BASA: Misson to Mars

De-Noising a Received Signal
with Periodic Noise Interference

Darcy Truman - n10517651

EGB242: Signal Analysis

August 12, 2021

1 Introduction

This report outlines the method and process of de-noising a speech signal received on the Mars-242 spaceship. The signal has been corrupted by an additive noise process comprised of two different periodic signals one in each half of the transmission. The complementary MATLAB code is appended to end of the report in Appendix, in section 6.5.

2 Received Signal

The noise signal received is shown below in Figure 1. It can be seen that signal has been corrupted with two different, periodic noise signals, one in each half of the transmission. The original speech was recorded at a rate of 44100 samples per second for 20 seconds. When listened to, the information in the speech signal is incoherent and undecipherable.

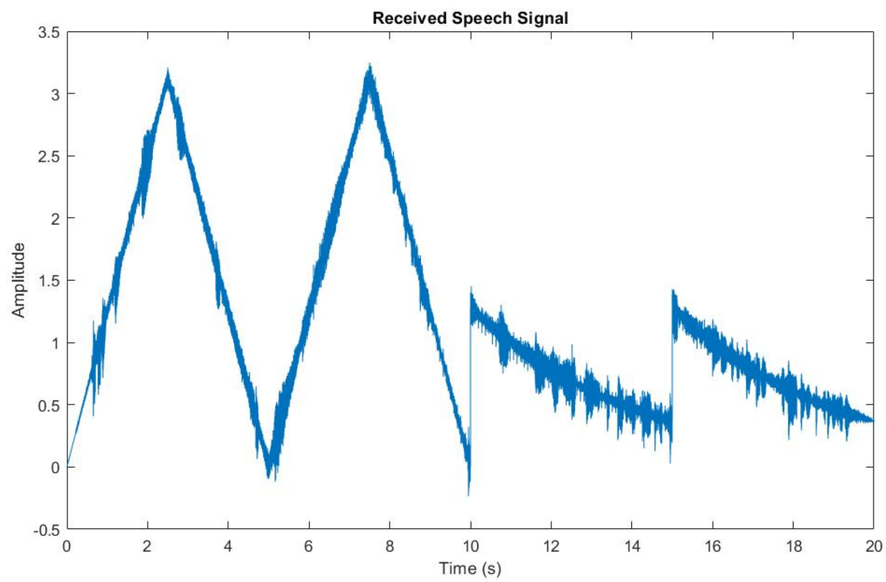

Figure 1: Speech Signal

The two signals that have been identified as the interfering with the transmission are shown below. With $s_3(t)$ corrupting the first half of the signal and $s_2(t)$ corrupting the second half of the signal.

$$s_3(t) = \begin{cases} \frac{5t}{4} & 0 \leq t < 2.5 \\ \frac{-5t+25}{4} & 2.5 \leq t < 5 \end{cases}$$

$$s_2(t) = e^{-\frac{t-1}{4}} \quad 0 \leq t < 5$$

The interference signals were identified using two factors; shape and period. As all functions started from 0 looking at the offset gave no indication to the identity of the noise signal. The first half of the noise signal has a distinct triangular shape which identified it as the piece-wise function $s_3(t)$. In addition to its shape, the period of $s_3(t)$ also fits with the start of the downward slope coinciding with the start of the second function in the piece-wise function. The four periods of the first noise signal comprises the first half of the speech signal. The second half of the noise signal is a downward sloping function which identified it as $s_2(t)$ as the other candidate function increased over the same period. As with the first function identified the two periods each 5 seconds long would comprise half of a 20 second noise signal.

The two additive noise functions are displayed below over one of there periods in figure 2 along with the MATLAB code used to generate the data.

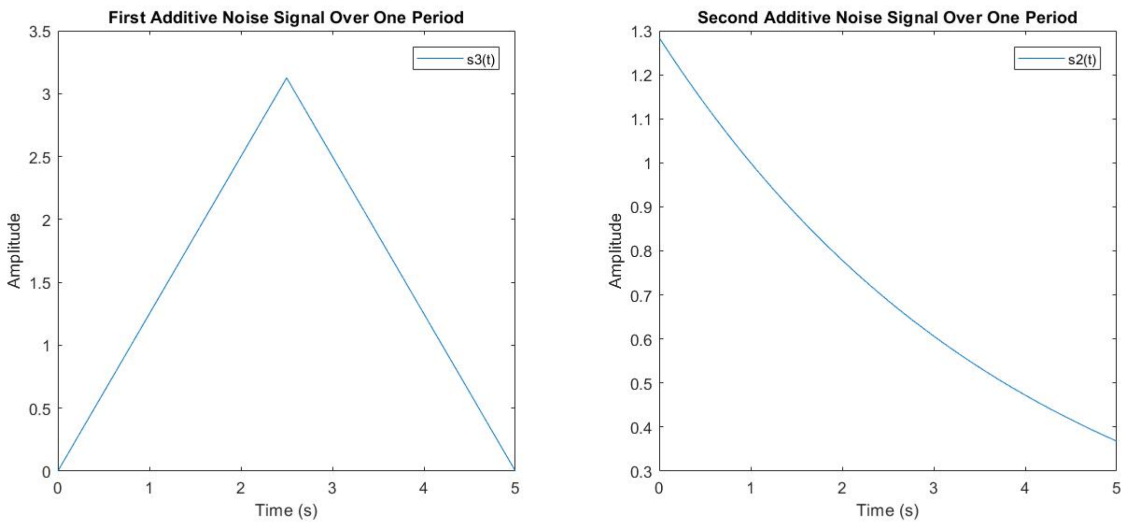

Figure 2: Noise Signals Over One Period

```
%%Noise Signals Plotted Over 1 Period
%period
T = 5;
%time vector
tPeriod = linspace(0,T, samples + 1);
tPeriod(end) = [];
%half periods to calculate piecewise function
tPeriodHalf1 = linspace(0, T/2, samples/2 + 1);
tPeriodHalf1(end) = [];
tPeriodHalf2 = linspace(T/2, T, samples/2 + 1);
tPeriodHalf1(end) = [];
%piecewise functions of s3(t)
PW1 = A * tPeriodHalf1 / 4;
PW2 = (- A * tPeriodHalf2 + 5 * A) / 4;
%calculating the additive noise vectors
additive_noise_first = [PW1 PW2];
additive_noise_second = exp(-1 * (tPeriod - B) / 4);
```

The second periodic noise signal, $s_2(t)$ has been determined to be neither an odd or even function. To show this, the mathematical definitions of even and odd functions can be used. A property of an even function is that it obeys the following rule.

$$x(t) = x(-t)$$

This rules states that for any value of time the function will equal the same value at negative value of that time. It represents symmetry across the y axis. Using this definition and substituting in $s_2(t)$ and an arbitrary value of time, it can be shown that this rule doesn't hold.

$$e^{-\frac{1-1}{4}} = e^{-\frac{(-1)-1}{4}}$$

$$1 \neq 1.65$$

As $s_2(t)$ doesn't obey the rule it can be determined that it isn't an even function.A similar rule for odd functions exists. At a negative value of time the function will equal the negative of that same value at same but positive value of time. It represent rotational symmetry after a 180 degree turn or symmetry flipped across the x, then y axis or vice-versa.

$$x(-t) = -x(t)$$

The same method can be used.

$$e^{-\frac{(-1)-1}{4}} = -e^{-\frac{1-1}{4}}$$

$$-1 \neq -1.65$$

As the rule doesn't hold, this proves that it is not an odd function. Therefore, the function $s_2(t)$ is neither an odd or even function.

3 Analysis

The Complex Fourier Series was chosen to model the second periodic noise signal, $s_2(t)$. The Complex Fourier Series was chosen over the Trigonometric Fourier Series as it requires less terms to evaluate meaning less code needs to be written and tested decreasing the chance of errors. Also as it is an exponential function making hand calculations to verify easier.

The harmonics: $-5 \leq n \leq 5$ were used in calculations. Displayed below is the Complex Fourier Series approximation plotted against the original function over one of its periods. The MATLAB code used to generate the Complex Fourier Series approximation data is also displayed below that.

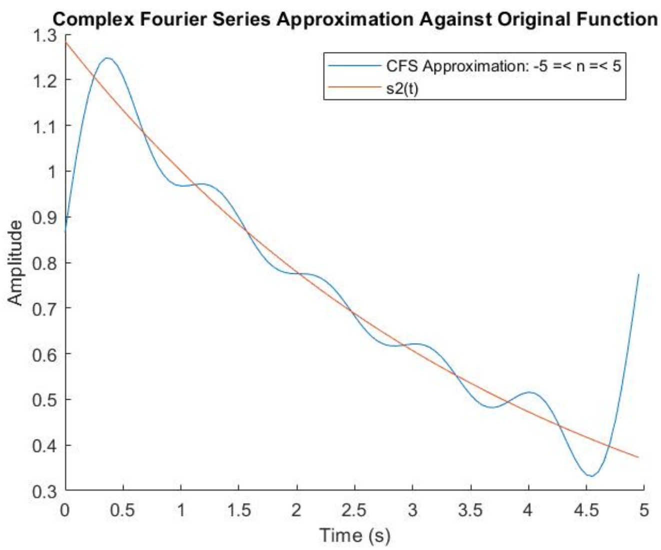

Figure 3: Complex Fourier Series Approximation

```
%%Calculating the Complex Fourier Series of Additive Noise Second
%number of hamonics
harm = 5;
%timestep value.
Ts = tPeriod(2)-tPeriod(1);
%frequency
f0 = 1/T;
%summation values
nVec = -harm:harm;
%removes n = 0 as c0 is already included
nVec(harm+1) = [];
%calculating the Complex Fourier Series coefficient
c0 = 1 / T * sum(additive_noise_second) * Ts;
cn = 1 / T * additive_noise_second * exp(1j * -2 * pi * f0 * nVec' *
    tPeriod).' * Ts;
CFS = c0 + cn * exp(1j * 2 * pi * f0 * nVec' * tPeriod);
```

The Complex Fourier Series was used to model the first periodic noise signal with harmonic values of $-5 < n < 5$. The calculations of the Fourier coefficients, c_0 and c_n, were done by hand. The Complex Fourier Approximation formula and its coefficients are calculated using the following equation.

$$f(t) = c_0 + \sum_{\substack{n=-\infty \\ n\neq 0}}^{\infty} c_n e^{j2\pi n f_0 t}$$

$$c_0 = \frac{1}{T}\int_{t_0}^{t_0+T} s_3(t)\,\mathrm{d}t$$

$$c_n = \frac{1}{T}\int_{t_0}^{t_0+T} s_3(t)e^{-j2\pi n f_0 t}\,\mathrm{d}t$$

The step by step calculations are displayed in the Appendix in section 6.3. The Fourier coefficients were calculated to be.

$$c_0 = 1.5625$$

$$c_n = \frac{250e^{-j\pi n} - 250}{80\pi^2 n^2}$$

The Fourier Series Approximation for $s_3(t)$ using harmonics: $-5 < n < 5$.

$$s_3(t) = 1.5625 + \sum_{\substack{n=-5 \\ n\neq 0}}^{5} \frac{250e^{-j\pi n} - 250}{80\pi^2 n^2} e^{\frac{j2\pi n t}{5}}$$

MATLAB was used to evaluate the expression. The plot of the approximate function against $s_3(t)$ is shown below with the code used to generate the function.

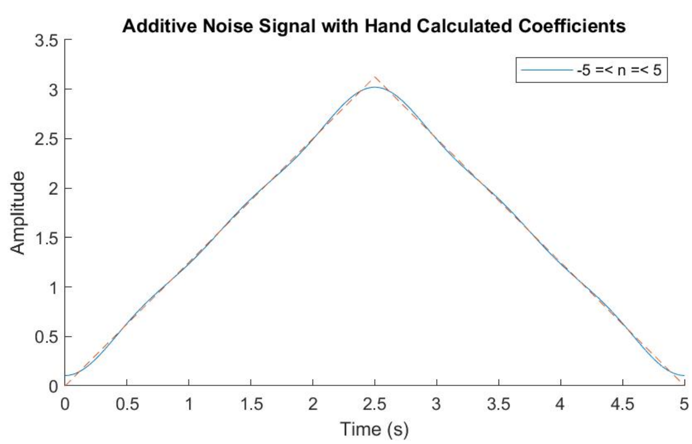

Figure 4: CFS with Hand Calculated Coefficients

```
%%Hand Calculated CFS
%hand calculated fourier coefficients
HCc0 = 1.5625;
HCcn = (250.*exp(1j.*-pi.*nVec)-250)./(80.*pi.^2.*nVec.^2);
%array of zeros to store for loop values
handcalcapprox = zeros(1,samples);
%nvector
nVec = -harm:harm;
nVec(harm+1) = [];
%for loop to sum up values
for n = nVec
    handcalcapprox = handcalcapprox + ((250.*exp(1j.*-pi.*n)-250)
        ./(80.*pi.^2.*n.^2)).* exp(1j.*2.*pi.*n.*f0.*tPeriod);
end
%add c0
handcalcapprox = handcalcapprox + HCc0;
```

4 De-Noising the Speech Signal

The speech signal was de-noised by identifying and modelling the individual noise signals over one of its period. This was then used to generate an approximation of the entire noise signal with a Fourier Series. This approximation included two periods of each noise signal and was the same length and had the same number of points as the received signal. The noise was additive, meaning it can be reversed by subtracting the modelled noise from the corrupted signal. The MATLAB code used to do this is shown below.

```
%caclulate the function values
pcWs1 = A * tPeriodHalf1 / 4;
pcWs2 = (- A * tPeriodHalf2 + 5 * A) / 4;
noiseOne = [pcWs1 pcWs2];
noiseTwo = exp(- (tPeriod - B) / 4);
%First noise signal
FSc0 = 1 / T * sum(noiseOne) * Ts;
FScn = 1 / T * noiseOne* exp(1j * -2 * pi * f0 * nVec' .* tPeriod).' * Ts;
FS1st = FSc0 + FScn*exp(1j*2*pi*f0*nVec' *tPeriod);
%second noise signal
SSc0 = 1 / T * sum(noiseTwo) * Ts;
SScn = 1 / T * noiseTwo * exp(1j * -2 * pi * f0 * nVec' * tPeriod).' * Ts;
FS2nd = SSc0 + SScn*exp(1j*2*pi*f0*nVec' *tPeriod);
%generate approximate noise vector using FS1st and FS2nd
FSTotal = [FS1st FS1st FS2nd  FS2nd];
%de-noise the signal
dnSnd = noiseSound - real(FSTotal);
```

After de-noising, the speech signal now looks as follows.

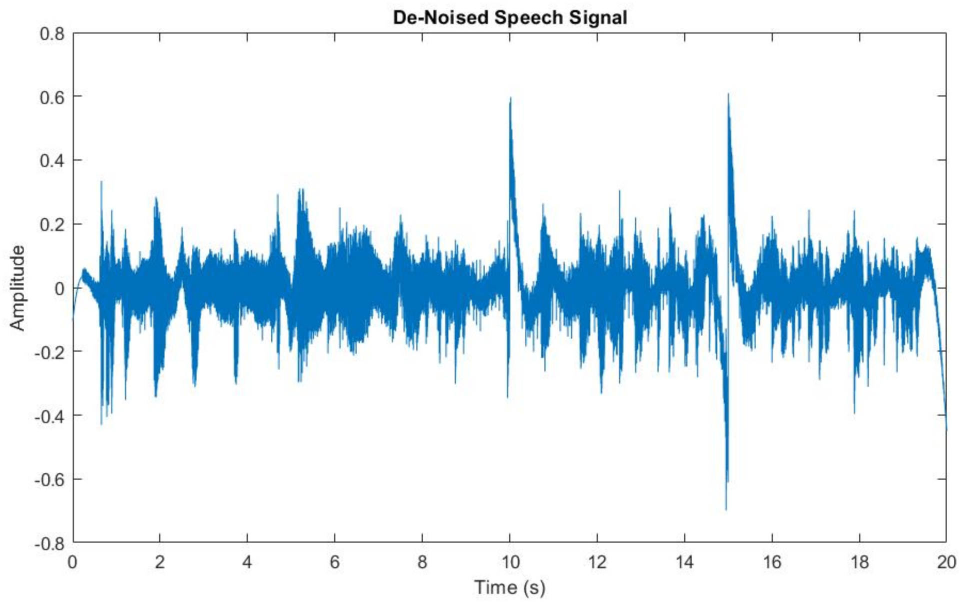

Figure 5: De-noised Speech Signal

The signal now is clearly different from the received one. The two different periodic functions have been removed and it is now centered about 0 on the y axis. It has the jagged expected look of a speech signal, with no points of discontinuity.

The message received is printed below.

T-minus 10, 9, 8, 7, 6, 5, all three engines up and burning, 2, 1, 0 and lift off. The final lift off of Atlantis, on the shoulders of the space shuttle, America will continue the dream.

After testing different numbers of harmonics used in calculating FS1st and FS2nd, the vectors used to generate the Complex Fourier Series approximation of the full length noise signal. It has been determined that increasing the number of harmonics has no effect on the quality of the message received in the speech signal. The de-noised signal with harmonics of $-1 \leq n \leq 1$ was compared with harmonics of $-500 \leq n \leq 500$. In both cases the message received was clear and easily understandable with no traces of noise. The significant difference between the two was the computation time needed; for harmonics of $-1 \leq n \leq 1$ the time taken to run the entire report code was 2.44 seconds. While harmonics of $-500 \leq n \leq 500$ to 48.98 seconds. It is recommended to use this de-noising module at a low number of harmonics, $-25 \leq n \leq 25$, taking 4.36 seconds to complete execution. This can be adjusted depending on how accurately the signal needs to be de-noised.

5 Reflection

During this assessment I demonstrated my understanding of the concepts in this report by getting all my MATLAB Grader questions correct. I demonstrated my understanding of concepts used in this report by identifying and approximating my two noise signals. Then using them to create a Fourier Series approximation of the entire noise signal to de-noise the received sound signal. Another way was by using a different method than what we had been shown to sum up and calculated the 'cn' values on MATLAB. In question 5 I used a 'for' loop to sum up the values instead of using discrete matrix multiplication method shown to us. This demonstrated my understanding of both MATLAB and mathematics. This for loop method was also used to check my MATLAB Grader values before submitting.

During this assignment I learnt that I should prepare more when writing MATLAB code. Once I had finished the code and de-noised the signal I was forced to spend a considerable amount of time figuring out, simplifying, and commenting my code. As when I had started the assignment I rushed into coding and figuring out what I should do as I went along. Instead I should have broken the task into steps and figuring out a plan beforehand, commenting and formatting the code as I went along. Another thing I learnt is I should keep my hand calculations neat and tidy so I can insert an image of them directly into the report. I had to type them into the report which used up time that could have been spent better elsewhere and risked mistakes when transcribing it across. The final lesson I learnt from this report was to thoroughly read and understand the tasks set out beforehand. I had a couple of occasions where I had to scrap my work for a question because I had misread what I needed to do.

6 Appendix

6.1 Calculating Trigonometric Fourier Series Coefficients

6.1.1 a_0

$$s_1(t) = \begin{cases} -2t & t \in (-0.5, 0) \\ 5t & t \in (0, 0.5) \end{cases}$$

$$T = 1 \quad f_0 = 1$$

$$a_0 = \frac{1}{T}\int_{t_0}^{t_0+T} s_1(t)\,\mathrm{d}t = \int_{-0.5}^{0} -2t\,\mathrm{d}t + \int_{0}^{0.5} 5t\,\mathrm{d}t$$

$$= -t^2\Big|_{-0.5}^{0} + 2.5t^2\Big|_{0}^{0.5}$$

$$a_0 = 0.25 + 0.625 = 0.875$$

6.1.2 a_n

$$a_n = \frac{2}{T}\int_{t_0}^{t_0+T} s_1(t)\cos(2\pi n f_0 t)\,\mathrm{d}t = 2\left[\int_{-0.5}^{0} -2t\cos(2\pi n t)\,\mathrm{d}t + \int_{0}^{0.5} 5t\cos(2\pi n t)\,\mathrm{d}t\right]$$

Let:

$$I_1 = \int_{-0.5}^{0} -2t\cos(2\pi n t)\,\mathrm{d}t$$

$$I_2 = \int_{0}^{0.5} 5t\cos(2\pi n t)\,\mathrm{d}t$$

$\underline{I_1}$

Using integration by parts the solution has the form.

$$I_1 = uv - \int vu'\,\mathrm{d}t$$

Where:

$$u = -2t$$

$$u' = -2$$

$$v' = \cos(2\pi n t)$$

$$v = \frac{\sin(2\pi n t)}{2\pi n}$$

Substituting the values into the equation gives.

$$I_1 = \frac{-2t\sin(2\pi n t)}{2\pi n}\Bigg|_{-0.5}^{0} - \int_{-0.5}^{0} \frac{-2\sin(2\pi n t)}{2\pi n}\,\mathrm{d}t$$

$$I_1 = \frac{-t\sin(\pi nt)}{\pi n}\bigg|_{-0.5}^{0} + \frac{1}{\pi n}\int_{-0.5}^{0} \sin(2\pi nt)\,\mathrm{d}t$$

$$I_1 = \frac{-t\sin(\pi nt)}{\pi n}\bigg|_{-0.5}^{0} + \left(\frac{-\cos(2\pi nt)}{2\pi^2 n^2}\right)\bigg|_{-0.5}^{0}$$

$$I_1 = \left[0 + \frac{(-0.5)\sin(2\pi n(-0.5))}{\pi n}\right] + \left[\frac{-1}{2\pi^2 n^2} + \frac{\cos(2\pi n(-0.5))}{2\pi^2 n^2}\right]$$

As n will always be an integer i.e. n $\in Z$, sine and cosine can be simplified.

$$I_1 = -\frac{1}{2\pi^2 n^2} + \frac{cos(\pi n)}{2\pi^2 n^2}$$

$$I_1 = \frac{1}{2\pi^2 n^2}(\cos(\pi n) - 1)$$

<u>I_2</u>

The integration by parts formula is used again.

$$I_2 = uv - \int vu'\,\mathrm{d}t$$

Where:

$$u = 5t$$

$$u' = 5$$

$$v' = \cos(2\pi nt)$$

$$v = \frac{\sin(2\pi nt)}{2\pi n}$$

Substituting the values into the equation gives.

$$I_2 = \frac{5t\sin(2\pi nt)}{2\pi n}\bigg|_{0}^{0.5} - \int_{0}^{0.5} \frac{5\sin(2\pi nt)}{2\pi n}\,\mathrm{d}t$$

$$I_2 = \frac{5t\sin(2\pi nt)}{2\pi n}\bigg|_{0}^{0.5} - \frac{5}{2\pi n}\int_{0}^{0.5} \sin(2\pi nt)\,\mathrm{d}t$$

$$I_2 = \frac{5t\sin(2\pi nt)}{2\pi n}\bigg|_{0}^{0.5} + \frac{5\cos(2\pi nt)}{4\pi^2 n^2}\bigg|_{0}^{0.5}$$

$$I_2 = \left[\frac{5(0.5)\sin(2\pi n(0.5))}{2\pi n} - 0\right] - \frac{5}{4\pi^2 n^2}\left[-\cos(2\pi n(0.5) + \cos(0)\right]$$
$$I_2 = \frac{5}{4\pi^2 n^2}(\cos(\pi n) - 1)$$

The solved values for I_1 and I_2 can be substituted back into the equation to solve for a_n.

$$a_n = 2(I_1 + I_2)$$
$$a_n = 2\left(\frac{1}{2\pi^2 n^2}(\cos(\pi n) - 1) + \frac{5}{4\pi^2 n^2}(\cos(\pi n) - 1)\right)$$
$$a_n = \frac{1}{\pi^2 n^2}(\cos(\pi n) - 1) + \frac{5}{2\pi^2 n^2}(\cos(\pi n) - 1)$$
$$a_n = \frac{7}{2\pi^2 n^2}(\cos(\pi n) - 1)$$

6.1.3 b_n

$$b_n = \frac{2}{T}\int_{t_0}^{t_0+T} s_1(t)\sin(2\pi n f_0 t)\,\mathrm{d}t = 2\left[\int_{-0.5}^{0} -2t\sin(2\pi nt)\,\mathrm{d}t + \int_{0}^{0.5} 5t\sin(2\pi nt)\,\mathrm{d}t\right]$$

Let:

$$I_3 = \int_{-0.5}^{0} -2t\sin(2\pi nt)\,\mathrm{d}t$$
$$I_4 = \int_{0}^{0.5} 5t\sin(2\pi nt)\,\mathrm{d}t$$

$\underline{I_3}$

Using integration by parts the solution has the form.

$$I_3 = uv - \int vu'\,\mathrm{d}t$$

Where:

$$u = -2t$$
$$u' = -2$$

$$v' = \sin(2\pi nt)$$
$$v = \frac{-\cos(2\pi nt)}{2\pi n}$$

Substituting the values into the equation gives.

$$I_3 = -\frac{-2t\cos(2\pi nt)}{2\pi n}\bigg|_{-0.5}^{0} - \int_0^{0.5} -\frac{-2\cos(2\pi nt)}{2\pi n}\,\mathrm{d}t$$

$$I_3 = \frac{t\cos(2\pi nt)}{\pi n}\bigg|_{-0.5}^{0} - \frac{1}{\pi n}\int_0^{0.5} \cos(2\pi nt)\,\mathrm{d}t$$

$$I_3 = \frac{t\cos(2\pi nt)}{\pi n}\bigg|_{-0.5}^{0} - \frac{\sin(2\pi nt)}{2\pi^2 n^2}\bigg|_{0}^{0.5}$$

$$I_3 = \left[0 - \frac{(-0.5)\cos(2\pi n(-0.5))}{\pi n}\right] - \left[\frac{\sin(2\pi n(0.5))}{2\pi^2 n^2} - \frac{\sin(0)}{2\pi^2 n^2}\right]$$

As n will always be an integer i.e. n $\in Z$, sine and cosine can be simplified.

$$I_3 = \frac{\cos(\pi n)}{2\pi n}$$

<u>I_4</u>

Using integration by parts the solution has the form.

$$I_4 = uv - \int vu'\,\mathrm{d}t$$

Where:

$$u = 5t$$
$$u' = 5$$

$$v' = \sin(2\pi nt)$$
$$v = \frac{-\cos(2\pi nt)}{2\pi n}$$

Substituting the values into the equation gives.

$$I_4 = \frac{-5t\cos(2\pi nt)}{2\pi n}\bigg|_{-0.5}^{0} - \int_0^{0.5} -\frac{5\cos(2\pi nt)}{2\pi n}\,\mathrm{d}t$$

$$I_4 = \frac{-5t\cos(2\pi nt)}{2\pi n}\bigg|_{-0.5}^{0} - \int_0^{0.5} -\frac{5\cos(2\pi nt)}{2\pi n}\,\mathrm{d}t$$

$$I_4 = \frac{-5t\cos(2\pi nt)}{2\pi n}\bigg|_{-0.5}^{0} + \frac{5}{2\pi n}\int_0^{0.5} \cos(2\pi nt)\,\mathrm{d}t$$

$$I_4 = \left.\frac{-5t\cos(2\pi nt)}{2\pi n}\right|_{-0.5}^{0} + \left.\frac{5\sin(2\pi nt)}{4\pi^2 n^2}\right|_{0}^{0.5}$$

$$I_4 = \left[0 + \frac{5(-0.5)\cos(2\pi n(-0.5))}{2\pi n}\right] + \left[\frac{5\sin(\pi n)}{4\pi^2 n^2} - \frac{5\sin(0)}{4\pi^2 n^2}\right]$$

As n will always be an integer i.e. n $\in Z$, sine and cosine can be simplified.

$$I_4 = -\frac{5\cos(\pi n)}{4\pi n}$$

The solved values for I_3 and I_4 can be substituted back into the equation to solve for b_n.

$$b_n = 2(I_3 + I_4)$$

$$b_n = 2\left(\frac{\cos(\pi n)}{2\pi n} - \frac{5\cos(\pi n)}{4\pi n}\right)$$

$$b_n = \frac{\cos(\pi n)}{\pi n} - \frac{5\cos(\pi n)}{2\pi n}$$

$$b_n = -\frac{3\cos(\pi n)}{2\pi n}$$

6.1.4 Harmonics: $0 < n < 2$

n = 1

$$s_1(t) = \frac{7}{2\pi^2(1)^2}[\cos(\pi(1)) - 1]\cos(2\pi(1)t) - \frac{3}{2\pi(1)}\cos[\pi(1)]\sin(2\pi(1)t)$$
$$s_1(t) = \frac{7}{2\pi^2}[\cos(\pi) - 1]\cos(2\pi t) - \frac{3}{2\pi}\cos(\pi)\sin(2\pi t)$$
$$s_1(t) = \frac{7}{2\pi^2}[-1 - 1]\cos(2\pi t) - \frac{3}{2\pi}(-1)\sin(2\pi t)$$
$$s_1(t) = -\frac{7}{\pi^2}\cos(2\pi t) + \frac{3}{2\pi}\sin(2\pi t)$$

n = 2

$$s_1(t) = \frac{7}{2\pi^2(2)^2}[\cos(\pi(2)) - 1]\cos(2\pi(2)t) - \frac{3}{2\pi(2)}\cos[\pi(2)]\sin(2\pi(2)t)$$
$$s_1(t) = \frac{7}{8\pi^2}[1 - 1]\cos(4\pi t) - \frac{3}{4\pi}\sin(4\pi t)$$
$$s_1(t) = -\frac{3\sin(4\pi t)}{4\pi}$$

6.2 Calculating Complex Fourier Series Coefficients

6.2.1 c_0

$$s_2(t) = 5t + 5e^{-t} \qquad t \in (0,1)$$

$$T = 1, f_0 = 1$$

$$c_0 = \frac{1}{T}\int_{t_0}^{t_0+T} s_2(t)\,\mathrm{d}t = \int_0^1 5t + 5e^{-t}\,\mathrm{d}t$$

$$c_0 = \frac{5t^2}{2} - 5e^{-t}\Bigg|_0^1$$

$$c_0 = \left[\frac{5(1)^2}{2} - 5e^{-1}\right] - \left[\frac{0}{2} - 5e^0\right]$$

$$c_0 = 2.5 - e^{-1} + 5 \approx 5.6606$$

6.2.2 c_n

$$c_n = \frac{1}{T}\int_{t_0}^{t_0+T} s_2(t)e^{-j2\pi nt}\,\mathrm{d}t = \int_0^1 (5t+5e^{-t})e^{-j2\pi nt}\,\mathrm{d}t$$

$$c_n = 5\int_0^1 te^{-j2\pi nt} + e^{-t-j2\pi nt}\,\mathrm{d}t$$

Let:

$$I_5 = \int_0^1 te^{-j2\pi nt}\,\mathrm{d}t$$

$$I_6 = \int_0^1 e^{(-1-j2\pi n)t}\,\mathrm{d}t$$

$\underline{I_5}$

The integration by parts formula is used again.

$$I_5 = uv - \int vu'\,\mathrm{d}t$$

Where:

$$u = t$$

$$u' = 1$$

$$v' = e^{-j2\pi nt}$$

$$v = \frac{e^{-j2\pi nt}}{-j2\pi n}$$

Substituting the values into the equation gives.

$$I_5 = -\frac{te^{-j2\pi nt}}{j2\pi n}\Bigg|_0^1 + \int_0^1 \frac{e^{-j2\pi nt}}{j2\pi n}\,\mathrm{d}t$$

$$I_5 = -\frac{te^{-j2\pi nt}}{j2\pi n}\Bigg|_0^1 + \frac{1}{j2\pi n}\int_0^1 e^{-j2\pi nt}\,\mathrm{d}t$$

$$I_5 = -\frac{te^{-j2\pi nt}}{j2\pi n}\Bigg|_0^1 - \frac{e^{-j2\pi nt}}{4\pi^2 n^2}\Bigg|_0^1$$

$$I_5 = \left[-\frac{(1)e^{-j2\pi n(1)}}{j2\pi n} + 0\right] - \left[\frac{e^{-j2\pi n(1)}}{4\pi^2 n^2} - \frac{e^0}{4\pi^2 n^2}\right]$$

$$I_5 = -\frac{e^{-j2\pi n}}{j2\pi n} - \left[\frac{e^{-j2\pi n}}{4\pi^2 n^2} - \frac{1}{4\pi^2 n^2}\right]$$

As n will always be an integer i.e. n $\in Z$, I_5 can be simplified.

$$I_5 = -\frac{1}{j2\pi n} - \left[\frac{1}{4\pi^2 n^2} - \frac{1}{4\pi^2 n^2}\right]$$

$$I_5 = -\frac{1}{j2\pi n}$$

$$I_5 = \frac{j}{2\pi n}$$

$\underline{I_6}$

$$I_6 = \int_0^1 e^{(-1-j2\pi n)t}\,\mathrm{d}t$$

$$I_6 = \left.\frac{e^{(-1-j2\pi n)t}}{-1-j2\pi n}\right|_0^1$$

$$I_6 = \left[\frac{e^{(-1-j2\pi n)(1)}}{-1-j2\pi n} - \frac{e^0}{-1-j2\pi n}\right]$$

$$I_6 = \frac{e^{-1-j2\pi n} - 1}{-1-j2\pi n}$$

As n will always be an integer i.e. n $\in Z$, I_5 can be simplified.

$$I_6 = \frac{e^{-1} - 1}{-1-j2\pi n}$$

The solved values for I_5 and I_6 can be substituted back into the equation to solve for c_n.

$$c_n = 5(I_5 + I_6)$$

$$c_n = 5\left(\frac{j}{2\pi n} + \frac{e^{-1} - 1}{-1-j2\pi n}\right)$$

$$c_n = \frac{5j}{2\pi n} - \frac{5e^{-1} + 5}{1 + j2\pi n}$$

6.2.3 Harmonics: $-2 < n < 2$

n = -2

$$c_n = \left[\frac{5j}{2\pi(-2)} + \frac{5 - 5e^{-1}}{1 + j2\pi(-2)}\right]e^{j2\pi(-2)t}$$
$$c_n = \left[-\frac{5j}{4\pi} + \frac{5 - 5e^{-1}}{1 - j4\pi}\right]e^{j-4\pi t}$$

n = -1

$$c_n = \left[\frac{5j}{2\pi(-1)} + \frac{5 - 5e^{-1}}{1 + j2\pi(-1)}\right]e^{j2\pi(-1)t}$$
$$c_n = \left[-\frac{5j}{2\pi} + \frac{5 - 5e^{-1}}{1 - j2\pi}\right]e^{-j2\pi t}$$

n = 1

$$c_n = \left[\frac{5j}{2\pi(1)} + \frac{5 - 5e^{-1}}{1 + j2\pi(1)}\right]e^{j2\pi(1)t}$$
$$c_n = \left[\frac{5j}{2\pi} + \frac{5 - 5e^{-1}}{1 + j2\pi}\right]e^{j2\pi t}$$

n = 2

$$c_n = \left[\frac{5j}{2\pi(2)} + \frac{5 - 5e^{-1}}{1 + j2\pi(2)}\right]e^{j2\pi(2)t}$$
$$c_n = \left[\frac{5j}{4\pi} + \frac{5 - 5e^{-1}}{1 + j4\pi}\right]e^{j4\pi t}$$

6.3 Q5: CFS Coefficients of First Periodic Noise Signal

6.3.1 c_0

$$s_1(t) = \begin{cases} \dfrac{5t}{4} & t \in (0, 2.5) \\ \dfrac{-5t+25}{4} & t \in (2.5, 5) \end{cases}$$

$$T = 5 \quad f_0 = 0.2$$

$$c0 = \frac{1}{5}\int_0^5 s_3(t)\,\mathrm{d}t$$

$$c0 = \frac{1}{5}\left\{\int_0^{2.5} \frac{5t}{4}\,\mathrm{d}t + \int_{2.5}^{5} \frac{-5t+25}{4}\,\mathrm{d}t\right\}$$

$$c0 = \frac{1}{5}\left\{\frac{5}{4}\int_0^{2.5} t\,\mathrm{d}t + \int_{2.5}^{5} \frac{-5t}{4} + \frac{25}{4}\,\mathrm{d}t\right\}$$

$$c0 = \frac{1}{5}\left\{\ \frac{5}{4}\left[\frac{t^2}{2}\right]_0^{2.5} + \left[\frac{-5t^2}{8} + \frac{25t}{4}\right]_{2.5}^{5}\right\}$$

$$c0 = \frac{1}{5}\left\{\ \frac{5}{4}\left[\frac{2.5^2}{2}\right] + \left(\left[\frac{-5(5)^2}{8} + \frac{25(5)}{4}\right] - \left[\frac{-5(2.5)^2}{8} + \frac{25(2.5)}{4}\right]\right)\right\}$$

$$c0 = \frac{1}{5}[3.91 + (15.63 - 11.72)]$$

$$c0 = 1.56$$

6.3.2 c_n

$$cn = \frac{1}{5}\Big\{\int_{0}^{2.5}\frac{5t}{4}e^{-j2\pi nf_0t}\,\mathrm{d}t + \int_{2.5}^{5}\frac{-5t+25}{4}e^{-j2\pi nf_0t}\,\mathrm{d}t\Big\}$$

$$cn = \frac{1}{5}(I_7 + I_8)$$

<u>Let</u>

$$I_7 = \int_{0}^{2.5}\frac{5t}{4}e^{-j2\pi nf_0t}\,\mathrm{d}t$$

$$I_8 = \int_{2.5}^{5}\frac{-5t+25}{4}e^{-j2\pi nf_0t}\,\mathrm{d}t$$

<u>I_7</u>

$$I_7 = \frac{5}{4}\int_{0}^{2.5} te^{\frac{-j2\pi nt}{5}}\,\mathrm{d}t$$

The integration by parts formula is used again.

$$I_7 = uv - \int vu'\,\mathrm{d}t$$

Where:

$$u = t$$

$$u' = 1$$

$$v' = e^{\frac{-j2\pi nt}{5}}$$

$$v = \frac{5e^{\frac{-j2\pi nt}{5}}}{-j2\pi n}$$

Substituting the values into the equation gives.

$$I_7 = \frac{5}{4}\Big\{\Big[\frac{5te^{\frac{-j2\pi nt}{5}}}{-j2\pi n}\Big]_0^{2.5} - \int_{0}^{2.5}\frac{5e^{\frac{-j2\pi nt}{5}}}{-j2\pi n}\,\mathrm{d}t\Big\}$$

$$I_7 = \frac{5}{4}\Big\{\Big[\frac{5te^{\frac{-j2\pi nt}{5}}}{-j2\pi n}\Big]_0^{2.5} + \frac{5}{j2\pi n}\int_{0}^{2.5} e^{\frac{-j2\pi nt}{5}}\,\mathrm{d}t\Big\}$$

$$I_7 = \frac{5}{4}\Big\{\Big[\frac{5te^{\frac{-j2\pi nt}{5}}}{-j2\pi n}\Big]_0^{2.5} + \frac{5}{j2\pi n}\Big[\frac{5e^{\frac{-j2\pi nt}{5}}}{-j2\pi n}\Big]_0^{2.5}\Big\}$$

$$I_7 = \frac{5}{4}\Big\{\Big[\frac{5te^{\frac{-j2\pi nt}{5}}}{-j2\pi n}\Big]_0^{2.5} + \frac{25}{4\pi^2n^2}\Big[e^{\frac{-j2\pi nt}{5}}\Big]_0^{2.5}\Big\}$$

$$I_7 = \frac{5}{4}\left\{\frac{12.5e^{-j\pi n}}{-j2\pi n} + \frac{25}{4\pi^2 n^2}\left[e^{-j\pi n} - 1\right]\right\}$$
$$I_7 = \frac{-62.5e^{-j\pi n}}{j8\pi n} + \frac{125}{16\pi^2 n^2}\left[e^{-j\pi n} - 1\right]$$
$$I_7 = \frac{-62.5e^{-j\pi n}}{j8\pi n} + \frac{125e^{-j\pi n} - 125}{16\pi^2 n^2}$$

$\underline{I_8}$

$$I_8 = \int_{2.5}^{5} \frac{-5t + 25}{4} e^{-j2\pi n f_0 t}\,\mathrm{d}t$$
$$I_8 = \int_{2.5}^{5} \frac{-5t}{4} e^{\frac{-j2\pi n t}{5}}\,\mathrm{d}t + \int_{2.5}^{5} \frac{25}{4} e^{\frac{-j2\pi n t}{5}}\,\mathrm{d}t$$

Let

$$I_{8.1} = \int_{2.5}^{5} \frac{-5t}{4} e^{\frac{-j2\pi n t}{5}}\,\mathrm{d}t$$
$$I_{8.2} = \int_{2.5}^{5} \frac{25}{4} e^{\frac{-j2\pi n t}{5}}\,\mathrm{d}t$$

$\underline{I_{8.1}}$

The integration by parts formula is used again.

$$I_{8.1} = uv - \int vu'\,\mathrm{d}t$$

Where:

$$u = \frac{-5}{4}t$$
$$u' = \frac{-5}{4}$$

$$v' = e^{\frac{-j2\pi n t}{5}}$$
$$v = \frac{5e^{\frac{-j2\pi n t}{5}}}{-j2\pi n}$$

$$I_{8.1} = \left[\frac{25te^{\frac{-j2\pi n t}{5}}}{j8\pi n}\right]_{2.5}^{5} - \int_{2.5}^{5} \frac{25e^{\frac{-j2\pi n t}{5}}}{j8\pi n}\,\mathrm{d}t$$

$$I_{8.1} = \left[\frac{25te^{\frac{-j2\pi nt}{5}}}{j8\pi n}\right]_{2.5}^{5} - \frac{25}{j8\pi n}\int_{2.5}^{5} e^{\frac{-j2\pi nt}{5}}\,\mathrm{d}t$$

$$I_{8.1} = \left[\frac{25te^{\frac{-j2\pi nt}{5}}}{j8\pi n}\right]_{2.5}^{5} - \frac{125}{16\pi^2 n^2}\left[e^{\frac{-j2\pi nt}{5}}\right]_{2.5}^{5}$$

$$I_{8.1} = \left[\frac{125}{j8\pi n} - \frac{62.5e^{-j\pi n}}{j8\pi n}\right] - \frac{125}{16\pi^2 n^2}\left[1 - e^{-j\pi n}\right]$$

$$I_{8.1} = \frac{125 - 62.5e^{-j\pi n}}{j8\pi n} - \frac{125 - 125e^{-j\pi n}}{16\pi^2 n^2}$$

$\underline{I_{8.2}}$

$$I_{8.2} = \int_{2.5}^{5} \frac{25}{4} e^{\frac{-j2\pi nt}{5}}\,\mathrm{d}t$$

$$I_{8.2} = \frac{25}{4}\left[\frac{5e^{\frac{-j2\pi nt}{5}}}{-j2\pi n}\right]_{2.5}^{5}$$

$$I_{8.2} = \left[\frac{125e^{\frac{-j2\pi nt}{5}}}{-j8\pi n}\right]_{2.5}^{5}$$

$$I_{8.2} = \frac{-125}{j8\pi n} + \frac{125e^{-j\pi n}}{j8\pi n}$$

$$I_{8.2} = \frac{-125 + 125e^{-j\pi n}}{j8\pi n}$$

$\underline{I_{8}}$

$$I_8 = I_{8.1} + I_{8.2}$$

$$I_8 = \frac{-125 + 125e^{-j\pi n}}{j8\pi n} + \frac{125 - 62.5e^{-j\pi n}}{j8\pi n} - \frac{125 - 125e^{-j\pi n}}{16\pi^2 n^2}$$

$$cn = \frac{1}{5}(I_7 + I_8)$$

$$cn = \frac{1}{5}\left(\frac{-62.5e^{-j\pi n}}{j8\pi n} + \frac{125e^{-j\pi n} - 125}{16\pi^2 n^2} + \frac{-125 + 125e^{-j\pi n}}{j8\pi n} + \frac{125 - 62.5e^{-j\pi n}}{j8\pi n} - \frac{125 - 125e^{-j\pi n}}{16\pi^2 n^2}\right)$$

$$cn = \frac{-62.5e^{-j\pi n}}{j40\pi n} + \frac{125e^{-j\pi n} - 125}{80\pi^2 n^2} + \frac{-125 + 125e^{-j\pi n}}{j40\pi n} + \frac{125 - 62.5e^{-j\pi n}}{j40\pi n} - \frac{125 - 125e^{-j\pi n}}{80\pi^2 n^2}$$

$$cn = \frac{-62.5e^{-j\pi n}}{j40\pi n} + \frac{250e^{-j\pi n} - 250}{80\pi^2 n^2} + \frac{-125 + 125e^{-j\pi n}}{j40\pi n} + \frac{125 - 62.5e^{-j\pi n}}{j40\pi n}$$

$$cn = \frac{-62.5e^{-j\pi n}}{j40\pi n} + \frac{250e^{-j\pi n} - 250}{80\pi^2 n^2} + \frac{-125 + 125e^{-j\pi n}}{j40\pi n} + \frac{125}{j40\pi n} - \frac{62.5e^{-j\pi n}}{j40\pi n}$$

$$cn = \frac{-125e^{-j\pi n}}{j40\pi n} + \frac{250e^{-j\pi n} - 250}{80\pi^2 n^2} + \frac{-125 + 125e^{-j\pi n}}{j40\pi n} + \frac{125}{j40\pi n}$$

$$cn = -\frac{125e^{-j\pi n}}{j40\pi n} + \frac{250e^{-j\pi n} - 250}{80\pi^2 n^2} - \frac{125}{j40\pi n} + \frac{125e^{-j\pi n}}{j40\pi n} + \frac{125}{j40\pi n}$$

$$cn = \frac{250e^{-j\pi n} - 250}{80\pi^2 n^2}$$

6.4 MATLAB Grader Code

```
%%%MATLAB Grader Answer
%%Question 1
function [a0, an, bn, s1t_approx, sid] = trigFS(sid)
    syms t, syms n, pi = sym('pi');
    % Enter your student ID as sid:
    sid = 10517651;
    % As a symbolic expression, save the expressions
    % for the trigonometric coefficients below:
    a0 =  0.875;
    an =  7/(2*pi^2*n^2)*(cos(pi*n)-1);
    bn =  -3/(2*pi*n)*cos(pi*n);
    % Find the trigonometric FS, evaluate the FS with
    % 2 harmonics and save the expression below:
    s1t_approx =  0.875 -7/pi^2*cos(2*pi*t)+3/(2*pi)*sin(2*pi*t) - 3/(4*pi)*sin(4*pi*t);
end

%%Question 2
function [c0, cn, s2t_approx, sid] = compFS(sid)
    syms n t; e = sym(exp(sym(1))); pi = sym('pi');
    % Enter your student ID as sid:
    sid = 10517651;
    % As a symbolic expression, save the expressions
    % for the complex coefficients below:
    c0 = 5*(0.5-exp(-1)+1);
    cn = ((5*1j)/(2*pi*n) + (5/(-1 -2*1j*pi*n))*(exp(-1)-1))
    % Find the complex FS, evaluate the FS up to
    % second harmonic and save the expression below:
    s2t_approx = 5*(0.5-exp(-1)+1) + (-(5*1j)/(4*pi) + (5 - 5*exp(-1))/(1 - 4*1j*pi))*exp(1j*-4*pi*t) + (-(5*1j)/(2*pi) + (5 - 5*exp(-1))/(1 - 2*1j*pi))*exp(1j*-2*pi*t) + ((5*1j)/(2*pi) + (5 - 5*exp(-1))/(1 + 2*1j*pi))*exp(1j*2*pi*t) + ((5*1j)/(4*pi) + (5 - 5*exp(-1))/(1 + 4*1j*pi))*exp(1j*4*pi*t);

end

%%Question 3
function [t, s2_hinf, sid] = noiseFunc(sid);
    % Enter you student ID below:
    sid = 10517651;
    % Save the appropriate outputs below as defined above:
    time = linspace(0,1,100+1);
    time(end) = [];
```

```
    t = linspace(0,5,500+1);
    t(end) = [];
    N = 5;
    s = 5.*time + 5.*exp(-time);
    s2_hinf = repmat(s, [1 N]);

end

%%Question 4
function [a0, an, bn, s_approx, T] = trigFS(s_hinf, t, N)

    Ts = t(2)- t(1);
    T = t(end) - t(1) + Ts;
    f0 = 1/T;
    Ts = t(2) - t(1);
    n_trig = 1:N;


    a0 = 1/T * sum(s_hinf) * Ts;
    an = 2/T * s_hinf * cos(2 * pi * f0 * n_trig' *t).' *Ts;
    bn = 2/T * s_hinf * sin(2 * pi * f0 * n_trig' *t).' *Ts;

s_approx = a0 + an* cos(2*pi*f0*n_trig'*t) + bn*sin(2*pi*f0*n_trig'*t);


end

%%Question 5
function [c0, cn, s_approx, T] = complexFS(s_hinf, t, N)
Ts = t(2)- t(1);
T = t(end) - t(1) + Ts;
f0 = 1/T;
Ts = t(2) - t(1);
n_comp = -N:N;

c0 = 1/T * sum(s_hinf) * Ts;

cn = 1/T * s_hinf * exp(1j * -2 * pi * f0 * n_comp' *t).' * Ts;

s_approx = cn * exp(1j * 2 * pi * f0 * n_comp' *t);
end
```

6.5 mission.m Code

```
%% Assignment 1 Part A - Section A2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Do not change before line 28.
% If you have not generated Data1A from GenerateDataAssignment1A.m,
% do that now.

% Clearing and preparing the workspace
clear; clc; close all;

% Load assignment data.
load Data1A;

% VARIABLES:
% t - Time vector
% T - Period
% additive noise - Your noise waveform
% a0, an, bn - Trig Fourier series variables
% OR
% c0, cn - Complex Fourier series variables
% FS1 - Fourier series approximation
% dnSnd - De-noised resulting wave

%=================================================================
% Refer to the assignment sheet for details on variable naming.
% Names of the variables are important,
% e.g. 'a1' is considered a different variable to 'A1'.
%===Enter your code below this line===============================


%%Question 1: Plotting the Recieved Signal
%length of noise signal
lgthNS = length(noiseSound);

%timevector
t = linspace(0,20, lgthNS + 1);
t(end) = [];

%plotting recieved noise signal
%sound(noiseSound,44100);
plot(t, noiseSound)
title("Received Speech Signal")
xlabel("Time (s)")
```

```
ylabel("Amplitude")
%==========================================================================

%%Question 2: Noise Signals Plotted Over 1 Period
%period
samples = lgthNS*0.25;
T = 5;
%time vector
tPeriod = linspace(0,T, samples + 1);
tPeriod(end) = [];
%half periods to calculate piecewise function
tPeriodHalf1 = linspace(0, T/2, samples/2 + 1);
tPeriodHalf1(end) = [];

tPeriodHalf2 = linspace(T/2, T, samples/2 + 1);
tPeriodHalf1(end) = [];


%piecewise functions of s3(t))
PW1 = A * tPeriodHalf1 / 4;
PW2 = (- A * tPeriodHalf2 + 5 * A) / 4;


%calculating the additive noise vectors
additive_noise_first = [PW1 PW2];
additive_noise_second = exp(-1 * (tPeriod - B) / 4);


%plotting the additive noise vectors
figure
hold on

subplot(1,2,1)
plot(tPeriod, additive_noise_first)
title("First Additive Noise Signal Over One Period");
xlabel("Time (s)");
ylabel("Amplitude");
legend("s3(t)");

subplot(1,2,2)
plot(tPeriod, additive_noise_second)
title("Second Additive Noise Signal Over One Period");
xlabel("Time (s)");
ylabel("Amplitude");
legend("s2(t)");
```

```
%===========================================================================

%%Question 3: classifying Noise Signal 2
%See page 4 in the report
%===========================================================================

%%Question 4: Calculating the Complex Fourier Series of Additive
    Noise Second
%number of hamonics
harm = 5;
%timestep value.
Ts = tPeriod(2)-tPeriod(1);
%frequency
f0 = 1/T;
%summation variable
nVec = -harm:harm;
%removes n = 0 as c0 is already included
nVec(harm+1) = [];


%calculating the Complex Fourier Series
c0 = 1 / T * sum(additive_noise_second) * Ts;
cn = 1 / T * additive_noise_second * exp(1j * -2 * pi * f0 * nVec'
     * tPeriod).' * Ts;
CFS = c0 + cn * exp(1j * 2 * pi * f0 * nVec' * tPeriod);


%plotting the CFS
figure
hold on
plot(tPeriod, CFS)
plot(tPeriod, additive_noise_second)
title("Complex Fourier Series Approximation vs Original Function")
    ;
xlabel("Time (s)");
ylabel("Amplitude");
legend("CFS Approximation", "s2(t)");
%===========================================================================

%%Question 5: Hand Calculated CFS
%hand calculated fourier coefficients
HCc0 = 1.5625;
HCcn = (250.*exp(1j.*-pi.*nVec)-250)./(80.*pi.^2.*nVec.^2);


```

```
%array of zeros to store for loop values
handcalcapprox = zeros(1,samples);

%nvector
nVec = -harm:harm;
nVec(harm+1) = [];


%for loop to sum up values
for n = nVec
    handcalcapprox = handcalcapprox + ((250.*exp(1j.*-pi.*n)-250)./(80.*pi.^2.*n.^2)).* exp(1j.*2.*pi.*n.*f0.*tPeriod);
end


%add c0
handcalcapprox = handcalcapprox + HCc0;


%plotting the function
figure
hold on
plot(tPeriod,handcalcapprox)
plot(tPeriod,additive_noise_first,'--')
title("Additive Noise Signal with Hand Calculated Coefficients");
xlabel("Time (s)");
ylabel("Amplitude");
legend("Hand Calculated Coefficients","Real Signal");
%==========================================================================

%%Q6: Using Trigonometric Fourier Series to Generate Noise Function
%caclulate the function values
pcWs1 = A * tPeriodHalf1 / 4;
pcWs2 = (- A * tPeriodHalf2 + 5 * A) / 4;
noiseOne = [pcWs1 pcWs2];
noiseTwo = exp(- (tPeriod - B) / 4);


%First noise signal
FSc0 = 1 / T * sum(noiseOne) * Ts;
FScn = 1 / T * noiseOne* exp(1j * -2 * pi * f0 * nVec' .* tPeriod).' * Ts;
FS1st = FSc0 + FScn*exp(1j*2*pi*f0*nVec' *tPeriod);

```

```
%second noise signal
SSc0 = 1 / T * sum(noiseTwo) * Ts;
SScn = 1 / T * noiseTwo * exp(1j * -2 * pi * f0 * nVec' * tPeriod).' * Ts;
FS2nd = SSc0 + SScn*exp(1j*2*pi*f0*nVec' *tPeriod);

%plotting the noise signals over 1 period
figure
hold on
subplot(1,2,1)
plot(tPeriod, FS1st)
title("Complex Fourier Series Representation of the First Noise Signal");
xlabel("Time (s)");
ylabel("Amplitude");
legend("-5 =< n =< 5");

subplot(1,2,2)
plot(tPeriod, FS2nd)
title("Complex Fourier Series Representation of the First Noise Signal");
xlabel("Time (s)");
ylabel("Amplitude");
legend("-5 =< n =< 5");
%==========================================================================

%%Q7: FS total
%generate approximate noise vector using FS1st and FS2nd
FSTotal = [FS1st FS1st FS2nd  FS2nd];

%plot FSTotal
figure
plot(t, FSTotal)
title("Full Noise Signal Approximation")
xlabel("Time (s)")
ylabel("Amplitude")
legend("-5 =< n =< 5");
%==========================================================================

%%Question 8: De-noising the Recieved Signal
%The additive noise can be subtracted from the recieved signal
dnSnd = noiseSound - real(FSTotal).';
%==========================================================================

%%Question 9:
```

```
%See page 9 in the report
%==========================================================================

%%Question 10: Plot and listen to the Recovered Speech Signal
figure
plot(t, dnSnd)
title("De-Noised Speech Signal")
xlabel("Time (s)")
ylabel("Amplitude")
sound(dnSnd,44100);
```