EGB242 – Assignment 1 (25%)

Individual assignment

Released: Friday 17th March, 11:59pm (Week 3)

Due: Friday 21st April, 11:59pm (Week 7)

Context

Congratulations on landing a summer internship at BASA, Brisbane's premiere (and only) space agency! Although such an achievement warrants an extended celebration, the launch date of the MARS-242 mission is rapidly approaching.

This crewed mission will deliver astronauts into orbit around the red planet in preparation for permanent residence, and land a rover onto the Martian surface to capture photographs of the proposed settlement location. To ensure both the astronauts and their scientific instruments aboard the spacecraft can maintain contact with mission control on Earth, a communication system must be designed.

Your first placement at BASA will involve finalising the design of several components of the communication system presented in Figure 1. You must detail your design process as part of a report to be presented to your supervisor, the head of communications engineering.

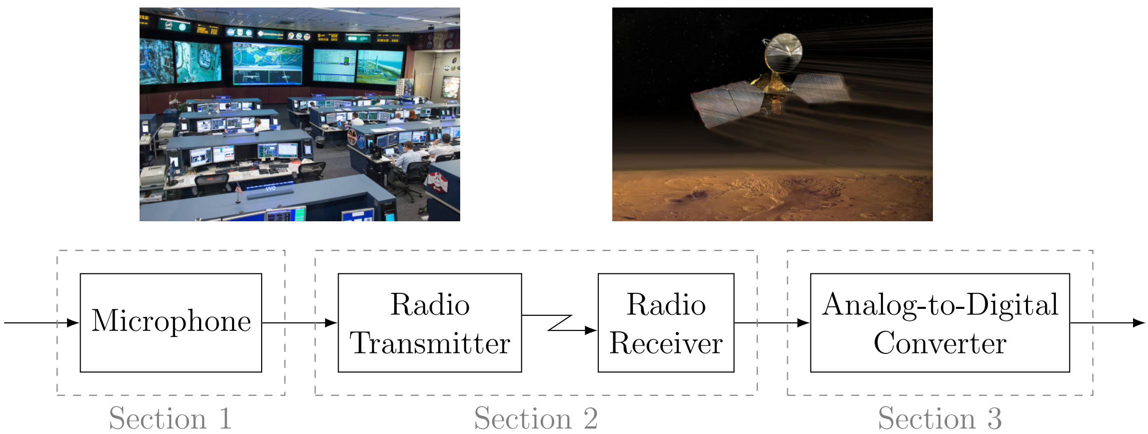


Figure 1: The communication system.

The system takes audio input through a microphone installed at mission control and uses a radio transmitter to send the audio signal to the astronauts. On the orbiter, a radio receiver will recover the audio signal which can then be prepared for digitisation by an analog-to-digital converter.

Section 1: Removing Periodic Noise

After inspecting the audio waveform produced by the newly installed microphone at mission control, it has been observed that a periodic signal produced by other nearby equipment is causing significant interference. Through discussions with the engineer who designed the problematic equipment, a function which describes the noise and a model for the noise process have been developed. This will be useful in removing the interfering noise.

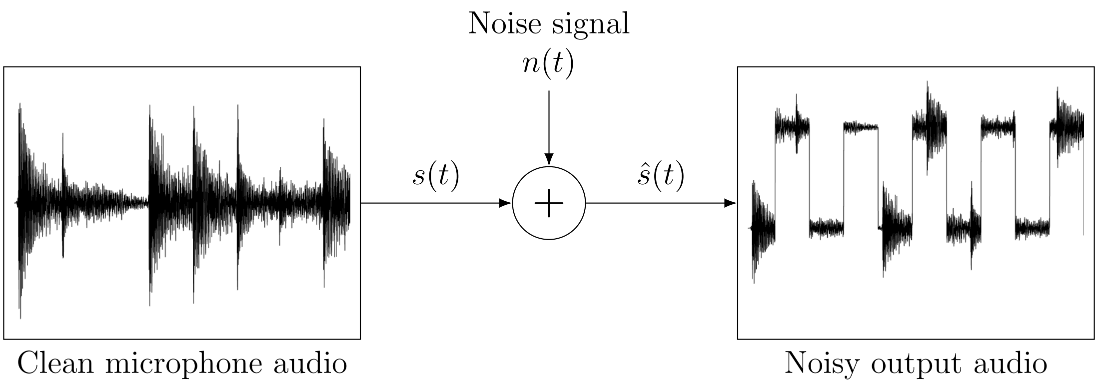


Figure 2: The additive noise model.

The interference of the periodic signal with the recorded audio can be modelled as an additive noise process (Figure 2). In order to produce clean, noise-free audio, this noise process must be reversed. Your task is to model the noise signal as a complex Fourier series, and use a Fourier series approximation to de-noise the signal.

1.1 Record 10 seconds of test audio and note down your noise function. To record your test audio, use the `record` function from the MATLAB Command Window:

```
record(sid)
```

where `sid` is replaced with your student number. Record the following message when prompted:

"Go for main engine start. T minus ten, nine, eight; All three engines up and burning. Two, one, zero, and liftoff!"

After recording your audio, write down the noise function $n(t)$ which is displayed. Ensure that the file `DataA1.mat` has been created.

You will only need to complete this step once.

1.2 Begin your MATLAB solution in the provided `missionA1.m` template. Listen to your recorded audio which has been loaded into the workspace with the `load DataA1` command by using `sound(audio, fs)`. Comment on your observations.

Note: you can stop a sound started with `sound()` from playing by typing `clear sound` into the MATLAB Command Window.

1.3 Create a time vector `t` for the audio signal. Plot the audio signal against `t`. Comment on your observations, and how they relate to any audible characteristics of the signal.

1.4 By hand, evaluate the complex Fourier series coefficients of your noise function. Typeset the noise function $n(t)$ and its complex Fourier series (including expressions for c_0 and c_n) into the body of your report. **Attach your hand-working as an appendix to your report**.

1.5 Using $-5 \leq n \leq 5$ (i.e., 5 harmonics), generate a vector `cn` in MATLAB which contains c_n evaluated at each value of n. List the values of these coefficients in your report.

1.6 Using the `cn` vector, generate an approximation of the noise signal `nApprox` for the full time vector `t`. Plot your recorded audio and your generated noise signal approximation.

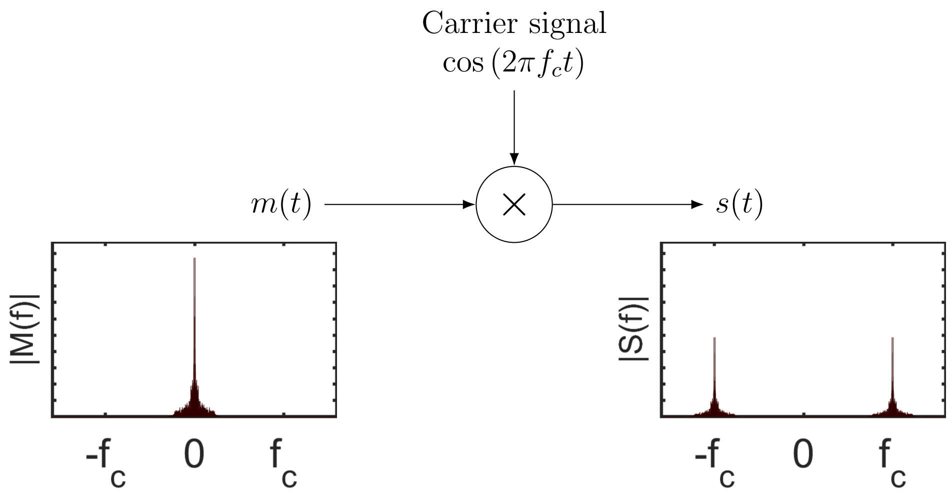
1.7 De-noise the recorded audio by reversing the additive noise process (Figure 2) using your Fourier series approximation, and store the de-noised signal in `audioClean`. Listen to the clean signal, and plot it.

1.8 Is using 5 harmonics in your noise signal approximation enough to adequately de-noise the audio? Experiment with the number of harmonics to determine a suitable value, and justify your choice both qualitatively and quantitatively.

Section 2: Transmitting and Receiving Signals

With the microphone audio now de-noised, it is ready to be fed into the radio transmitter and sent from the ground station to the MARS-242 orbiter. In order to transmit the signal, it must be modulated onto a carrier.

Modulation is the process of "moving" a signal in the frequency domain. It is only possible to transmit an audio signal as a radio wave once it is modulated and thus shifted to a much higher band of frequencies. However, this requires the shift to be removed when the radio wave is received in order to recover the audio signal. The *modulation property* of the Fourier transform allows these shifts to be achieved.



Audio signal magnitude spectrum, before modulation, centered at 0 Hz.

Radio signal magnitude spectrum, after modulation, centered at $\pm f_c$ Hz.

Figure 3: Using the modulation property to move a signal up to a carrier frequency of f_c Hz.

You have also been provided with a MATLAB function `channel` which simulates the transmission of a signal `input` between the ground station and the orbiter. The function is utilised as follows:

```
output = channel(sid, input);
```

where `sid` is your student number, `input` is a row vector representing the signal which should be transmitted, and `output` is the signal received by the orbiter.

2.1 Plot the magnitude spectrum of the clean audio signal, using an appropriate frequency vector `f`.

2.2 "Listen" to the channel before transmitting your signal through it. You can "listen" to the channel before transmitting anything by passing a vector of zeroes through the `channel` function.

```
channelQuiet = channel(sid, zeros(size(t)));
```

2.3 Plot the time and frequency domain of `channelQuiet` in order to find an empty band of frequencies that you can transmit your audio on. State your selected range of frequencies and the center frequency. Justify these parameter choices.

2.4 Modulate your audio signal using the carrier frequency you have selected.

2.5 Simulate the transmission of your modulated signal, providing it as input to the `channel` function, and plot the frequency domain of the input and output signals.

2.6 Demodulate your audio signal from the channel output created in 2.5. View the demodulated signal in the frequency domain. Filter the demodulated signal to isolate your audio signal. Use the `lowpass` function in MATLAB to simulate an analogue filter, and store the received audio as `audioReceived`.

2.7 What pre-existing signals were visible in the output of the channel when you investigated it in 2.2? Modify your code to instead demodulate each of the other signals visible in the channel and listen to these signals.

Note: revert the changes you have made in 2.7 before proceeding.

Section 3: Analog-to-Digital Conversion

Although the demodulation process now allows the crew to listen to transmissions from an analogue speaker, a digital audio signal is required to distribute audio to the astronauts' headphones and other ship systems. Thus, the signal must be digitised through a sampling and quantisation process.

3.1 Use MATLAB's `resample` function to resample your received audio signal at a valid sampling rate (from Table 1) closest to its Nyquist rate. Store your new sampling rate as `fs2` and your resampled audio as `audioResampled`.

Although theoretically any positive number is a valid sampling rate, digital systems have settled on a common list of sampling rates which are most likely to be supported by hardware, shown in Table 1.

Table 1: Valid sampling rates, known to be widely supported by computer systems.

Valid sampling rates [Hz]						
8,000	11,025	22,050	44,100	48,000	96,000	192,000

3.2 Listen to and comment on the resampled audio.

3.3 With an appropriate quantiser (mid-tread or mid-riser), quantise `audioResampled` using 16 quantisation levels and store the result as `audioQuantised`. Listen to and plot the quantised audio and comment on any changes.

3.4 Experiment with using 2, 4, 8 and 32 quantisation levels. Listen to the quantised audio for each case, and select an appropriate number of quantisation levels for the final system. Justify your choice.

Reflection

A two paragraph reflection is to be written and appended at the end of your report. In the first paragraph, summarise how you have demonstrated your understanding of the concepts used in this assignment. The second paragraph should be a discussion/professional reflection that covers any lessons learned from doing this assignment, and things that you would have done differently. Each paragraph should not exceed 250 words. Marks for this are included as part of the criteria available on the appended CRA sheet.

Submission Requirements

There are several components of Assignment 1 which must be submitted to the "Assessment 1 – Problem Solving Task: Submission" assignment on Canvas before the due date.

- Your report, in PDF format
- Your complete MATLAB solution (`missionA1.m`)
- Your recorded test audio (`DataA1.mat`)

After submitting, re-download each file to ensure you have submitted a complete and correct version of your assignment.

Report

To present your solution to the assignment tasks, you are required to write a report.

Your report should demonstrate clear knowledge and understanding of the subject through a combination of visual, mathematical, and coding elements. Your report should have a logical flow which guides the reader through your solution process, incorporating relevant explanations and justifications for the steps taken. Correct information which is not articulated clearly will be awarded a lower grade, as will inaccurate or vague justification. Ensure you include, at minimum, all requested plots/figures and justification. **Remember that you are writing to inform.**

It is highly recommended to follow the below report structure. You may adapt the structure to suit your needs, but ensure you include all required aspects.

- Title page - *must state your name, student number, the unit name and unit code*
- Introduction
- *Further headings, splitting up your solution as appropriate*
- Conclusion
- Reflection
- References - *if required*
- Appendices
  1. Full MATLAB source code - *include only raw source code, no figures*
  2. *Any other appendices as appropriate*

Note:

- Do not include a table of contents, list of figures, or a list of tables.
- Integrate code and figures throughout your report. Do not simply state "refer to appendix".

Interview

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 on Canvas for the guidelines of what is expected.