ECE 3101L - EXPERIMENT 9

Frequency Analysis

OBJECTIVES

The student will be able analyze continuous-time and discrete-time signals and systems using Fourier series and transforms

LIST OF EQUIPMENTS AND PARTS

1 PC with MATLAB

PRELAB:

1. INTRODUCTION

We will use Fourier series and transforms to analyze continuous-time and discrete-time signals and systems. The Fourier representations of signals involve the decomposition of the signal in terms of complex exponential functions. These decompositions are very important in the analysis of linear time-invariant (LTI) systems. The response of an LTI system to a complex exponential input is a complex exponential of the same frequency! Only the amplitude and phase of the input signal are changed. Therefore, studying the frequency response of a LTI system gives complete insight into its behavior.

In this experiment, we will use the Simulink extension to MATLAB. Simulink is an icon-driven dynamic simulation package that allows the user to represent a system or a process by a block diagram. Once the representation is completed, Simulink may be used to digitally simulate the behavior of the continuous or discrete-time system. Simulink inputs can be MATLAB variables from the workspace, or waveforms or sequences generated by Simulink itself. These Simulink-generated inputs can represent continuous-time or discrete-time sources. The behavior of the simulated system can be monitored using Simulink's version of common lab instruments: scopes, spectrum analyzers and network analyzers. The outputs of these "devices" are displayed in graph windows. The output of a system can be viewed using one of the monitoring devices listed above, or it may be saved to a variable created in the MATLAB workspace.

2 Background Exercises

2.1 Synthesis of Periodic Signals

Fourier series expansion in the form of

$$x(t) = c_0 + \sum_{n=1}^{\infty} C_k \cos(2\pi n f_0\, t + \theta\_k)$$

where $f_0 = \frac{1}{T_0}$.

Compute the cosine form of the Fourier series, of the following signal.

Sketch the signal on the interval [0, T_0].

a. Period $T_0 = 2$ For $t[0-2]$

$$x(t) = rect\left(t - \frac{1}{2}\right)$$

b. Period $T_0 = 2$ For $t\left[-\frac{1}{2}, 1/2\right]$

$$x(t) = rect(2t) - \frac{1}{2}$$

2.2 Magnitude and Phase of Discrete-Time Systems

For the discrete-time system described by the following difference equation,

$$y(n) = 0.9y(n-1) + 0.3x(n) + 0.24x(n-1)$$

a. Computer the impulse response
b. Draw a system diagram
c. Take the Z-transform of the difference equation using the linearity and the time shifting properties of the Z-transform
d. Find the transfer function.

$$H(z) = \frac{Y(z)}{X(z)}$$

e. Use MATLAB to compute and plot the magnitude and phase responses, $|H(e^{j\omega})|$ and $\angle H(e^{j\omega})$), for $-\pi < \omega < \pi$ You may use MATLAB commands phase and abs

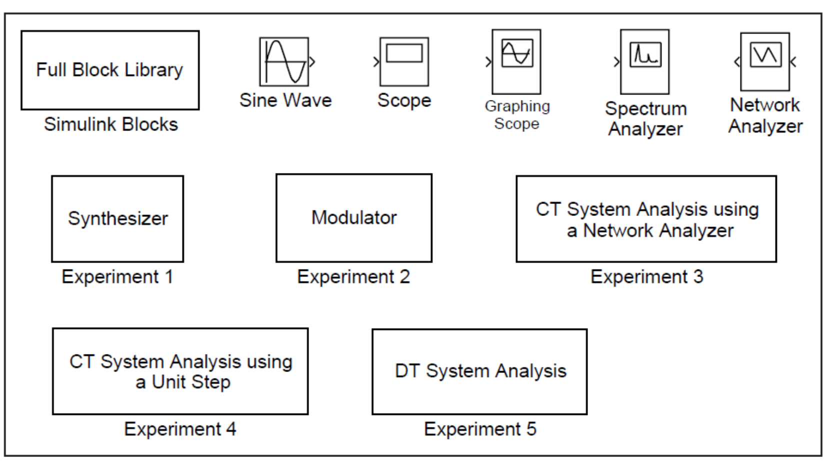


Figure 1: Simulink utilities for lab 9.

3. In Lab

3.1 Getting Started with Simulink

In this section, we will learn the basics of Simulink and build a simple system To get the library of Simulink functions for this laboratory, download the Lab9Utilities. Once MATLAB is started, type "Lab9" to bring up the library of Simulink components shown in Fig. 1. This library contains a full library of Simulink blocks, a spectrum analyzer and a network analyzer designed for this laboratory, a sine wave generator, a scope, and pre-design systems for each of the experiments that you will be running.

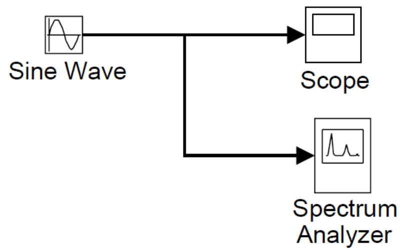


Figure 2: Simulink model for the introductory example.

In order to familiarize yourself with Simulink, you will first build the system shown in Fig. 2. This system consists of a sine wave generator that feeds a scope and a spectrum analyzer.

1. Open a window for a new system by using the *New* option from the *File* pull-down menu, and select *Model*.
2. Drag the *Sine Wave*, *Scope*, and *Spectrum Analyzer* blocks from the *Lab9* window into the new window you created.
3. Now you need to connect these three blocks. With the left mouse button, click on the output of the *Sine Wave* and drag it to the input of the *Scope*. Now use the right button to click on the line you just created, and drag to the input of the *Spectrum Analyzer* block. Your system should now look like Fig. 2.
4. Double click on the *Scope* block to make the plotting window for the scope appear.
5. Set the simulation parameters by selecting *Simulation parameters* from the *Simulation* pull-down menu. Under the *Solver* tab, set the *Stop time* to 50, and the *Max step size* to 0.02. Then select *OK*. This will allow the Spectrum Analyzer to make a more accurate calculation.
6. Start the simulation by using the *Start* option from the *Simulation* pull-down menu. A standard MATLAB figure window will pop up showing the output of the *Spectrum Analyzer*.
7. Change the frequency of the sine wave to 5*pi rad/sec by double clicking on the *Sine Wave* icon and changing the number in the *Frequency* field. Restart the simulation. Observe the change in the waveform and its spectral density. If you want to change the time scaling in the plot generated by the spectrum analyzer, from the MATLAB prompt use the subplot(2,1,1) and axis() commands.

3.2 Continuous-Time Frequency Analysis

In this section, we will study the use and properties of the continuous-time Fourier transform with Simulink. The Simulink package is especially useful for continuous-time systems because it allows the simulation of their behavior on a digital computer.

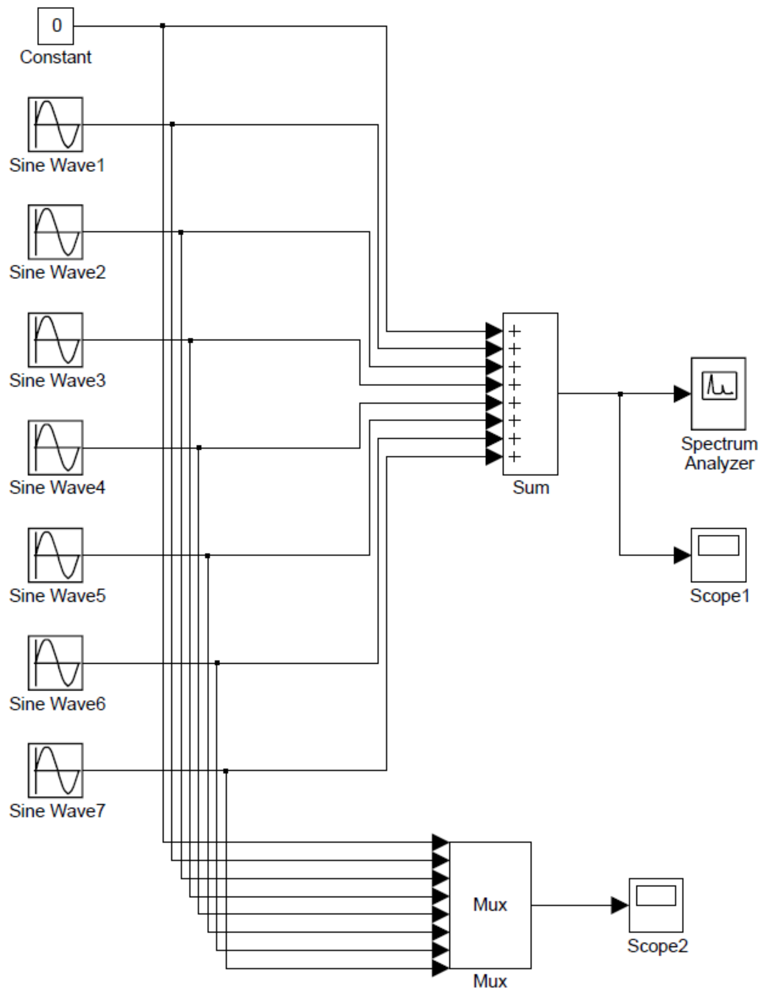


Figure 3: Simulink model for the synthesizer experiment

3.2.1 Synthesis of Periodic Signals

Double click the icon labeled *Synthesizer* to bring up a model as shown in Fig. 3. This system may be used to synthesize periodic signals by adding together the harmonic components of a Fourier series expansion. Each *Sin Wave* block can be set to a specific frequency, amplitude and phase. The initial settings of the *Sin Wave* blocks are set to generate the Fourier series expansion

These are the first 8 terms in the Fourier series of the periodic square wave shown in Fig. 4. Run the model by selecting *Start* under the *Simulation* menu. A graph will pop up that shows the synthesized square wave signal and its spectrum. This is the output of the *Spectrum Analyzer*. After the simulation runs for a while, the *Spectrum Analyzer* element will update the plot of the spectral energy and the incoming waveform. Notice that the energy is concentrated in peaks corresponding to the individual sine waves. Print the output of the *Spectrum Analyzer*.

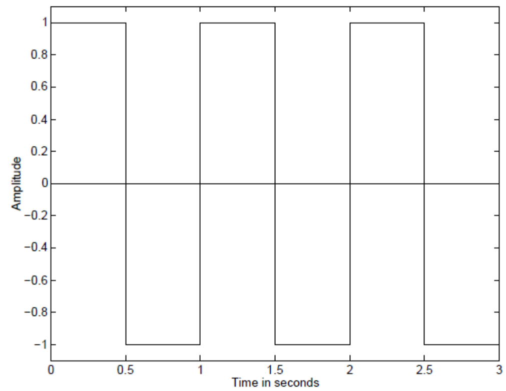


Figure 4: The desired waveform for the synthesizer experiment.

You may have a closer look at the synthesized signal by double clicking on the *Scope1* icon. You may see the simultaneous plot of all the sine waves by double clicking on the *Scope2* icon.

Synthesize the two periodic waveforms listed in section 2.1 of the background exercises.
Do this by setting the frequency, amplitude, and phase of each sinewave generator to the proper values. For each case, print the output of the *Spectrum Analyzer*.

3.2.2 Modulation Property

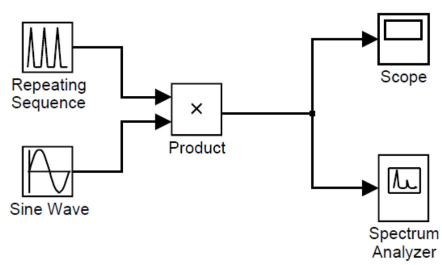


Figure 5: Simulink model for the modulation experiment.

Double click the icon labeled *Modulator* to bring up a system as shown in Fig. 5. This system modulates a triangular pulse signal with a sine wave. You can control the duration and duty cycle of the triangular envelope and the frequency of the modulating sine wave. The system also contains a spectrum analyzer which plots the modulated signal and its spectrum.

Generate the following signals by adjusting the *Time values* and *Output values* of the *Repeating Sequence* block and the *Frequency* of the *Sine Wave*. The *Time values* vector contains entries spanning one period of the repeating signal. The *Output values* vector contains the values of the repeating signal at the times specified in the *Time values* vector. Note that the *Repeating Sequence* block does NOT create a discrete time signal. It creates a continuous time signal by connecting the output values with line segments. Print the output of the *Spectrum Analyzer* for each signal.

1. Triangular pulse duration of 1 sec; period of 2 sec; modulating frequency of 10 Hz (initial settings of the experiment).
2. Triangular pulse duration of 1 sec; period of 2 sec; modulating frequency of 15 Hz.
3. Triangular pulse duration of 1 sec; period of 3 sec; modulating frequency of 10 Hz.
4. Triangular pulse duration of 1 sec; period of 6 sec; modulating frequency of 10 Hz.

Notice that the spectrum of the modulated signal is composed of a comb of impulses in the frequency domain arranged about a center frequency.

3.3 Discrete-Time Frequency Analysis

In this section, we will study the use of the discrete-time Fourier transform.

3.3.1 Discrete-Time Fourier Transform

The DTFT (Discrete-Time Fourier Transform) is the Fourier representation used for finite energy discrete-time signals. For a discrete-time signal, $x(n)$, we denote the DTFT as the function $X(e^{j\omega})$ given by the expression

$$X\left(e^{j\omega}\right) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}$$

Since $X(e^{\omega})$ is a periodic function of ω with a period of 2π, we need only to compute $X(e^{\omega})$ for $-\pi < \omega < \pi$.

Write a MATLAB function X = DTFT(x,n0,dw) that computes the DTFT of the discrete- time signal x. Here n0 is the **time** index corresponding to the 1^{st} element of the x vector, and dw is the spacing between the samples of of the MATLAB vector X. For example, if x is a vector of length N, then its DTFT is computed by

$$X(\omega) = \sum_{n=1}^{N} x(n)e^{-j\omega(n+n0-1)}$$

where w is a vector of values formed by w=(-pi:dw:pi).

Hint: In MATLAB, j or i is defined as $\sqrt{-1}$. However, you may also compute this value using the MATLAB expression **i** = sqrt(-1).

For the following signals use your DTFT function to

a. Compute $X(e^{j\omega})$

b. Plot the magnitude and the phase of $X(e^{j\omega})$ in a single plot using the *subplot* command.
   Hint: Use the abs() and angle()

commands.

1. $x(n) = \delta(n)$
2. $x(n) = \delta(n - 5)$
3. $x(n) = (0.5)^n u(n)$

INLAB REPORT: Hand in a printout of your MATLAB function. Also hand in plots of the DTFT's magnitude and phase for each of the three signals.

3.3.2 System Analysis

Double click the icon labeled *DT System Analysis* to bring up an incomplete block diagram as shown in Fig. 8. It is for a model that takes a discrete-time sine signal, processes it according to a difference equation and plots the multiplexed input and output signals in a graph window. Complete this block diagram such that it implements the following difference equation given in section 2.2 of the background exercises.

$$y(n) = 0.9y(n - 1) + 0.3x(n) + 0.24x(n - 1)$$

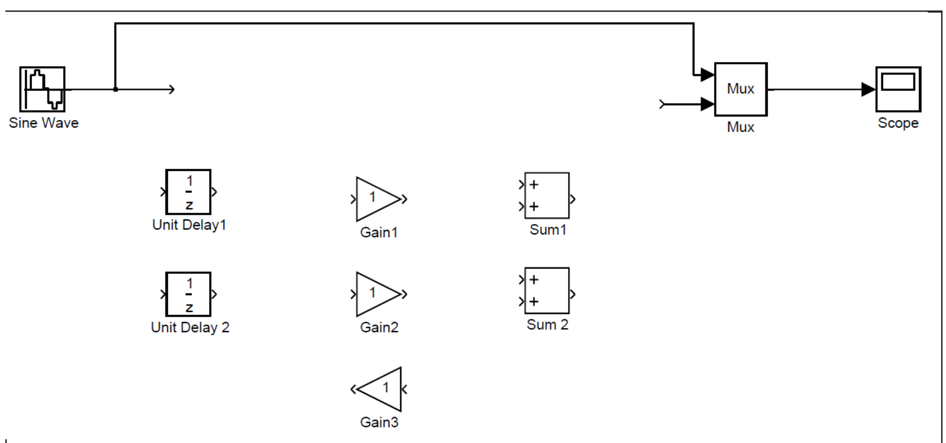


Figure 8: Incomplete Simulink setup for the discrete-time system analysis experiment.

You are provided with the framework of the setup and the building blocks that you will need. You can change the values of the *Gain* blocks by double clicking on them. After you complete the setup, adjust the frequency of *Sine Wave* to the following frequencies: $\omega = \pi/16$, $\omega = \pi/8$, and $\omega = \pi/4$. For each frequency, make magnitude response measurements using the input and output sequences shown in the graph window. Compare your measurements with the values of the magnitude response $|H(e^{j\omega})|$ which you computed in the background exercises at these frequencies.

An alternative way of finding the frequency response is taking the DTFT of the impulse response. Use your DTFT function to find the frequency response of this system from its impulse response. The impulse response was calculated in section 2.2 of the background exercises. Plot the impulse response, and the magnitude and phase of the frequency response in the same figure using the *subplot* command.

Example

Consider the system described by the difference equation

$$y(n) = -0.5y(n-1) + 0.5x(n) - 0.25x(n-1)$$

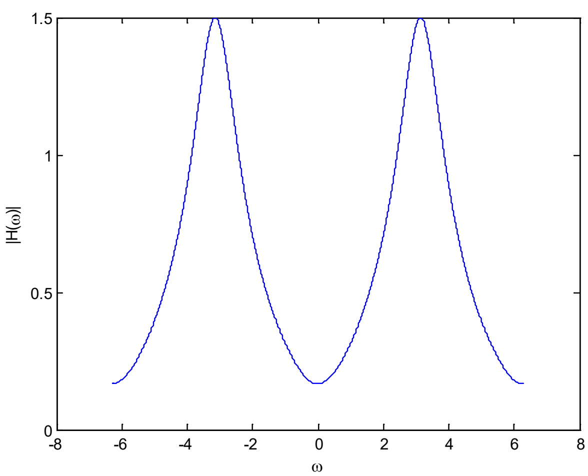
a. Determine its frequency response H(ω).
b. Sketch | H(ω)|. What kind of filter is this?
c. Determine its response to the following input.

$$x(n) = 2\cos(\frac{\pi}{4}n) + \cos(\frac{\pi}{2}n) + 3\sin(\pi n)$$

Solution:

a. $$H(\omega) = \frac{0.5 - 0.25e^{-j\omega}}{1 + 0.5e^{-j\omega}}$$

b.



HP Filter

c.

$$H(\frac{\pi}{4}) = \frac{0.5 - 0.25e^{-j\frac{\pi}{4}}}{1 + 0.5e^{-j\frac{\pi}{4}}} = 0.2633\measuredangle 43.3^\circ$$

$$H(\frac{\pi}{2}) = \frac{0.5 - 0.25e^{-j\frac{\pi}{2}}}{1 + 0.5e^{-j\frac{\pi}{2}}} = 0.5\measuredangle 53.13^\circ$$

$$H(\pi) = \frac{0.5 - 0.25e^{-j\pi}}{1 + 0.5e^{-j\pi}} = 1.5\measuredangle 0^\circ$$

$$y(n) = 2\,|H(\frac{\pi}{4})|\cos(\frac{\pi}{4}n + \angle H(\frac{\pi}{4})) + |H(\frac{\pi}{2})|\cos(\frac{\pi}{2}n + \angle H(\frac{\pi}{2})) + |H(\frac{\pi}{4})|\,3\sin(\pi n + \angle H(\pi))$$

$$y(n) = 0.5266\cos(\frac{\pi}{4}n + 43^\circ) + 0.5\cos(\frac{\pi}{2}n + 53.13^\circ) + 4.5\sin(\pi n)$$

You build same blocks on Simulink and test the results.