Lab 0: Introduction to TIMS AND MATLAB

1. INTRODUCTION

The TIMS (Telecommunication Instructional Modelling System) system was first developed by Tim Hooper, then a senior lecturer at The University of New South Wales in early 1970s. It is now a very advanced system for telecommunications training (http://www.tims.com.au/main.htm). TIMS is a rack and module system, in which each module performs a basic communication or signal processing function, eg: oscillator, multiplier, filter, etc. Modules can be plugged into each of the twelve slots in the rack and then connected with other modules to create a variety of systems. Several permanent modules are located in the lower part of the rack. It provides for students an easy way of modelling communication and signal processing systems in the laboratory.

In addition to the TIMS system, each experimental bay contains a personal computer (PC) and oscilloscope. The oscilloscope is connected to the TIMS system Scope Selector module, a permanent module, by BNC terminated cables. The Scope Selector module enables any two from four inputs to be viewed simultaneously on the oscilloscope. The PC provides measurement and signal processing functions. The Analog to Digital converter interface between the PC and TIMS is a Soundblaster ASP16 system. The Soundblaster and TIMS are connected by the stereo audio cable terminated with RCA plugs. For stereo recording the RCA plugs should be connected to the sockets at the X and Y inputs of the oscilloscope, but for mono recording only the red RCA plug should be connected. Only mono recording is required for this laboratory.

Data acquisition and processing software is provided by MATLAB. MATLAB is a useful software package designed for scientific and technical computation. It is a convenient numeric computation and visualization tool that may be used for a wide range of applications in engineering and science. MATLAB may also provide several tool boxes (additional software packages) specially developed for telecommunication, signal processing, control, applied mathematics, image processing etc. You may easily extend and customize the capabilities of MATLAB by writing your own functions and procedures.

This laboratory is intended to familiarize you with MATLAB and TIMS, but the experiments also illustrate some signal processing concepts and methods. Both facilities will be new to most of you, but before you graduate you can expect to use them for several core and elective subjects in the signal processing, control and communication area. The laboratory introduces some basic TIMS building blocks: adder, multiplier, filter etc, and connects up a few simple systems.

NB: The School has a classroom Teaching Kit licence for the MATLAB software which runs under Microsoft Windows. This permits students enrolled in this subject to obtain a copy of the software for home use provided the licence agreement is signed. See the Laboratory Manager, Mr Noel Kaarsberg in Room 402 for details.
Some Rules for Connecting TIMS Modules

For all the modules used in the TIMS system the following rules apply:

<table>
<thead>
<tr>
<th>Left hand socket</th>
<th>input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Right hand socket</td>
<td>output</td>
</tr>
<tr>
<td>Red socket</td>
<td>TTL level (0 and 5 volts).</td>
</tr>
<tr>
<td>Yellow socket</td>
<td>analog level (nominally 4 volts peak to peak, maybe larger)</td>
</tr>
<tr>
<td>Green socket</td>
<td>common, ground or earth level.</td>
</tr>
</tbody>
</table>

DO NOT CONNECT YELLOW OUTPUTS TO RED INPUTS!
2. PREPARATION

NB: It is important that the preliminary work for each experiment is done before you come to the laboratory. It will be marked at the beginning of each laboratory period.

P1. Using Fourier transform tables or otherwise, write down the Fourier transform of the sinusoidal signal,

\[ x(t) = 1 + A \sin \omega_0 t, \quad \text{all } t, \]

where \( \omega_0 = 2\pi f_0 \) is the angular frequency and \( A \) is a constant.

Sketch \( x(t) \) and the magnitude and phase of its frequency spectrum.

P2. Using simple trigonometry (or otherwise) write down the Fourier series expansion/representation of the periodic signal,

\[ y(t) = x^2(t), \]

where \( x(t) \) is defined in question P1.

Sketch \( y(t) \) and the magnitude and phase of its frequency spectrum.

Explain the spectrum symmetry of signals: \( x(t) \) and \( y(t) \).

P3. A nonideal lowpass filter has a frequency response specified by,

\[ H(f) = \frac{1}{(1 + jf / f_c)^2} \]

where \( f_c \) is the filter cutoff frequency.

Sketch \( |H(f)| \) as a function of frequency.

Using Fourier transform tables write down and sketch the filter impulse response,

\[ h(t) = F^{-1}\{H(f)\} \]
3. EXPERIMENTS

3.1 Sinusoidal Signals and Frequency Measurement

Refer to the TIMS Laboratory Manual for details about the Audio Oscillator, Frequency Counter and Scope Selector modules. Connect the Audio Oscillator to the oscilloscope via the Scope Selector and use the Frequency Counter to set the signal frequency to about 3 kHz. Observe the waveform on the oscilloscope and also note the effect of the Gate Time of the Frequency Counter measurement.

a) Note the Frequency Counter reading.

b) Estimate the signal frequency from the oscilloscope. How well does this measurement agree with that in Part a)?

c) Use the PC with the following MATLAB commands to record and display 400 samples of the signal sampled at 44,100 Hz (and quantized at 16 bits per sample),

\[
\begin{align*}
x &= \text{getrec}(400, 44100, 16) \quad \{\text{acquire the data}\} \\
t &= (0:399)/44.1; \quad \{\text{set the time scale}\} \\
\text{plot}(t, x), \text{grid} \quad \{\text{plot with grid}\}
\end{align*}
\]

Note: *On–line information is available for MATLAB functions; simply type help function_name, for example help getrec. Also see Appendix 1 of this Laboratory experiment.*

The amplitude (or magnitude) of the frequency spectrum of the signal record \(x\) is computed and displayed using the MATLAB command,

\[
\text{ampspec}(x, 44100);
\]

Sketch this amplitude spectrum and compare it with the theoretical frequency spectrum of a sinusoidal signal (of infinite duration).

3.2 A Simple System: Signal Cancellation or Nulling

Connect up the TIMS modules for the system shown below. (Refer to the TIMS Manual for module details).

Set the Audio Oscillator frequency to about 2 kHz and

1. Observe the inputs of the Adder on the oscilloscope.
2. Show that the phase of the upper signal (cosωt) and the amplitude of the lower signal can be adjusted so that the output from the Adder can be reduced (almost) to zero, ie nulled.

3. Observe, record and think
   a) Are the amplitude and phase adjustments in Part 2. independent (ie: not interactive)?
   b) What minimum voltage level at the Adder output can you achieve?
   c) Is the null adjustment upset if the frequency of the Audio Oscillator is changed? If so, why?

3.3 Multiplier Connected as a Squarer

Connect up the Audio Oscillator and Multiplier modules as shown below.

![Diagram of Audio Oscillator and Multiplier](image)

Set the Audio Oscillator frequency to about 2 kHz and
   a) Observe and sketch the Multiplier input and output signals on the oscilloscope which should be dc coupled.
   b) Write down an algebraic expression for the multiplier output. Is the observed output consistent with this?
   c) The TIMS Manual indicates that for input signals x(t) and y(t), the Multiplier output signal is
      \[ z(t) = k \ x(t) \ y(t) \]
      where k is the multiplier scaling parameter or factor.
      Determine the value of k including its units.
   d) Use the PC with the following MATLAB commands to obtain a sample of the multiplier output signal and display its amplitude spectrum.
      \[ x = \text{getrec} (400, 44100, 16); \]
      \[ \text{ampspec} (x, 44100); \]
   e) Sketch the amplitude spectrum signal. Is it consistent with the results from Part a)? If not, why not?
3.4 Tuneable Lowpass Filter

Read the TIMS Manual entry for the Tunable Lowpass Filter and then connect up the modules shown in the block diagram below.

Choose the Audio Oscillator TTL (red) output and set the frequency to obtain a square wave signal with a fundamental frequency of approximately 1 kHz. Use the Adder module to reduce the square wave amplitude to about 2 volts peak–peak. Set the Tuneable Lowpass Filter cutoff frequency to 1.5 kHz (use the Frequency Counter to set the LPF clock frequency).

a) Observe the filter input and output signals on the oscilloscope and sketch their waveforms.

b) Use the PC with the following MATLAB commands to obtain samples of the filter input and output signals (separately) and display their amplitude spectra.

\[
x = \text{getrec}(400, 44100, 16);
ampspec(x, 44100);
\]

c) Sketch the amplitude spectra of the square wave and the filtered signal. How do they compare with (theoretically) expected results?

d) Repeat the `getrec` and `ampspec` commands with 2048 points or samples. Does the frequency spectrum look different at all?
4. APPENDIX

Note on Signal Acquisition and Spectrum Analysis using a PC with MATLAB Software

Signal acquisition and spectrum analysis will be carried out digitally by means of an analog–to–digital converter and a personal computer (PC) with the MATLAB software package. Analog–to-digital conversion is provided by a Soundblaster 16 ASP card in the PC. The discrete Fourier transform (DFT) of the sampled signal is computed by means of a MATLAB Fast Fourier Transform (FFT) algorithm. The sampled signal and the amplitude spectrum can be processed and displayed using the appropriate MATLAB commands.

1. Signal Acquisition

Use the MATLAB function `getrec` to obtain a sampled record of a signal. eg: The following MATLAB statement will cause the Soundblaster to acquire 1024 samples of a signal at a sampling rate of 44100 samples per second with 16 bits per sample,

\[ x = \text{getrec} (1024, 44100, 16); \]

The result is stored in the vector \( x \).

To display the sampled signal \( x \) versus a time scale \( t \) in milliseconds use the following MATLAB commands (Note that the number of points plotted can be varied and is 100 in the present case).

\[ t = (0:\text{length}(x) - 1)/44.1; \]
\[ \text{plot} (t (1:100), x(1:100)), \text{grid} \]

2. Spectrum

The MATLAB function `ampspec` computes the amplitude spectrum of the signal \( x \) and displays the result,

\[ X_{\text{amp}} = \text{ampspec} (x, 44100); \]

where the vector \( X_{\text{amp}} \) contains the amplitude spectrum values. The function `ampspec` uses a fast Fourier transform algorithm, the MATLAB function `fft`, to compute the discrete Fourier transform (DFT) of the data vector \( x \). The computation is faster if the length of \( x \) is a power of 2; alternatively the data can be padded with zeros to make the length a power of two. For example: if \( N = \text{length} (x) = 1000 \), we might write \( \text{fft}(x, 1024) \) instead of \( \text{fft}(x) \). The difference in computation time is significant for longer vectors eg: \( N = 10^4 \) or greater.

3. Average Powers

The average power \( P_{av} \) of the signal \( x \) can be estimated from the amplitude spectrum using the command

\[ P_{av} = 2 * \text{sum}((X_{\text{amp}} * F_s / N .^2) \]

where \( N = \text{length} (x) \) is the number of samples in the signal record vector \( x \) and \( F_s \) is the sample rate in Hz.
The average dc and ac power, (in a 1 ohm resistor) \( P_{dc} \) and \( P_{ac} \), of the signal \( x \) can also be estimated in the time domain from the mean and variance of \( x \) using the appropriate MATLAB commands:

\[
P_{dc} = \text{mean}(x)^2 \quad (**)
\]

\[
P_{ac} = \text{std}(x)^2
\]

** Note: \( P_{dc} \) cannot be measured when the data is acquired using the Soundblaster because it is AC coupled.

