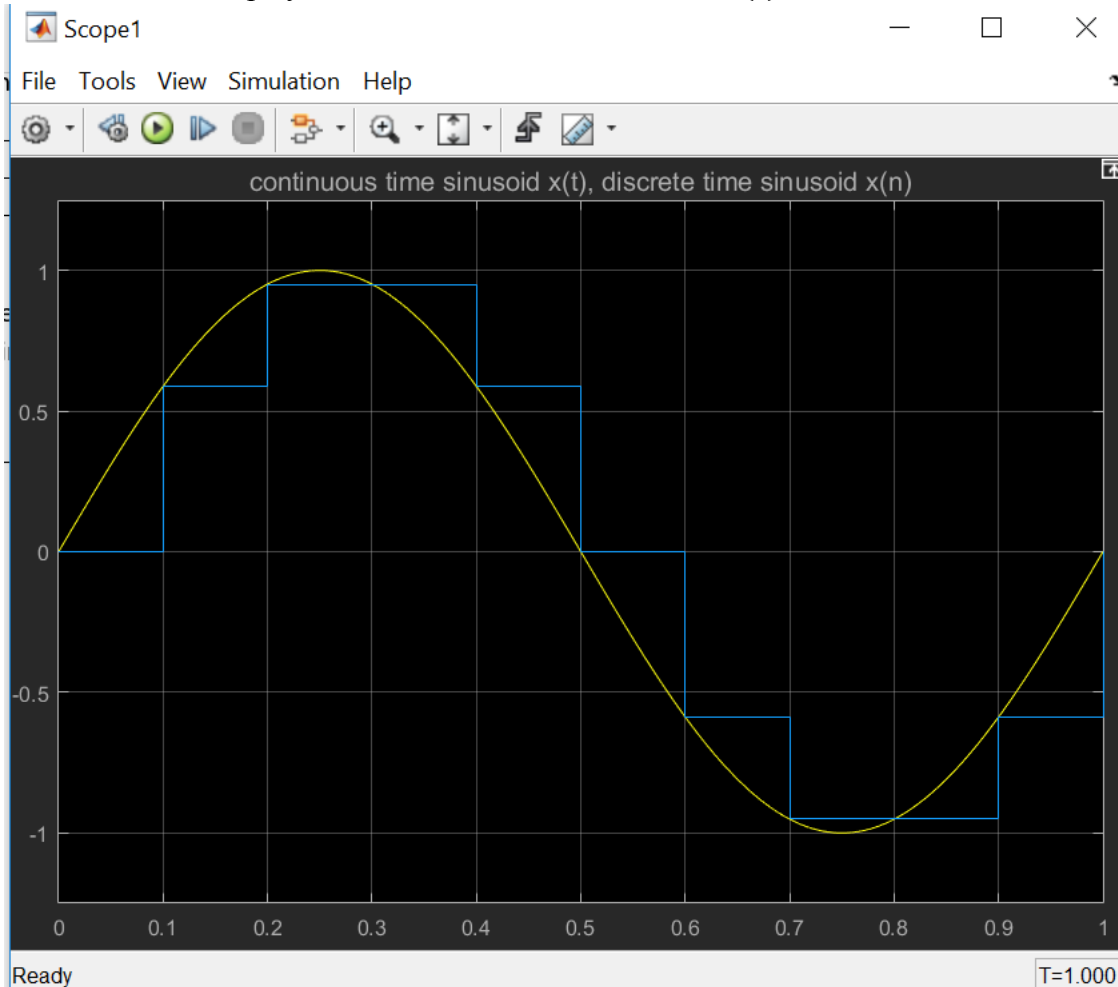


Fig. 2: Oscilloscope display of simulation results

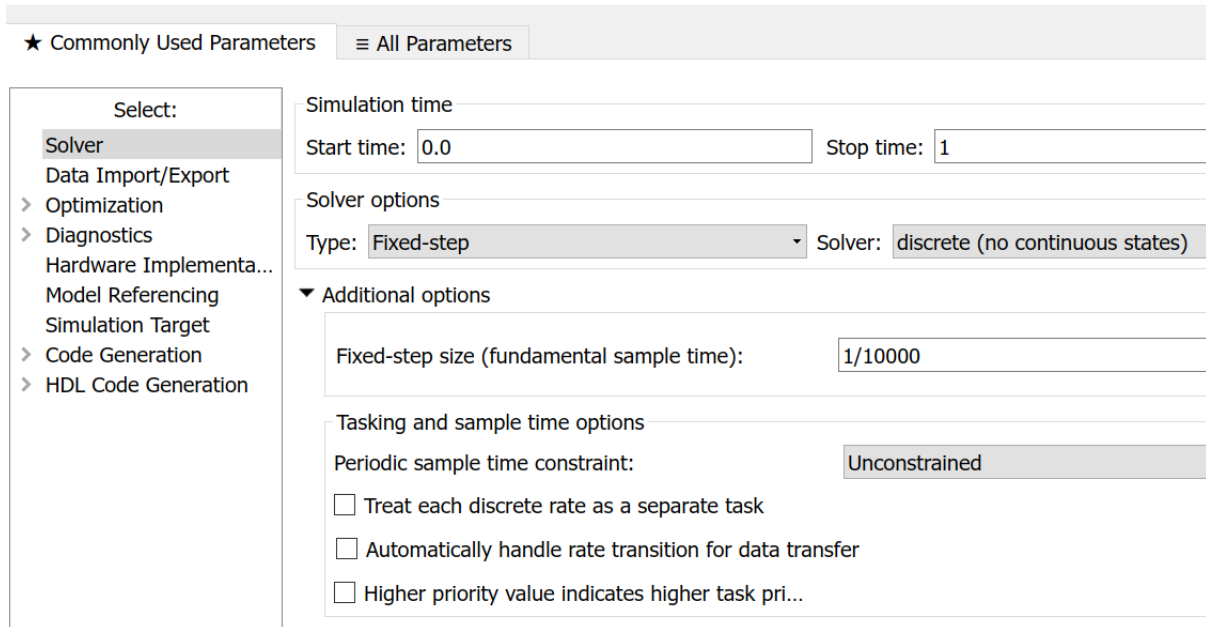


Fig. 3: Model Configuration Parameters for the simulation

Simulink exercise

Create the signal $x(t) = 2 \sin(2\pi t) + \sin(4\pi t)$ in Simulink. Sample the signal at a sampling rate $F_s = 20$ samples/sec (sampling period $T = 1/20 = 0.05$ s) to produce the discrete-time signal $x(n)$. Run the simulation for a duration of 1 sec and display both these waveforms on an oscilloscope. Upload a Word or PDF file containing screenshots of the Simulink block diagram and oscilloscope traces. Your block diagram must include text showing the settings applied to all blocks. Also upload your Simulink model file with .slx extension. You will need an “Add” block to add the two sinusoids; you can find the “Add” block using the search bar of the library browser (or by looking in the Math Operations library). Your Simulink block diagram will end up looking similar to that of Fig. 4. You will need to add text to show the settings of the different blocks.

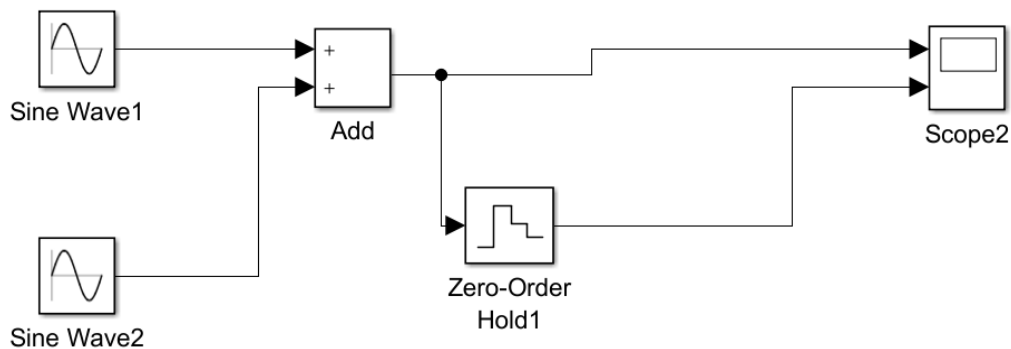


Fig. 4: Form of the block diagram for the Simulink exercise

Simulink lab 2: Exploration of Aliasing

To explore aliasing using Simulink we will consider two sinusoidal signals having analog frequencies $F_1 = 1$ kHz and $F_2 = 11$ kHz. We will sample both these sinusoids at a sampling rate $F_s = 10000$ samples/sec. The corresponding sampling period is $T = 10^{-4}$ s, which is entered in Matlab or Simulink as $1e-4$. We know from the Sampling Theorem that the 11kHz sinusoid experiences aliasing. The digital frequencies of the sampled sinusoids are $f_1 = \frac{1\text{kHz}}{10000} = 0.1$ cycles/sample and $f_2 = \frac{11\text{kHz}}{10000} = 1.1$ cycles/sample. We also know that two sinusoids that differ in frequency by 1 cycle/sample (or 2π rad/sample) are identical. Thus $f_2 = 1.1$ cycles/sample $= 0.1$ cycles/sample. Thus $f_2 = f_1$, and the 11kHz sinusoid gets aliased onto 1kHz.

The Simulink block diagram of Fig. 1 is used to visualize the sampling and aliasing process. The 1kHz and 11kHz sinusoid are created and sampled with a sampling period of 0.1ms (sample time $T = 1e-4$). The sampled signals are then viewed on a scope.

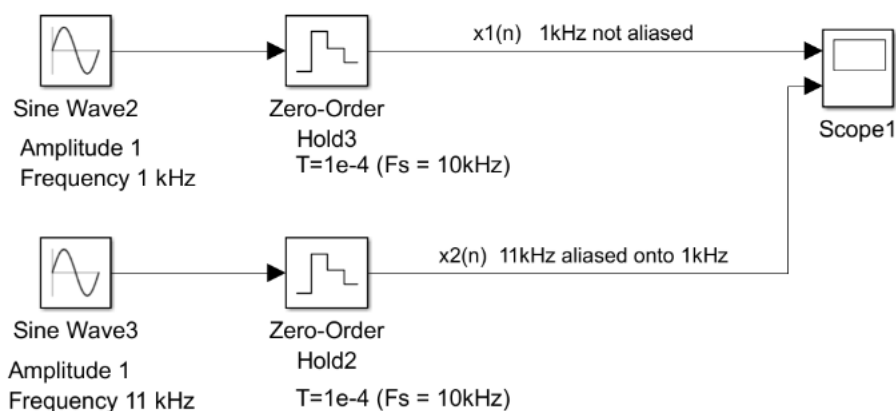


Fig. 1: Block diagram for exploration of aliasing

Fig. 2 shows the output of the Scope, with layout adjusted such that the two traces appear one below the other. The figure shows that the samples of the 1kHz signal and the 11kHz signal are identical (the 11kHz signal aliases to 1kHz). With the default layout, both traces lie on top of each other (the waveforms are identical), and it appears that only one waveform is present.

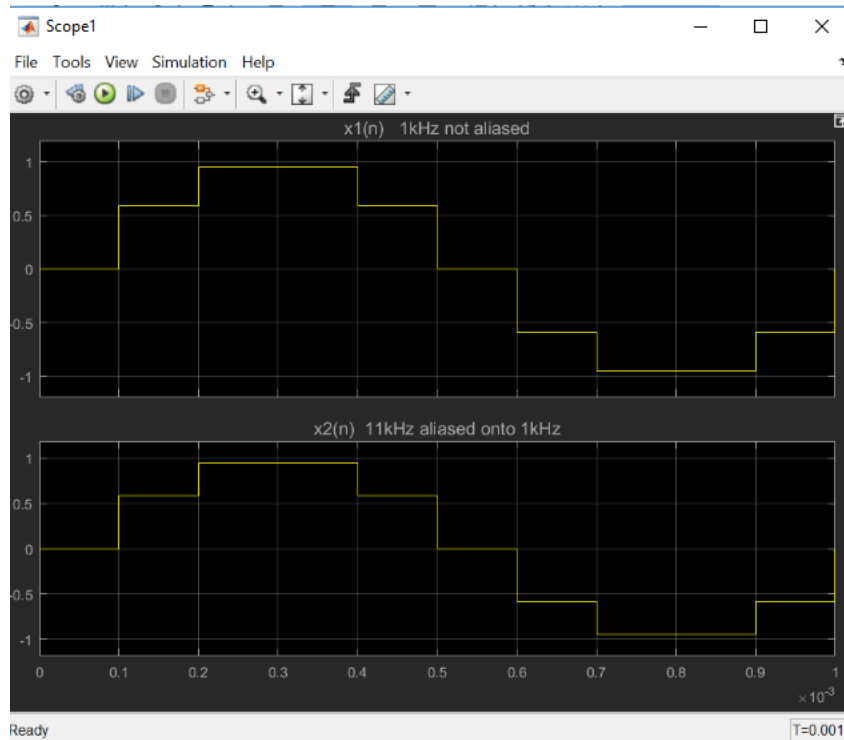


Fig. 2: Oscilloscope display showing aliasing of the 11kHz signal

Simulink exercise

Identify one more analog frequency in the range 5 to 10kHz that gets aliased onto 1kHz when sampled at $F_s = 10\text{kHz}$. Recall: a sinusoid of negative frequency $f = -0.1$ cycles/sample also oscillates at the same rate and has the same period as a sinusoid of frequency $f = 0.1$ cycles/sample. Show your work illustrating how you identify this analog frequency F_3 .

Create the Simulink block diagram of Fig. 1. Add a third sinusoid of frequency F_3 and sample it using the same sampling period. Run the simulation using a simulation stop time of 1ms (this will show one period of the 1ms sinusoid). Use the same simulation configuration parameters as in the Introduction to Simulink lab (fixed-step solver with only discrete states, fixed step-size = $1e-4$). View all three sampled waveforms on a scope. Create a scope view in which all three waveforms appear one below the other. Do you agree that frequency F_3 gets aliased onto F_1 ? Why?

Submit a Word or PDF file showing your block diagram, scope waveforms and calculations / discussion. Also submit your .slx Simulink file.

Simulink lab 3: Impulse and Step Response of Linear, Time-Invariant Systems

In this lab we will learn how to represent a system described by a linear, constant coefficient difference equation in Simulink. We will also learn how to obtain the impulse response and step response of such systems.

Let us consider the system $y(n) - y(n - 1] + 0.25y(n - 2) = x(n) + 2x(n - 1) + x(n - 2)$, where $x(n)$ is the input to the system, and $y(n)$ is the resulting system output. The block diagram of Fig. 1 shows how Simulink is used to obtain the impulse response and step response of the system. New Simulink blocks used in this lab are the Discrete Impulse, Step, Discrete Filter, and To Workspace. All these blocks can be located using Search in the library browser. The Discrete Impulse part is located under Sinks in the DSP System Toolbox; it is not available with a base edition of Matlab / Simulink.

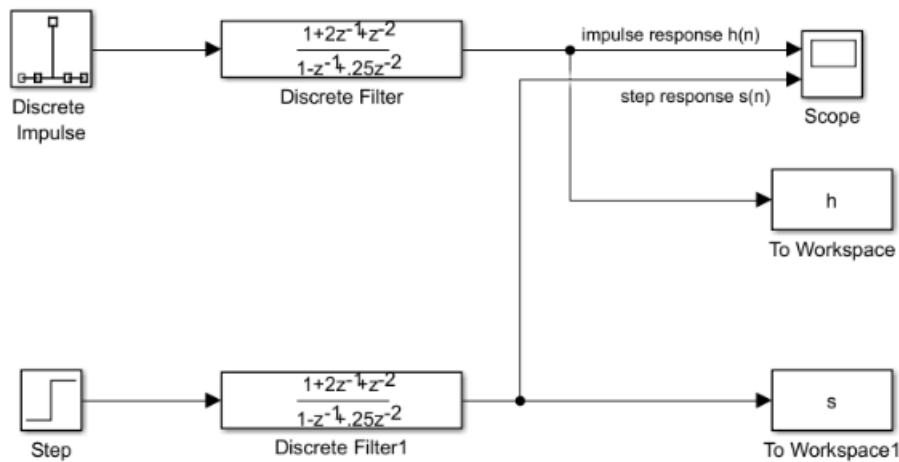


Fig. 1: Block diagram used to obtain impulse and step response of a system

Fig. 2 shows the settings for the Discrete Impulse and the Discrete Filter blocks (these settings are made by double clicking on the block).

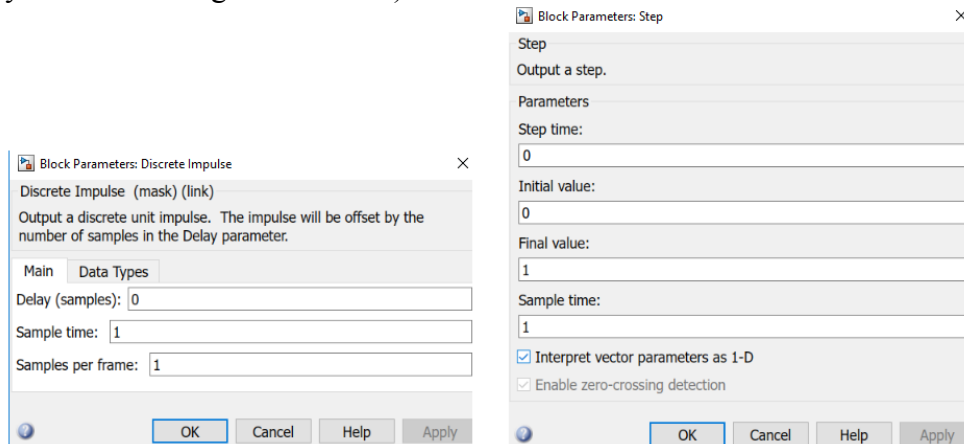


Fig. 2: Settings for the Discrete Impulse and Step blocks

For the Discrete Impulse, the delay is set to 0 since we want an impulse $\delta(n)$ located at $n = 0$. The sample time is set to 1 since we are only interested in the output of the system at discrete time instants $n = 0, 1, 2, \dots$. For the Step block, we use Step time = 0 since we are interested the step input $u(n)$ without any delay. We want the step to vary between values of 0 and 1 and the initial value and final value of the step are thus set accordingly.

Fig. 3 shows the settings for the Discrete Filter block and the To Workspace block. The filter difference equation is $y(n) - y(n - 1) + 0.25y(n - 2) = x(n) + 2x(n - 1) + x(n - 2)$. The Numerator for the Discrete Filter is the vector [1 2 1] of right-hand-side coefficients of the difference equation, while the Denominator is the vector [1 -1 0.25] of left hand side coefficients of the difference equation. The Sample time is set to -1: this inherits the sample time from the previous block (we had set the sample time to 1).

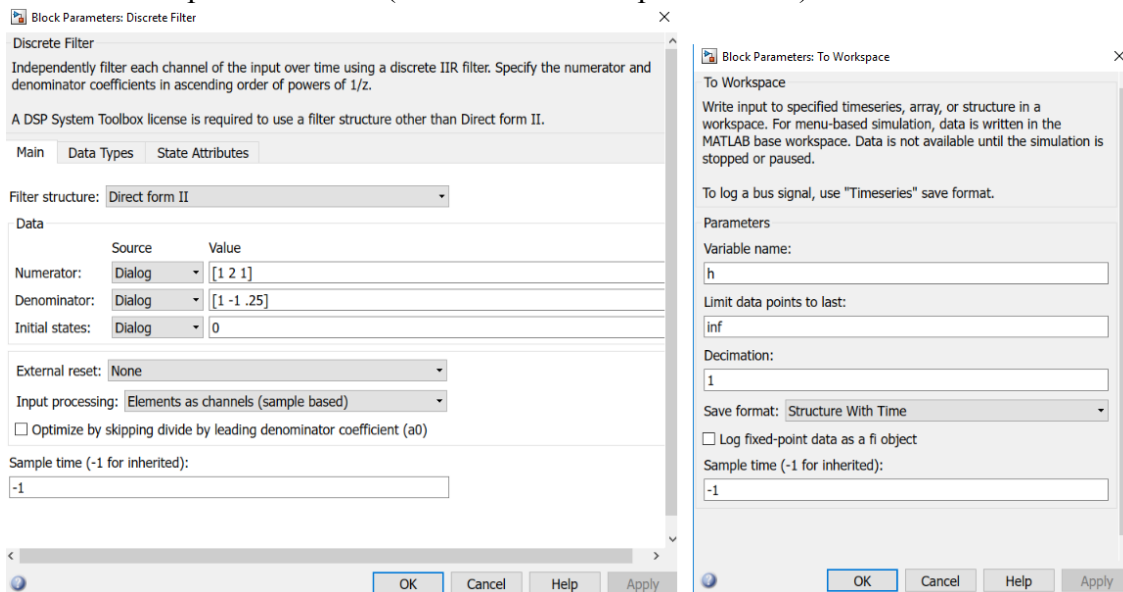


Fig. 3: Settings for the Discrete Filter and To Workspace blocks

In the block diagram of Fig. 1, the filter output goes to a Scope as well as to a To Workspace block. The scope displays the impulse response, while the To Workspace block sends the samples of the impulse response to the Matlab workspace so the actual samples can be examined in Matlab.

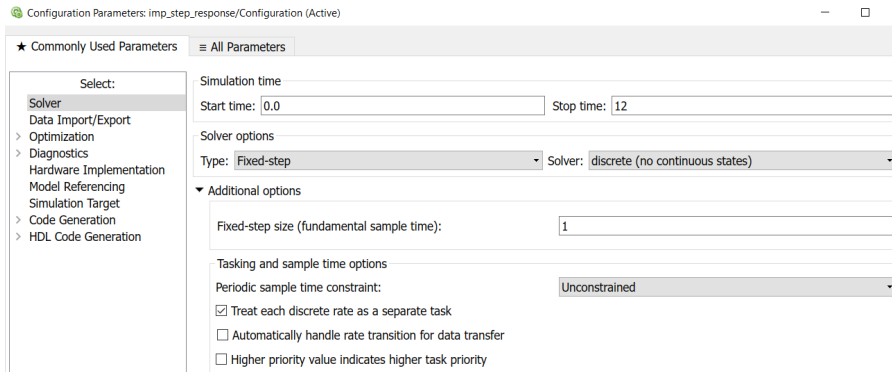


Fig. 4. Simulation configuration parameter settings

Fig. 4 shows the simulation configuration settings for the simulation (the settings window can be opened using CTRL E). As in previous labs, we use only discrete states for the simulation. The only difference from previous labs is that the step size is set to 1 since we are only interested in outputs at integer time instants. The simulation is run for a duration of 12 seconds (or 12 samples).

When the simulation of the system of Fig. 1 is executed using the green play button, the waveforms of Fig. 5 appear on the scope. Clicking on the Scale Axes button highlighted in Fig. 5 resizes the plots so they occupy most of the window.



Fig. 5. Impulse response and step response of the discrete filter

The impulse response h that Simulink sends to the workspace is a structure. Going to the Matlab command window, the command below allows one to view the impulse response in compact form:

```
imp = squeeze(h.signals.values)
```

imp = Columns 1 through 8: 1.0000 3.0000 3.7500 3.0000 2.0625 1.3125 0.7969
0.4688
Columns 9 through 13: 0.2695 0.1523 0.0850 0.0469 0.0256

The samples of imp correspond to the impulse response sample values of Fig. 5.

Simulink exercise

Obtain the impulse response and step response of the system $y(n) + 1.4y(n - 1) + 0.49y(n - 2) = x(n)$. The oscilloscope must be set so the impulse response and step response are properly titled and the plots have the format of Fig. 5.

Submit a Word or PDF file showing your block diagram and scope waveforms. Also submit your .slx Simulink file.

Simulink lab 4: Power spectra of signals

In this lab we will learn how to use a spectrum analyzer in Simulink to obtain the power spectrum of a signal. The power spectrum refers to distribution of signal power as a function of frequency.

Let us consider the sinusoidal voltage signal $v(t) = A \sin(2\pi Ft)$. We know from Circuits that the RMS (root-mean-square) or effective value of the signal $v(t)$ is $V = \frac{A}{\sqrt{2}}$. If $v(t)$ were connected across a resistance R Ohms, the average power delivered by the sinusoidal signal to the resistor would be $\frac{V^2}{R} = \frac{A^2}{2R}$. If we consider a resistance $R = 1\Omega$, the average power delivered is $P = \frac{A^2}{2}$. In the signal processing field, the power spectrum shows the average power of the signal as a function of frequency, assuming the signal was a voltage, and that the load was 1Ω .

Let us now consider a sinusoid of amplitude $A = 1$ and frequency $F = 2\text{kHz}$. Based on the above discussion, its average power is $P = \frac{A^2}{2} = 0.5$ Watt. We will use the block diagram of Fig. 1 to estimate the power spectrum of such a sinusoid: $v(t) = \sin(2\pi * 2000t)$. Both blocks are from the DSP System Toolbox: the discrete-time sine wave is from the *Sources* folder of this toolbox, and the spectrum analyzer from the *Sinks* folder of the toolbox. Fig. 1 also shows the settings of the discrete-time sinusoid (set by double clicking the block).

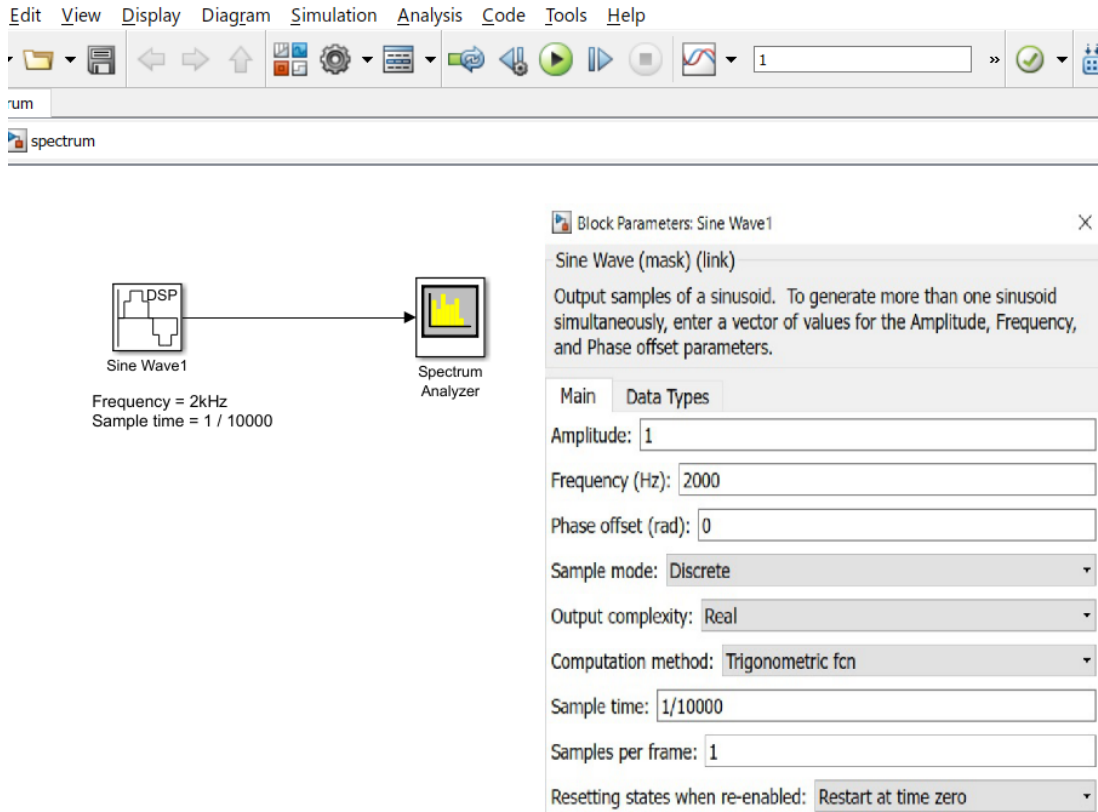


Fig. 1: Block diagram to evaluate power spectrum of a sinusoidal signal

Fig. 2 shows the simulation configuration parameters: the settings window can be opened using CTRL+E. A fixed-step solver is used with a simulation stop time of 1s (a long data record is needed to get an accurate power spectrum).

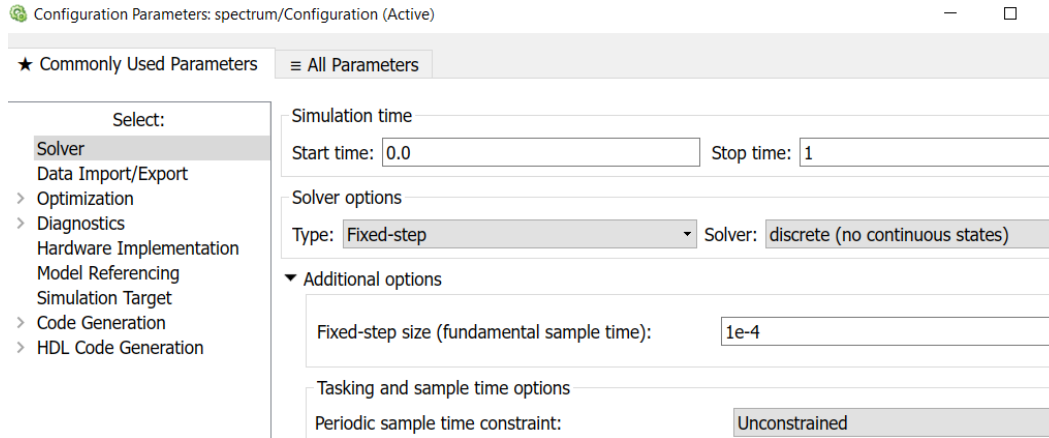


Fig. 2: Simulation Configuration Parameters

The simulation can be run by pressing the green Play button or by typing CTRL+T. The spectrum analyzer can be opened by double clicking it. Spectrum analyzer settings are set by going to View and then selecting Spectrum Settings. Fig. 3 shows the spectrum analyzer settings and the spectral plot resulting from the simulation.

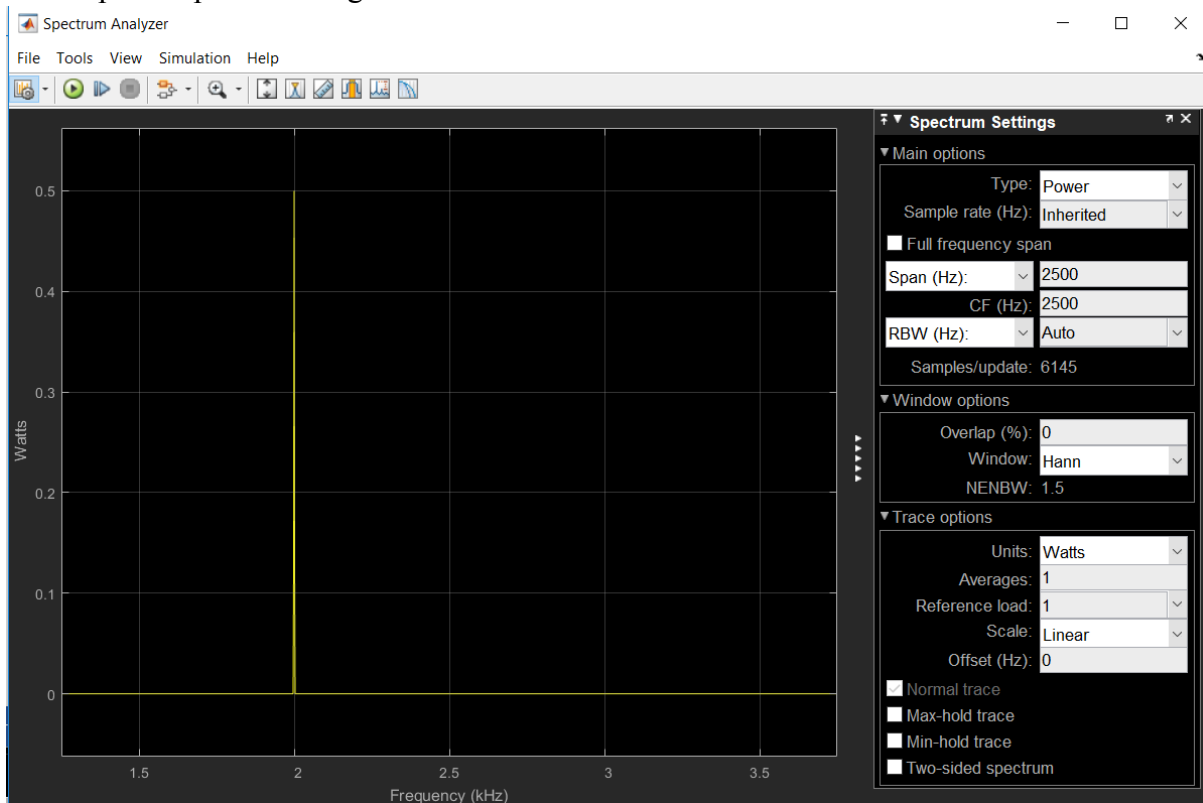


Fig. 3: Spectrum Analyzer settings and simulation results

The sampling rate employed in the simulation is $F_s = 10\text{kHz}$, which allows for examination of frequencies in the range 0 to 5kHz. The center frequency (CF) is set to 2500Hz (the center of this frequency range), and the Span to 2500Hz. This allows for evaluation of the frequency range 0 to 5kHz. Altering the center frequency or span alters the range of frequencies visible in the spectral plot of Fig. 3 (run the simulation and experiment with different center frequency and span settings).

As expected and predicted by the theoretical discussion at the beginning of the lab, Fig. 3 shows a large spectral spike at $F = 2\text{kHz}$ (the frequency of the sinusoid). The figure also shows that the power of the sinusoid is 0.5W, as predicted.

Simulink exercise

Begin by completing the simulation of Fig. 1 and make sure that you obtain the expected results. Then go on to complete the exercise below.

We wish to examine the spectrum of the signal $x(t) = \sin(2\pi * 1500t) + 2 \sin(2\pi * 2000t)$ that contains a 1.5kHz component of amplitude 1 and a 2kHz component of amplitude 2. Create the sampled version of $x(t)$ using a sampling rate $F_s = 10\text{kHz}$ (you will need two discrete-time sinusoids and an Add block). Then use the spectrum analyzer to obtain the spectrum of the signal. Record the frequencies present in the spectrum and the corresponding signal power for each of the frequencies (varying the center frequency and span can produce slightly different results; more accurate power results can be obtained if you know what particular frequencies you are looking for). Calculate by hand the expected power for each frequency component and compare with the simulation results.

Submit a Word or PDF file showing your block diagram and spectrum analyzer waveforms. Your document should contain the theoretical power calculations and make comparisons with the simulation results. Also submit your .slx Simulink file.