Orbiting the Red Planet

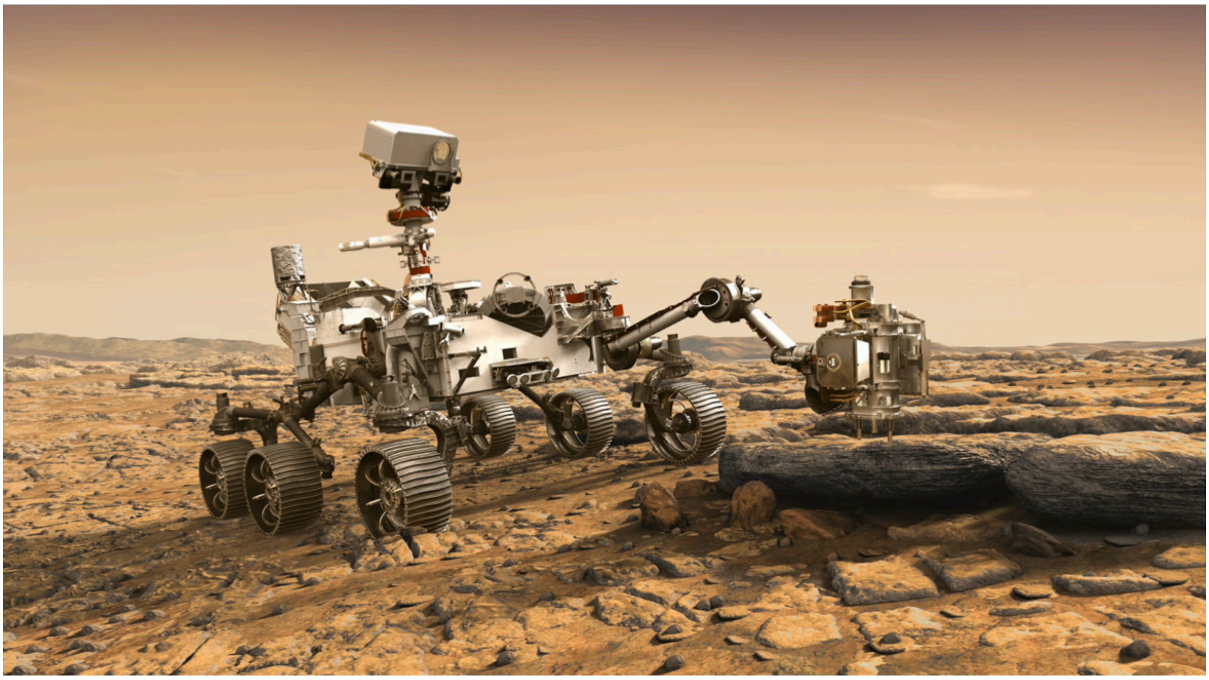


Fig 0. [1]

Group 95
Billy Henderson n1051387
Deon Ajay n10524177
Don Kaluarachchi n10496262

Introduction

After the successful completion of the previous tasks the chief supervisor of BASA has decided to allow a group of three of the interning engineers to join the main engineering team that supervisors the MARS-242 mission.

Two high priority objectives have been provided which will build upon and rely on the simulated Mars mission scenarios provided in previous tasks at BASA. These two high priority tasks are,

1. Monitor the Astronauts' physical and psychological health over the rest of the mission.
2. Identify an appropriate landing site for the landing module as the spaceship orbits Mars.

Section 1 – Mars Rover

The communication system of the Mars rover that has been gathering data in preparation of the spaceships landing has been damaged. The rover cannot communicate long distance due to this and the only solution is for the astronauts to do the analysis of the data themselves. This data contains images of the planet which are important for the landing and we must mathematically model the data so they can be received and processed.

The Spaceships Communication Channel

When communications channel seems to be compromised as it is believed that the spectrum of the rovers' transmitted signals may interfere with the spectrum of the spaceship's communication module. Using the generated data, it was determined that the f_{bw} is 2900Hz. This was modelled along $-f_{bw}$ to $+f_{bw}$ as a gate function with constant magnitude of K. Using this it was mathematically expressed as $S_{sship}(f)$ below.

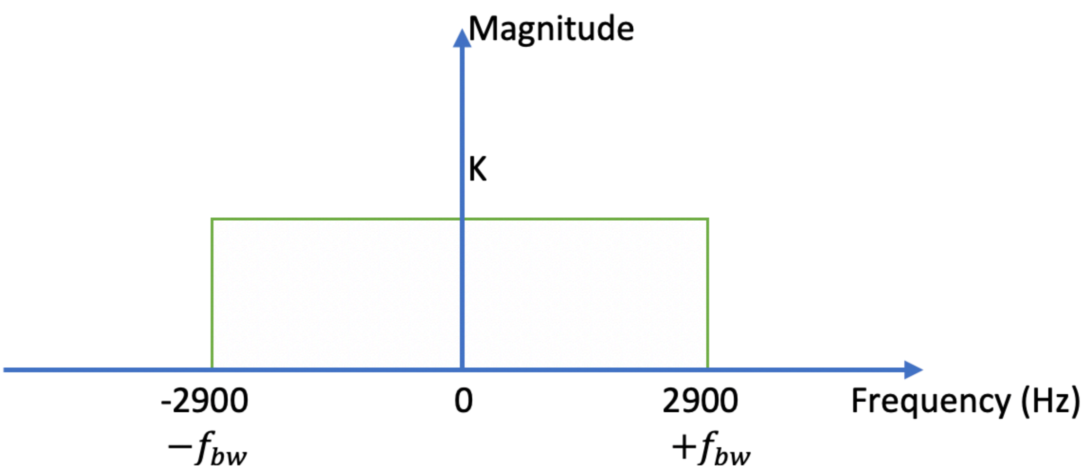


Fig 1.0 Modelling the spaceships communication channel.

$S_{sship}(f)$ can be given in the form $K.rect\frac{f}{T}$, K is the magnitude and T is the period

$$T = 2\,.\,f_{bw}$$
$$T = 2\,.\,2900\ Hz$$
$$T = 5800\ Hz$$
$$S_{sship}(f) \ = K.rect\frac{f}{5800}$$

Modelling the Rover's Transmission Function

The rover's transmission function in the time domain can be expressed as,

$$S_{rov} = A.Sinc(6000t).Sin(10\pi.10^3t + \tfrac{\pi}{2})$$

The data generated provided the 'A' value which was 12000. Using this value and expression is derived in the frequency domain using knowledge of Fourier and Laplace transforms.

i. The function can be divided into two parts,

$$S_{rov} = \underbrace{12000.Sinc(6000t)}_{\text{x(t)}}.\underbrace{Sin(10\pi.10^3t + \tfrac{\pi}{2})}_{\text{y(t)}}$$

ii. Converting to the required form for transform,

$$12000.Sinc(6000t) \longrightarrow 2.6000Sinc(6000t)$$
$$FSinc(Ft) \ \text{ where F = 6000}$$

iii. Using the transform table,

$$FSinc(Ft) \longleftrightarrow rect(\tfrac{f}{F})$$
$$X(f) = Fourier\ Transform\{x(t)\} = 2.6000Sinc(6000t)$$
$$X(f) = 2.rect(\frac{f}{6000})$$

iv. Using trigonometric identities to solve for y(t) as shown in part "i",

$$Sin(\theta) = Cos(\frac{\pi}{2} - \theta)$$
$$\therefore Sin\left(10\pi.10^3t + \frac{\pi}{2}\right) can\ be\ written\ as\ Cos(\frac{\pi}{2} - 10\pi.10^3t + \frac{\pi}{2})$$
$$given\ that\ \cos(-\theta) = \cos\,(\theta)$$
$$Cos(10\pi.10^3t)$$
$$Rearranging\ the\ above\ expression\ as\ \cos(2\pi.5000t)\ in\ the\ form\ \cos\,(2\pi.f_0t)$$

v. Using the modulation property,

$$x(t).\cos(2\pi.f_0t) \longleftrightarrow \frac{1}{2}[X(f-f_0)+X(f+f_0)]$$

vi. Writing out the entire transform

$$S_{rov}(f) = Fourier\ Transform\{S_{rov}(t)\}$$
$$Fourier\ Transform\{x(t).\cos(2\pi.5000t)\}$$
$$\frac{1}{2}[X(f-5000)+X(f+5000)]$$
$$X(f-5000) = 2.rect(\frac{f-5000}{6000})$$
$$X(f+5000) = 2.rect(\frac{f+5000}{6000})$$
$$S_{rov}(f) = \frac{1}{2}[2.rect\left(\frac{f-5000}{6000}\right)+2.rect(\frac{f+5000}{6000})]$$

$$S_{rov}(f) = rect\left(\frac{f-5000}{6000}\right)+rect(\frac{f+5000}{6000})$$

vii. Plotting the magnitude spectrum of the rover,

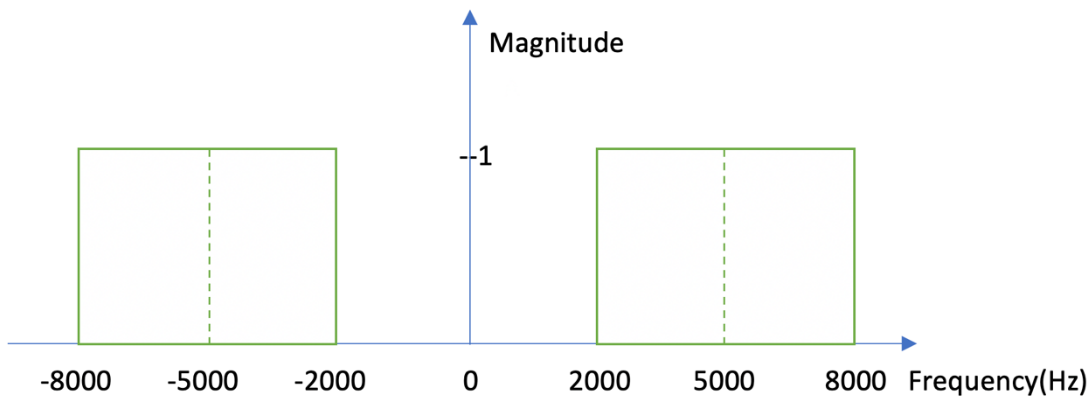


Fig 1.1 Graph of the magnitude spectrum of the rover

Determining the Transfer Function of The Ideal Filter

The graphs in Fig 1.0 and Fig 1.1 are sketched on the same axis so that the interference between the 2 signals could be identified.

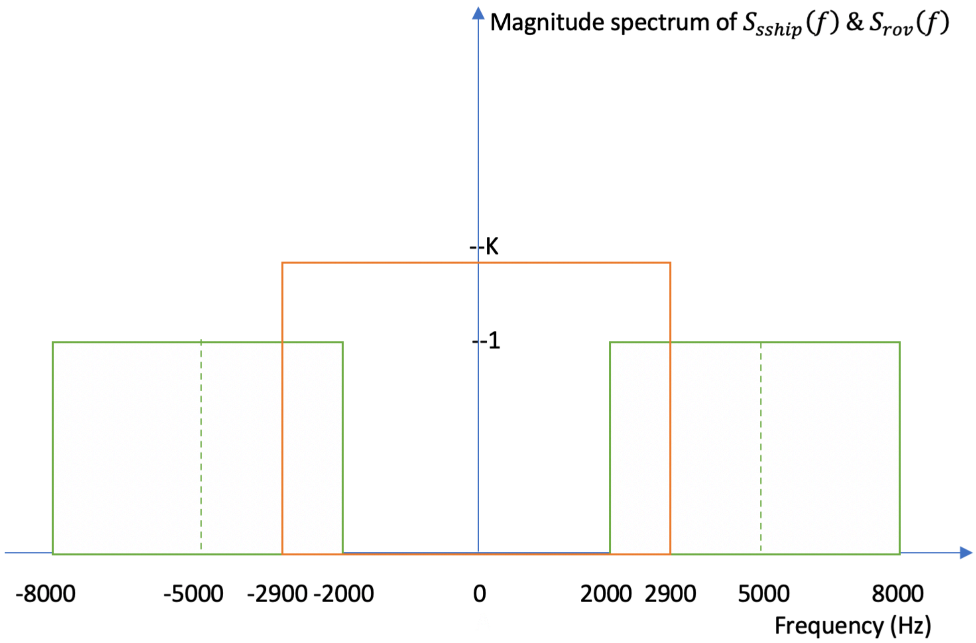


Fig 1.2 Graph of the magnitude spectrum of the rover and the spaceship

Observing the frequencies present in the graph we can determine that the interference from the spaceship is in the range of 2000 to 2900 Hz and a filter must be used to allow the rest of the signal to be extracted.

A high pass filter was used for this process as it allows the frequency above the cut off to pass through but attenuates all the frequencies below the cut off f_c [1]. From the graph the cut off was evaluated to be 2900Hz. This means that the interference from 2000 to 2900 Hz will be removed but frequencies in the range of 2900 to 8000 Hz.

The transfer function of the high pass can be determined from the generalised LTI model.

$$S_{rov} \longrightarrow \boxed{H_{filter}(f)} \longrightarrow S_{rovOut}$$

S_{rov} is the input transmission function and S_{rovOut} is the output transmission function.

The transfer function can then be found by using the equation shown below,

$$H_{filter}(f) = \frac{S_{rovOut}}{S_{rov}}$$

The transfer function of the ideal high pass filter can then be mathematically expressed in the equation as shown

$$H_{filter}(f) = 1 - rect(\frac{f}{5800})$$

The Final Frequency Domain of The Rover

The final frequency after the ideal high pass filter has been applied is given below.

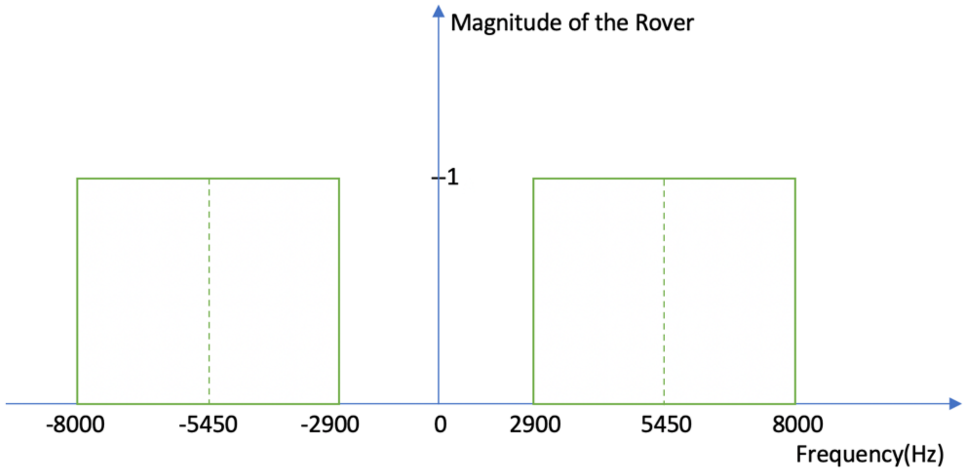


Fig 1.3 Graph of Final Frequency of Rover's Transmission function

Exploring the Impacts of the Filter

An ideal high pass filter is used for the above task as it has the ability to perform the task required. The figure below shows the characteristics of a high pass filter.

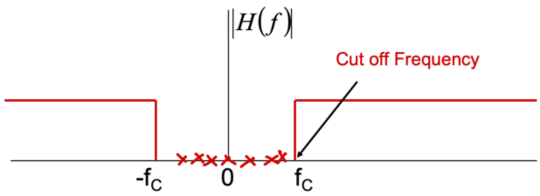


Fig 1.4 Ideal high pass filter characteristics [2]

When using an ideal high pass filter, it causes the image to be sharper and the quality will decrease. However, there are different effects when using a practical filter. The signal isn't completely attenuated in stop bands, they don't have an immediate transition between stop and pass bands but rather they have a roll off at the cut off frequency and they do have an effect on phase.[2]

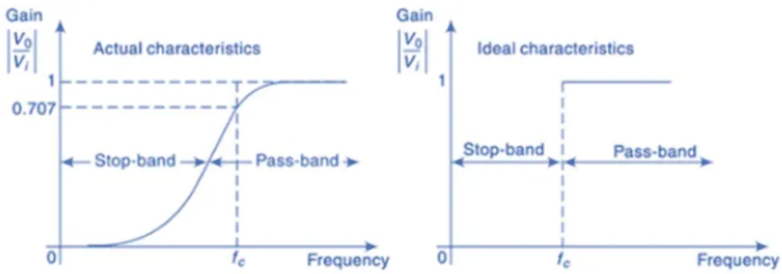


Fig 1.5 Practical high pass filter vs Ideal high pass filter [3]

Section 2 – EEG Signal analysis

Due to the longevity of the MARS-242 mission, BASA psychologists have recommended more quantitative tests of brain function for the astronauts aboard the mission. Electroencephalograms (EEG) scans of each astronaut have been collected via sensors within their spacesuits and multiplexed into a frequency division multiplexed (FDM) data stream that has been forwarded to mission command. The multiplexed signal is usually transmitted to a modular hardware system which de-constructs the FDM signal, removes frequency shifts, filters the signals and provides a clear output for each of the multiplexed streams of data. Due to the malfunction of the module, this section of the report will cover the steps of analysing the data streams manually and examining the final EEG signals.

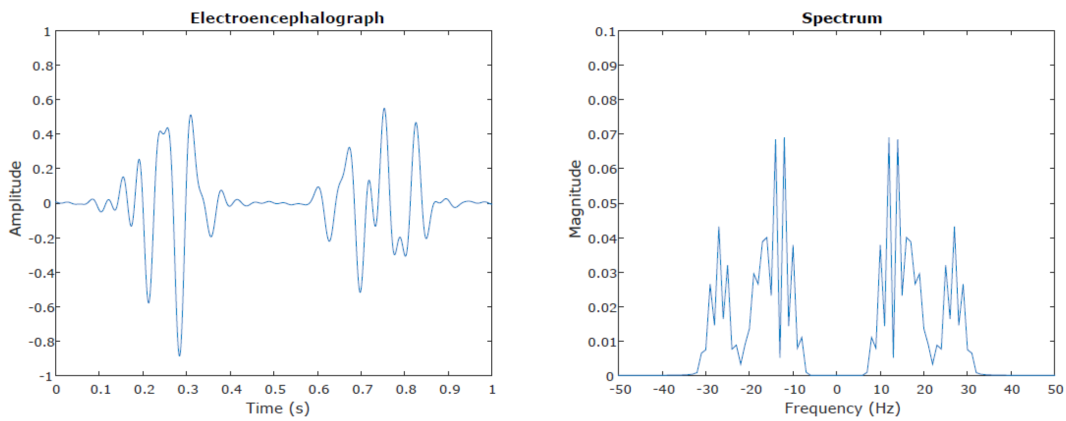


Fig 2.1 Electroencephalogram signal. Left: time domain, Right: Frequency Domain. [1]

Represent the spectrum analyser

he first step in analysing FDM signal is determining the sampling period of the multiplexed signal and computing the Fourier transform to manipulate the signal into the frequency domain, as shown below in figure 2.1.

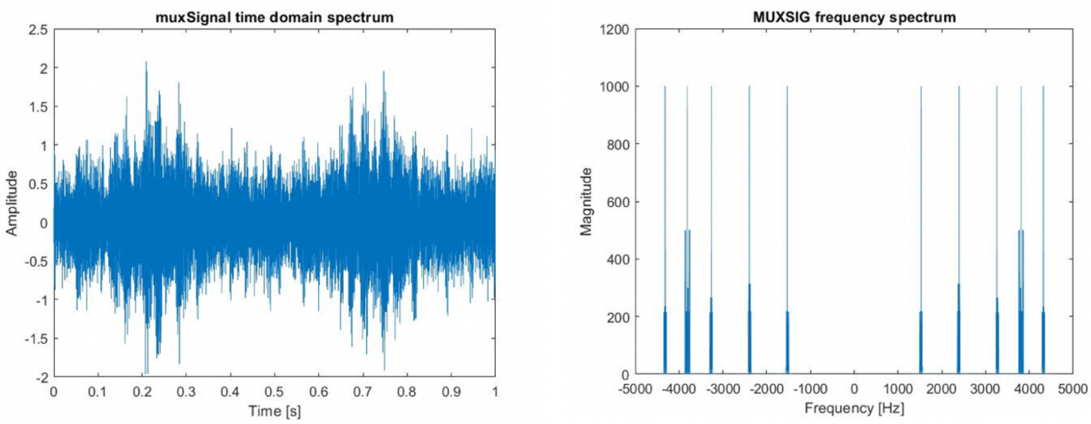


Fig 2.1 Received signal. Left: Time domain, Right: Frequency domain

Determine Demultiplexing parameters

The next step of the analysis involves determining the parameters required to demultiplex the data. This involves identifying the frequency shifts and storing them in a vector '*fshift'* and finding the corresponding magnitude and phase respectively. This is achieved by finding the locations of the peaks in the frequency domain signal after it has been scaled, storing these locations in the vector *'fshift'* and finding the magnitude and phase of the signal at these locations.

```
%Shifting and scaling MUXSIG%
MUXSIG = fftshift (MUXSIG)/fs;

%identifying frequency shifts%
%noting peak heights above 0.06 magnitude%

[PKS,LOCS] = findpeaks(abs(MUXSIG), 'MinPeakHeight', 0.06);
LOCS = LOCS(6:end);

%storing frequency shifts in vector 'fshift'%
fshift = k(LOCS);

%finding corresponding magnitude and phase%
Mag = abs(MUXSIG(LOCS))';
Phase = angle(MUXSIG(LOCS))';
```

Remove the Frequency Shifts for All Five Astronauts

After obtaining the corresponding frequency shift locations, phase and magnitude vectors, the frequency shifts must be removed from each signal in the multiplexed data stream. This is achieved by inputting this data, plus the corresponding time vector and time domain signal into the provided *'FDMDemux'* function which returns a matrix *'xdm',* in which each row contains the EEG data for one astronaut. As shown in the figure below, once the Fourier transform is computed on each data stream the results can be individually reviewed before applying any filtering.

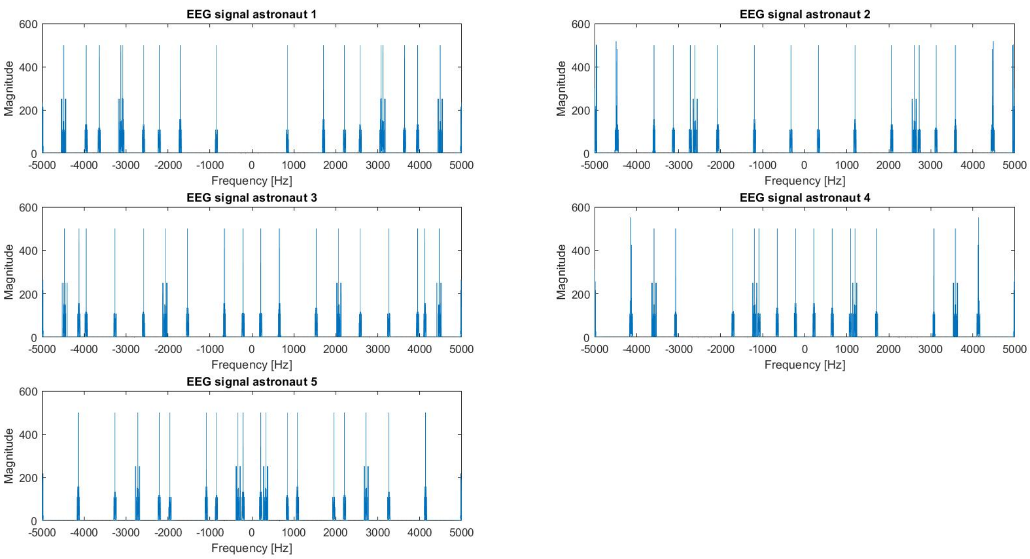
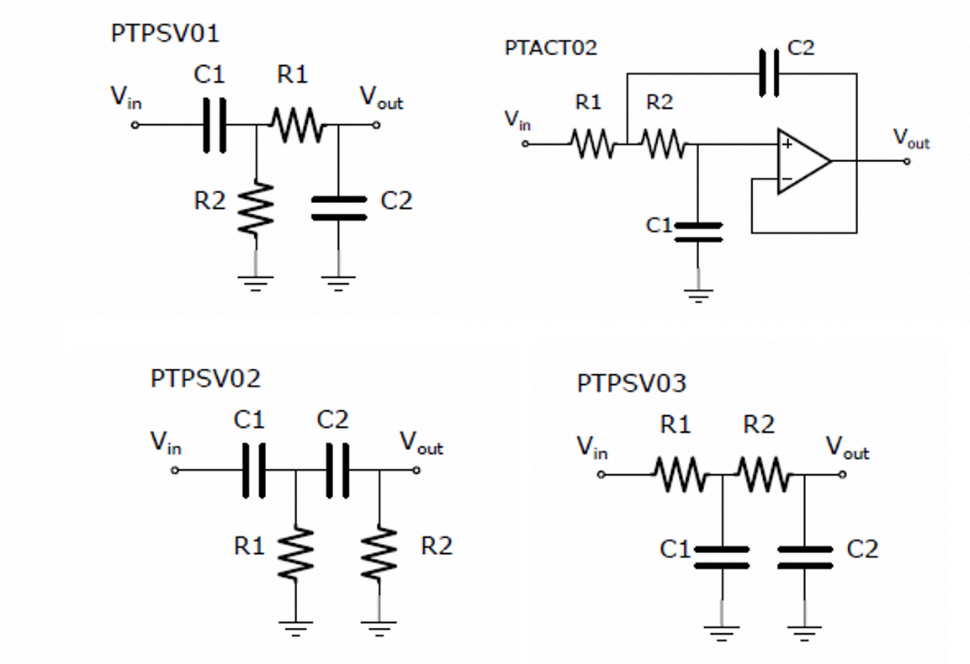


Fig 2.2 Individual EEG Signals in the frequency domain

Review

Some basic filters have been provided as shown below to apply to the signals. To decide on the suitability of the filters for the signals, some basic analysis must take place first.



Filter information	Filter 1: PTPSV01	Filter 2: PTACT02	Filter 3: PTPSV02	Filter 4: PTPSV03
Resistor 1 [Ohms]	191.29	1000	2295.65	130.34
Resistor 2 [Ohms]	299.16	1000	2295.65	130.34
Capacitor 1 [Farads]	1e-6	1.45966e-6	1e-6	1e-6
Capacitor 2 [Farads]	1e-6	2.91931e-6	1e-6	1e-6

Fig 2.3 Generic filters that have been provided including impedance function

Mathematical Analysis

By factorizing the transfer functions one by one and finding the roots of the numerator and denominator polynomials, this will return the poles and zeros for the transfer function and determine suitability for application. When analysing the stability of these filters, it is important to ensure the positions of poles are situated in the “stable” region of the s-plane as shown in the figure below. If the poles are in the “unstable” region of the s-plane, the system will never settle from an impulse or step response resulting oscillations or a magnitude that gradually increases to infinity.

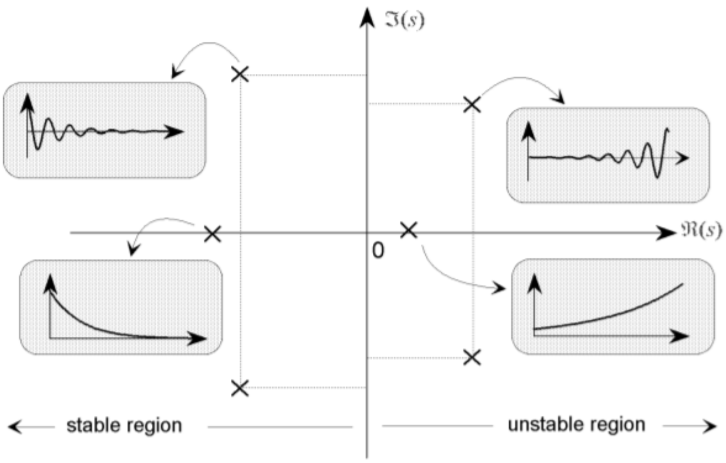


Fig 2.4 S plane, pole locations showing resulting step response. [5]

The 4 provided filters return the following values corresponding to their roots and poles:

	Transfer function (H(s))	Pole/zero locations
Filter 1	$\frac{3342.6546s}{(s+5227.6102)(s+3342.6546)}$	Zeros: $s_1 = 0, s_2 = 0$ Poles: $s_1 = -3.34e3,\ s_2 = -5.23e3$
Filter 2	$\frac{234675.9004}{(s+342.5463+j342.5463)(s+342.5462-j342.5463)}$	Zeros: NaN Poles: $s_1 = -343 + j343,$ $s_2 = -343 - j343$
Filter 3	$\frac{s^2}{(s+435.6056+j5.0085e-6)(s+435.6056-5.0085e-6)}$	Zeros: $s_1 = 0, s_2 = 0$ Poles: $s_1 = -436 + j5.01e - 6,$ $s_2 = -436 - j5.01e - 6$
Filter 4	$\frac{58865636.719}{(s+7672.3945+j0.00011826)(s+7672.3945-j0.00011826)}$	Zeros : NaN Poles : $s_1 = -7.67e3 + j0.000118,$ $s_2 = -7.67e3 - j0.000118$

Given the pole locations are all situated in the stable region of the s-plane, some further analysis must take place.

System Analysis

Investigating the systems further requires some more intuitive analysis. By viewing the LTI system response, analysis of the step response, impulse response and Bode plots can help narrow down the selection of a suitable filter. Upon analysis of each systems step response, filters 2 and 4 can be removed from consideration due to their amplitudes not settling after being hit with a step change. Upon further investigation of filters 1 and 3, their impulse response also settles at 0 making them both candidate filters still but their Bode plots require further analysis. Filter 1 appears almost as a bandpass filter but its magnitude (dB) never reaches -3[dB] (frequency cut-off point) making it not ideal to use. Filter 3 however, appears to be a high-pass filter and a good candidate to use to filter the EEG signals.

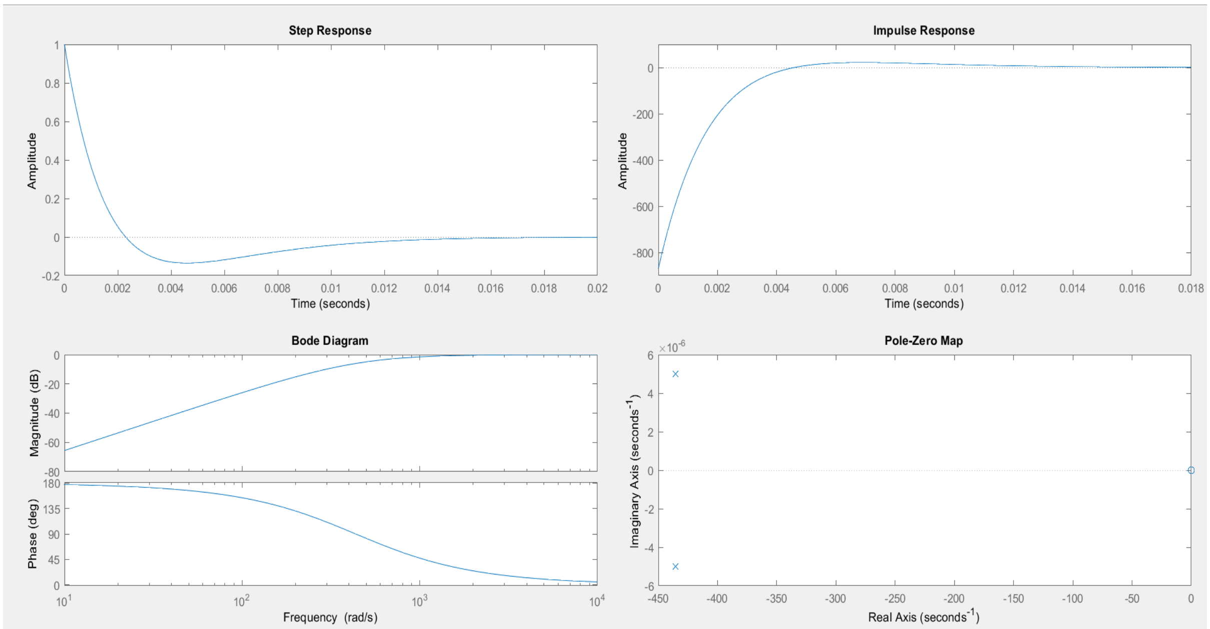


Fig 2.5 LTI system response of chosen filter

Recommend

After evaluating all candidate filters its settled that filter 3 (PTPSV02) is the most desirable filter to apply to the EEG signals. This can be shown clearly by the systems response to an impulse or step change with the amplitude settling to 0 and based on the numerator and denominator polynomials, the root locations are in the stable region of the s-plane. Implications associated with applying this filter will just result in not an ideal filtering of the signals which is due to cut off frequencies being pre-determined when applying a generic filter.

Filter the signals

Applying the LTI system to the EEG signals to filter the results requires creating a variable (h_f) which is the impulse response of the dynamic system (filter 3) in the time domain. Once the impulse response data is obtained, the Fourier transform can be performed on this signal and then further shifted and scaled in the frequency domain. Since both signals are in the frequency domain, the convolution equivalent of multiplication can be performed on the input signal (EEG) and the LTI system (filter 3) to return a desired output of filtered EEG signals as shown below.

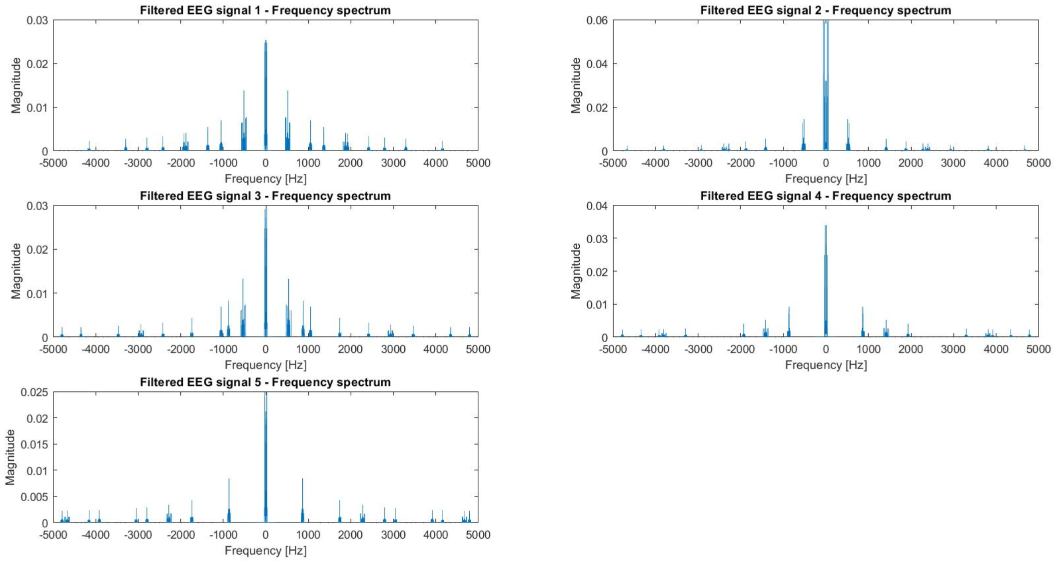


Fig 2.6 Filtered EEG signals in the frequency domain

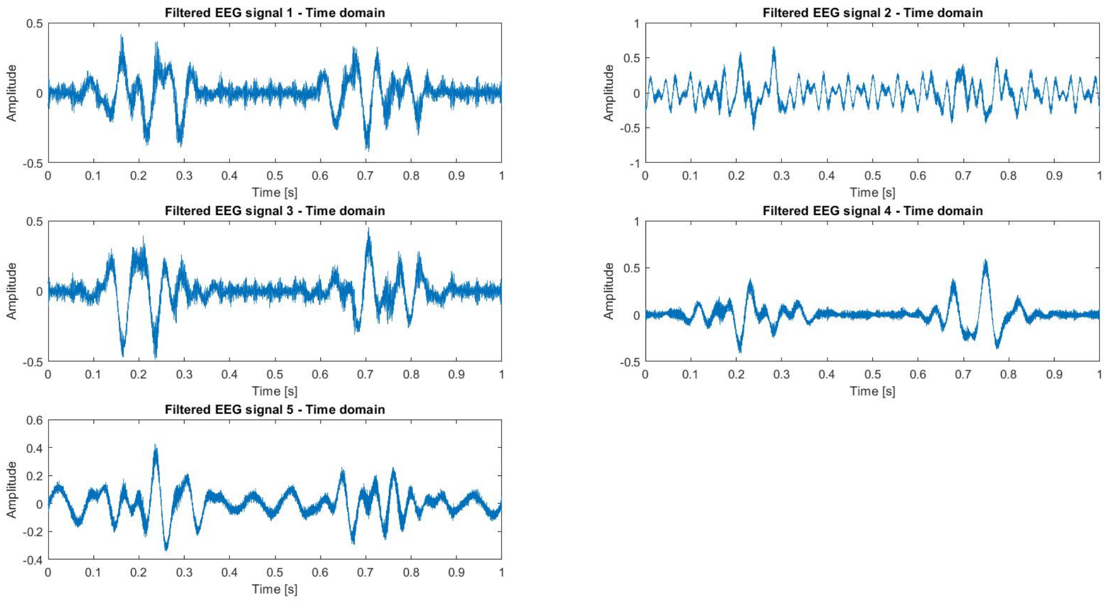


Fig 2.7 Filtered EEG signals in the time domain

As clearly shown in the figures above, the EEG signals still require further filtering which will be covered further down in this report.

Equivalence with convolution

While multiplication of an LTI system with an input provides an output in the frequency domain, it can also be proven that these are correct representations of the signals by performing convolution in the time domain of the EEG signals and the LTI system. As shown in the figure below, when convoluted in the time domain the returned signal is identical to the returned signal 1 from further up in this report.

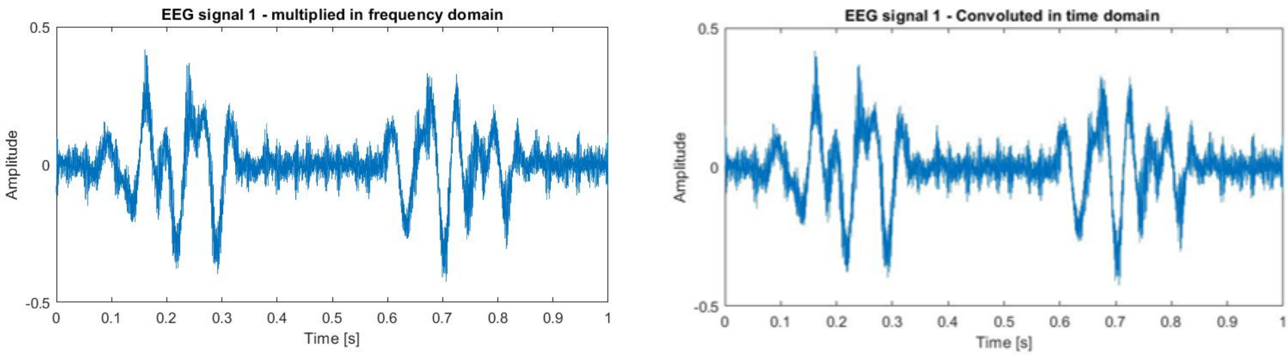


Fig 2.8 EEG signals in frequency (left) and time(right) domain

```
%impulse response
impresp = h_f;

%convoloution of signalS
Conveq = conv(impresp, xdm(1,:))/fs;
%removing extra samples
Conveq = Conveq(1:length(xdm));

figure (6)
subplot(2,2,1)
plot (t, eeg(1,:))
xlabel('Time [s]')
ylabel('Amplitude')
title('EEG signal 1 - multiplied in frequency domain')
subplot(2,2,2)
plot(t, Conveq)
xlabel('Time [s]')
ylabel('Amplitude')
title('EEG signal 1 - Convoluted in time domain ')
```

Compare

When comparing the time domain EEG signals with the example provided in the task brief its clear the signals need further filtering to return an accurate result. Some further issues can also be seen in signals 2 and 5 not following a similar trend within the plot which can be evaluated once the signals are filtered accurately

Digital Denoising

Finally, applying an ideal filter to the signals will allow the BASA medical team to evaluate the EEG signals of the astronauts confidently. By examining the shifted and scaled plots in frequency domain, it's established a bandwidth of 120kHz will be sufficient for filtering all signals. By creating an ideal rectangular function consisting of value 1 for the bandwidth and zero elsewhere, the signals can be filtered through and return an accurate representation of the EEG signal by multiplying the signal by the rectangular function. Upon observation of the filtered signals, it became apparent that there was a couple of single frequency additive noises in EEG signal 2 still. After modifying the rectangular function to cut out the additive noise signals, an accurate representation of the EEG signals is returned as shown in the figure below.

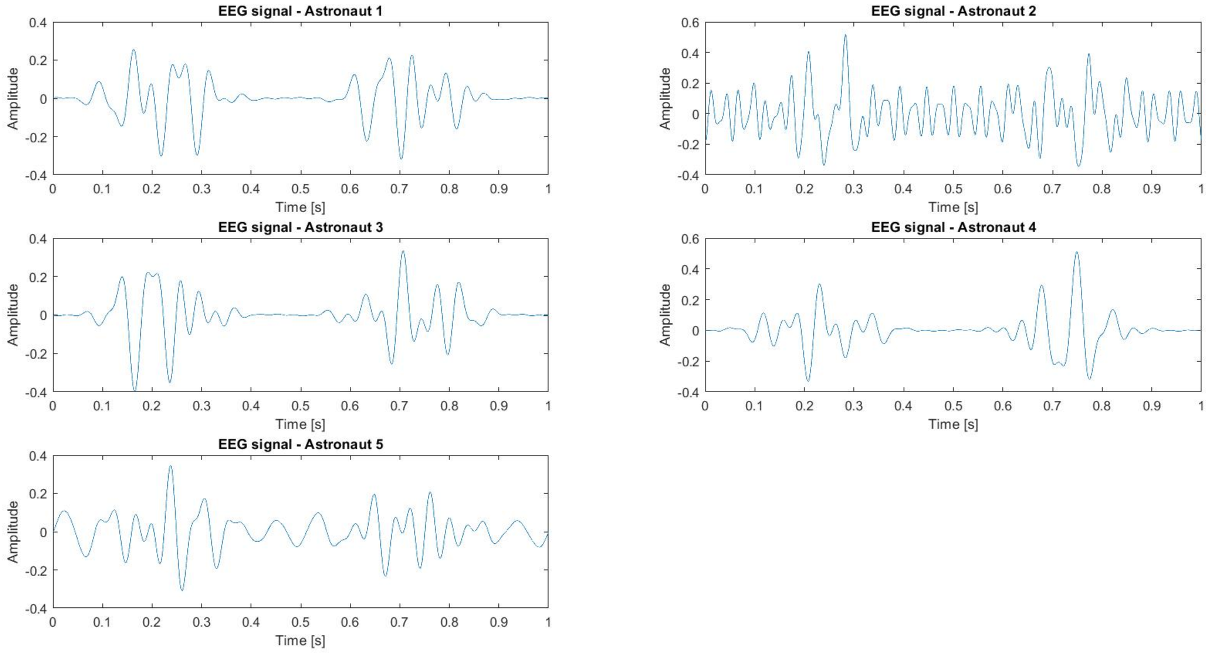


Fig 2.9 EEG signals of the astronauts after noise is removed.

Visual Analysis

While it appears astronauts 1,2 and 4 are quite uniform and stable, be advised that astronauts 2 and 5 are possibly in a significantly suboptimal mental state and further medical attention may be required.

Section 3 – Choosing a landing Site

The astronauts are now nearing Mars and will need to identify a suitable site to land. To help identify a suitable landing site the surface-based rover has identified potential landing sites and has transmitted them to the spaceship. However, the transmission channel between the rover and spaceship has produced periodic and bandlimited random noises, which as a result rendered the image making it useless. Therefore, as BASA engineers we have been assigned the task to de-noise the images and recommend a suitable landing site. This section of the report will outline the processes taken and a step-by-step analysis in de-noising these images through the use of Fourier Series and Fourier Transforms.

Viewing Noisy Image

The first step is to view the rendered image that needs to de-noised and analysed. The image data is stored in rows inside a matrix called *'sig'*. 'Figure 3.1' shows the rendered image of the first image data through using *'imshow'* and *'reshape'* functions. The code used to implement this has also been attached below.

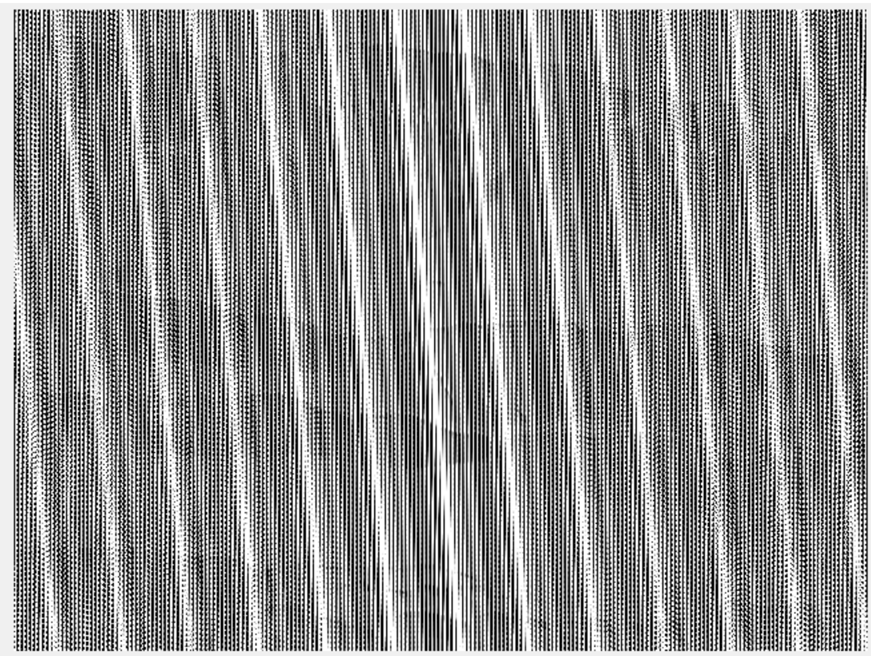


Fig 3.1

```
data = matfile('A2P3Data.mat');
figure(1)
title('Corrupted image1');
first_image_data = data.sig(1,:);
imshow(reshape(first_image_data, 480, 640));
```

Reference Vectors

The next step is representing the rendering in time and frequency domains. To do this, time and frequency vectors are created. The fs is set to be 1000 as the data received is at 1000 pixels per second.

```
fs = 1000;
Ts = 1/1000;
% Creating the time vector
t = linspace(0, Ts * length(first_image_data),
length(first_image_data) + 1); t(end) = [];
% Creating the Frequency vector
k = linspace(-fs/2,fs/2, length(first_image_data) + 1); k(end) = [];
```

Visualise the received signal

To understand how to proceed with the signal analysis the signals are plotted on the frequency and time domain using the vectors created in the previous part.

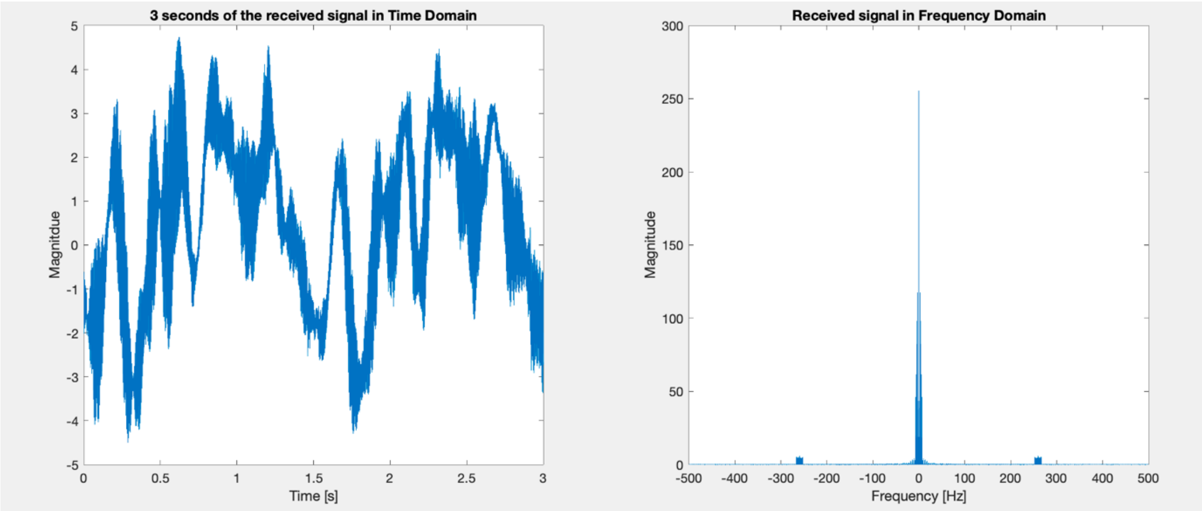


Fig 3.2 Left: Time Domain plot, Right: Frequency Domain plot

Observing the plots we can identify that the periodic noise occurs around 0.4 and continues till 1.8 seconds. In the Frequency plot we can identify that the bandlimited noise occurs between 250 and 270Hz. This is zoomed in on and has been attached below.

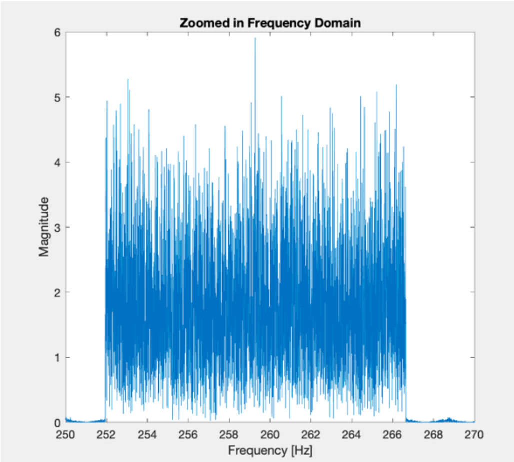


Fig 3.3 Zoomed in plot

Estimate your periodic noise

The periodic noise in the image can be accurately removed if the period is accurately estimated. A set of possible values for the period of the corrupting noise has been identified, and these values are stored in a vector called *'canditateT'*.

candidateT	[1392,1191,1360,1630,1469,1591,1172,1460,1270,1314]

Fig 3.4 Possible values stored in candidateT

These values are then compared to the approximated period identified earlier to determine the period, and as a result, the value of 1469 milliseconds or 1.469 seconds has been identified as the closest to the approximated period. Next *'estimateNoise'* function is implemented to estimate the noise profile through using the estimated period, which is then stored in a variable called *'Noisesig'*. The code implementation of this is shown below.

```
T = candidateT(1,5);
Noisesig = estimateNoise(sig(1,:), T);
j = t(1:2*length(Noisesig));
figure(3)
%plot(j, Noisesig, 'r');
y = repmat(Noisesig.', [1, 2]);
plot(j, y,'r');
hold on
plot(t, first_image_data, 'b.');
xlim([0 3])
```

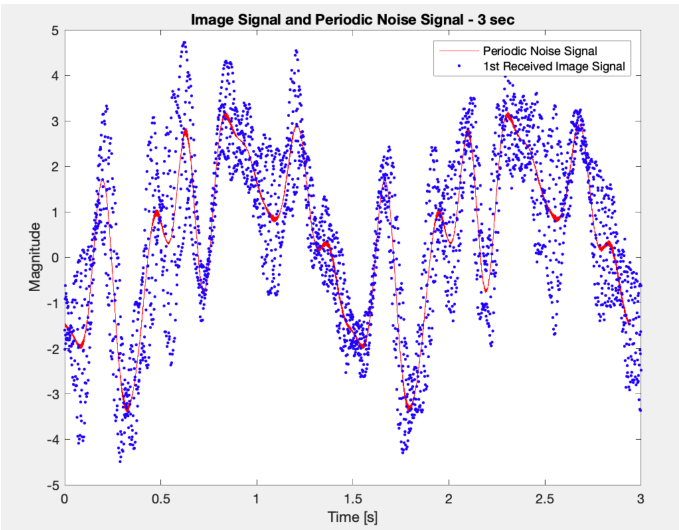


Fig 3.5 Plot of the correctly selected periodic noise signal

Model the periodic noise

The noise signal can be represented by determining the Fourier coefficients within *'Noisesig'*. To determine the coefficients, either the Trigonometric or Complex Exponential Fourier Series can be used. It has been decided that the Trigonometric Fourier Series will be used to determine the coefficients a0, an and bn. The code used to implement this has been attached below.

```
T = 1.469;
f = 1/T; % Fundamental frequency
a0 = (1/T).*sum(Noisesig'.*Ts);
j = t(1:length(Noisesig));

% Using Fourier Series
FTSignal = a0;
N = 6;
for n = 1:N
    an = (2/T).*sum(Noisesig'.*cos(2.*pi.*f.*n.*j))*Ts;
    bn = (2/T).*sum(Noisesig'.*sin(2.*pi.*f.*n.*j))*Ts;
    FTSignal = FTSignal + an.*cos(2.*pi.*f.*n.*j) +
bn.*sin(2.*pi.*f.*n.*j);
end
```

a0	0.5261
an	0.0562
bn	-0.6018

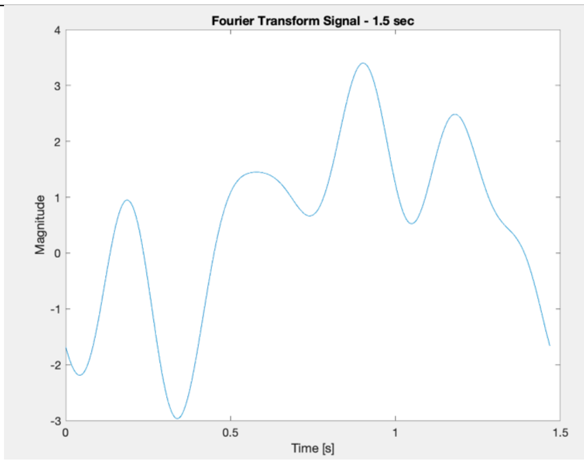


Fig 3.6 Data produced from code

Bias 3.6

Mission control has informed the team that the mean periodic noise is zero. This means that in the Fourier series expression the term a0 can be modelled as zero.

Generate the approximation

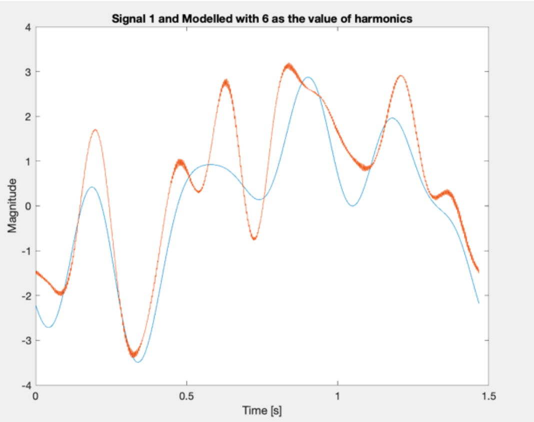
After a0 has been made zero due to the DC component present in the '*Noisesig'* the approximation is regenerated with an increased number of harmonics.

```
Noisesig_fs = 0; %initialise Noisesig_fs
N = 20; % Increased number of harmonics

for n = 1:N
    an = (2/T).*sum(Noisesig'.*cos(2.*pi.*f.*n.*j))*Ts;
    bn = (2/T).*sum(Noisesig'.*sin(2.*pi.*f.*n.*j))*Ts;
    Noisesig_fs = Noisesig_fs + an.*cos(2.*pi.*f.*n.*j) +
bn.*sin(2.*pi.*f.*n.*j);
end
% To calculate how many time the signal was repeated
Rep = ceil(length(t) / 1469);
Noisesig_fs = repmat(Noisesig_fs, [1, Rep]);
Noisesig_fs = Noisesig_fs(1: length(t));
```

Compare the approximation

The generated approximation was thought to be inefficient therefore the number of coefficients is increased to 20. Once this was done the graph was produced again and it can be observed a much clearer signal is obtained when increased to 20 harmonics.



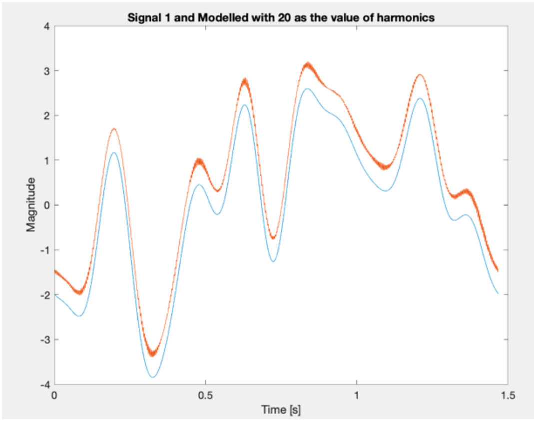


Fig 3.7 Left: Plot with 6 harmonics, Right: Plot with 20 Harmonics.

De-noise

Using the knowledge gained in BASA training the image can now be de-noised. The result of this is then stored in the first row of the image matrix called 'im1'. 'Figure 3.8 shows the recovered image and spectrum.

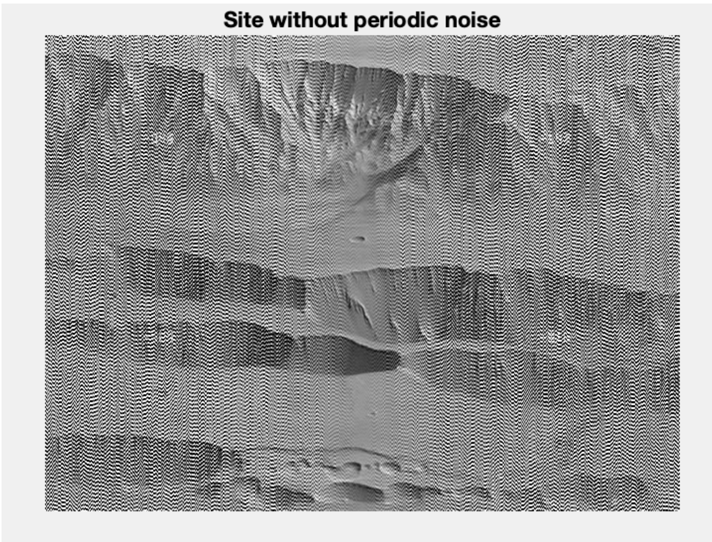


Fig 3.8 Image without periodic noise

Remove the bandlimited random noise

From the Fig 3.8 we can observe that the bandwidth noise is still present since we identified that the noise was present between 250 and 270Hz a filter was created to remove the signals in this range. Code is implemented which uses Fourier transform to change the signal to frequency domain and once the bandwidth noise is removed it's converted back to time domain. The signal isn't stored in the required variable (explained in following section).

```
De_noisefreq = fft(De_noise);
De_noisefreq = fftshift(De_noisefreq)/fs;
sigfilt = zeros(1,length(De_noisefreq));

idealfilt = ones(1, length(sigfilt));
idealfilt(-270 < k & k <-250) = 0;
idealfilt(250 < k & k < 270) = 0;

sigfilt = idealfilt.*De_noisefreq;

figure(7)
im1 = ifftshift(sigfilt)*fs;
im1 = ifft(im1);
```

Once this code has been implemented, we are able to receive a clear signal image. This has been attached below.

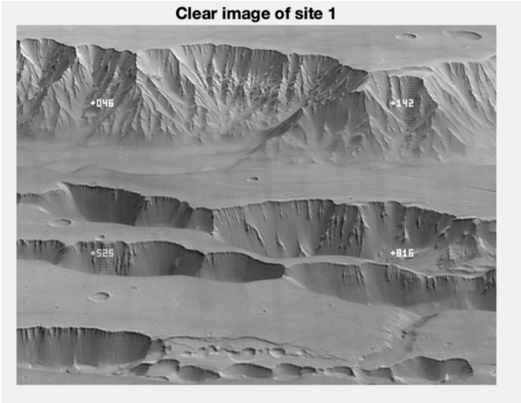


Fig 3.9 Clear image of site 1

Choose a site

Even though it was stated by Mission control that the periodic noise profile is consistent when code was implemented there was slight irregularities in the image. Custom filters are therefore implemented, and the filters are applied to the images to produce clear landing sites. The code for the filters has been attached below along with the four landing sites.

```
im2 = zeros(4, length(sig));

im2(1,:) = im1;
im2(2,:) = sig(2,:) - Noisesig_fs;
im2(3,:) = sig(3,:) - Noisesig_fs;
im2(4,:) = sig(4,:) - Noisesig_fs;

% Performing Fourier Transform
im1f = zeros(3,length(im2));
im1f(1,:) = fft(im2(2,:));
im1f(1,:) = fftshift(im1f(1,:))/fs;
im1f(2,:) = fft(im2(3,:));
im1f(2,:) = fftshift(im1f(2,:))/fs;
im1f(3,:) = fft(im2(4,:));
im1f(3,:) = fftshift(im1f(3,:))/fs;
```

```
% Removing the custom bandwidth
finalfilt1 = ones(1, length(sigfilt));
finalfilt1(-245 < k & k < -225) = 0;
finalfilt1(225 < k & k < 245) = 0;

finalfilt2 = ones(1, length(sigfilt));
finalfilt2(-240 < k & k < -220) = 0;
finalfilt2(220 < k & k < 240) = 0;

finalfilt3 = ones(1, length(sigfilt));
finalfilt3(-255 < k & k < -235) = 0;
finalfilt3(235 < k & k < 255) = 0;


filt_im1 = zeros(3,length(im1f));

filt_im1(1,:) = im1f(1,:).*finalfilt1;
filt_im1(2,:) = im1f(2,:).*finalfilt2;
filt_im1(3,:) = im1f(3,:).*finalfilt3;


%Chaning back to time domain
im1t(1,:) = ifftshift(filt_im1(1,:))*fs;
im2(2,:) = ifft(im1t(1,:));

im1t(2,:) = ifftshift(filt_im1(2,:))*fs;
im2(3,:) = ifft(im1t(2,:));

im1t(3,:) = ifftshift(filt_im1(3,:))*fs;
im2(4,:) = ifft(im1t(3,:));
```

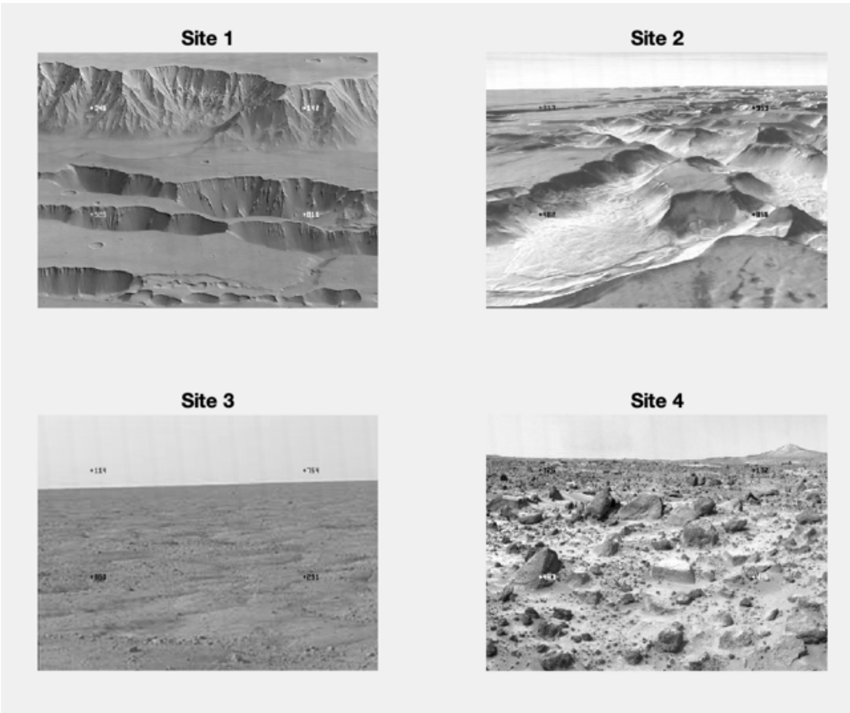


Fig 3.10 Landing sites in one image

Looking at the sites in Fig 3.10, Site 3 looks like an optimal landing zone. Site 1 and 2 seems to have an uneven terrain and site 4 seems the worst as it has rocks which will damage the shuttle.

Resolution

Site	Top Left	Top Right	Bottom Left	Bottom Right
1	046	142	525	816
2	317	953	482	856
3	114	752	860	291
4	326	132	437	426

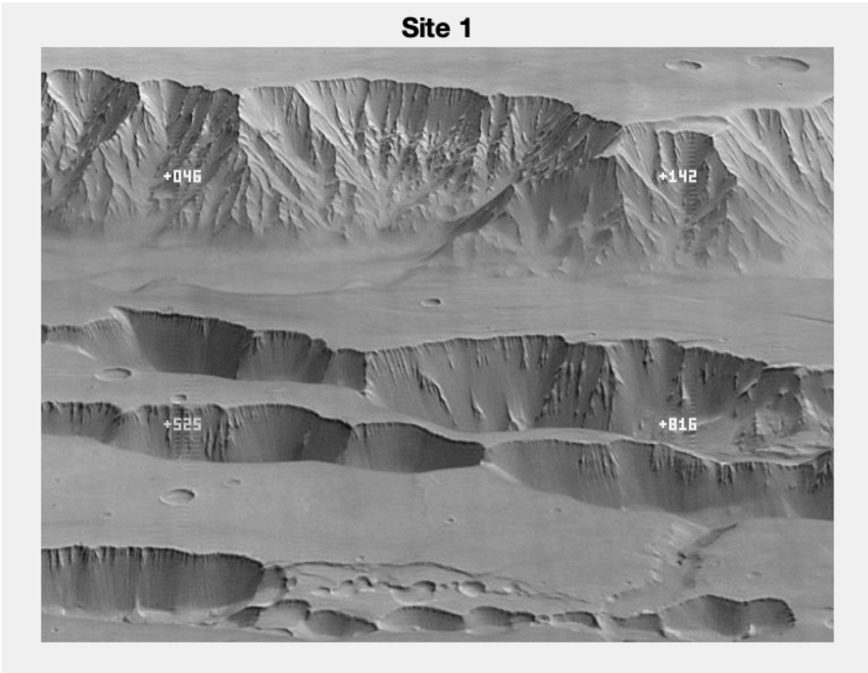


Fig 3.11 Landing site 1

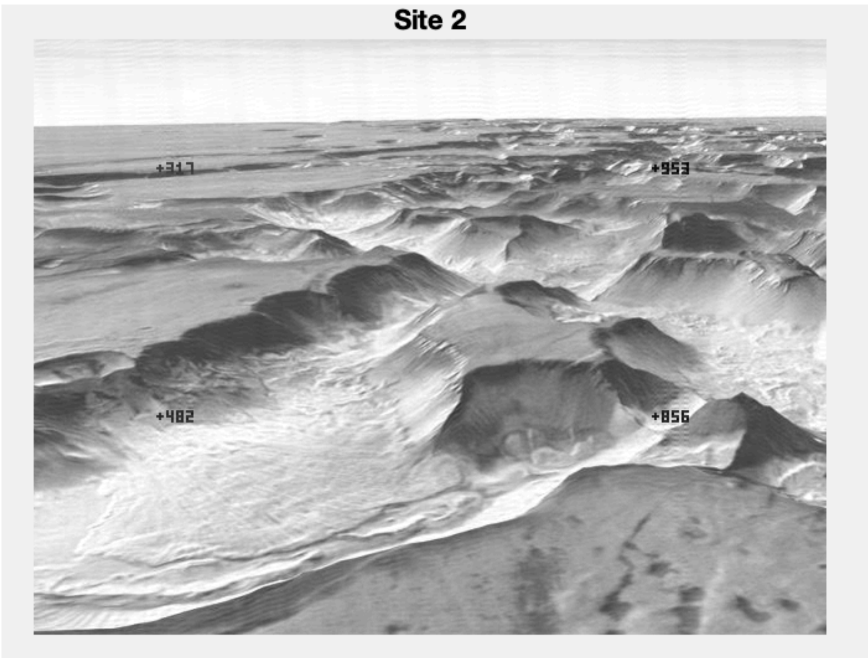


Fig 3.11 Landing site 2

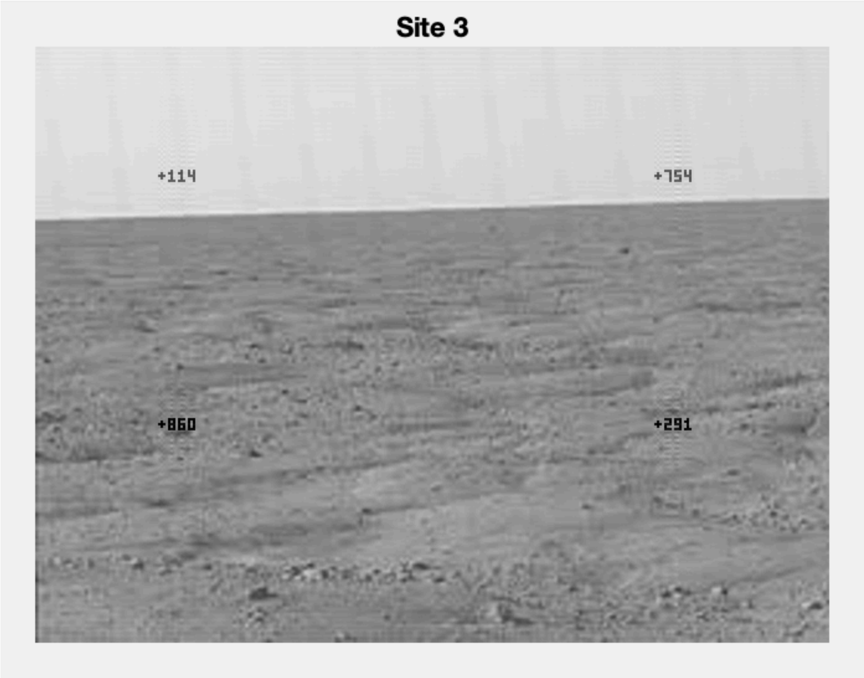


Fig 3.11 Landing site 3

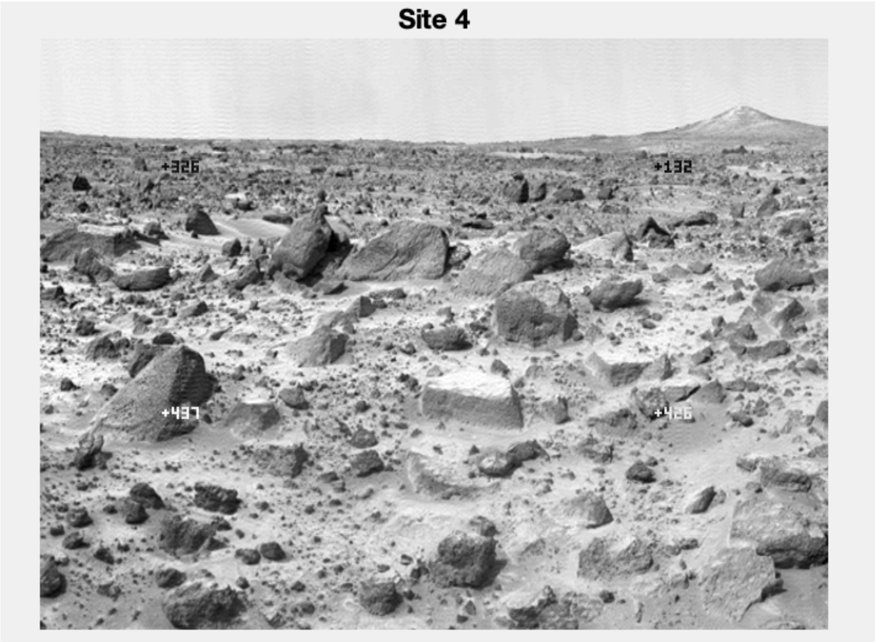


Fig 3.12 Landing site 4

Reflection

The assignment builds upon the technical skills in the earlier assignments and gives a more practical approach to using the skills learned in signal analysis. Fourier series, Fourier transforms and LTI systems were implemented in MATLAB we were allowed to make appropriate choices as to how to implement them.

As usual the feedback given from previous assignments was helpful in understanding how we can approve. The tutorials and supplementary videos helped to understand concepts better and help was available from the tutors whenever a difficult problem was faced.

Since this was a group assignment it also built upon our professional skills such as teamwork, effective communication and time management. This assignment managed to bring about a "real world" experience which we believe will be beneficial to us as engineers.

References:

[1] Assignment 2 Brief. Available: https://blackboard.qut.edu.au/bbcswebdav/pid-8717307-dt-content-rid-34199132_1/courses/EGB242_20se2/Assignment2-Brief%287%29.pdf

[2] Week 8 Annotated Lecture slides. Available: https://blackboard.qut.edu.au/bbcswebdav/pid-8717289-dt-content-rid-33978089_1/courses/EGB242_20se2/EGB242%20-%20Lecture%208E-Annotated.pdf

[3]"Filters - Classification, Characteristics, Types, Applications & Advantages", *electricalfundablog.com*, 2020. [Online]. Available: https://electricalfundablog.com/filters-classification-characteristics/. [Accessed: 13- Oct- 2020].

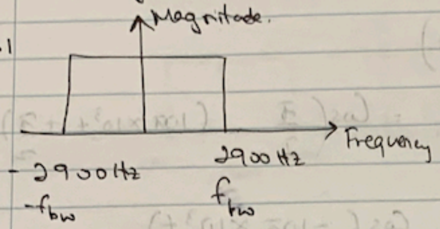
[4] "What is High Pass Filter? Its response curve, types, design | SM Tech", *SM Tech*, 2020. [Online]. Available: https://somanytech.com/high-pass-filter/. [Accessed: 14- Oct- 2020].

[5] *Web.mit.edu*, 2020. [Online]. Available: https://web.mit.edu/2.14/www/Handouts/PoleZero.pdf. [Accessed: 16- Oct- 2020].

Appendix A: Hand calculations

Assignment 2.

1.1



1.1 ⟶ (f_{bw}) ⟶ 2900Hz

1.2 ⟶ A ⟶ 1200

$S(f) \longrightarrow B \cdot \text{rect}\left(\frac{f}{T}\right)$

$T = 2 \times f_{bw}$

$T = 2 \times 2900$

$T = 5800$

1.2 $S_{rov}(t) = A\,\text{Sinc}(6000t)\,\text{Sin}\left(10\pi \times 10^3 t + \frac{\pi}{2}\right)$

where $A = 12{,}000$

$S_{rov}(t) = 12000\,\text{Sinc}(6000t) \cdot \text{Sin}\left(10 \times 10^3 t + \frac{\pi}{2}\right)$

$12000\,\text{Sinc}(6000t) \rightsquigarrow 2 \times 6000\,\text{Sinc}(6000t)$

$F\,\text{Sinc}(Ft)$

$F = 6000$

Using the transform table

$F\text{Sinc}(Ft) \xleftrightarrow{F} \text{rect}\left(\frac{f}{F}\right)$

$X(f) = \mathcal{F}\{x(t)\} = 2 \times 6000\,\text{Sinc}(6000t)$

$X(f) = 2 \times \text{rect}\left(\frac{f}{6000}\right)$

for $y(t)$ using trig identities.

$\sin(\theta) = \cos\left(\frac{\pi}{2} - \theta\right)$

$\therefore \sin\left(10\pi \times 10^3 t + \frac{\pi}{2}\right) = \cos\left(\frac{\pi}{2} - \left(10\pi \times 10^3 t + \frac{\pi}{2}\right)\right)$

$= \cos(-10\pi \times 10^3 t)$

using $\cos(-\theta) = \cos\theta$

$\Rightarrow \cos(10\pi \times 10^3 t)$

Rearranging the terms $\cos(2\pi \times 5000t)$ $\Big\}$ $\cos(2\pi f_0 t)$

Using the modulation property.

$x(t) \cdot \cos(2\pi f_0 t) \xleftrightarrow{F} \frac{1}{2}\left[X(f - f_0) + X(f + f_0)\right]$

$f_0 = 5000$

$S_{row}(f) = F\{S_{row}(t)\}$

$= F\{x(t) \cdot \cos(2\pi \cdot 5000t)\}$

$= \frac{1}{2}\left[X(f - 5000) + X(f + 5000)\right]$

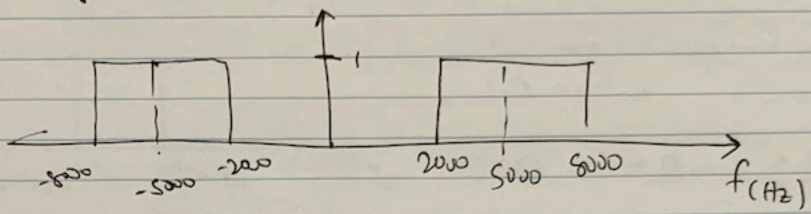
$X(f - 5000) = 2\,\text{rect}\left(\frac{f - 5000}{6000}\right)$

$X(f + 5000) = 2\,\text{rect}\left(\frac{f + 5000}{6000}\right)$

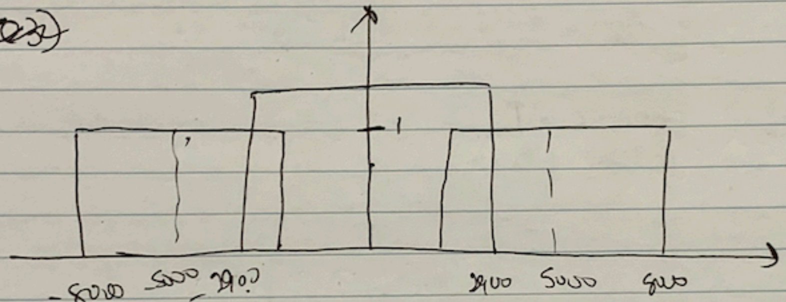
$S_{row}(f) = \frac{1}{2}\left[2\,\text{rect}\left(\frac{f - 5000}{6000}\right) + 2\,\text{rect}\left(\frac{f + 5000}{6000}\right)\right]$

$= \text{rect}\left(\frac{f - 5000}{6000}\right) + \text{rect}\left(\frac{f + 5000}{6000}\right)$

Plotting the mag spect.



1.3 (Q3?)

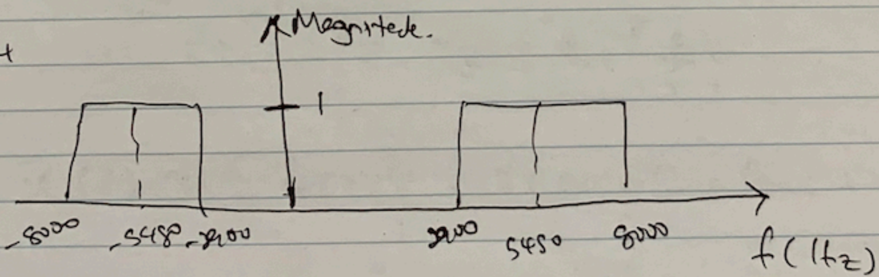


2900 is the cut off

$$H_{filter}(f) = \frac{S_{nv}(f)'}{S_{nv}(f)}$$

$$H_{filter}(f) = 1 - rect\left(\frac{f}{5800}\right)$$

1.4



1.5