3C1 SIGNALS AND SYSTEMS LABORATORY[1]

Department of Electronic and Electrical Engineering

The aims of this laboratory are:

1. To determine the frequency response of a linear time-invariant (LTI) system
2. To examine the steady-state response and the transient response of an LTI system
3. To investigate the relationship between the values of poles and zeros of the transfer function and the response of the system

At the end of this laboratory you should have learned the answers to the following questions:

1. What does frequency and phase mean with respect to a sinusoidal signal?
2. What is the definition of an LTI system?
3. How does an LTI system alter the amplitude and phase of a sinusoidal signal?
4. What do the terms 'phase shift' and 'gain' mean?
5. How are the step and impulse response of system characterised?

The laboratory is punctuated by questions. The answers to these questions constitute your write up.

IN THIS LABORATORY HANDOUT SPECIFIC INSTRUCTIONS ARE INDICATED WITH THIS SYMBOL : □ . INSTRUCTIONS WHICH REQUIRE A RESPONSE IN YOUR WRITE UP ARE ENUMERATED. .

[1] This lab is an amended version of the lab created by Prof. Anil Kokaram and Dr. David Corrigan.

1 MATLAB review

Programs written in Matlab are stored in files called Matlab scripts. Matlab script files have the extension **.m** after the filename.

☐ Visit matlab.mathworks.com and log in with your account. Copy the file 3C1LabOnline.zip (available in the 3C1 laboratory folder on Blackboard) into the Matlab online current directory. Extract this file and type cd S1Online in the command window to change to the S1Online folder.

Matlab commands are run from the command window. This is a window in which you will see a prompt which look looks like this: >>. It is the only window that opens when Matlab is first run.

IN WHAT FOLLOWS, COMMANDS FOR TYPING INTO THE MATLAB COMMAND WINDOW ARE WRITTEN PREFIXED WITH >>. THIS DENOTES THAT YOU SHOULD WRITE WHAT FOLLOWS AFTER THE MATLAB PROMPT, WHICH LOOKS LIKE >>. AFTER TYPING A LINE YOU SHOULD HIT RETURN TO EXECUTE THE COMMAND.

☐ To see one of the basic Matlab commands in operation, create a matrix **A** as follows.

```
>> A = [1 5 ; 7 8]
```

The name to be given to the matrix is thus **A** and the numbers in the matrix are placed inside SQUARE brackets [ ]. Each row of the matrix is separated from the next one with a SEMI-COLON like this one – ; . Thus you have defined matrix **A** to have 2 rows (because there are 2 sets of numbers separated by a semi-colon) and 2 columns (because there are 2 numbers in each row).

If you did not type in the same number of numbers in each row, Matlab would complain.

☐ Now type in the command to create another matrix **B** such that

$$B = \begin{bmatrix} 2 & 3 \\ -3 & 2 \end{bmatrix} \tag{1}$$

☐ Check that Matlab holds B in memory by typing

```
>> B
```

and make sure it is as you expect.

☐ Do the same for **A**

Matlab can manipulate these matrices just as you would do for scalar quantities (i.e. single numbers, not matrices) if you were writing a C-program.

☐ Use Matlab to calculate **AB** by typing

```
>> A*B
```

☐ Write the answer here and verify that it is as you expect.

☐ Use Matlab to calculate the inverse of **A** by typing

```
>> inv(A)
```

☐ Verify that it is as you expect.

Note: During this laboratory, you may find that you have typed several things erroneously and your results look confusing. You can clear all the variables from memory by typing

```
>> clear
```

1.1 Plotting with Matlab

In Matlab, making graphical plots is done with the command `plot` which takes some arguments. Plotting takes place in the current figure window unless you specify otherwise. Matlab is a numerical package, therefore you cannot ask it to plot a function like $f(x) = x^2 + 6$ for instance without specifying a range and a number of points in x over which to evaluate the function $f(x)$. You will now do this for the function $x^2 + 6$.

☐ Create a vector of equally spaced points in x from -5 to 5 with a spacing of 0.1 as follows.

```
>> x = (-5 : 0.1: 5);
```

The semi-colon after the command prevents Matlab from printing out the result of the command. To see what Matlab does otherwise you could repeat the command but without the semi-colon.

☐ See what x is set to. (Type x by itself at the Matlab prompt).

You should see that it is set to a row vector having numbers from -5 to 5 incremented in steps of 0.1. You will observe many numbers and they may fill the screen.

☐ Now calculate the corresponding points in $y = x^2 + 6$ by typing

```
>> y = x.^2 + 6;
```

Now y is set to another row vector having the values of $x^2 + 6$ for each value in the vector x. The Matlab operator . causes the subsequent mathematical symbol to operate on each element of a matrix (here a vector) independently. Thus `x.^2` means calculate the square of each element in x if x is a vector or a matrix. Similarly $A. * B$ means multiply by each element in A by the corresponding element in B. Without the . then $A * B$ implies a matrix multiply, as you have found previously. If a constant is added to a vector (as in this case with +6) then Matlab assumes that that value is to be added to every element of the vector. This is the appropriate behaviour for our case here.

☐ Examine the first 5 elements of y and verify that they are what you expect. Do this with the command

```
>> y(1:5)
```

To examine elements from the 40th to the 45th you would type `y(40:45)`. Note that in Matlab the first element of a vector is $y(1)$ and NOT $y(0)$. $y(0)$ does not exist and Matlab will return an error if you try to display it.

☐ Use matlab to plot the graph of y vs x using:

```
>> figure(1);plot(x,y);title('My first PLot');xlabel('This is the x-axis');
>> ylabel('This is the y-axis');
```

This plots y on the vertical axis and x on the horizontal axis in figure window 1 and joins up the points plotted with a smooth line. Note the use of `title()`, `xlabel()` and `ylabel()` commands to annotate the plot and axes.

☐ Use matlab to plot a more informative graph of y vs x using:

```
>> figure(2);plot(x,y,'g-',x,y,'*r');
```

This plots the graph as a green line through the points `x,y,'g-'` (green = `g`, straight line = `-`), then it draws a second graph with just the points `x,y,'r*'` as asterixes (`*`) plotted in red (`r`).

You will also see that both figure 1 and figure 2 are available for you to compare the results of your two commands. Just drag the figure 2 window away from where it was generated on top of the figure 1 window.

There was no real reason for us to create a variable y to plot the graph. We could have also used:

```
>> plot(x,x.^2+6);
```

But that way you would learn less Matlab speak and so we did not do it that way. Any time you get stuck on the syntax for a command, you should use the help facility in Matlab. Just type `>> help plot` for instance to get a quick listing of help on the `plot` command.

Finally it is often the case that you might want to plot more than one line in a plot. Matlab allows you to do this by putting the arguments for each line in the same command. Try this.

```
>> figure(3); plot(x,y,'g-*',x,y+3,'r-+');
```

And you will see there are now two lines on the plot. The red line is $y + 3$ while the green line is y. Note how the points on the line are different for each line as specified in the `plot` function.

2 Signals

A signal may be deterministic or stochastic. Deterministic signals are described by 'well behaved' explicit functions. $x(t) = \sin \omega t$ is an example of a deterministic signal . The value of $x(t)$ at any time t is known exactly.

The state or value of a stochastic signal at any time t can only be determined with some 'probability'. So if $x(t)$ was a binary 'random' variable. One might say that it has value 1 with a probability 0.2 and value 0 with a probability 0.8, say. More formally a stochastic signal is generated by a random (or stochastic) process.

Most real world signals can be thought of as stochastic (eg. sound, images, text etc). The study of stochastic signals is central to Signal Processing as well as Telecommunications. However, the behaviour of deterministic signals is easier to analyse.

2.1 The Sine wave: a deterministic signal

1. Using Matlab, generate the deterministic signal $x_1(t) = 3 \sin 5t$ and plot it over the range $t = 0 : 6$ seconds. This is a sine wave. Label the x-axis as *Seconds* and the y-axis as *Volts*. Let us assume this signal is the measurement of the voltage from an AC power supply. Thus the plot you have made is the plot which you would see if you hooked up an oscilloscope to the terminals of the power supply to measure voltage.

2. What is the maximum and minimum value of the sinusoid (in Volts), the frequency of the sinusoid in Hertz, and the period of the wave in seconds?

   The angular frequency of the the sine wave is ω rad/sec. $\omega = 2\pi f$ where f is the frequency of the sinusoidal signal in Hertz.

   Now you should understand why sometimes you will have seen sine waves written like $\sin(\omega t)$ where ω is in rad/sec. It is because it is a more compact representation than always having to write $\sin(2\pi f t)$ where f is in Hertz. For example in the expression $y(t) = \sin(4.5t)$, the frequency of the sine wave is 4.5 rad/sec or $4.5/(2\pi)$ Hertz.

   Conversely if you want to write the expression for a sine wave having a frequency of 10 Hertz, you would write it as $\sin(2 \times \pi \times 10 \times t) = \sin(20\pi t)$. Also, if you have an expression like $\sin(4\pi t)$ then the frequency of the sine wave must be 2 Hertz.

   Finally, because $\sin(\ldots)$ varies between -1 and 1, the value of x_1 varies between ± 3 Volts.

3. Use Matlab to plot the graph of $x_2(t) = A \sin(\omega_1 t + \phi)$ with $A = 3$, $\omega_1 = 5$, and $\phi = -3$ (green colour), label the axes as previously and show x_1 (in red) on the same plot. Two lines should now be plotted on the graph. Include this figure in your write up.

   ☐ What is the frequency of x_2 in Hertz? What is the difference between the signals x_2 and x_1? Is there a phase lag? Is this a delay or an advance?

When signals are *phase lagged* with respect to each other it means that one is a time shifted version of the other. When a signal is shifted *to the right* on the time axis it has been *delayed*. You can see this from the expression for the signals because the phase difference would **negative**. Assume that what you are plotting is the output of two voltage sources from the time they were switched on since $t = 0$ secs. How long does it take from $t = 0$ for you to observe the same point in the signal from the two sources? Take the first positive zero-crossing for example (since this point is easy to spot). You will see that the +ve zero-crossings for $x1$ are at $0,\ 1.25,\ 2.5, \ldots$ secs, while the corresponding points on the $x2$ curve occur at about $0.6,\ 1.2,\ 1.8, \ldots$ secs. Thus the whole $x2$ signal is arriving *after* the corresponding points on the $x1$ signal. Thus $x2$ is *delayed* with respect to $x1$.

It turns out that it is much easier to see this phenomenon when you observe real i.e. stochastic signals since those sorts of signals are not periodic and so contain distinct shapes whose position is easy to spot.

4. By how much is x_2 delayed (in seconds) with respect to x_1? Clearly indicate which points you used to calculate the the delay. How does this value relate to the constant offset term, ϕ in the argument for the sin function in x_2? Show exactly how ϕ can be used to calculate the phase lag in *seconds*. (Hint: ϕ has units of radians.)

5. Use Matlab to plot the graph of $x_3(t) = A \sin(\omega_1 t + \phi)$ with $\omega_1 = 5,\ \phi = -3 + 2\pi,\ A = 3$. Plot the function in Fig. 3 superimposed on the graph for x_1 (as in previous instructions). By how much is x_3 delayed (in seconds) with respect to x_1? Use measurement from the displayed graph only. How does this value relate to the constant offset term in the argument for the sin function in x_3? Explain any difference or similarity with x_2 in terms of the properties of the sine function.

3 Linear Time Invariant Systems

In this section you will examine some fundamental properties of Linear Time Invariant (LTI) Systems using a simple GUI (Graphical User Interface) called `lab1` which was written for this experiment using the Matlab GUIDE tool.

☐ Run the GUI by typing $>> lab1$. In the rest of this document this GUI will be referred to as LAB1.

You will now see a window resembling that shown in Figure 1.

☐ Click on the $\boxed{\text{Sine}}$ radio button on the lower left hand side of the GUI to start the experiment.

The GUI represents the action of putting a signal (shown as an oscilloscope trace or plot on the left hand plot) through a system (there are 4 possible user-selectable systems shown in the centre of the window). The right hand plot window shows the output signal which results. You can change the time scale for viewing by changing the `Tmax, Tmin` values on the lower right hand corner. You can also select several different input signals, and set their amplitudes and frequencies on the lower left hand set of option buttons and edit boxes. The radio button in the centre of the window allows you to superimpose the input signal on the same plot window in which the output signal is displayed so you can make comparisons more easily. Also note that if you place the mouse pointer over the output plot and press and hold down the left mouse button, the coordinates of the points under the mouse pointer are displayed. This will help you to make readings more accurately. You can also drag the pointer across the signals in the output box and the coordinates of the corresponding points will also be displayed.

The default system used in LAB1 is `SYSTEM1`. This system is LTI. You will now examine some of its properties.

☐ Set the time axis to approximately $0 \leftrightarrow 1.5$ secs.

1. Examine the input and and output signals corresponding to the default settings of LAB1. The input signal is a Sine wave. What is the output signal? (Is it a Sine wave or a Cosine wave or something else?)

2. What is the difference between the input and output signals using system 1? What is the same? (Superimpose the input on the output to see this more clearly.)

3. Increase the amplitude of the input signal by a factor of 2. What is the corresponding increase in the amplitude of the output signal, after initial transients have decayed? How long does the system take to settle into a steady state response?

4. For input amplitudes of 0.5, 1.0, 1.5, 2.0, 2.5, measure the corresponding output amplitudes in the output signal and plot a graph of i/p amplitude vs output. Label your axes carefully and include the plot in your write up.

5. Switch the input signal to the Pulse Wave and set its frequency to 0.8 Hz. and repeat the above experiment. Take the output amplitude to be the maximum value of the output pulse after initial transients have decayed.

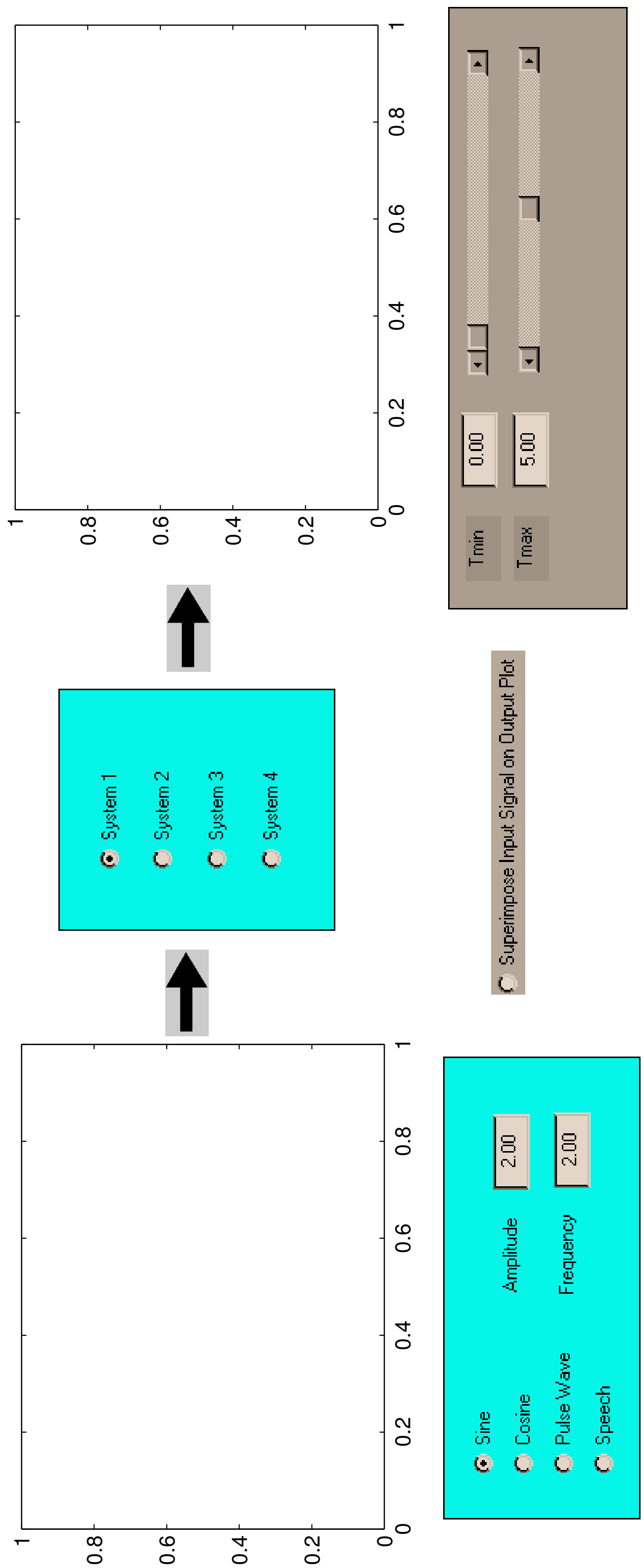

Figure 1: The GUI for LTI systems.

You have just observed a property of LTI systems, scaling the input signal by a factor of A, causes a similar scaling in the output. Let $x_1(t)$ and $y_1(t)$ denote the input and output signals respectively.

$$\begin{aligned} \text{IF } x_1(t) &\rightarrow y_1(t) \\ \text{THEN } ax_1(t) &\rightarrow \boxed{ay_1(t)} \end{aligned}$$

where a is *any* complex constant. (i.e. a can be either real or complex).

☐ Change the time axis to observe some other window in time after 5 secs, say. Is there any difference in the output signal?

This last experiment has verified (rather loosely) the property of time invariance (the **TI** bit in L**TI**). Essentially, time invariance implies that the behaviour of the system remains the same over time.

$$\begin{aligned} \text{IF } x_1(t) &\rightarrow y_1(t) \\ \text{THEN } x_1(t-\tau) &\rightarrow \boxed{y_1(t-\tau)} \end{aligned}$$

The final property of LTI systems is one which LAB1 is not set up to verify, but it will be needed later, so it is stated here. A Linear system possesses the important property of *superposition*. The response to a weighted sum (superposition) of several inputs is a weighted sum of the responses to each of the inputs.

$$\begin{aligned} \text{IF } x_1(t) &\rightarrow y_1(t) \\ \text{AND } x_2(t) &\rightarrow y_2(t) \\ \text{THEN } x_1(t)+x_2(t) &\rightarrow \boxed{y_1(t)+y_2(t)} \end{aligned}$$

6. Select the Cosine as the input signal and set the frequency to 2Hz. What is the difference between the input and output signals using system 1? What is the same?

7. For both the Sine and Cosine input signals (at Frequency 2Hz and amplitude 2) with system 1, write mathematical expressions for the input and output signals using LAB1 to help you make measurements.

   This final set of experiments has shown (albeit indirectly) another important property of LTI systems. Complex exponential functions are *eigenfunctions* of LTI systems. **The output signal has the same frequency as the input** although changed in phase and amplitude.

8. Using what you now know about LTI systems, classify Systems 2,3,4 as LTI or non–LTI. Remember to check input amplitudes covering a wide range (e.g. $1 \leftrightarrow 6$).

3.1 Gain and Phase as a function of frequency

1. In your report, complete the table below for the Sine input at various frequencies using system 1 (Note the units!)

INPUT			OUTPUT				
Frequency (rad/sec)	Frequency (Hz)	Amplitude (x)	Frequency (rad/sec)	Amplitude (y)	Phase Lag (sec)	Phase Lag (rad)	Gain (y/x)
1		1	1				
5		1	5				
8		1					
10		1	10				
12		1					
15		1					
20		1					
30		1	30				
50		1					
100		1	100				

2. Plot a graph of Gain vs Frequency (rad/sec) and Phase (rad) vs Frequency (rad/sec) for system 1 using the information above. Label axes carefully.
3. Is the effect of the system the same at all frequencies? How does the system behaviour change with frequency?
4. Discuss the significance of this plot with respect to the effect of system 1 on the Pulse Wave and the Speech signal.

3.2 Characterising the Transient Responses of a System

In the last section you saw how the output of a system can be broken down into a transient and steady state response (If you are not sure what this means ask a demonstrator). You also learned how to characterise the steady state respone to a sinusoidal input. In this section, the basic concepts in the characterisation of the transient response are introduced. A focus is placed on the characterisation of the transient response to a step input signal.

Consider the following system (Fig. 2).

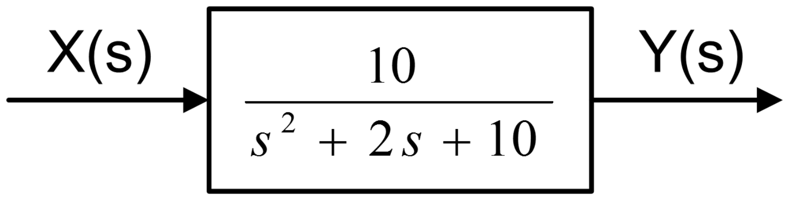

Figure 2: A 2^{nd} order LTI System.

This system can be created in Matlab using the `tf()` function (in fact the LTI systems in the lab1 GUI were created in this way). It takes two arrays as input parameters. The first is an array that contains the coefficients of the numerator polynomial in order of descending power and the second contains the coefficients of the denominator polynomial.

For example, a system with a transfer function $1/(s+3)$ can be created by the line

```
>> the_sys = tf([1], [1 3]);
```

☐ Using the tf function, create a transfer function object that represents the system shown in Figure 2. Verify your answer by omitting the semi-colon at the end of the line used to create the system (this will cause Matlab to print the transfer function on screen).

`the_sys` is not an array or a numeric type but is instead an object of the transfer function class. The object can be used to simulate the step and impulse responses of the system using the `step` and `impulse` functions respectively.

1. Create two new figures. In the first generate the impulse response and in the second plot the step response. Sketch the plots in your report and label your axes carefully.

   The response of a system to a given input signal consists of the transient response and the steady-state response. The transient response describes the initial response of the system to the input signal, while the steady state response describes the behaviour of the output signal as time approaches infinity.

   Higher Order Systems or systems whose transfer functions are unknown can be compared by measuring parameters from their step and impulse responses. Some of the important parameters are

Impulse Response

- peak time - the time at which the peak value of the output signal occurs.
- peak value - the maximum absolute value.
- settling time - the time after which the impulse response is entirely bounded by a user-defined threshold. (eg. ± 0.1). It is effectively the time at which the signal transitions from the transient to steady state behaviours.

Step Response

- steady state value - the value of the output signal as $t \to \infty$.
- rise time - the time taken for the output to go from 10% to 90% of the steady state value.
- % overshoot - $\frac{\text{peak value} - \text{steady state value}}{\text{steady state value}} \times 100$.
- settling time - the time taken for the signal to be entirely bounded to within a tolerance of the steady state value.

2. Using the plots generated previously, record values for all of the parameters listed above.

   Often systems are specified in terms of the poles and zeros of the transfer function of the system, $H(s)$. The poles are the roots of the denominator polynomial of the transfer function. The zeros are the roots of the numerator polynomial of the transfer function. The location of the poles and zeros governs the behaviour of a linear time-invariant system.

3. Determine the poles and zeros of the transfer function shown in Fig. 2.

   Matlab can specify systems in terms of poles and zeros using the `zpk()` function. The first two inputs are arrays specifying the zeros and poles (in any order). The final parameter specifies a gain value for the system. For example if a system has the transfer function

$$H(s) = \frac{10(s+3)}{(s+1)(s+2)},$$

then the gain is 10 and there are poles at $s = -1,\ -2$ and a zero at $s = -3$. The correct call is then

```
>> the_sys = zpk([-3], [-1 -2], 10);
```

☐ use the `zpk()` function to specify a transfer system for the system shown in Fig. 2 and verify your result by comparing the step or impulse response with the ones you generated earlier. Note if there are no zeros an empty array must be passed to `zpk`. An empty array can be created using the following syntax:

```
>> A = [];
```

4. Create four 2^{nd} order systems, each having a gain of 1, no zeros in the transfer function, and with the poles specified below:

   - System 1 - poles at $s = -1 + j,\ s = -1 - j$
   - System 2 - poles at $s = -0.5 - j,\ s = -0.5 + j$
   - System 3 - poles at $s = -0.5 + 3j,\ s = -0.5 - 3j$
   - System 4 - poles at $s = -1,\ s = -0.5$

   Using the first three systems investigate the effect the position of the poles has on the transient response of a system to a unit-step input $u(t)$. Consider only the frequency of oscillation and settling time of the output in your answer. What are the general differences between response of the system that only has real poles compared to the response of the other three systems which have pairs of complex conjugate poles?

5. Using System 1 above investigate what happens to the step response of a system when a zero is added at $s = -1$.

4 The Write Up

In your report you should provide a response to each of the enumerated questions asked during the laboratory. Include plots from Matlab as directed or where you think it assists your arguments.