EGB242 Assignment 2 (25%)

Orbiting the Red Planet

Released: Friday, 20th of September at 11:59pm (Week 9)
Group Plan Due: Friday, 27th of September at 11:59pm (Week 10)
Final Assignment Due: Friday, 18th of October at 11:59pm (Week 12)

Message from MARS-242 Mission Control

The chief engineers at BASA Headquarters are confident in your demonstrated skills dealing with the simulated Mars mission scenarios you have worked on so far. You are now ready to be part of the main engineering team supervising the 'live' MARS-242 mission.
Monitoring the Astronauts' physical and psychological health will continue to be critical to the success of this first mission to Mars. Furthermore, as the spaceship is now orbiting Mars, identifying an appropriate landing site for the landing module is also a pressing issue. Fuel reserves on-board the spaceship and the effects of planetary alignment require that this process be carried out efficiently and precisely.
You are now called upon to assist with achieving these two high priority objectives. Your contributions will strongly depend on the techniques you have practiced in the simulated tasks code-named Assignments 1A and 1B.

Section 0 - Preparation

Read the entire document before attempting the tasks

There is important information in the end of this assignment brief regarding how the assignment is to be presented.

A unique set of data is provided to you based on the student members of your group. You will need to generate this data in order to complete the assignment.

Extract the files from `EGB242_Assignment2.zip`
Open and read `GenerateAssignment2Data.m`
Enter your group members' student numbers in the appropriate location.
If your group has fewer than 3 members, enter 0 for the missing member(s).
Ensure all necessary files are in the same folder and run the script.
Two (2) files will be generated that you will use for this assignment - `A2P2Data.mat` and `A2P3Data.mat`.

Two (2) additional template files, `A2part2.m` and `A2part3.m`, have been provided to you, corresponding to sections 2 and 3. You are to write all code for this assignment for each section in the respective file.

Section 1 - Mars-242: Mars Rover

The maths for this section can be either typed or handwritten neatly.

In anticipation for landing on Mars soon, a rover has been gathering and collecting important data. The rover has been communicating directly with stations situated on Earth where BASA Engineers conduct their analysis on the received data and send the results to the astronauts for action.
Unfortunately, a malfunction with the communication system on the rover has meant that this very long distance communication is no longer possible. Because the rover's data consists of information and images of the planet, which are vital for the success of the anticipated Mars landing, a solution has to be found. The best alternative is for the rover to send the data directly to the spaceship and for the astronauts do the analysis themselves.
However, this can only succeed with your help and knowledge of how Time Invariant systems work. You are called upon to assist the astronauts to mathematically model a system that allows reception and processing of the rover data.

Figure 1: A Mars Rover (Source: NASA)

Your task is to mathematically determine how communication between the spaceship and the rover can occur.

1.1 The spectrum of the rover's transmitted signals may interfere with the spectrum of the Spaceship's own communication module. The Spaceship's communication channel can be modelled as a constant magnitude function in the frequency domain, between $-f_{bw}$ and $+f_{bw}$. The f_{bw} for your group will be printed out when you generate your data. Mathematically express and sketch this function. (CR1a)

1.2 The rover's transmission function can be expressed as: $s_{\text{rov}}(t) = (-1)^n A \,\text{sinc}(6000t) \sin(10\pi \times 10^3 t - m\frac{\pi}{2})$. The magnitude variable A, n and m are printed when you generate your data. Derive an expression for the corresponding representation of this function in the frequency domain, $S_{\text{rov}}(f)$, and sketch the magnitude spectrum. (CR1a, 1c)

1.3 Determine the Transfer Function of the ideal filter, $H_{\text{filt}}(f)$, that is required to eliminate the spaceship's broadcast information and ensure the rover's signal is received. Additionally, name the filter type. Ensure that you select the simplest type of filter for this situation and justify your selection. (CR1a,1d)

1.4 Evaluate and sketch the final frequency domain function that defines the range of frequencies you will receive from the rover after applying your filter above. (CR1a,1d)

1.5 Discuss what impacts applying the filter could have on receiving the rover's information. How might you mitigate these issues? (CR1d)

Section 2 - Mars-242: EEG signal analysis

All code for section 2 should be entered into the file A2part2.m.
Make sure to include appropriate scaling and informative axis labels.

Despite the crew's outwardly high morale, an analysis of the astronauts' song choices, which you have helped extract from your work in Assignment 1B, pointed to a possible problem. This has prompted BASA psychologists to recommend more quantitative tests of their brain function. Biometric data has been collected in the form of Electroencephalograms (EEG) via sensors in their spacesuits. The EEG signals have been multiplexed at the spaceship using the Fourier Transform frequency shift property. The result is a Frequency Division Multiplexed (FDM) data stream transmitted to mission command, similar to the ones you have successfully dealt with in your previous assignment.

> *Your task is to extract the EEG signals of each individual astronaut, remove any noise or interference effects from them, and comment on any abnormalities you might be able to observe.*

Your expertise in examining the EEG signals to identify any abnormalities will assist our BASA medical team with diagnosis as well as provide data for research into human physiology in microgravity environments.

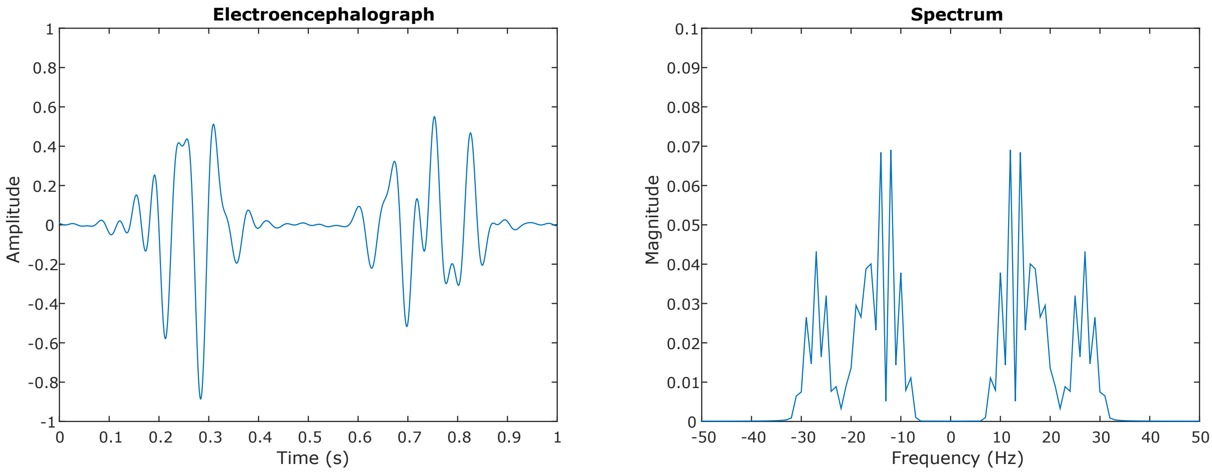


Figure 2: Electroencephalogram signal - (*Left*: Time domain, *Right*: Frequency domain)

An example of an EEG signal in the time domain, with its corresponding spectrum, is shown in Fig. 2. Notice that most of the energy in the signal is concentrated between ±30 Hz and that there is very little spectral leakage outside of that range. We can say that this particular signal is band limited and that it has a bandwidth of $B = 30$ Hz.

The multiplexed signal is normally received at command by a modular hardware system which consists of receiver (with real-time spectrum analyzer) followed by a frequency shifting module and analogue filter. The output of this filter is sampled by an analogue-to-digital converter for further manipulation. The block diagram is shown below in Fig. 3.

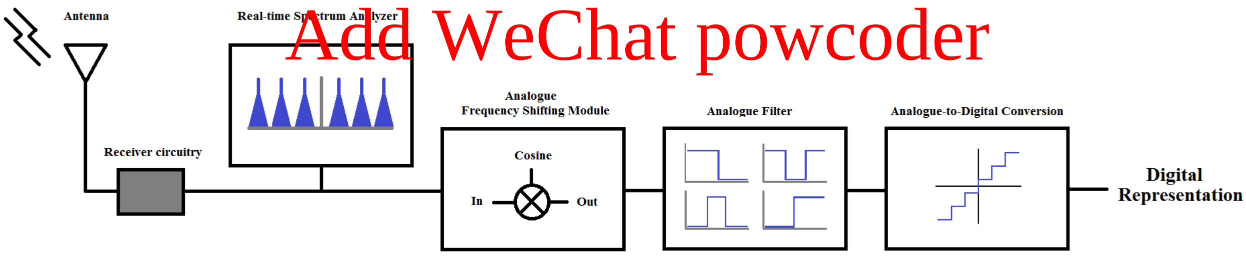


Figure 3: Block diagram of the modular receiver

2.1 *Represent the spectrum analyzer* - Determine the sampling period, `Ts`, and construct a time domain vector, `t`, that corresponds to `muxSignal`. Plot `muxSignal` against this time vector. Compute the Fourier transform of `muxSignal`, `MUX`, and construct a suitable frequency vector, `k`. Plot the magnitude spectrum of `MUX`. (CR 1a, 2a, 2c)

2.2 *Determine demultiplexing parameters* - Identify the frequency shifts and store them (in an ascending order) in row vector `fshift`. Find the corresponding magnitude and phase and store in row vectors `Mag` and `Phishift` respectively. *Note:* Remember to scale `Mag` to account for the sampling rate. (CR 1b, 2b)

2.3 *Remove the frequency shifts for all five astronauts* - Input variables `muxSignal`, `t`, `Mag`, `fshift`, and `Phishift` should be inserted as row vectors. Each row of the `xdm` matrix will contain the EEG data for one astronaut with frequency shift removed upon successful application of the module. (CR 1b, 1d)

The frequency shifting module is provided in `FDMDemux.p`. This is mathematically equivalent to the hardware module being used and shares the same functionality as `FDMDemux.m` from Assignment 1B.
The syntax for its usage is

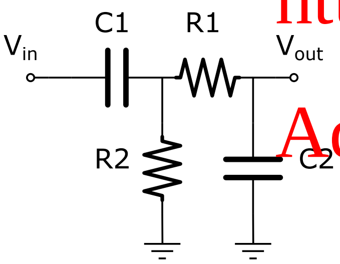
```
xdm = FDMDemux(muxSignal,t,Mag,fshift,Phishift)
```

2.4 *Review* - Compute the Fourier transform for each data stream in *`xdm`*. Store the result in the matrix *`XDM`*. Plot the magnitude spectrum for each data stream. (CR 1a, 2a, 2b)

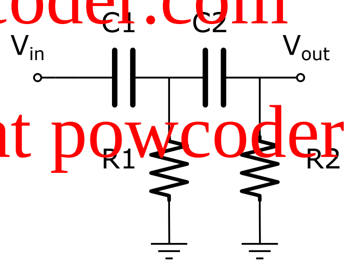
At this time, the filter module is non-operational and your expertise in using the Laplace Transform for circuit analysis will be required to select and implement an appropriate filter. Circuit diagrams from the unit's technical documents have been supplied below along with their corresponding transfer functions. You must analyse the response of the circuits to determine which filter is the most appropriate for this application. Once your signal has been filtered you should remove any additional unwanted frequency components and proceed with your analysis.

You have been given the four most appropriate filters out of the six shown below. The transfer functions for these four systems are stored in *`sys`*, with the corresponding circuit diagram reference in *`sysInfo`*. Each row of the matrix is the description for each of the systems. To display the part name and parameters for the first system, type *`sysInfo(1,:)`*.

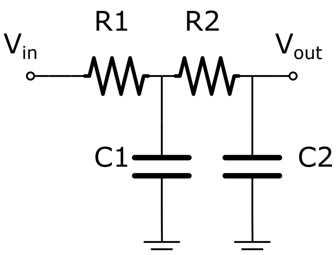
PTPSV01



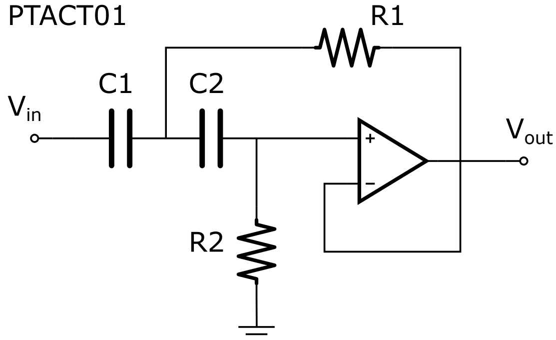
PTPSV02

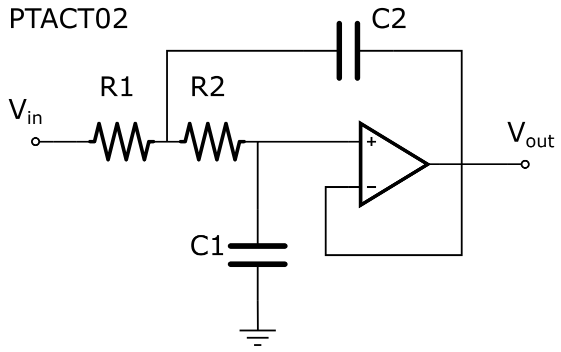


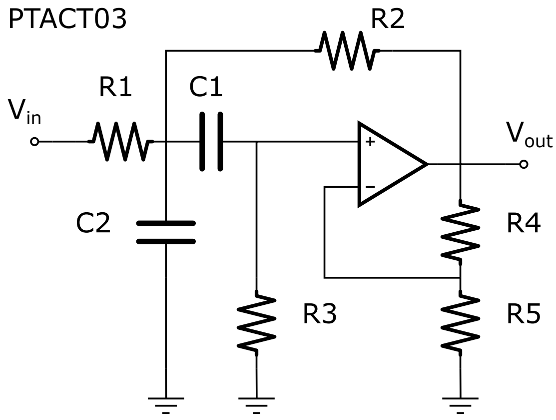
PTPSV03



PTACT01







2.5 *Mathematical analysis* - Display the values of each transfer function in MATLAB. Determine the suitability of each system (for the task) based solely on its numerator and denominator polynomials. Justify your choices. The function `factorTF.p` has been provided to perform factorisation of the polynomials if needed. Use it by passing a transfer function to it, *eg.* `factorTF(sys(1))`. (CR 1a, 1b, 1d)

2.6 *System Analysis* - Analyse each system, to help you understand their behaviour. You may use the LTI viewer or other functionality in MATLAB to do this. You should find information such as the: impulse response, step response, bode plot and pole zero map. (CR 1a, 1c, 1d, 2b)

2.7 *Recommend* - Make your final filter selection and provide a short summary (between 50 and 100 words) justifying your choice based on your mathematical and system analysis performed previously. Describe any implications that may come from using the chosen type of filter. (CR 1d, 2b)

2.8 *Filter the signals* - Use the system you have selected to extract the EEG signal for each of the astronauts. The same filter parameters can be used for each signal. Do this by first multiplying the frequency response of the system by each data stream in the frequency domain (`XDM`). Store the result in the matrix `EEG`.
Convert these back to the time domain and store the result in a matrix called `eeg`.
Each row of both `eeg` and `EEG` should contain the time domain and frequency domain of the EEG signal for one astronaut, respectively. Organise the rows so that they correspond to each frequency in `fshift`. Plot the signals for each of the astronauts in both the time and frequency domains. (CR 1b, 1c, 1d)

2.9 *Equivalence with convolution* - Convolution in the time domain is equivalent to multiplication in the frequency domain (and vice versa). Demonstrate this by convolving the impulse response (`imp`) of your chosen system (you can use the impulse function to calculate this) with `xdm(1,:)` and saving this to variable `eqConv`. Compare this to the result obtained in 2.8 of the same signal (`eeg(1,:)`). Comment on and explain any differences you see. (CR 1b, 1c, 2b)

2.10 *Compare* - Compare the time domain signals and spectra with the example in Fig. 2. Outline any differences, if any, and provide your justification as to why they should be the same or different. (CR 1b, 1d)

Now that the demultiplexer has been fixed, the EEG can now be received and fed into BASA Headquarters' computer network.

2.11 *Digital de-noising* - It appears that some of the signals have been corrupted by single-frequency interference noise. Implement code to remove this noise. Explain what you have done and provide justifications. (CR 1b, 1d, 2b)

2.12 *Visual analysis* - Compare the EEG signals of the astronauts to each other (as well as to the example in Fig 2) and state for each astronaut whether you have identified any variations that may signify a suboptimal mental state. This information will be provided to BASA's medical team for their professional assessment and prescription of any remedial action. (CR 1a, 1d)

MATLAB variables that should be included in your workspace for section 1 (A2part2.m),

Ts - Sampling period
t - Time domain vector
MUX - Frequency domain representation of mux
k - Frequency vector
fshift - Frequency shifts
Mag - Magnitude shifts
Phishift - Phase shifts
xdm - EEG data
XDM - Frequency domain representation of xdm
freqResponse - Frequency response of systems
imp - Impulse response of chosen system
EEG - Frequency domain representation of Filtered signals
eeg - Time domain of filtered signals
eqConv - Convolution equivalence

Section 3 - Mars-242: Choosing a landing site

All code for section 3 should be entered into the file A2part3.m.
Make sure to include appropriate scaling and informative axis labels.

The Mars-242 crew must now identify a suitable site for their landing module. The surface-based rover has transmitted images of some areas it has scouted to the spaceship for human inspection. After all your efforts to enable communication between the rover and the Spaceship (in section 1: Mars Rover), you now discover that, the transmission channel between the rover and the spaceship has introduced both periodic and bandlimited random noise to the signal stream. As you can see from the example image shown in Fig. 4 the noise has rendered the image useless for identifying any surface features.

Your task is to de-noise the images and recommend an appropriate landing site.

Figure 4: A corrupted image from the Mars exploration rover.

Images in MATLAB are stored as 2D matrices. Each element in the matrix represents a pixel of the image. The numeric value at each matrix index describes the colour intensity of each pixel. Matrix elements for grayscale images are floating point numbers between 0 and 1; which correspond to the colours black (zero intensity) and white (maximum intensity) respectively. You will be working with grayscale images in this assignment.

Even though the images are represented by 2D matrices, they are received as a 1D data stream. The first received pixel is placed at the top left corner of the image. Subsequent pixels are used to fill the image column-wise (top to bottom) from left to right *(shown below)*. The size of the images being received are 480×640 pixels.

$$1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10 \quad 11 \quad 12 \quad \longrightarrow \quad \begin{matrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{matrix}$$

The received signals are provided in the rows of the matrix `sig`. There are 4 images received, and hence 4 signals, with the first stored in the first row of the matrix.

NOTE: For sections 3.1-3.10 only use the first received image

3.1 *View the noisy image* - Display the first image using the `imshow` function. You will need to first convert the signal into a 2D matrix using `reshape`, *ie.* `imshow(reshape(sig(1,:), 480, 640))`. (CR 2a)

3.2 *Reference vectors* - Image data is received from the rover at 1000 pixels/sec. Consider each pixel a single sample. Construct a time vector for the input signal (i.e. a single image) and store this in `t`. Create an appropriate frequency vector and store in `k`. (CR 2a)

3.3 *Visualise the received signal* - Plot the first received signal (`sig(1,:)`) in both the time and frequency domains. Scale the time domain plot to show the first 3 seconds of the signal. Designate the frequency domain data with capitals i.e `SIG`. Data in `SIG` should correspond with time domain data in `sig` i.e. row-wise. Identify (graphically) both the periodic and bandlimited random noise in both frequency domain plots. Ensure you state the approximate period of the periodic noise and the bandwidth of the bandlimited noise. (Rescale the plots if necessary). (CR 1b, 2b)

The periodic noise in the image can be removed if it is accurately estimated. Engineers at mission control have identified a set of possible values for the period of the corrupting noise. These values are stored in the vector `candidateT`. Averaging the shape of the waveform for all periods will emphasise the noise component, smoothing away the underlying image into an offset value. The function `estimateNoise.p` has been provided to you to perform this. It will return a vector one period long representing your periodic noise.

3.4 *Estimate the periodic noise* - Determine the period of the noise from the given options and store your selected value as `T`. The units for `T` should be *seconds*. Use `estimateNoise` to estimate the noise profile and store the result in the variable `sigNoise`. The syntax for its usage is

```
output = estimateNoise(inputSignal,periodInSamples)
```

Plot and compare a periodic version `sigNoise` to the received signal (`sig(1,:)`), showing only the first 3 seconds. You can use `repmat` to repeat a vector many times. *If the overall shape of the signals do not closely match, you may have an incorrect estimate of the period.* (CR 1a, 1c, 2d)

3.5 *Model the periodic noise* - Model the noise signal by computing the Fourier coefficients of `sigNoise`. You can choose to perform this using either the trigonometric or complex exponential Fourier series. Start by computing the DC (a_0) and the first 6 harmonics (a_n, b_n for $1 \leq n \leq 6$ or C_n for $-6 \leq n \leq 6$). Remember to account for sampling rate. List these coefficients and explain your process for computing them. (CR 1a, 1b, 2a, 2b)

3.6 *Bias* - Mission control has determined that the mean of the periodic noise signal is 0. Any DC component present in `sigNoise` is due to the underlying image. Make any changes necessary to your coefficients to take this information into account. List the new coefficients and explain the reasoning behind any modifications you have made. (CR 1a, 1d)

3.7 *Generate the approximation* - Using these Fourier Series coefficients, generate the approximation of the noise component to correspond with the received signal (ie. all $307,200$ samples) and store it in `sigNoise_fs`. (CR 1a, 2b)

3.8 *Compare the approximation* - Compare the first T samples of `sigNoise_fs` to `sigNoise`. Does the number of coefficients used result in a good representation of `sigNoise`? (CR 2b)

3.9 *De-noise* - Use your previous BASA training from Assignment 1A to de-noise the image. Store the result in the first row of an image matrix `im1`. Display the image and its spectrum in MATLAB. Include the recovered image and spectrum in your report. Comment on how the spectrum has changed compared with the spectrum of the received image. Also comment on the quality of the de-noised image. Has a sufficient amount of noise been removed to identify details of the image? If not, experiment with using more Fourier coefficients to improve the image quality. Justify the number of coefficients you decide on using for removing the noise. (CR 1c, 2b)

3.10 *Remove the bandlimited random noise* - Remove the bandlimited random noise. Decide whether to do this in the time or frequency domain, but the final outcome must be in the time domain. Store the filtered signal in the first row of `im2`. (CR 1d)

3.11 *Choose a site* - Mission control has determined the periodic noise profile is consistent across all received data. Using `sigNoise_fs`, repeat the de-noising process from 3.9 for the remaining 3 images and store in the corresponding rows of `im1`. View the spectrum and remove the bandlimited random noise using the same method used in 3.10. Store the results in corresponding rows of `im2`. Display the images contained in `im2` in a single figure in MATLAB. Of the 4 landing sites photographed, which is an appropriate site to send our lander? Justify your choice. (CR 2b)

3.12 *Resolution* - Navigational numbers were marked onto each image at the time they were photographed. What are these numbers? List these in your report. Also include the filtered images in your report, at a scale that allows the navigational numbers to be read. (CR 3d)

MATLAB variables that should be included in your workspace for section 3 (A2part3.m),

t - Time domain vector
k - Frequency vector
SIG - Fourier Transform
T - Selected candidate T value
sigNoise - Estimated noise signal
c0, cn - Complex Fourier series variables, OR, **a0, an, bn** - Trig Fourier series variables
sigNoise_fs - Approximation of noise
im1 - Image matrix 1
im2 - Image matrix 2

Reflection (CR 3d)

Include a short discussion (maximum 300 words long) that covers any lessons learned and things that you would have done differently. Marks for this are included as part of the criteria available at the end of this brief.

Academic Integrity Declaration (Mandatory)

The provided Academic Integrity Declaration and contribution online form must be completed and submitted along with the assignment. Each student from the group will need to complete their own form. Marks may be moderated depending on contributions. Assignments with incomplete or missing declarations will not be marked. *Familiarise yourself with the university's policy regarding **plagiarism** and **collusion**. See the file "Academic Honesty Slides.pdf" posted with this assignment for some useful details.*

Report and Code Presentation

This assignment includes elements of writing and coding. This is a group assessment item and you are expected to generate and submit:

- One assignment report, "The Report",
- One set of MATLAB code, "The Code" (including at least A2part2.m and A2part3.m), and
- Two data files (A2P2Data.mat and A2P3Data.mat).

The Criteria Reference Assessment (CRA) sheet has the outlines of the marking standards of this assignment.
The teaching team has put together some pointers for you to consider:

The Report (CR3)

An outstanding report demonstrates clear knowledge and understanding of the subject through a combination of visual, mathematical and coding elements. Correct information that is not articulated clearly will attract deductions. **Remember that you are writing to inform.**

- Present the report so it can be understood without reference to the assignment brief.
- Figures or code referenced should be no more than 1 page turn away.
- Avoid the use of "see appendix" and "refer to .m file".
- Full working is required in mathematics-based sections.
- Ensure legibility in any handwritten working.
- Include a title page that states the unit name, unit code, and your names and student ID numbers.
- **Do not include a table of contents, list of figures, nor a list of tables.**

The MATLAB Code (CR 2)

Working MATLAB code is expected to be submitted, alongside your report to Blackboard. The code needs to be executable (in *.m) and **without** run-time errors. No error correction will be made to make your code "run."

Code should be fully commented to describe intent. Quality comments encapsulate your understanding of the topic.

You may use the code provided in the weekly tutorials to check your solutions. However, you are expected to generate your own code for your assignment. Submitting supplied **.p** code as your own work constitutes academic misconduct and will not be awarded any marks.

Code for this assignment will be marked with the assistance of an automated marking system. Ensure that you follow given instructions carefully, including naming conventions. Your code submitted will also be checked for academic misconduct.

Interview

Group interviews will take place (at the discretion of the teaching team) to ensure demonstrated understanding and skills required for this assignment, by the group, and the individual members. You may be selected and contacted to attend an interview if the teaching team requires clarification about how you arrived at your solutions. Interviews will be a casual discussion. These interviews are compulsory and grades are withheld until they are completed. Marks may be deducted for poor demonstration of understanding of content or assignment knowledge. Consult the CRA sheet for the guidelines of what is expected.

Submission Protocol

Assignments are to be submitted in soft-copy through QUT Blackboard in three parts

- A completed academic integrity and group contribution online form. This form is to be completed individually by every student.
- The report. Only **ONE** group member is required to submit the report to the Turnitin link. Coordinate within your group who this will be.
- Your data and code files. Include everything here that your code needs to run. You may submit as either a single zip file, or attach your required files individually.

Some further points:

- Submission deadline is on Thursday 18th of October at 11:59pm.
- This will be a hard deadline, and late submission will not be accepted. As per QUT policy, late assignments receive 0 marks, unless you have applied for and received approval for extension, as per the university policy.
- You do not need to assign your submission with a special name.
- You will need to be registered to a group before you can submit your assignment.

- You may submit as many times as you like before the deadline. New submissions overwrite old submissions. Therefore, only the latest submission will be marked.
- All documents can be reviewed after submission, and thus it is your responsibility to verify the uploaded documents.
- Be aware that the electronic time stamp is placed only after all files have been uploaded successfully.

Don't risk the late penalty and submit early.