EGB242 Assignment 2

Charlie Turner: n10752846 Joel Kemp: n10746862
Viet Truong Minh: n10440151

May 31, 2022

Introduction

After the chief at Brisbane-based Australian Space Agency (BASA) has reviewed previous tasks, they have decided to assign the team 'live' supervision of the MARS-242 mission.

Communication between the ground and the mission must be clear and uninterrupted to ensure the mission is safe and smooth. There are three important tasks to be completed:

- Identifying an appropriate landing site for the crew
- Ensuring the safety of the Martian habitats for the current and future crews

To achieve these objectives, the team must be able to properly interpret and analyse data streams from a rover scouting the surface prior to landing. The landing site must be identified despite periodic and bandlimited noise disrupting signal streams. Once a landing site is determined, the acoustic properties of the habitat's rooms must be analysed, removing interference and noise arising from the communication channel and method.

1 Mars Rover

With the crew landing soon, a rover has been gathering data regarding possible landing sites. Until a malfunction on the rover, data has been communicated directly with Earth, where it can be analysed and the results sent directly to astronauts on board Mars-242. The rover's data is vital for the survival of the mission, so a solution has been found to send the raw data directly to astronauts, but there are complications in this process.

Communication Channels

$s_{rov}(t) = 8000 \times \text{sinc}^2(4000t)\cos(2\pi \times 8000t)$

Rover Transmission Function

Filtering Information

Signal Recovery

2 Choosing a Landing Site

The surface-based rover has collected and transmitted images of possible landing sites. Unfortunately, the transmission encountered both periodic and bandlimited noise, so the signals must be denoised to observe the intended images. One of the noisy images is visualised in Fig. 1. This is achieved in MATLAB using the `reshape` function, reshaping the incoming 1D data stream into a matrix, knowing that the images are all of the same size (640×480).

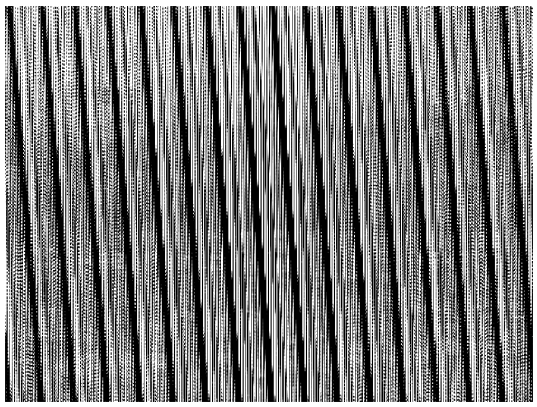


Figure 1: Noisy image of possible landing site

2.1 Modelling Periodic Noise

To understand the characteristics of relevant noise, the signal is plotted in both the time and frequency domain as in Fig. 2, understanding the importance of the sampling rate for the incoming signal:

```
1  Fs = 1000;                                    % Pixel Sampling Rate [Hz]
2  T = pixels/Fs;                                % Time to recieve an image
3  t = linspace(0, T, length(sig)+1);            % Time vector for full image
4  f = linspace(-Fs/2, Fs/2, length(sig)+1);     % Frequency Vector
```

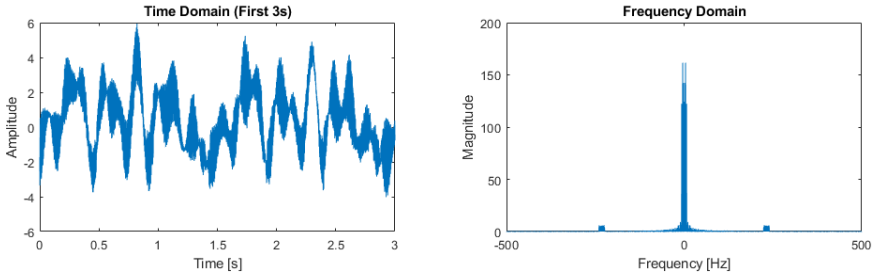


Figure 2: Time and Frequency Domain of Incoming Signal

While the full signal for an image is over 300 seconds long, plotting only the first three seconds shows that there is a significant periodic noise present in the time domain. Engineers at mission control have identified a set of possible periods for the noise, and graphically we can determine that it appears to be approximately 1.477 seconds long, confirming the engineers approximations. Using MATLAB and the provided `estimateNoise` function, a vector representing one period of this noise was computed:

```
T = candidateT(1);                                    % period of noise
periodInSamples = T * Fs;                             % samples in noise period
Noisesig = estimateNoise(sig(1, :), periodInSamples);  % estimated noise function
```

Comparing the estimated noise to the received signal in Fig. 3, we can see that the overall shape of the signals is similar.

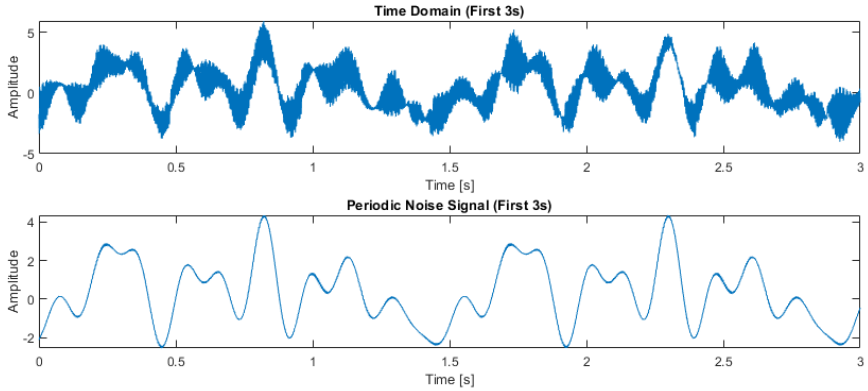


Figure 3: Comparison of Estimated Noise and Recieved Signal

With the correct discrete representation of periodic noise determined, we can model `Noisesig` with a complex fourier series as follows:

$$x(t) = \sum_{n=-\infty}^{\infty} c_n \exp(j2\pi n f_0 t)$$
$$\text{with } c_n = \frac{1}{T}\int_{t_0}^{t_0+T} x(t)e^{-j2\pi n f_0 t}dt$$

In MATLAB, integrals are substituted for Riemann sums on the discrete vector, computing c_n for $-6 \leq n \leq 6$ taking into account an accurate time vector and sampling rate:

```
t1= t(t<T); t1(end) = [];                             % Noise Time vector
ts = t1(2)-t1(1);                                     % Sampling interval [s]
f0 = 1/T;                                             % Fundamental Frequency
N = 6;                                                % Number of harmonics
n = (-N:N).';                                         % Vector of harmonics
cn = f0 * Noisesig * exp(-1j*2*pi*f0*n*t1).' * ts;    % Fourier coefficients
c0 = cn(N+1);                                         % DC coefficient
```

The DC coefficient is computed as $c_0 = +0.3571$, and the complex coefficients for the first 6 harmonics as follows:

$$\begin{array}{lll} c_{-6} : -0.2208 + 0.4813i & c_{-5} : +0.2479 - 0.2237i & c_{-4} : +0.0260 - 0.1276i \\ c_{-3} : -0.4525 + 0.1048i & c_{-2} : -0.2916 + 0.2814i & c_{-1} : -0.3702 + 0.1522i \\ c_{+1} : -0.3702 - 0.1522i & c_{+2} : -0.2916 - 0.2814i & c_{+3} : -0.4525 - 0.1048i \\ c_{+4} : +0.0260 + 0.1276i & c_{+5} : +0.2479 + 0.2237i & c_{+6} : +0.2208 + 0.4813i \end{array}$$

BASA has confirmed that there should be no DC component to the periodic noise. Because c_0 represents the DC component of the Fourier series, this term can simply be equated to 0 in order to remove this bias:

```
90   c0 = 0;                                          % DC coefficient
91   cn(N+1) = c0;                                    % Corresponding cn coefficient
```

With the correct Fourier coefficients determined, the signal can be reconstructed for the total period of the image signal, as per the following:

```
93   Noisesig_fs = cn * exp(1j*2*pi*f0*n*t);          % Fourier series for total t
```

To test the accuracy of the model, this Fourier approximation is plotted against the noise profile for one period, as seen in Fig. 4:

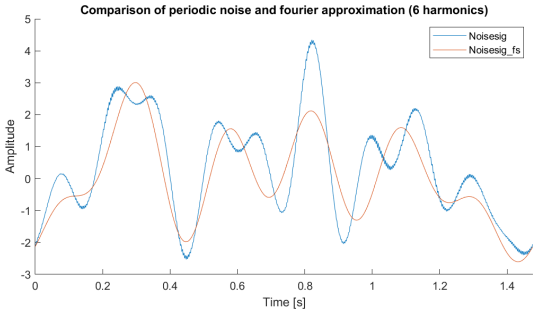


Figure 4: Estimated noise and Fourier series approximation

Visually, the Fourier approximation with 6 harmonics doesn't provide a clear representation of the signal, as it is unable to model the more detailed noise at the peaks and such. The overall shape of the noise is represented, but it is expected that the denoising may not be sufficient. This approximation can be tested by subtracting the periodic noise from the incoming image signal.

```
130  im1(1,:) = sig(1,:) - Noisesig_fs;               % First image with periodic noise removed
```

The above reshaping and Fourier transform of the denoised image was then computed to view the image and its magnitude spectrum at this stage of the denoising process:

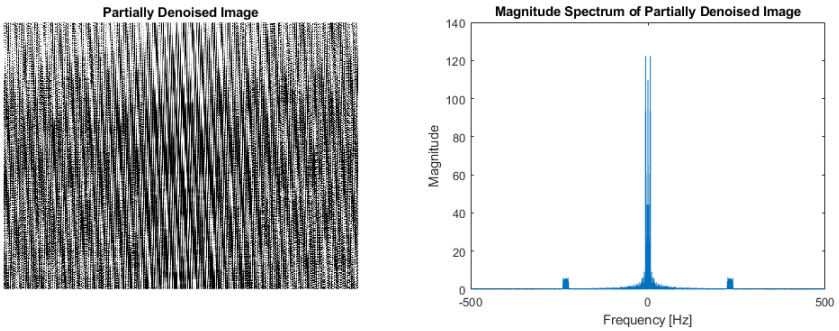


Figure 5: Image and Magnitude Spectrum with periodic noise removed

As expected, the image is not noticeably clearer than the original, with strong lines still obscuring the intended picture. The magnitude spectrum confirms this noise, with large frequency spikes present at specific low frequencies. This implies that there is still considerable periodic noise. To improve the noise approximation, a more suitable number of harmonics would be 10, as it captures the details much more accurately. It

also requires very little extra computational power, taking only about 0.03 seconds extra to compute the Fourier approximation than with 6 harmonics. The added accuracy is shown below in Fig. 6, with 10 harmonics required to achieve a considerably more accurate approximation.

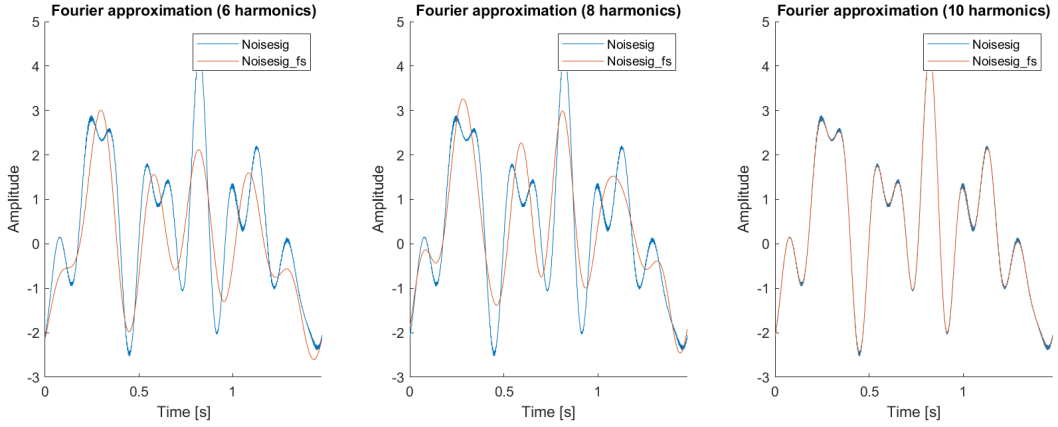


Figure 6: Comparison of fourier series (6, 8, and 10 harmonics)

Now with 10 harmonics to better approximate the periodic noise, we can once again denoise the image, shown in Fig. 7:

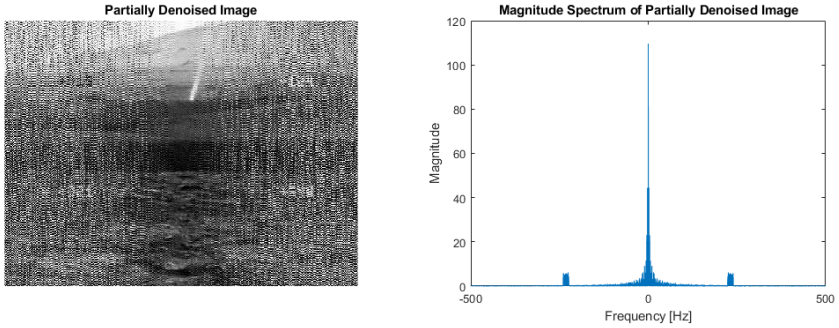


Figure 7: Improved Denoised Image and Magnitude Spectrum

The strong predictable lines corrupting the image are now much less pronounced, and the outline of what resembles a mountain and rocky terrain can be distinguished. However, the image is still much too noisy to discern the specific location.

Removing Bandlimited Noise

The frequency band for bandlimited noise was determined visually as in Fig. 8, with a lower limit of ±223Hz, and upper limit of ±240Hz. A band stop filter was created to remove the noise. This is achieved in MATLAB in the frequency domain, which is much more easy to visualise. The filter was constructed as a vector corresponding to the full frequency vector, multiplying frequency components within the band by zero, while leaving other data untouched. Finally, the inverse Fourier

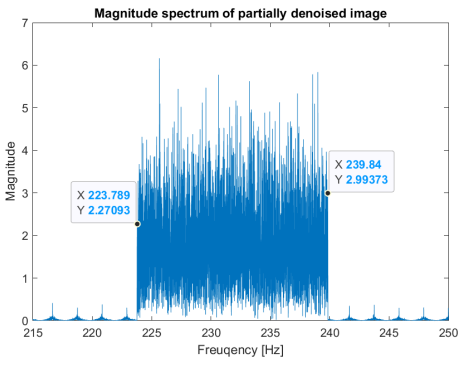


Figure 8: Magnitude spectrum of bandlimited noise

transform must be computed to convert the de-noised signal back to the time domain for viewing:

```
150  B_low = 223;                                          % Lower bandwidth frequency
151  B_high = 240;                                         % Upper bandwidth frequency
152  filter = ones(size(f));                               % Initiallise with ones
153  filter((abs(f) > B_low) & (abs(f) < B_high)) = 0;     % Assign 0 to elements in band
154  filtered = fftshift(IM1(1,:)) .* filter;              % Filter in frequency domain
155  im2 = zeros(size(sig));                               % Initialise for performance
156  im2(1,:) = ifft(ifftshift(filtered));                 % Filtered image signal in time domain
```

Showing this final image in Fig. 9, we can see that the picture has been recovered, along with navigational numbers present on the image:

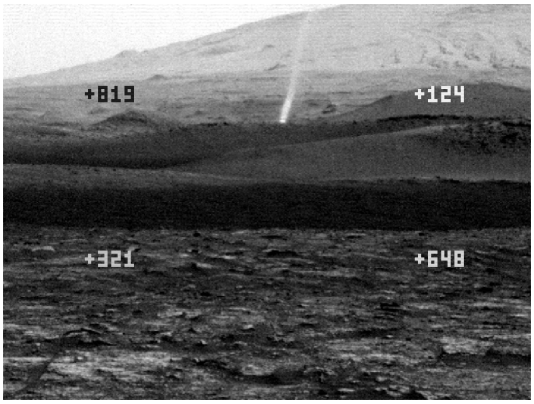


Figure 9: Recovered image from the rover

2.2 Choosing A Site

Knowing the process for denoising incoming images, the other three sites were recovered. Periodic noise is the same for each image so was subtracted as above, however removing bandlimited noise required reconstructing the band-stop filter for each image. To do this, the frequency bands were determined visually for each image in the frequency domain, and then a for loop was used to filter the corresponding images:

```
1   % Visually get bands of noise
2   bands = {[223, 240], [232, 246], [234, 251], [253, 268]};
3   % Remove bandlimited noise from each image
4   filter = ones(size(f));                                          % Initialise for performance
5   im2 = zeros(size(im1));                                          % Initialise for performance
6   for k = 1:length(bands)
7       filter(1:end) = 1;                                           % Reallocate ones
8       filter(abs(f) > bands{k}(1) & abs(f) < bands{k}(2)) = 0;% Construct filter
9       filteredIm = fftshift(IM1(k, :)) .* filter;                  % Apply in frequency domain
10      im2(k, :) = ifft(ifftshift(filteredIm));                     % Store in time domain
11  end
```

The result, as seen in Fig. 10, is four clear images complete with navigational numbers:

Site	Top Left	Top Right	Bottom Left	Bottom Right
Site 1	819	124	321	648
Site 2	246	979	916	321
Site 3	111	371	958	351
Site 4	841	331	362	615

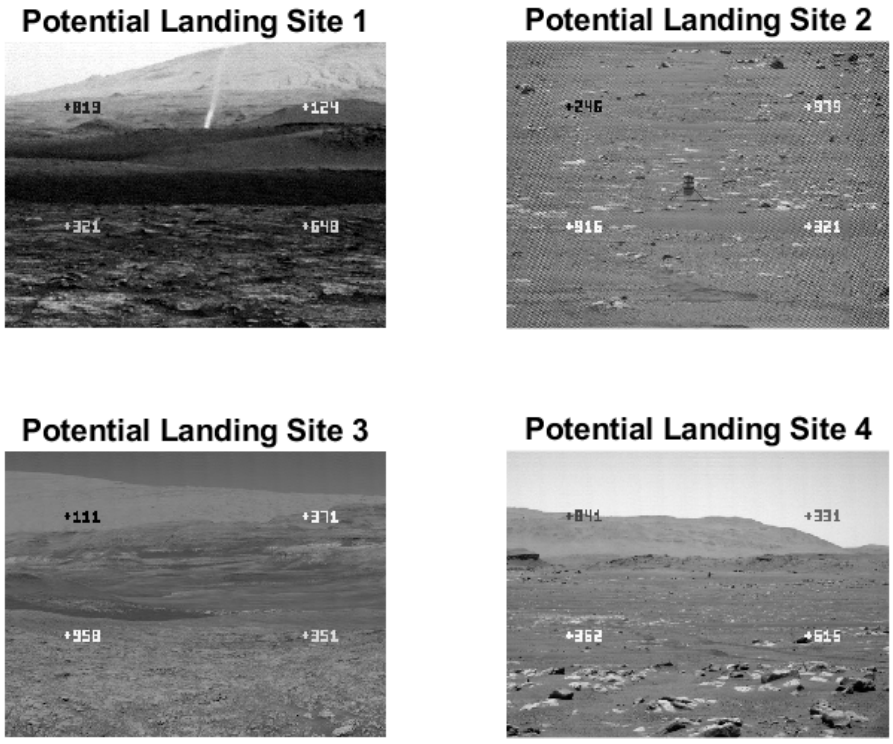


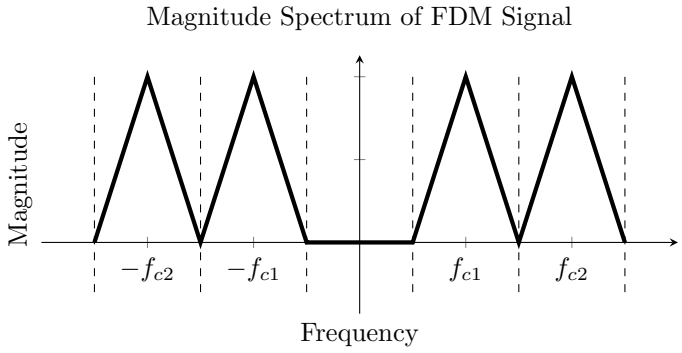
Figure 10: All four sites recovered

It appears as though site 4 may be the ideal landing site, as it is the only image showing a large landscape that is almost entirely flat except for the far away mountains. Site 2 appears as though It could be a good site, however there is not a large field of view in the image. Sites 1 and 3 have large mountains and valleys, and are therefore unsuitable for ensuring a safe landing.

3 Habitat Impulse Analysis

With a landing site chosen, the base of operations has been set up. In order to ensure safety of occupants, acoustic properties of rooms in the habitat must be analysed. This will allow the team to determine astronauts' locations in an emergency if they cannot identify it themselves. For this analyses, the team has recorded the impulse responses of different rooms, and sent them back to the team at BASA headquarters. The signal is in the form of a Frequency Division Multiplexed (FDM) data stream, containing the impulse responses as well as a text message indicating the astronaut and room.

The result is a signal which is unintelligible, with each part of the data modulated onto different, non-overlapping frequency bands. This is demonstrated in the simplified figure below:



In the case of the habitat's acoustic responses, all intended messages are multiplexed to occupy 8kHz of bandwidth at differing carrier frequencies. In order to recover the impulse responses and text messages from the FDM signal, these different carrier frequencies must be determined, and each signal must be demodulated.

Spectrum Analyser

To determine the carrier frequencies for the multiplexed signal, the Fourier transform of the signal is used to plot the magnitude spectrum in MATLAB. This is done by first determining the time and frequency vectors for the signal, knowing that incoming signal is sampled at `fs=576`kHz.

```
Ts = 1/fs;                                % Sampling Period [s]
t = 0:Ts:0.5; t(end) = [];                % Time vector
f = linspace(-fs/2, fs/2, length(t)+1); % Frequency Vector
f(end)=[];
MUXSIG = fft(muxSignal);                  % Fourier Transform
```

Plotting the signals,

Demultiplexing

Filtering

Resampling for Analogue to Digital Conversion

Resampling for Analogue to Digital Conversion

Practical Analysis

4 Reflection