

# **Lab 3(Part A)**

## **Symbolic expressions**

## **1.1 Motivation:**

MATLAB can create and handle symbolic expressions

## **1.2 Introduction and background theory:**

### **1.2.1 Symbolic Variables / Expressions:**

As matrices, symbolic variables should be defined prior to their usage in symbolic expressions:

- Sym: constructs symbolic numbers, variables and objects.
- Syms: symbol1 symbol2 type is a short-cut for only constructing symbolic objects.
- Symbol name must begin with a letter and can contain only alphanumeric characters.
- To define a symbolic expression we simply write it using defined symbols and functions.
- MATLAB has several built-in functions for building mathematical expressions, some of these functions have already been defined in Lesson 1 such as sin,tan,cot, sqrt, exp, log, log10, conj, abs, angle, etc.

#### **Notes:**

- ❖ When using sym to define a symbol (e.g. pi) to be the symbolic representation for a number (e.g. pi), the symbol avoids the floating point approximations inherent in the value of pi.
- ❖ For symbolic expressions, matrix and array operations are the same.

### **1.2.2 Differentiation of symbolic expressions:**

MATLAB can differentiate symbolic expressions for n times using the diff command:

- diff(SE): differentiates symbolic expression with respect to its free variable determined by 'findsym'.
- diff(SE,var): differentiate symbolic expression with respect to the variable 'var' one time.
- diff(SE,n): differentiate symbolic expression with respect to its free variable n times.
- diff(SE,var,n): differentiate symbolic expression with respect to the variable 'var' n times.

### **1.2.3 Integration of symbolic expressions:**

MATLAB can handle definite, indefinite (anti-derivative) and as well as numerical integrals. Numerical integrals are conventionally used as an approximation for integrating symbolic

expressions that do not have a closed form integrals, on the other hand, it can also be used for symbolic expressions with closed form integrals.

- `Int(SE)`: indefinite integral of symbolic expression with respect to variable found by `'findsym'`.
- `Int(SE,var)`: indefinite integral of symbolic expression with respect to variable `'var'`.
- `Int(SE,a,b)`: definite integral of symbolic expression with respect to its free variable from `a` to `b`.
- `Int(SE,var,a,b)`: definite integral of symbolic expression with respect to variable `'var'` from `a` to `b`.

Notes:

- ❖ To include minus infinity `"- ∞"` and / or infinity `"∞"` in the limits of integration, use their symbolic representation in MATLAB `-inf` and / or `inf`.
- ❖ For discrete functions we can use the command `sum` instead of `int`.

### 1.2.4 Learning more

There are many useful symbolic operations that you can perform using the symbolic math toolbox. To learn more go to `help> Contents>Symbolic Math Toolbox`. Below we will be introducing a few more useful symbolic math operations. We will give you the needed function but you will need to go to the help and figure out how to apply them to solve the exercise problems.

**Answer Sheet Name:** \_\_\_\_\_ **#=** \_\_\_\_\_ **Section** \_\_\_\_\_

1. Compute the partial derivatives of  $f(x,t)=\cos(x)+\sin(t)+e^t$

2. Using Matlab symbolic math calculate the definite integral  $\int_0^{\infty} t e^{-t} dt$

3. Calculate the double indefinite integral  $\iint x^3 e^t dx dt$

Does it matter what order you do the integrations?

4. Using the symbolic summations function (symsum) compute the infinite summation

$$\sum_{k=0}^{\infty} \frac{x^k}{k!}.$$

5. Using the symbolic function numden express in rational form the function

$$f(x) = \frac{3}{x+2} + \frac{x}{x^2+1}$$

6. Using the solve operator find the solutions of the equation

$$f(x) = x^3 + 2x^2 - x - 2 = 0$$

7. Finally compute and plot the solution of the differential equation  $y''(t) + y(t) = 1$  for the initial conditions  $y(0) = 0, y'(0) = 0$  over the interval  $-5 \leq t \leq 5$ . The plotting is a little tricky. To plot the symbolic expression it needs to be evaluated and plotted over a discrete set of time points. To do this define a time vector like you did in the last lab with a fine enough time step so that the plot will not appear “jaggy”. Then use the function “subs” (Symbolic Substitution function) in the Symbolic math toolbox to substitute the  $t$  values into the symbolic express to create a vector which you can then plot against  $t$ . Check work, look at your plot, does the function as plotted look like it satisfies the initial conditions?

# **Lab 3(Part B)**

## **Simulink**

## **Motivation:**

1. To introduce Simulink environment.
2. To identify various types of linear modulated waveforms in time and frequency domain representations.
3. To implement theoretically functional circuits using Simulink.

## **Brief Overview of Simulink:**

In this experiment, we will use Simulink rather than Matlab script. It is thus important to be familiar with Simulink both in concept, advantages and how to use. In the prelab you should have familiarized your self with the basics of using Simulink

### **(a) Advantages of simulink:**

Simulink can be considered as a graphical based library of Matlab that is capable of implementing systems in a manner very similar to using hardware.

Simulink has many advantages, including powerful function libraries and display tools. Also, Simulink is compatible with many VHDL and DSP kits so the design can be downloaded to real-time hardware implementation.

### **(b) Disadvantages of Simulink**

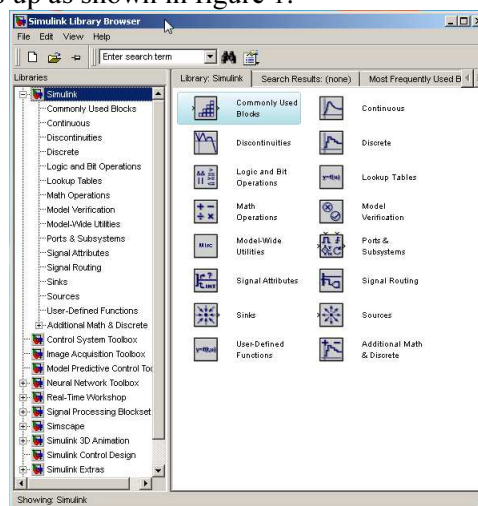
Simulink often will conceal from the user exactly what is going on so strange things can happen, operating with the MATLAB code directly provides great transparency and control over just how the calculations are being done.

## **Example of how to implement communication hardware using Simulink:**

### **Spectrum Analyzer and Function Generator.**

This section deals with looking at the spectrum of simple waves. We first look at the spectrum of a simple sine wave. Write down the questions and your answers and turning with your lab report.

To Start Simulink: Start Matlab then type simulink on the command line. A Simulink Library Browser Window opens up as shown in figure 1.



**Figure 1**

Click the blank document (new model) icon to open up a new Simulink Workspace as shown below in figure 2. Enter the design shown there to generate and display a simple sine wave in both the time and frequency domain much as we did in lab 1. Hint the signal generator is a “source” the oscilloscope is a Simulink sink. Hunt around in the library browser to see what sort of blocks you can build systems out of. The Spectrum Scope is located in the “Signal Processing Blockset” as one of the “Signal Processing Sinks”.

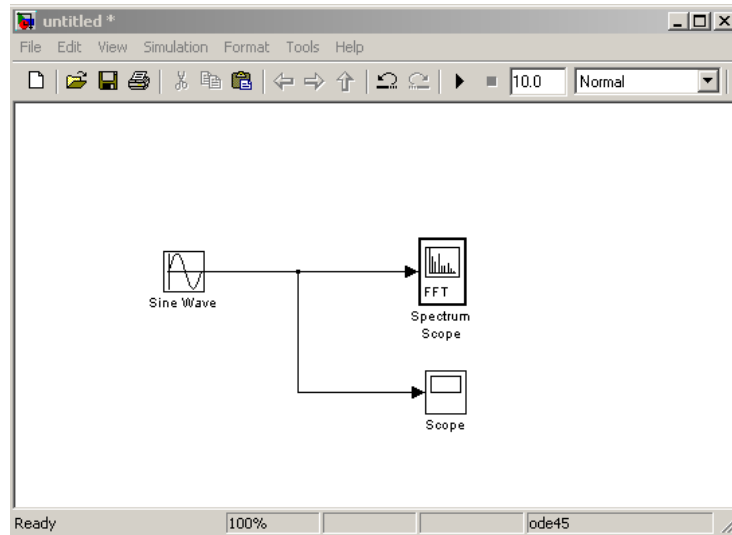


Figure 2.

The model defaults to a 10 second simulation period, this and other parameters can be adjusted from the Simulation/ConfigurationParameter Panel.

Double click on the Sine Wave Block to open up the Block Parameters Window. You will see the defaults for this block. Set the frequency to 5 rad per second and the Sample time to .01 sec.

How many samples do you have per each cycle of the sine wave?

How many samples do you have over the entire 10 seconds of the wave form?

Typically the shorter the interval between samples the more accurate the high frequency parts of the spectrum will be, the longer the interval that you measure the more accurate the low frequency parts of the spectrum will be. Later in the course you will learn the quantitative relations for sampled signals and why this is so. For now just remember that you need to have a lot of samples per cycle and lots of cycles to get good spectra.

Next double click on the Spectrum scope block and set its parameters as shown in figure 3. The buffer size setting controls how many samples are used to calculate the spectrum. Here we want a lot of samples to take advantage of most of the samples that are generated in the 10 seconds of data. This number needs to be set to a power of 2.



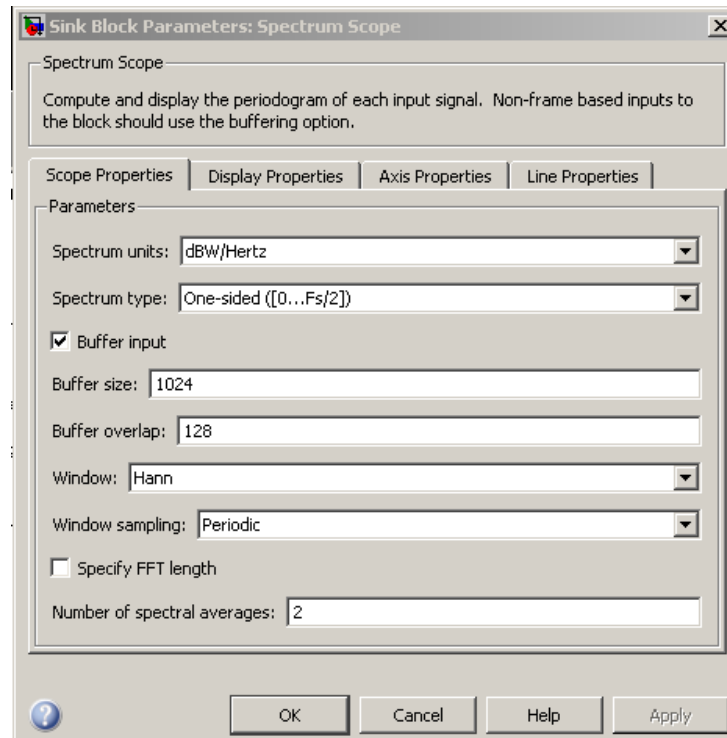


Figure 3. This is the property box of the Spectrum Analyzer

Now you are ready to start your simulation, choose Simulation-Start, you should see the time and frequency displays as shown below, play with the display options to see how you can adjust how the waveform and spectrum appears, zoom in and zoom out, find out what autoscale does etc. Zoom in to measure power levels. You can also set the default display and axis properties in the Spectrum Scope Block parameters window. You may need to set the maximum and minimum axis values to be able to see the whole trace.

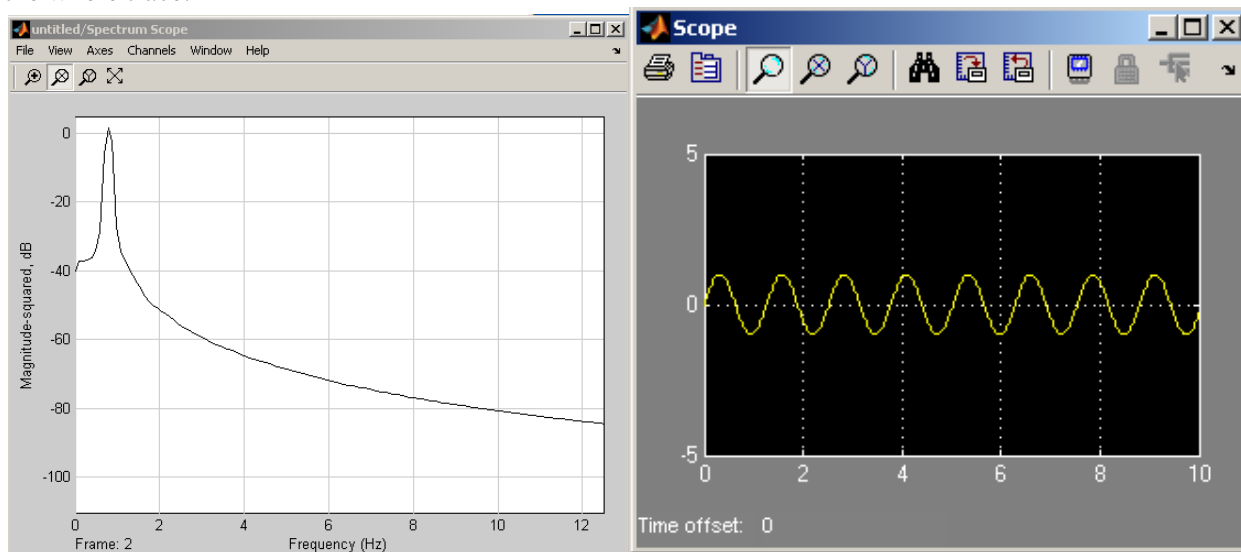


Figure 4 Power Spectra and waveform of the Sine wave.

Just like with the real instruments the spectrum is not just a line (delta function) at the frequency of the sine wave. The departure from this ideal is due to the finite amount of data and how fast you sample. If we samples infinitely fast and took data forever we would get a Delta function for the power spectral density.

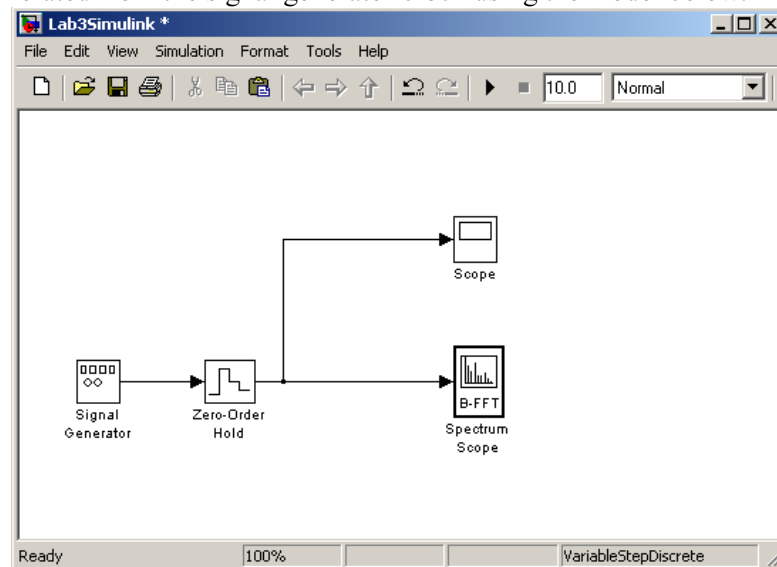
Change the sample time on the Sine wave generator so that it is 5 times longer. What happens?

The signal on the scope starts to look jaggy and the spectrum disappears. The spectrum can't be calculated any more because you now don't have 1024 points. Change the buffer size to 128 (always a power of 2) and the overlap to 32 and look at the spectrum, you will need to play with the display to get it to look right. Notice that now with fewer samples how the spectrum is less like a delta function with more power at high and low frequencies and how you have no data at frequencies much higher than your sample rate. Change the sample rate back to where it was and return the buffer and buffer overlap to where it was and explore what the effect of change of amplitude of the sine wave, the frequency of the sine wave, and spectrum units, and the spectrum type have on the wave form and spectra.

## Lab exercises

**Calculating the Power:** The power can be calculated by squaring the value of the voltage of the Waveform, the units don't work but we do this because in general we don't know the resistance of the system. Note: If the signal analyzer is chosen to display the single sided spectrum then the power in the spectrum is the total power, if the double sided spectrum is chosen then the power in the positive and negative frequencies must be added to get the total power.

Similar operations can be done for other waveforms – like the square wave, triangular and random noise. These signals can be generated from the signal generator block using the model below.



**Figure 5.**

The zero order hold is sampling the output of the signal generator and changing its continuous output to a set of discrete samples in time that the Spectrum Scope needs to compute the spectrum. It is located under Simulink/Discrete in the Simulink browser window. Wire up this model and set the parameters of the blocks as follows:

Signal generator: Wave form-Sine, time- use the simulation time, amplitude-1, frequency-2, units-Hertz and “interpret vector parameters as 1-d” should be checked.

Zero-Order Hold: Set the Sample time to .005 (How many samples will you have per period? How many total samples?) Will this give you a more accurate spectra than you obtained in the first example?

Spectrum Scope: Set the Spectrum units to dBW and the spectrum type to One-sided, set the buffer size to 1024 and the buffer overlap to 128.

### **Sine Wave**

Run the simulation. How much power do you measure at the spectra peak, zero frequency and 10 Hz?

Now change the Spectrum Scope to display a two sided spectra, how much power do you now measure in each peak. Does the total power measured by these three methods all agree?

### **Square Wave**

Change the Signal generator to produce a square wave. Measure and record the power in the fundamental, 3<sup>rd</sup> and fifth harmonic. What fraction of the total power (from the Scope trace) is in these first 3 harmonics?

### **Sawtooth**

Now change to a sawtooth waveform. What is the measured ratio of the power in the fundamental to that in the 3<sup>rd</sup> and 5<sup>th</sup> harmonics.

### **Random Noise**

Finally set the signal generator to generate a random waveform (white noise, “static” on your radio or TV). Estimate the average power that is displayed in the spectrum analyzer with the “number of spectral averages set to 1. Now set the number of spectral averages to 64 and remeasure the average power. By how much did the noise power drop?

Averaging the spectrum of a signal like a sine wave has little effect on the appearance beyond smoothing it out a bit as we saw in lab 1. Averaging the noise however drops the total power observed, you will learn why in lecture. Averaging is an important method of observing signals in the face of noise, the noise is reduced but the signal stays roughly the same thus increasing the ratio of the signal to the noise and making the signal easier to accurately hear, see, measure etc.