

Digital Signals Processing and Communications

Digital Signals Processing.

Richard Woodhams

Gower College Swansea (UWTSD)

<u>Contents</u>	<u>Page</u>
Title page	1
Contents	2
Introduction	3
Task 1	4 - 6
Task 2	7 - 10
Task 3	11 - 13
Task 4	14 - 16
References	17
Appendix A – Task 2 Full Code	18

Introduction

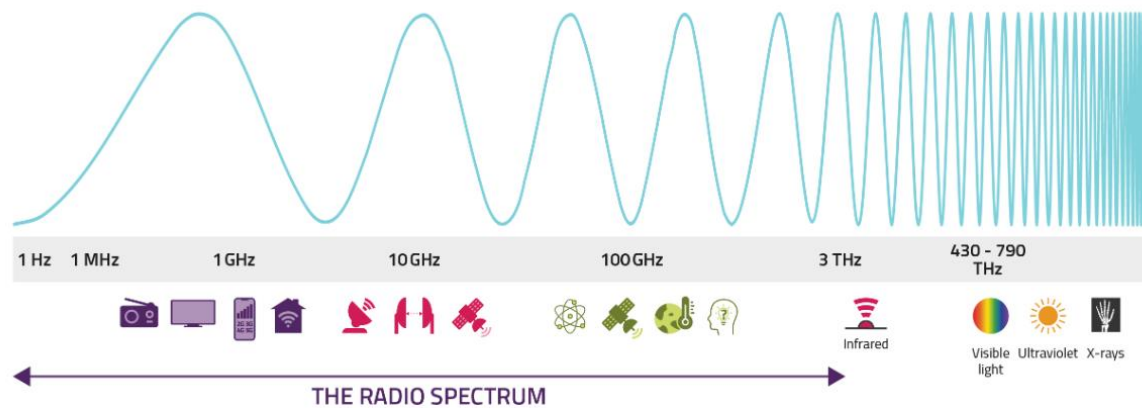
This report presents a comprehensive investigation into four key areas of digital signal processing and mobile communications. It begins with an analysis of how mobile networks optimise capacity through techniques such as TDMA and WCDMA, including interference management strategies. Following this, the report explores filtering techniques applied in MATLAB to clean a corrupted ringing signal (TMPRS), and to process biomedical and communication signals. Each section includes custom MATLAB code and theoretical analysis. These tasks collectively demonstrate practical and theoretical competence in DSP and communication systems.

Task 1

Capacity Optimisation

Capacity optimisation in the context of telecommunications refers to maximising the effectiveness, availability, and reliability of wireless networking. Broadly speaking, different types of devices will use different ranges of the radio spectrum to communicate wirelessly.

The electromagnetic spectrum



(Ofcom, 2023)

The capacity at each frequency is dependant on the infrastructure that is in place to broadcast it. Telecoms infrastructure is made up of transmitters and receivers ranging from femtocells that are typically the size of a home router and used within buildings, to small cells that have a range of around 100 meters. The next size are macrocells which are on towers or masts. They have a range of a few miles and will typically form the wireless grid from which most signals will be broadcast and received (Goss, 2023).

Macrocells vs. small cells vs. femtocells

	 MACROCELLS	 SMALL CELLS	 FEMTOCELLS
SIZE	Around 50 to 200 feet tall	The size of a pizza box	The size of a paperback book
AVERAGE COVERAGE RANGE	A few miles	A football field -- 100 yards	A home or small business
AVERAGE COST TO INSTALL	\$200,000	Under \$10,000	Around \$100
DEPLOYMENT	The U.S. has about 200,000 macrocells	The U.S. will have 5 to 10 times more small cells than macrocells once fully deployed	Anyone can purchase a femtocell for their home or small business



(Techtarget.com, 2020).

Ultimately, the best way to increase network capacity is to *densify networks by adding additional macro sites as needed*. Operators need to take a view on the most appropriate mode of densification to pursue in practice, according to the characteristics and anticipated demand of a particular area. In dense urban areas and busy locations such as a sports stadium, small cells with mmWave or existing mid-band spectrum might be preferred. Small cells could be deployed in busier rural or suburban areas, but, in some contexts, operators might prefer to deploy new macro sites. (Ofcom, 2022). This would increase capacity simply by making more spectrum available.

In terms of optimising and managing the current mobile network capacity, a technique called network slicing is particularly effective in 5G networks. Network slicing is the process of splitting up the 5G network into virtual elements that can be customised to work for specific types of connection. *This network architecture is based on the principles of network functions virtualisation (NFV) and software-defined networking (SDN), two infrastructure models that allow traditional networks to be broken down into virtual (software-driven) elements. In practice, each virtual network represents a portion of the larger network, is made up of a set of independent functions and has its own network architecture, provisioning and engineering mechanism. In turn, the general network hosts other slices, to which it assigns individually the network topology and resources they require to run (connectivity, coverage, speed, bandwidth, etc.), keeping the traffic of each of them completely insulated so that they do not interfere with each other* (Telefonica.com, 2023).

Time Division Multiple Access (TDMA) in Global System for Mobile Communications (GSM) networks

Time division multiple access in global system for mobile communications is a technology that splits a single frequency channel into eight time slots. Each time slot is allocated to one user. *Users take turns in using the channel in an efficient and timely manner. Guard times of 30-50 microseconds between time slots are commonly built in to prevent overlapping and interference between users* (Awati, 2021). The duration of each timeslot is so quick and is repeated cyclically so the user does not notice any drop in connection. In reality, the user's connection is picked up and dropped off at the beginning and end of their allocated time slot. This type of technology allows for the full frequency to be used by each user during their time slot. An advantage of TDMA is that it offers better spectral efficiency than analogue systems. *Spectral efficiency refers to the amount of data that can be transmitted over a specific bandwidth* (Taranawireless.com, 2021). A disadvantage of TDMA is the need for guard intervals, mentioned previously. These are essentially time slots that are not used for any kind of transmission and so can be seen as lost time for communications.

Wideband Code Division Multiple Access (WCDMA) in Universal Mobile Telecommunications System (UMTS) networks

Wideband code division multiple access is a universal mobile telecommunications system standard and is based on CDMA. WCDMA is an air-interface technology that uses radio spectrum for data transfer. A different modulation scheme and a wider bandwidth allows WCDMA to provide improved data transfer at higher speeds when compared to other older 2G technologies.

The International Mobile Telecommunications-2000 (IMT 2000) recognised WCDMA as a standard network model to create multimedia services and global roaming. It is a direct spread technology which can spread its transmissions over a 5MHz carrier (Deepti, 2024).

Wideband code division multiple access is similar to time division multiple access in that it allows multiple users access to the same frequency band. An important difference is that it assigns unique digital codes, known as spreading codes, to each user which allows all users to access the full frequency band at the same time rather than having to assign time slots. This has the advantage of not having to create guard intervals and losing transmission time for each user. The higher frequency of 5MHz also allows for faster downloads.

Frequency Reuse and Interference Mitigation

Frequency reuse is a principle in mobile networks that allows the same radio frequencies to be used in different locations, but only where there is adequate separation between cells to avoid interference. In GSM networks that use time division multiple access, frequency reuse usually involves a reuse factor of 3 or 4 to prevent interference between adjacent cells. Wideband code division multiple access networks, on the other hand, are designed with a reuse factor of 1, meaning every cell can use the same frequency. This is possible because of the digital coding that WCDMA systems apply to each user, meaning that the same frequencies cannot cause interruption to each other, even if they overlap as the user can only access one cell at a time.

To maintain clear and stable connections, various interference mitigation techniques are used. Power control is essential, especially in WCDMA, to ensure that all users transmit at the minimum power necessary to

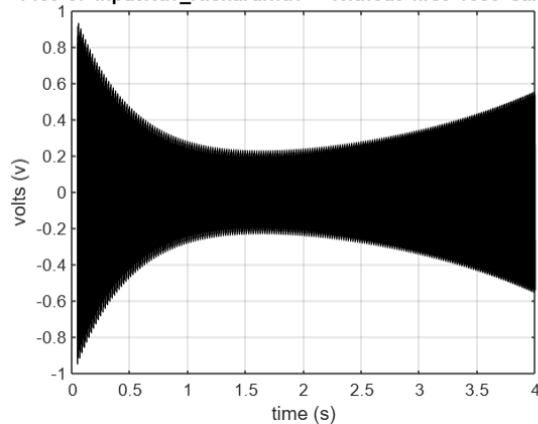
maintain signal quality. This reduces the overall interference within the network. Interference cancellation techniques, such as advanced signal processing and filtering, help isolate desired signals from unwanted ones. GSM networks rely heavily on frequency planning to avoid overlap between neighbouring cells, while newer systems also use adaptive modulation and coding schemes to adjust transmission parameters in real time based on network conditions. Together, these methods help ensure reliable communication and optimal network performance in increasingly complex and dense mobile environments.

Task 2

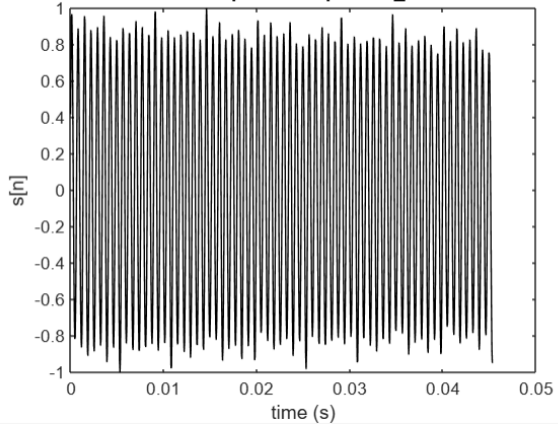
a) Read the TMPRS .wav file provided.

```
% DSP_Assignment_1_Task_2.m
% Evaluation and Filtering of a Telephone Mobile Power Ringing System (TMPRS)
clear all
close all
clc
newfilename = 'NewWav_Richard.wav'; % Output sound file name
d=4; % Pause time (s)
[s,fs]=audioread('InputWav_Richard.wav'); % Input sound file name
fs % display sample frequency
t = (0:length(s)-1)/fs;
figure(1)
w=length(t);
plot(t(1000:w-1),s(1000:w-1),'k')
grid
xlabel('time (s)')
ylabel('volts (v)')
title('Plot of InputWav\_Richard.wav - Without first 1000 samples')
figure (2)
n = 1000; % Plot first 1000 samples of InputWav_Richard.wav
plot(t(1:n),s(1:n),'k')
xlabel('time (s)')
ylabel('s[n]')
title('First 1000 samples of InputWav\_Richard.wav')
figure (3)
plot(t,s,'k')
grid
xlabel('time (s)')
ylabel('Volts (v)')
title('Plot of InputWav\_Richard.wav')
```

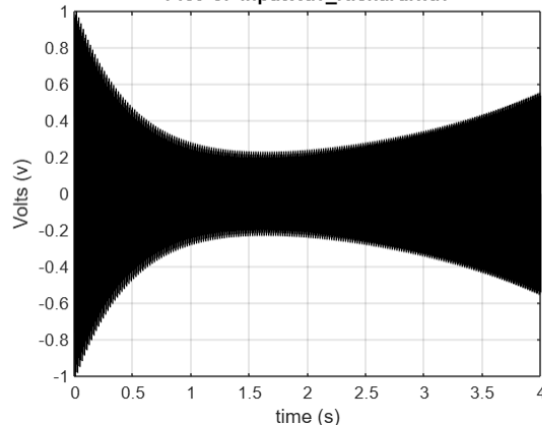
Plot of InputWav_Richard.wav - Without first 1000 samples



First 1000 samples of InputWav_Richard.wav



Plot of InputWav_Richard.wav



b) Identify the frequencies of each tone in the signal.

```
sound(s, fs); % Play InputWav_Richard.wav
pause(d + 0.5); % Wait for sound end
p=fft(s);
L=length(s);
p2=abs(p/L);
p1=p2(1:L/2+1);
k=fs*(0:(L/2))/L;
figure(4)
plot(k,p1)
plot(k(1:15000),p1(1:15000),'k')
grid
xlabel('frequency (Hz)')
ylabel('magnitude')
title('FFT of InputWav\_Richard.wav')
```

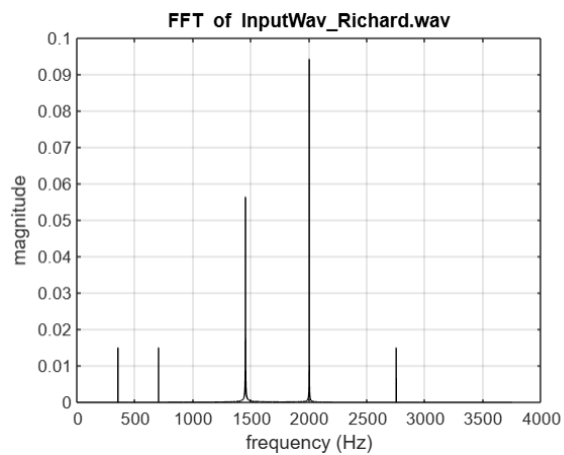


Figure 4 shows that

f1 = 350Hz

f2 = 700Hz

f3 = 1450Hz

f4 = 2000Hz

f5 = 2750Hz

This helps determine the high and low cut off frequencies to remove f3 and f4.

c) Design a filter to remove the source of the unpleasant sound.

I chose to cut off frequencies of 1300Hz – 1600Hz for f3 and 1800Hz – 2200Hz for f4.

```
% eliminate the first frequency (f3) from the signal
LCoff1 = 1300/(fs/2); % low cutoff frequency
HCoff1 = 1600/(fs/2); %high cutoff frequency
ORD1=500; % order of FIR bandstop filter.
filtcoeff1=fir1(ORD1,[LCoff1, HCoff1],'stop');% returns the filter coefficients in length ORD1+1
vector filtcoeff1
%y = filter( b, a, x);
% filters the data in vector 'x' {in this case s} with the filter described by coefficient vectors 'a'
and 'b' (filtcoeff1 vectors b, a=1 FIR filter]
% to create the filtered data vector 'y' [in this case voic1] .
voic1=filter(filtcoeff1,1,s); % generate a new voice with only four frequencies
sound(voic1,fs);
pause(d + 0.5); % waiting for sound end
% eliminate the second frequency (f4) from the voice signal
LCoff2 = 1800/(fs/2); % low cutoff frequency
HCoff2 = 2200/(fs/2); %high cutoff frequency
ORD2=500; % order of FIR bandstop filter.
filtcoeff2=fir1(ORD2,[LCoff2, HCoff2],'stop'); % returns the filter coefficients in length ORD2+1
vector filtcoeff2
% filters the data in vector 'x' {in this case voic1] with the filter described by coefficient vectors
'a' and 'b' (filtcoeff1 vectors b]
% to create the filtered data vector 'y' [in this case voic2] .
voic2=filter(filtcoeff2,1,voic1);
sound(voic2,fs); % generate a new voice with only three frequencies
```

d) Generate a clean version of the tone that retains only the three desired frequencies.

```
figure(5)
plot(t(1000:w-1),voic2 (1000:w-1),'k')
grid
xlabel('time-secs')
ylabel('signal value - volts')
title('A Plot of the new voice')
pause(d + 0.5); % waiting for sound end
figure(6)
L1=length(voic2);
mag1=abs(fft(voic2));
kk=fs*((0:(L1-1))/L1);
plot(kk(1:15000),mag1(1:15000),'k')
grid
xlabel('frequency (Hz)')
ylabel('magnitude')
title ('FFT of the newVoice')
```

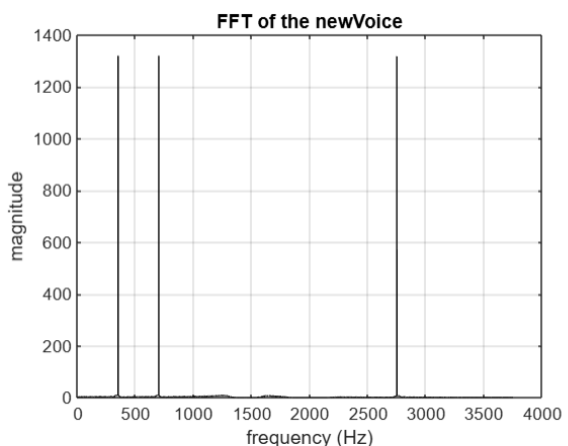
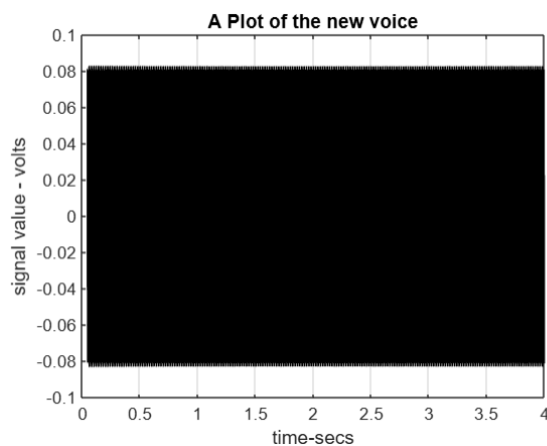


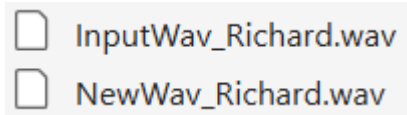
Figure 5 shows the plot of the new voice with no undesired tones. Figure 6 shows the undesired frequencies having been removed entirely.

e) Produce a new version of the TMPRS without the corrupted tones.

The code below creates a new .wav file using the new file name that I input during step one – NewWav_Richard.wav.

```
% Save wav file to disk  
audiowrite(newfilename,voic2,fs);
```

Below is a screenshot of all the .wav files now contained in my MATLAB folder.

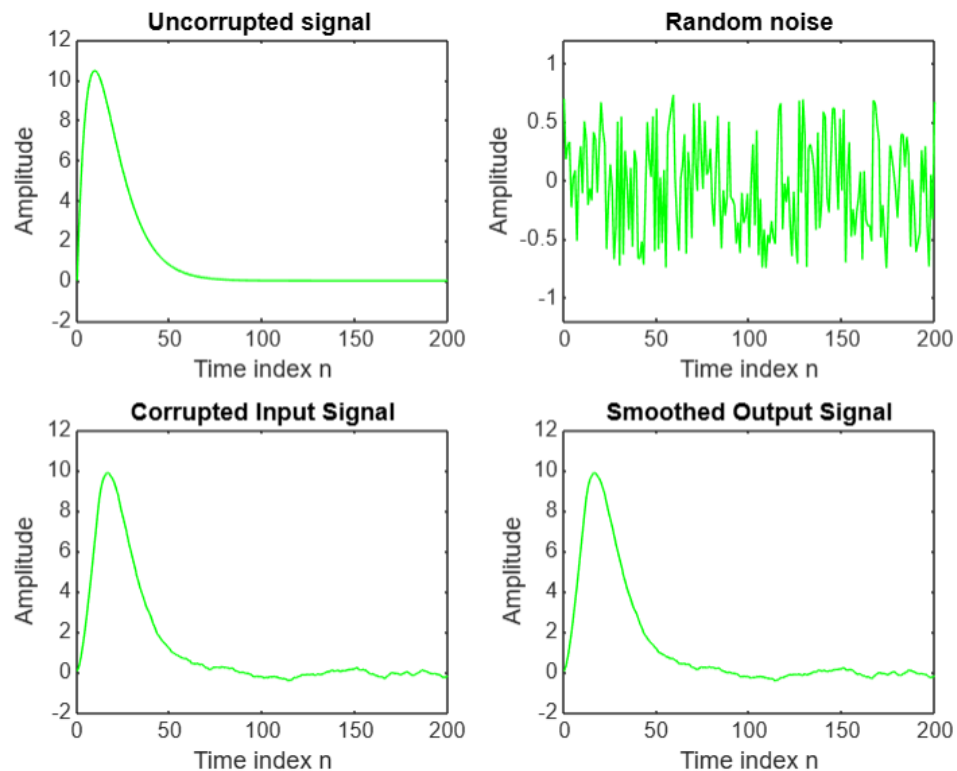


Task 3

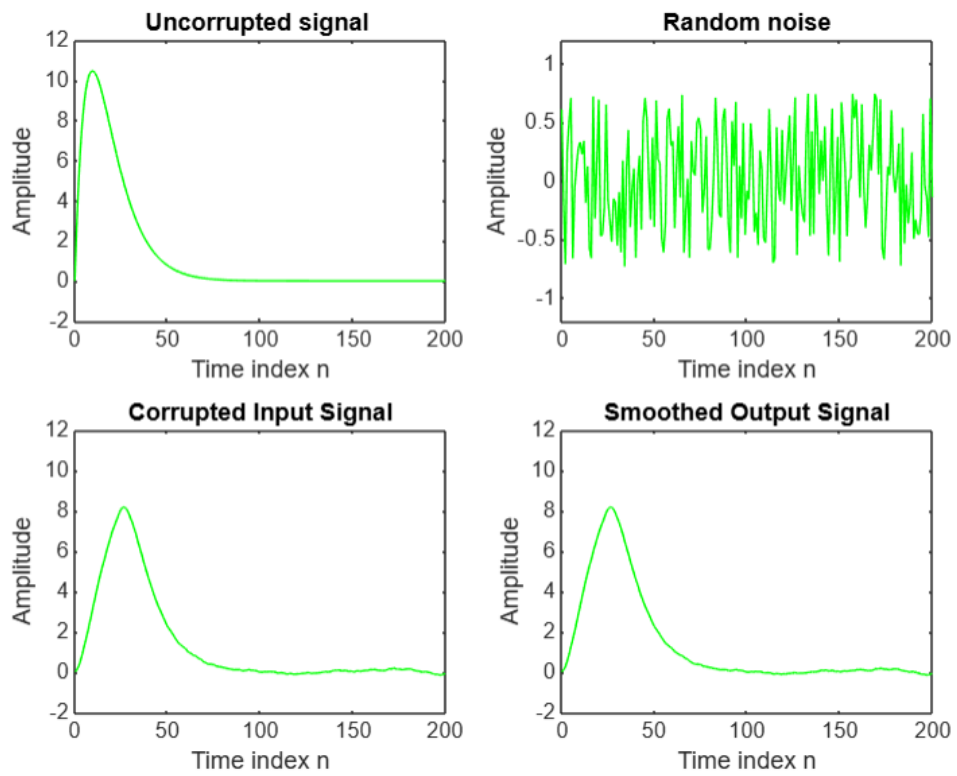
a. Biomedical Signal Processing Using Moving Average Filtering

```
% Biomedical Signal Processing Using Moving Average Filtering
R = 201;
noi = 1.5*(rand(R,1) - 0.5); % Random noise generation
n = 0:R-1;
s = 3*n.*(0.9.^n); % Uncorrupted signal
y = s + noi; % Uncorrupted signal + Random noise
M = input('Choice of M value (filter length) = ');
num = ones(1,M);
y = filter(num,1,y)/M;
% Display the input and output signals
clf;
subplot(2,2,1);
plot(n,s,'g');
axis([0, 200, -2, 12]);
xlabel('Time index n'); ylabel('Amplitude');
title('Uncorrupted signal ');
subplot(2,2,2);
plot(n, noi,'g');
axis([0, 200, -1.2, 1.2]);
xlabel('Time index n'); ylabel('Amplitude');
title('Random noise');
subplot(2,2,3);
plot(n, y,'g');
axis([0, 200, -2, 12]);
xlabel('Time index n'); ylabel('Amplitude');
title('Corrupted Input Signal');
subplot(2,2,4);
plot(n, y,'g');
axis([0, 200, -2, 12]);
xlabel('Time index n'); ylabel('Amplitude');
title('Smoothed Output Signal');
axis;
```

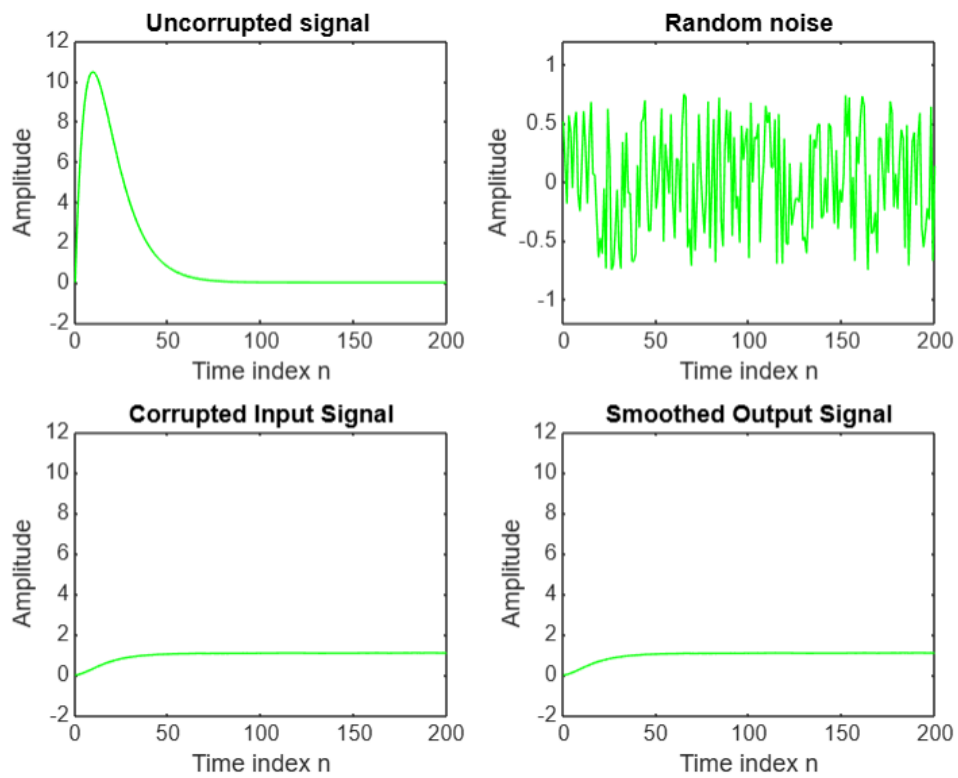
M = 12



M = 25



M = 250



The above plots demonstrate how increasing the filter length will smooth out an input signal. The first example, where I have chosen $M=12$, shows that there is very little filtering. The greatest amplitude is reduced slightly and some of the ripple along the input signal is smoothed out. In the world of biomedical science, this filter length would not be particularly useful as it has not cleaned up the input signal to anything near the original, uncorrupted signal.

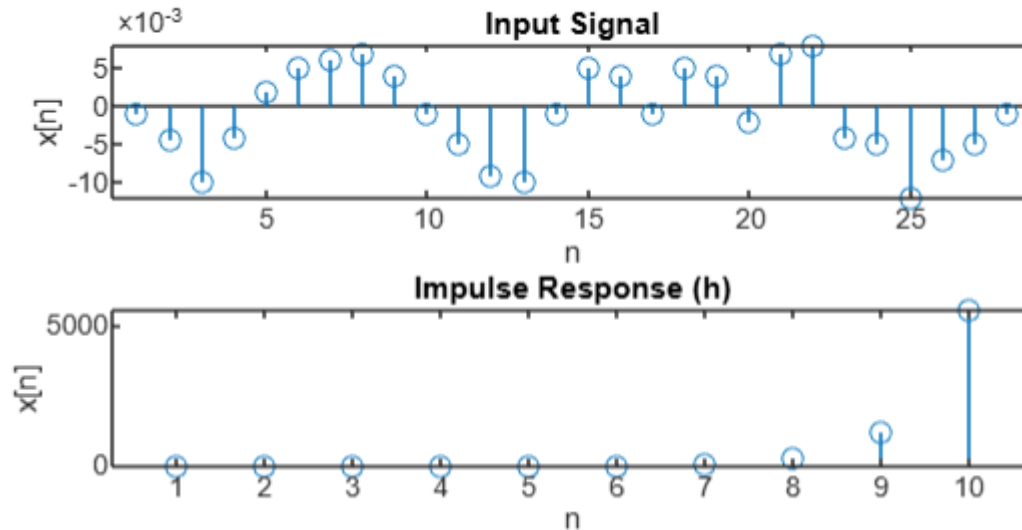
The second example, where I have chosen $M=25$, shows that slightly more filtering has taken place. The output signal is smoothed more than previously. Compared to the original, uncorrupted signal, the output signal has been distorted slightly with a reduced amplitude and slightly delayed response.

The third example, where I have chosen $M=250$, shows a vastly smoothed and distorted output. The signal has gone almost completely flat. This has filtered out any useful information within the signal which would be detrimental to its use within biomedical science as there is no resemblance to the clean, original, uncorrupted signal.

Task 4

A and B.

```
x=[-0.001 -0.0045 -0.01 -0.0042 0.002 0.005 0.006 0.007 0.004 -0.001 -0.005 -0.009 -0.01 -0.001 0.005
0.004 -0.001 0.005 0.004 -0.002 0.007 0.008 -0.004 -0.005 -0.012 -0.007 -0.005 -0.001];
n=10; %number of responses to plot.
%y[n]=3.6y[n-1]+5y[n-2]-3.2y[n-3]+0.82y[n-4]+0.013x[n]-0.032x[n-1]+0.044x[n-2]-0.033x[n-3]+0.013x[n-4]
%y[n]-3.6y[n-1]-5y[n-2]+3.2y[n-3]-0.82y[n-4]=0.013x[n]-0.032x[n-1]+0.044x[n-2]-0.033x[n-3]+0.013x[n-4]
num=[0.013 -0.032 0.044 -0.033 0.013]; %x-coefficients
den=[1 -3.6 -5 3.2 -0.82]; %y-coefficients
h=impz(num, den, n);
subplot(3,1,1);
stem(x);
title("Input Signal");
ylabel("x[n]");
xlabel("n");
%Task 4 A
subplot(3,1,2);
stem(h);
title("Impulse Response (h)");
ylabel("x[n]");
xlabel("n");
%Task 4 B
disp(h'); %show first 10 response values of h as an array
%Task 4 Finally...
Y1=filter(h',1,x); %output of low-pass filter
subplot(3,1,3);
stem(Y1);
title("Output of low-pass filter (Y1)");
ylabel("x[n]");
xlabel("n");
```



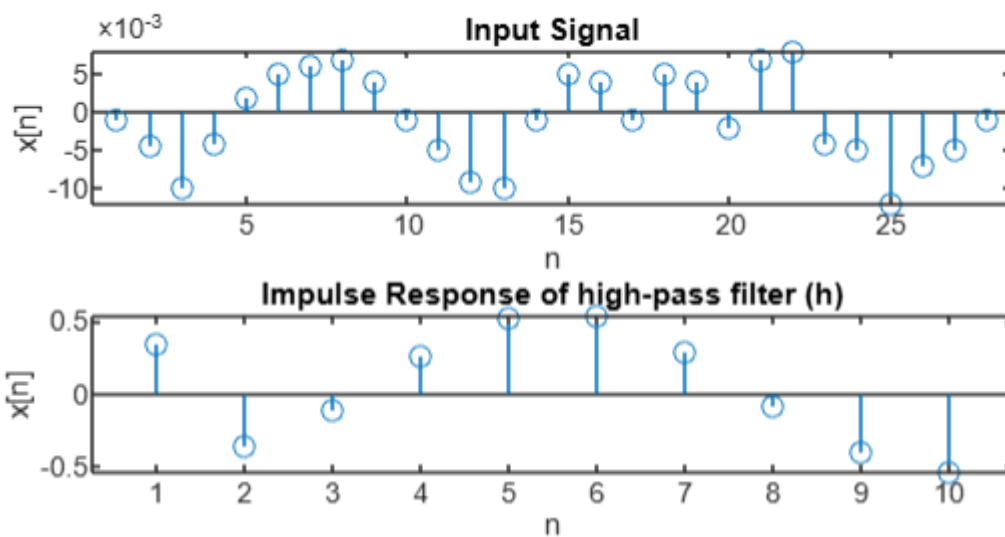
Command Window

```
>> DSP_Assignment_1_Task_4_LPF
1.0e+03 *

    0.0000    0.0000    0.0002    0.0006    0.0029    0.0128    0.0588    0.2671    1.2169    5.5384
```

C and D)

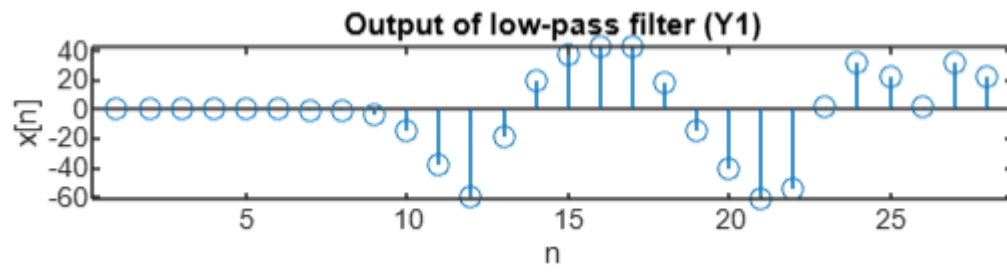
```
x=[-0.001 -0.0045 -0.01 -0.0042 0.002 0.005 0.006 0.007 0.004 -0.001 -0.005 -0.009 -0.01 -0.001 0.005
0.004 -0.001 0.005 0.004 -0.002 0.007 0.008 -0.004 -0.005 -0.012 -0.007 -0.005 -0.001];
n=10; %number of responses to plot
%y[n]=2.7y[n-1]-3.3y[n-2]+2y[n-3]-0.57y[n-4]+0.35x[n]-1.3x[n-1]+2x[n-2]-1.3x[n-3]+0.35x[n-4]
%y[n]-2.7y[n-1]+3.3y[n-2]-2y[n-3]+0.57y[n-4]=0.35x[n]-1.3x[n-1]+2x[n-2]-1.3x[n-3]+0.35x[n-4]
num=[0.35 -1.3 2 -1.3 0.35]; %x-coefficients
den=[1 -2.7 3.3 -2 0.57]; %y-coefficients
h=impz(num, den, n);
subplot(3,1,1);
stem(x);
title("Input Signal");
ylabel("x[n]");
xlabel("n");
%Task 4 C
subplot(3,1,2);
stem(h);
title("Impulse Response of high-pass filter (h)");
ylabel("x[n]");
xlabel("n");
%Task 4 D
disp(h'); %show first 10 response values of h as an array
%Task 4 Finally...
Y1=filter(h',1,x); %output of low-pass filter
subplot(3,1,3);
stem(Y1);
title("Output of high-pass filter (Y1)");
ylabel("x[n]");
xlabel("n");
```



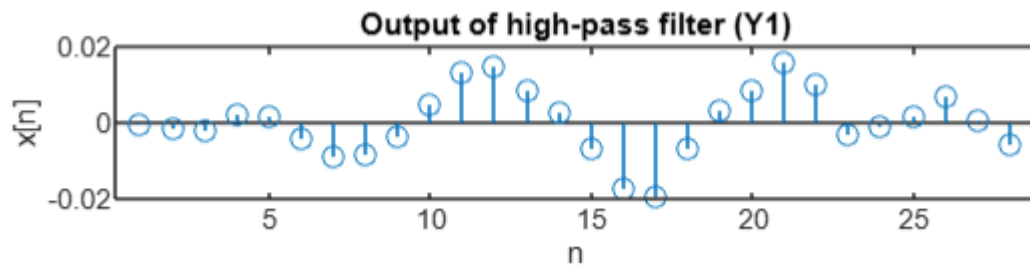
Command Window

```
>> DSP_Assignment_1_Task_4_HPF
0.3500 -0.3550 -0.1135 0.2650 0.5307 0.5335 0.2841 -0.0834 -0.3980 -0.5353
```

Output of the low pass filter.



Output of the high pass filter.



References

- Awati, R. 2021 – *What is Time Division Multiple Access (TDMA)?* [Online]
Available at: <https://www.techtarget.com/searchnetworking/definition/TDMA>
Accessed: 18 May 2025
- Deepti, Y L, 2024 – *Difference Between UMTS and WCDMA* [Online]
Available at: <https://www.tutorialspoint.com/difference-between-umts-and-wcdma>
Accessed: 18 May 2025
- Goss, M, 2023 – *Macrocell vs. small cell vs. femtocell: A 5G introduction.* [Online]
Available at: <https://www.techtarget.com/searchnetworking/feature/Macrocell-vs-small-cell-vs-femtocell>
Accessed: 19 May 2025
- Ofcom, 2022 – *Mobile networks and spectrum. In: Meeting future demand for mobile data.* [Online]
Available at: <https://www.ofcom.org.uk/siteassets/resources/documents/consultations/category-3-4-weeks>
Accessed 19 May 2025
- Ofcom, 2023 – *What is spectrum?* [Online]
Available at: <https://www.ofcom.org.uk/spectrum/radio-equipment/what-is-spectrum>
Accessed: 20 May 2023
- Taranawireless.com, 2021 – *The Importance of Spectral Efficiency* [Online]
Available at: <https://www.taranawireless.com/the-importance-of-spectral-efficiency/>
Accessed: 18 May 2025
- Techtarget.com, 2020. *Macrocell vs. small cell vs. femtocell: INFOGRAPHIC.* [Online]
Available at: <https://www.techtarget.com/searchnetworking/feature/Macrocell-vs-small-cell-vs-femtocell>
Accessed: 19 May 2025
- Telefonica.com, 2023 – *What is network slicing and what is it used for?* [Online]
Available at: <https://www.telefonica.com/en/communication-room/blog/what-is-network-slicing>
Accessed: 20 May 2025

Appendix A – Task 2 Full Code

```
% DSP_Assignment_1_Task_2.m
% Evaluation and Filtering of a Telephone Mobile Power Ringing System (TMPRS)
clear all
close all
clc
newfilename = 'NewWav_Richard.wav'; % Output sound file name
d=4; % Pause time (s)
[s,fs]=audioread('InputWav_Richard.wav'); % Input sound file name
fs % display sample frequency
t = (0:length(s)-1)/fs;
figure(1)
w=length(t);
plot(t(1000:w-1),s(1000:w-1),'k')
grid
xlabel('time (s)')
ylabel('volts (v)')
title('Plot of InputWav\_Richard.wav - Without first 1000 samples')
figure(2)
n = 1000; % Plot first 1000 samples of InputWav_Richard.wav
plot(t(1:n),s(1:n),'k')
xlabel('time (s)')
ylabel('s[n]')
title('First 1000 samples of InputWav\_Richard.wav')
figure(3)
plot(t,s,'k')
grid
xlabel('time (s)')
ylabel('Volts (v)')
title('Plot of InputWav\_Richard.wav')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Process 1, 2 and 3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sound(s, fs); % Play InputWav_Richard.wav
pause(d + 0.5); % Wait for sound end
p=fft(s);
L=length(s);
p2=abs(p/L);
p1=p2(1:L/2+1);
k=fs*(0:(L/2))/L;
figure(4)
plot(k,p1)
plot(k(1:15000),p1(1:15000),'k')
grid
xlabel('frequency (Hz)')
ylabel('magnitude')
title ('FFT of InputWav\_Richard.wav')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% fir1 filter B =fir1(N, Wn, 'filter type')designs an N'th order lowpass, high pass, band pass and band stop FIR digital filter
% and returns the filter coefficients in length N+1 vector B.Wn is cutoff frequency
% B = fir1(N,Wn,'high') designs an N'th order highpass filter
% If Wn is a two-element vector, Wn = [W1 W2], FIR1 returns an order N bandpass filter with passband W1 < W < W2.
% By default FIR1 uses a Hamming window
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% eliminate the first frequency (f3) from the signal
LCoff1 = 1300/(fs/2); % low cutoff frequency
HCoff1 =1600/(fs/2); %high cutoff frequency
ORD1=500; % order of FIR bandstop filter.
filtcoeff1=fir1(ORD1,[LCoff1, HCoff1],'stop'); % returns the filter coefficients in length ORD1+1 vector filtcoeff1
%y = filter( b, a, x);
% filters the data in vector 'x' {in this case s} with the filter described by coefficient vectors 'a' and 'b' (filtcoeff1 vectors b,
a=1 FIR filter]
% to create the filtered data vector 'y' [in this case voic1] .
voic1=filter(filtcoeff1,1,s); % generate a new voice with only four frequencies
sound(voic1,fs);
pause(d + 0.5); % waiting for sound end
% eliminate the second frequency (f4) from the voice signal
LCoff2 = 1800/(fs/2); % low cutoff frequency
HCoff2 =2200/(fs/2); %high cutoff frequency
ORD2=500; % order of FIR bandstop filter.
filtcoeff2=fir1(ORD2,[LCoff2, HCoff2],'stop'); % returns the filter coefficients in length ORD2+1 vector filtcoeff2
% filters the data in vector 'x' {in this case voic1} with the filter described by coefficient vectors 'a' and 'b' (filtcoeff1 vectors
b]
% to create the filtered data vector 'y' [in this case voic2] .
voic2=filter(filtcoeff2,1,voic1);
sound(voic2,fs); % generate a new voice with only three frequencies
figure(5)
plot(t(1000:w-1),voic2 (1000:w-1),'k')
grid
xlabel('time-secs')
ylabel('signal value - volts')
title('A Plot of the new voice')
pause(d + 0.5); % waiting for sound end
figure(6)
L1=length(voic2);
mag1=abs(fft(voic2));
kk=fs*(0:(L1-1))/L1;
plot(kk(1:15000),mag1(1:15000),'k')
grid
xlabel('frequency (Hz)')
ylabel('magnitude')
title ('FFT of the newVoice')
% Save wav file to disk
audiowrite(newfilename,voic2,fs);
```