UCD School of Electrical, Electronic
& Communications Engineering

EEEN30110 Signals & Systems

Lab 3 Signals and Systems Report

Fergal Lonergan 13456938

Declaration:
I declare that the work described in this report was done by the person named above, and that the description and comments in this report are my own work, except where otherwise acknowledged. I have read and understand the consequences of plagiarism as discussed in the EECE School Policy on Plagiarism, the UCD Plagiarism Policy and the UCD Briefing Document on Academic Integrity and Plagiarism. I also understand the definition of plagiarism.

Signed: ______________________________

Date: ______________________________

Lab 3 Signals and Systems Report

Objective:

To investigate applications of the Fourier Transform and filtering.

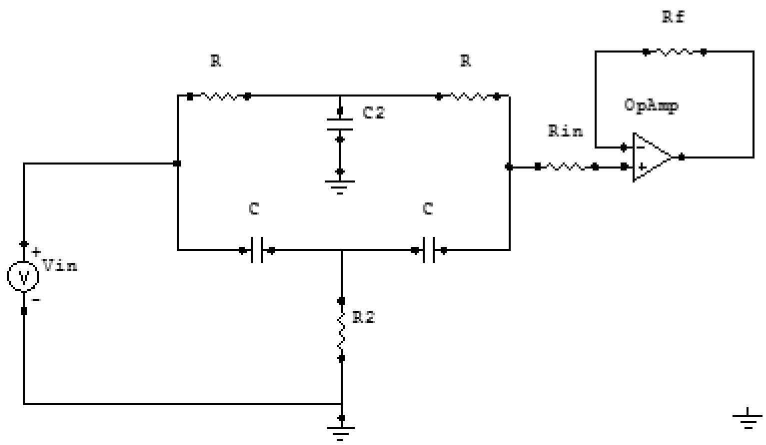


Figure 1: Twin Tee Filter

The Twin Tee filter completely blocks a particular positive frequency. Find an expression for this frequency in terms of the parameters K, R and C of the transfer function.

$$H(j\omega) = \frac{K\left((j\omega)^2 + \left(\frac{1}{RC}\right)^2\right)}{(j\omega)^2 + \left(\frac{4}{RC}\right)(j\omega) + \left(\frac{1}{RC}\right)^2}$$

Question 1

The Twin Tee filter completely blocks a particular positive frequency. Find an expression for this frequency in terms of the parameters K, R and C of the transfer function.

In order to this this we must take our formula for the transfer function above and set it equal to 0. After doing this we can then find our expression of ω in terns of K, R *and* C.

$$H(j\omega) = \frac{K\left((j\omega)^2 + \left(\frac{1}{RC}\right)^2\right)}{(j\omega)^2 + \left(\frac{4}{RC}\right)(j\omega) + \left(\frac{1}{RC}\right)^2}$$

$$0 = \frac{K\left((j\omega)^2 + \left(\frac{1}{RC}\right)^2\right)}{(jw)^2 + \left(\frac{4}{RC}\right)(j\omega) + \left(\frac{1}{RC}\right)^2}$$

Therefore we can just let the top equal to 0 so:

$$0 = K\left((j\omega)^2 + \left(\frac{1}{RC}\right)^2\right)$$

And dividing both sides by K, squaring our terms and getting ω on its own we get:

$$\omega = \frac{1}{RC}$$

Question 2:

Import the file AudioData.wav into Matlab. What is the sampling frequency?

After importing the AudioData.wav file into MATLAB you just read the value for f_s which is equal to 11025Hz.

Question 3:

The Audio file has been corrupted by the addition of a relatively large amplitude jamming sinusoid of a certain frequency. If possible confirm this by playing the corrupted file. You should hear a tone. Plot the corrupted file to see what it looks like in the time domain. By determining the spectrum of the resulting corrupted signal estimate the frequency of the jamming signal.

Using MATLAB I opened and played the sound using the sound() command. I then proceeded to plot the signal in the time domain (see Figure 1) and the in the frequency domain (see Figure 2). After analyzing Figure 2 further (see Figure 3) we can see that the jamming signal appears to be 400Hz.

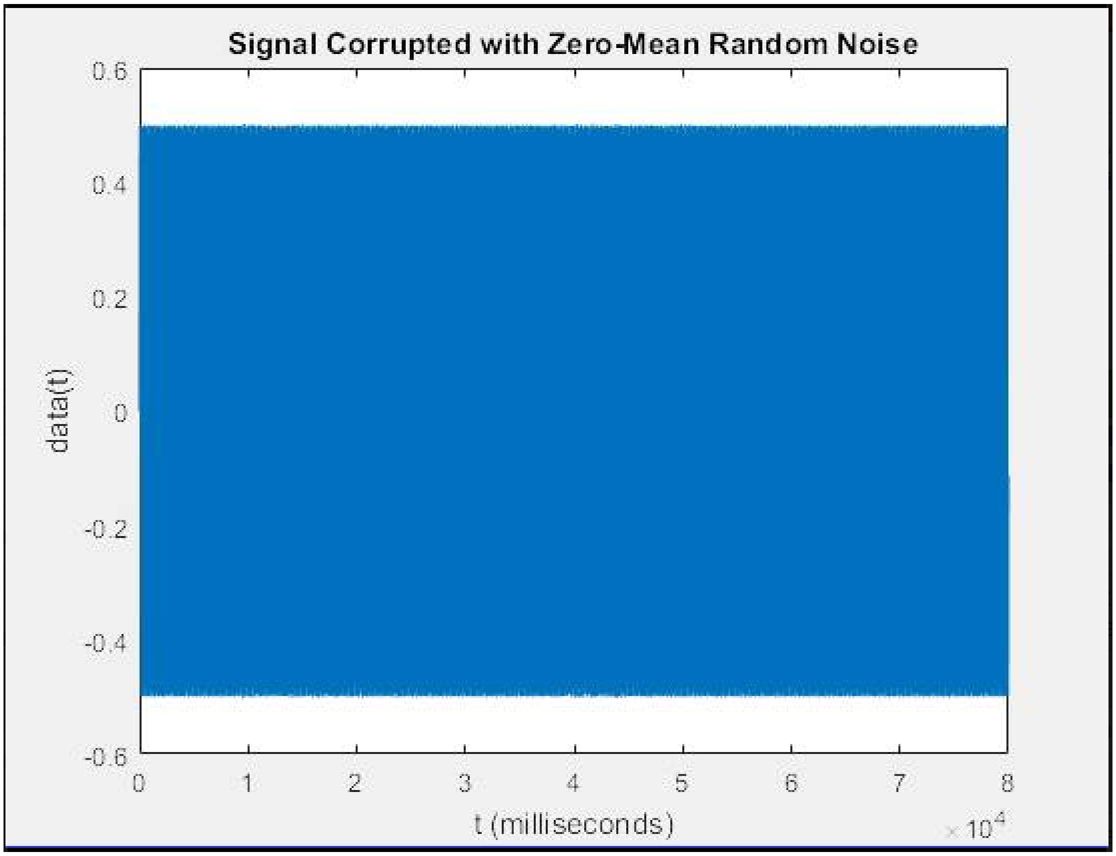

Signal Corrupted with Zero-Mean Random Noise
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
data(t)
0
1
2
3
4
5
6
7
8
t (milliseconds)
× 10^4

Figure 1

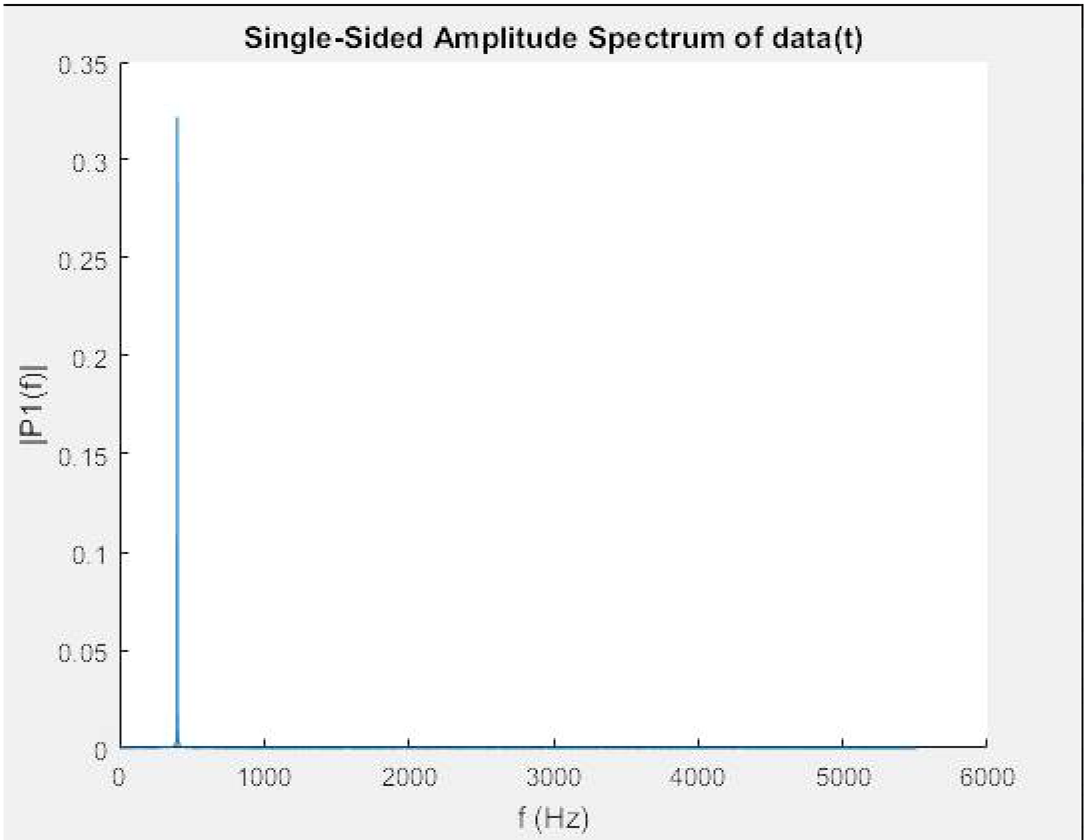

Single-Sided Amplitude Spectrum of data(t)
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
|P1(f)|
0
1000
2000
3000
4000
5000
6000
f (Hz)

Figure 2

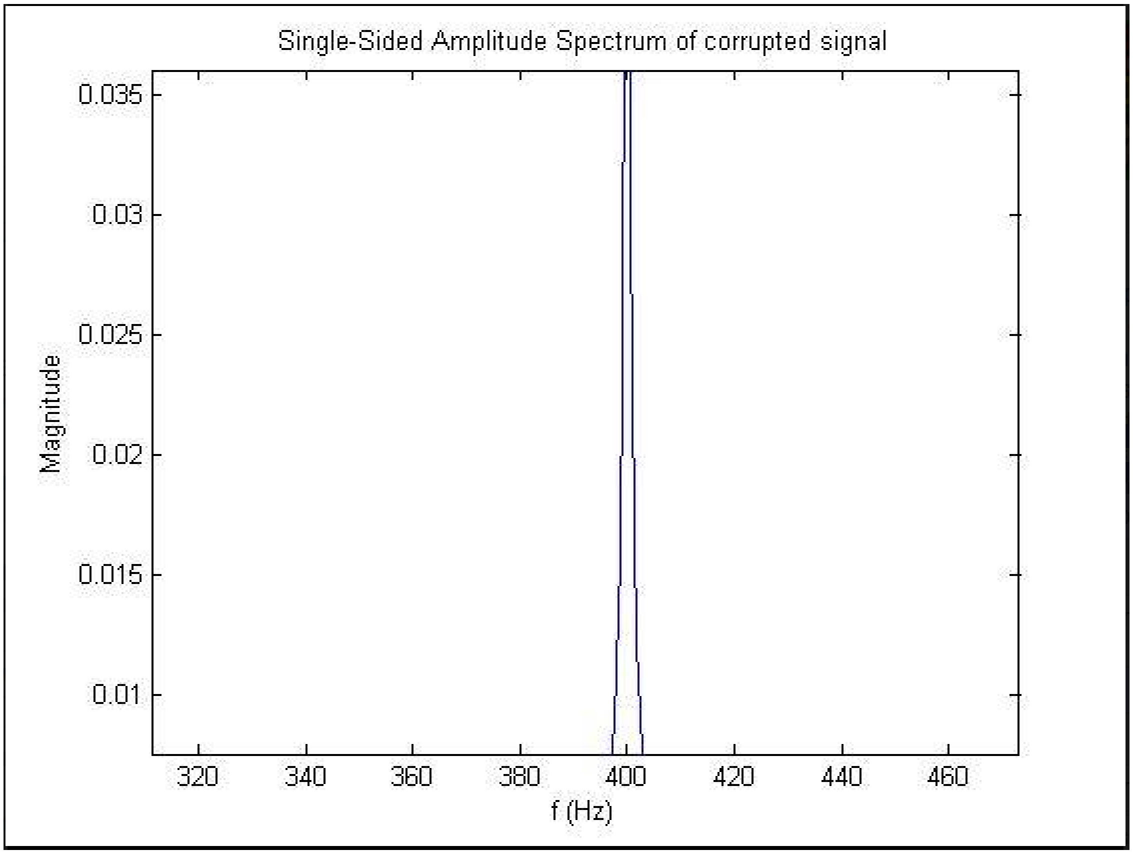


Figure 3

Question 4:

Design a Twin Tee filter (i.e. specify its transfer function) so that the jamming sinusoidal frequency should be blocked. You will have to experiment a little with gain value K so that you can hear the signal. Do *not* make this gain too large, the resulting signal may be too loud to be safely played over insertion-style headphones. For now set it to one.

In order to design a Twin Tee filter that blocks out our jamming signal of 400Hz we must let our $\frac{1}{RC}$ be equal to ω for that frequency. Therefore for every $\frac{1}{RC}$ we sub in $2\pi * 400Hz$ or $800\pi Hz$.

$$H(j\omega) = \frac{K((j\omega)^2 + (800\pi Hz)^2)}{(jw)^2 + (4 * 800\pi Hz)(j\omega) + (800\pi Hz)^2}$$

$$H(s) = \frac{K((s)^2 + (800\pi Hz)^2)}{(s)^2 + (4 * 800\pi Hz)(s) + (800\pi Hz)^2}$$

Then utilizing the OKAWA online Twin-Tee simulation tool I found resistance and capacitances that could be implemented to design our circuit.

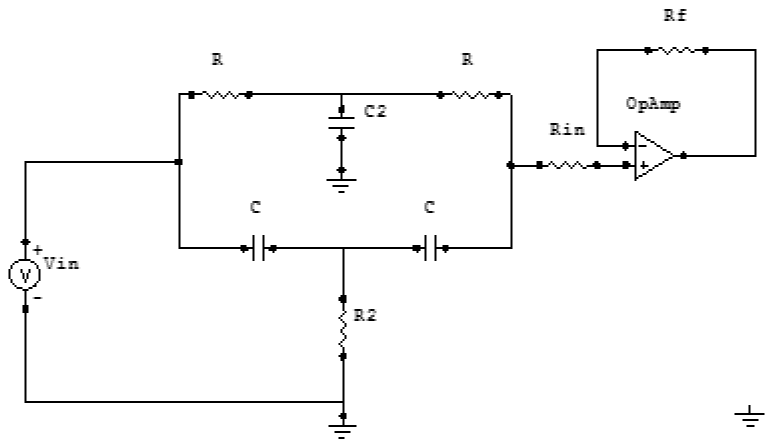


Figure 1: Twin Tee Filter

COMPONENT	VALUE
R	5.6kΩ
R_2	1.5kΩ
C	0.1μF
C_2	0.1μF

R_{in} *and* R_f *must be of a significant ratio so that the output signal is visible.*

V_{in} *is our input signal*

We can now simulate the circuit and find that the center rejection frequency lies between:

f0 = 401.926926856[Hz]
f0 = 388.298301434[Hz]

As our circuit is supposed to reject frequencies of 400Hz this circuit is a very good model.

Therefore the poles and zeros of our circuit will be:

Pole(s)

p = -628.903260402

$|p|$= 100.09306262[Hz]

p = -9693.48413655

$|p|$= 1542.76591612[Hz]

p = -3487.13641257

$|p|$= 554.994997296[Hz]

Zero(s)

z = -40.9183702141 +2467.85329518i

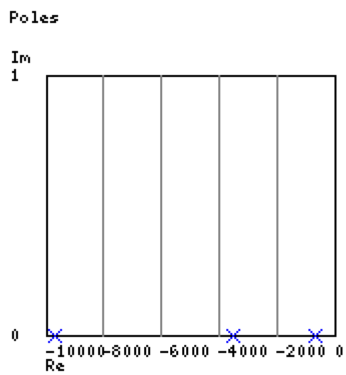
$|z|$= 392.825036312[Hz]

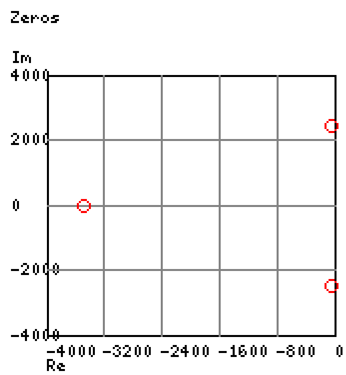
z = -3489.591831

$|z|$= 555.385789277[Hz]

z = -40.9183702141-2467.85329518i

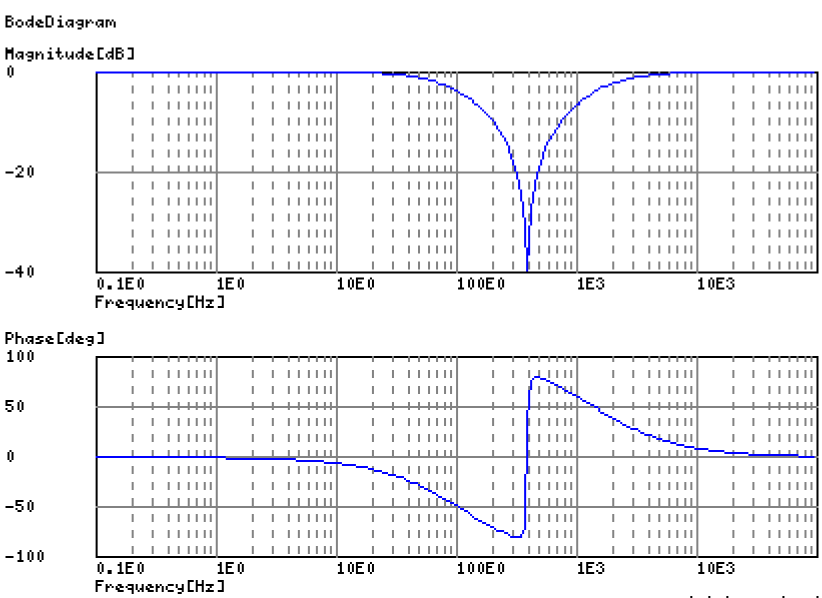
$|z|$= 392.825036312[Hz]



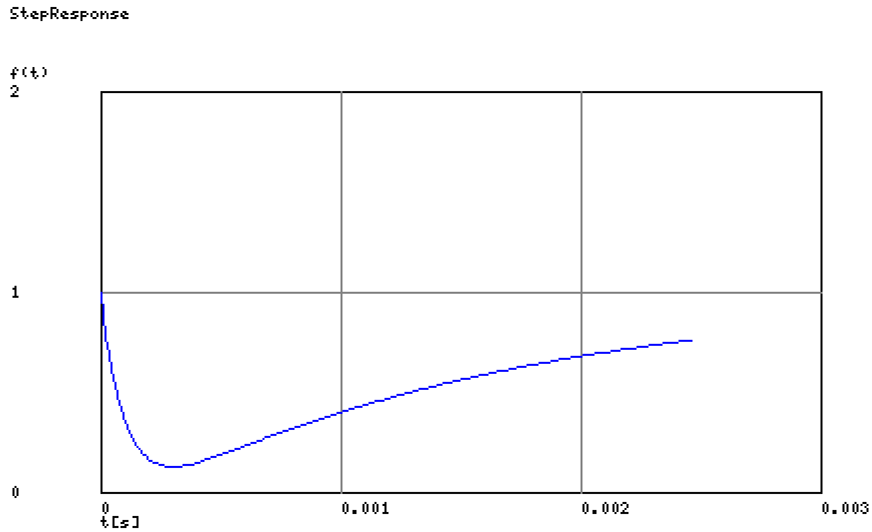


Therefore our system doesn't oscillate.

And we get the following Bode diagrams:



Finally it provides us with the following transient analysis:



Question 5:

Plot the frequency response of the resulting Twin Tee filter and confirm that it should be expected to block the undesired jamming frequency but pass most other frequencies with essentially constant gain.

Then using MATLAB we plot the magnitude of the transfer function from Question 4 in relation to frequency we get the following graph (Figure 4) which shows the filtering of frequencies at 400Hz and the relative passing of all other frequencies.

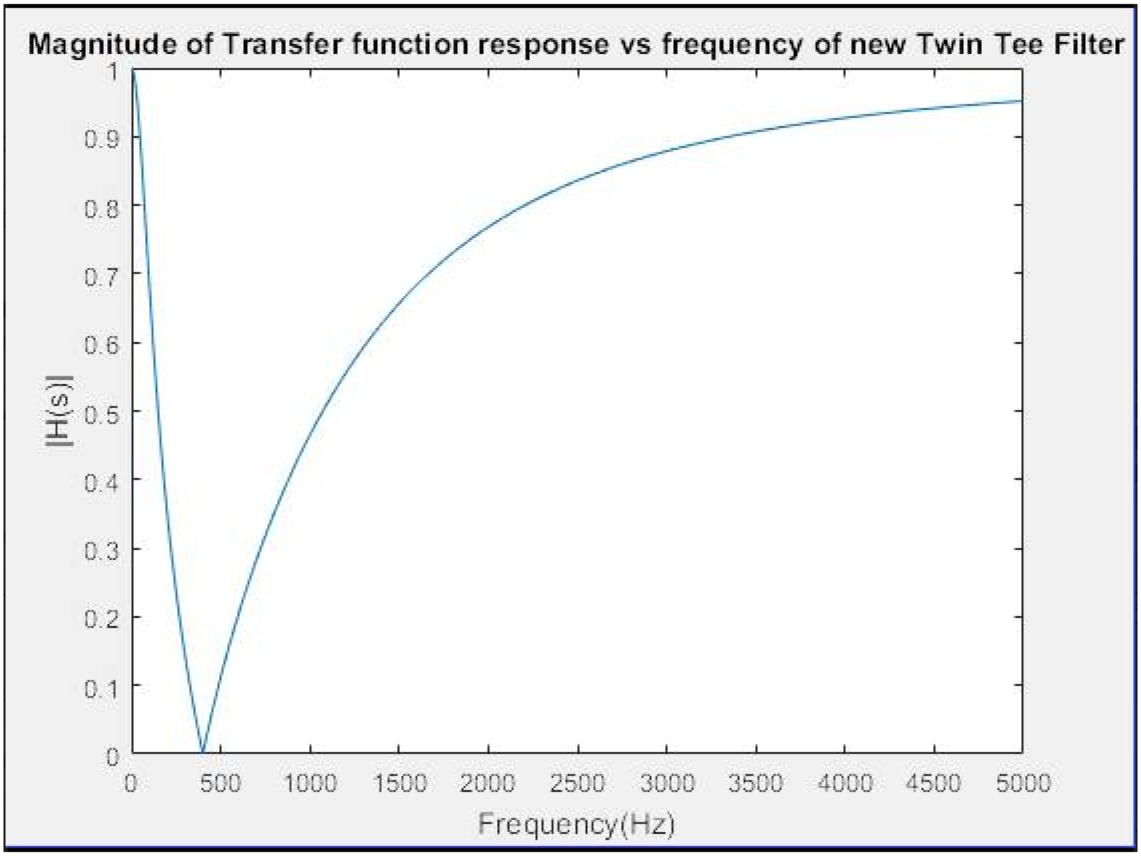


Figure 4

Question 6:

> Using Matlab's **lsim** command find the output of the Twin Tee filter for input equal to the corrupted data. If possible listen to the resulting output signal, increasing the gain in increments until it is audible. Although the elimination of the jamming signal is far from perfect the underlying, original audio file should now be recognisable. Finally plot the magnitude spectrum of the output of the Twin Tee and, by comparing it with the magnitude spectrum of the input, explain why the audio file is now (or should now be) recognisable.

Now utilizing MATLAB's lsim() command, I plot the output of my sound signal when passed through my filter. (See Figure 5) However after using the sound() command it is clear that the amplitude is much to low so I redefine my Transfer function and set K to be equal to 50 (See Figure 6) After then utilizing the lsim() and sound() commands again I am now able to hear Mike Meyers, in character as Austin Powers, say "Yeah Baby".

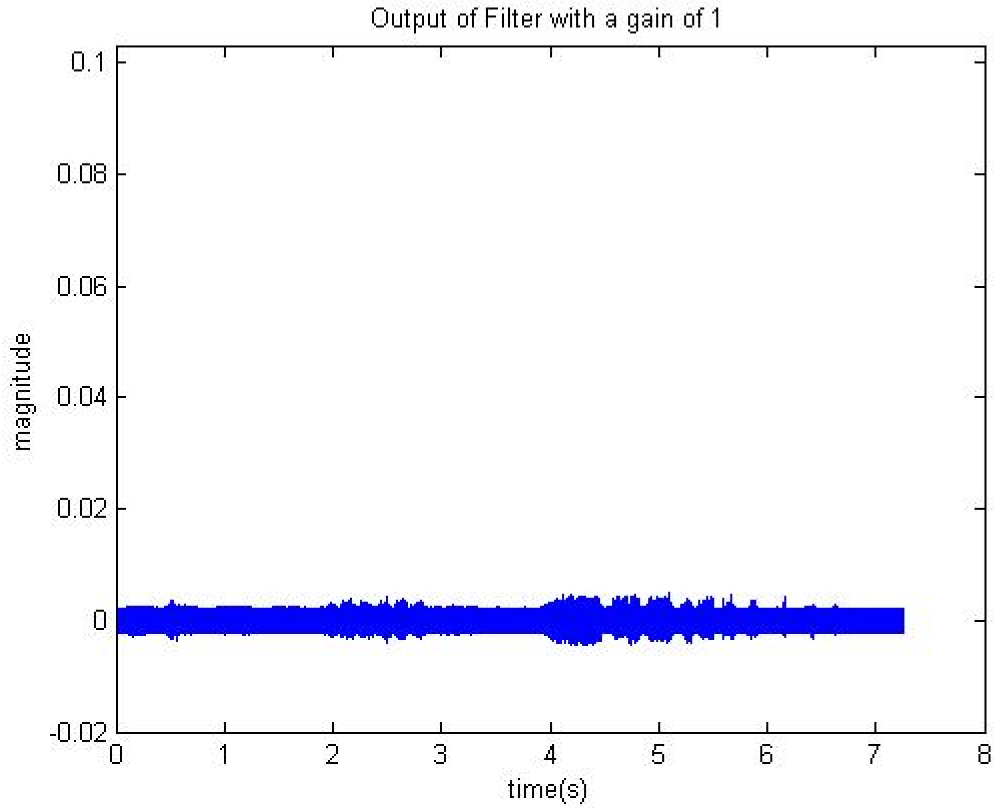

Output of Filter with a gain of 1
magnitude
time(s)
0.1
0.08
0.06
0.04
0.02
0
-0.02
0
1
2
3
4
5
6
7
8

Figure 5

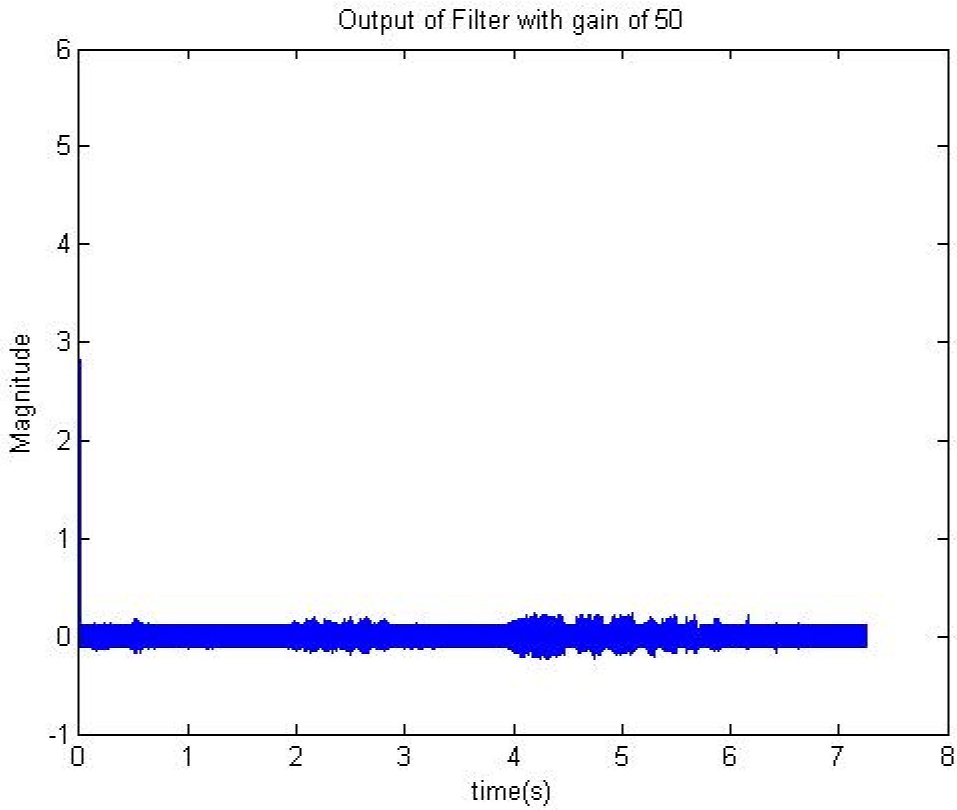


Figure 6

Finally I plot the magnitude spectrum of the output of my Twin Tee filter. (See Figure 7). I then plot the input versus the output (Figure 8). We can see that the filter has filtered the jamming signal greatly, from a magnitude of 0.16 to 0.069 and that in relation to the input signal the magnitude of the output signal at other frequencies appear to be greater than that of the input signal. This would explain why the sound clip is now audible where it wasn't before.

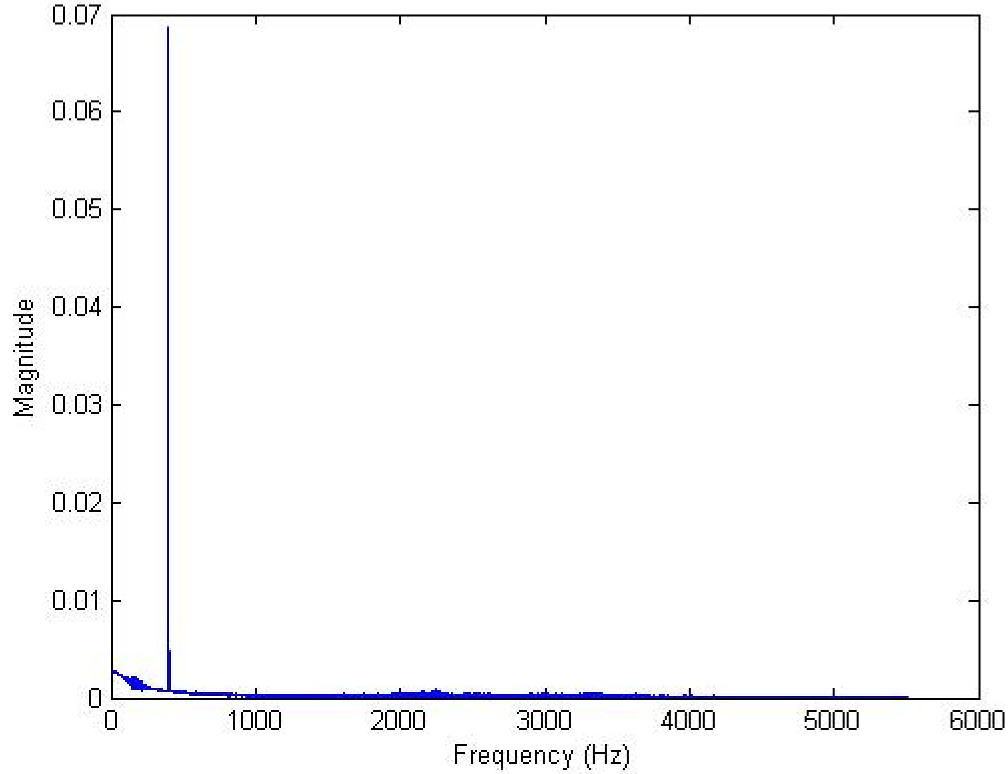
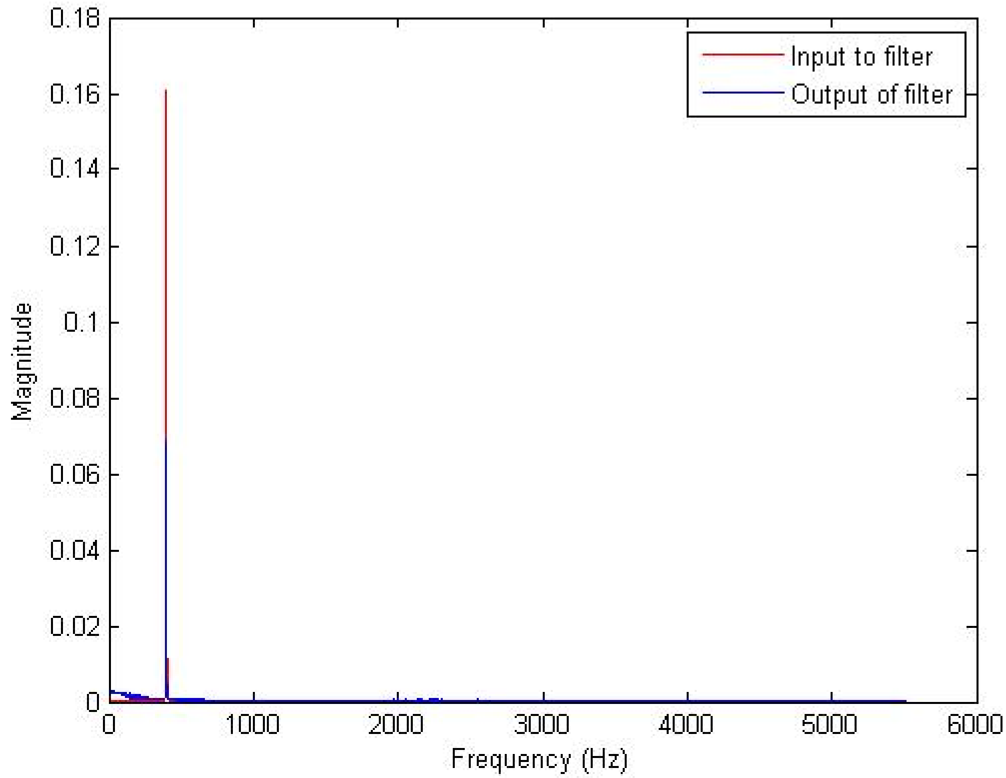

0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
Magnitude
0
1000
2000
3000
4000
5000
6000
Frequency (Hz)

Figure 7



0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
Magnitude
0
1000
2000
3000
4000
5000
6000
Frequency (Hz)
Input to filter
Output of filter

Figure 8