



DSP and Communications

Student Name: Joshua Smith

Student ID: SMI16836261

Module Code: GOEE5003

Lecture: Dr Rabya Kadhim

Table Of Contents

Table Of Contents	2
Investigating Mobile Network Capacity and Interference Management	3
Network Slicing	3
Small Cells and Dense HetNet (Heterogeneous Networks)	3
Wireless Backhaul using mmWave and Satellite	4
High capacity Fiber Backhaul	4
Edge computing	5
Content Caching (Content Delivery Networks)	5
Comparing TDMA and WCDMA Systems	6
TDMA	6
WCDMA	6
Comparison	6
Frequency Reuse in TDMA	7
Frequency Reuse in WCDMA	7
Evaluation and Filtering of a Telephone Mobile Power Ringing System (TMPRS)	8
Biomedical Signal Processing Using Moving Average Filtering	12
Generating and Filtering a Complex Signal Using MATLAB	13
Signal Analysis and Filtering in a Communication System	14
Low-pass Filter	14
Low-Pass Filter Output	15
Hi-Pass Filter	16
High-pass Filter Output	18
References	19

Investigating Mobile Network Capacity and Interference Management

Network Slicing

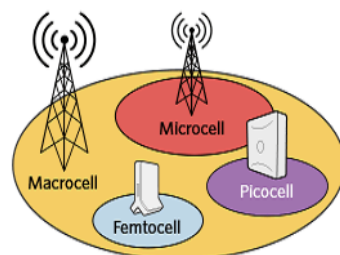
Network slicing is a way of splitting a network into multiple virtual networks made for specific use cases, Used mostly in 5G networks because operators can allocate different parts of the network to specific types of traffic such as video calling, fast responders and special events. Each of these slices can be optimized to suit the specific use case. This enhances the overall capacity of the network maximizing the use of available resources.



(Network slicing – enabling differentiated connectivity - ericsson 2019)

Small Cells and Dense HetNet (Heterogeneous Networks)

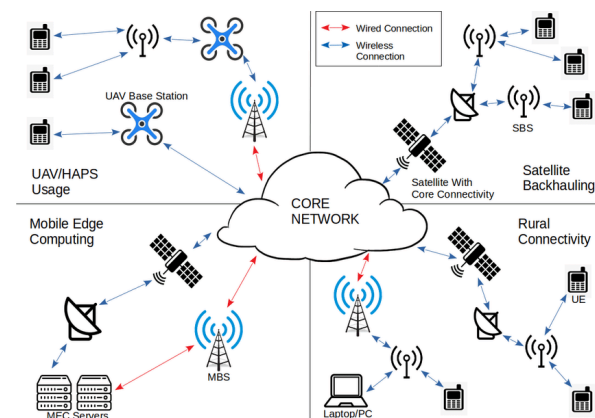
To increase Network capacity, mobile operators deploy low power short range access points designed to offload traffic from macro cells; these cells are typically installed in high traffic areas like in city centres, stadiums and in airports. The combination of small cells and macro cells forms a heterogeneous network also known as a HetNet. HetNets boost coverage and capacity by using a mix of different types of cells as well as an unlicensed spectrum.



(Gbenga-Ilori, Energy efficiency techniques in ultra-dense wireless heterogeneous networks: An overview and outlook 2020)

Wireless Backhaul using mmWave and Satellite

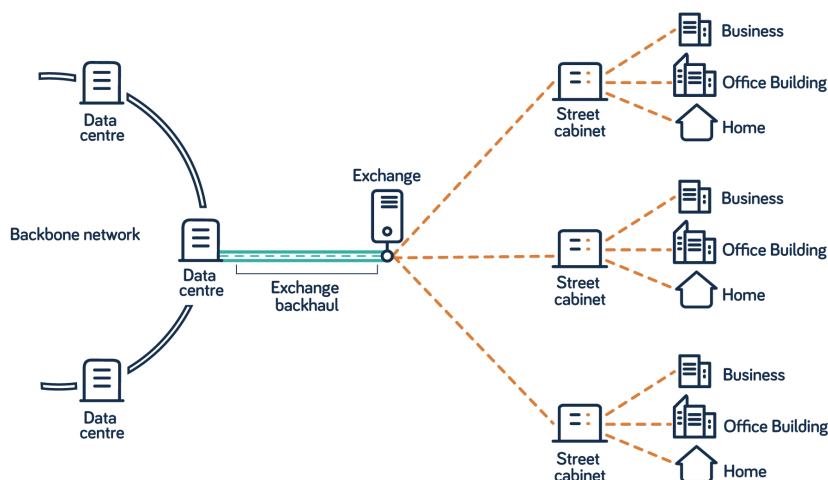
Sub-6 gNBs or LTE eNBs can be wirelessly connected to your current core network system via 5G mmWave wireless backhaul, also known as wireless transport. The lack of cable infrastructure in contrast to conventional fiber-based solutions enables quick and affordable deployment. By extending or enabling greenfield deployments, 5G mmWave backhaul is the ideal method to significantly boost the capabilities of current LTE/5G Sub-6 GHz infrastructure.



(Microamp. The most powerful 5G mmwave backhaul 2024)

High capacity Fiber Backhaul

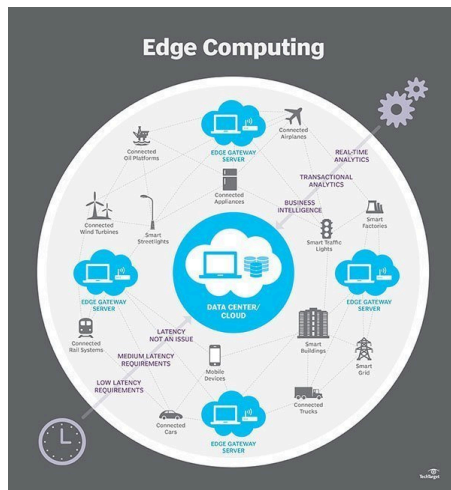
Fiber is able to transfer data as light allowing for Gigabit to terabit bandwidth which is a lot more bandwidth than you would get with copper or wireless data links, this large bandwidth is crucial for handling the large amounts of data used by 4G/5G users. Fiber also allows for a latency as low as 1 millisecond which is essential for autonomous industrial use cases. Fiber backhaul will enable mobile networks to keep up with the growing demand in data usage and demand for high speed connectivity.



(Neos Networks. What is backhaul in networking? 2024)

Edge computing

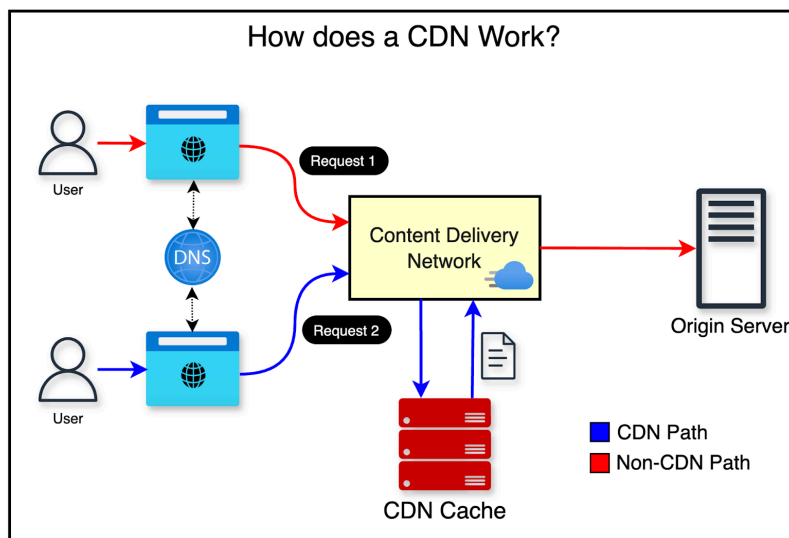
Multi- access edge computing is a way of bringing the data processing closer to the end user, this reduces the need to send data to central cloud servers. By processing data locally it reduces traffic load on backhaul links. Because the data happens closer to the user it helps to reduce latency and offload the traffic from the core network . Edge computing has a positive effect on security cameras and sensors as they process data locally on a 5g network.



(Bigelow. What is edge computing? everything you need to know 2021)

Content Caching (Content Delivery Networks)

Content delivery networks also known as CDN's work by storing frequently accessed content more local to the user which reduced the need for repeated data transfer Content caching helps enhance network capacity by storing popular web pages videos or frequently used software updates in local caches that helps reduce traffic and minimises the load on the core network. Because data is cached locally users can experience faster loading times this is due to the data being fetched from the local caches rather than distant data centres



(Caching static and dynamic content | how does it work? | cloudflare 2022)

Comparing TDMA and WCDMA Systems

TDMA

Time Division Multiple Access, also known as TDMA, is a core technology used in the general system for mobile networks (GSM) that allows networks to efficiently allocate bandwidth and users access to the frequency channel. It also allows for users to use the same frequency as it divides the frequency into eight different time slots.

These time slots are repeated in a cycle every 4.6ms, fast enough for the user not to realise it's even happened, allowing for clear phone calls and smooth data transition.

The advantages of TDMA are listed below

- 8 Users can use the same frequency at once thanks to the allocated time slots
- Time based separation reduces the risk of overlapping signals
- Time slot structure aids in call transfers

However there are some limitations of TDMA like for example each user only has access to one eighth of the channels capacity and the fact that it is not optimal for big bursts of data that happens on 4g and 5g networks.

(Perez. Time Division Multiple Access 1998)

WCDMA

WCDMA also known as wideband code division multiple access is a core radio access used in UTRAN also known as universal mobile telecommunication systems in 3g mobile networks it is designed to support higher rates of data, multimedia services and greater user capacity, WCDMA is completely different to TDMA in how it allows multiple users to share the same frequency as WCDMA uses codes to separate users allowing for instant transmission over the same frequency band.

WCDMA uses direct-sequence spread spectrums. A user's data signal is multiplied by a high rate pseudo-random code sending it across a wider bandwidth which is usually 5MHz per user. Allowing for users to transmit on the same frequency at the same time separated by unique codes.

The advantages of WCDMA are listed below:

- Has a high bandwidth allowing for higher data output per user
- Allows multi user access on the same frequency separated by a unique code
- It has more flexible data that's supported voice video and mobile broadband

Comparison

There are a few differences between TDMA and WCDMA one of these differences is the way they allow other users to use the same frequency TDMA allows users to use the same frequency by giving each user a different time slot the way WCDMA does it is by giving users an unique code allowing for a better use of available spectrum. WCDMA also allows for a lower latency which is better for all services but TDMA's latency is a lot higher due to the rigid slot timing.

(Olamide. The comparison of Time Division Multiple Access (TDMA) ... 2014)

Frequency Reuse and Interference Management in TDMA and WCDMA

Frequency Reuse in TDMA

The way that frequency reuse works in TDMA is that the coverage area is divided into hexagonal cells each has its own set of frequencies; these frequencies are reused systematically across distant cells.

In order to prevent interference when doing this methods like frequency hopping or power control can be used. Frequency hopping reduces interference and improves the quality of a link by switching frequencies regularly and power control works by adjusting the transmission power of the mobile and base station in order to minimize interference to neighboring cells.

Frequency Reuse in WCDMA

Because WCDMA uses individual codes to separate users the frequency reuse factor actually equals to one, this is because all the cells use the same frequency band made possible by the individual codes and the fact that all users transmit on the same 5 MHz channel.

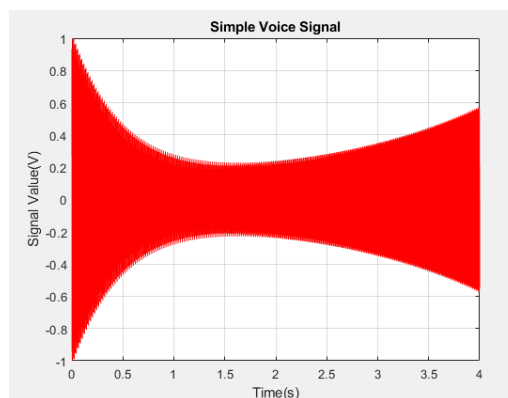
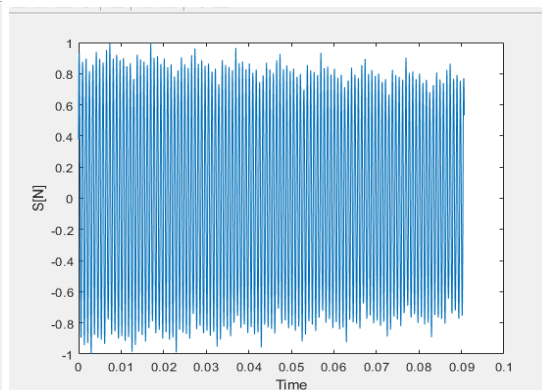
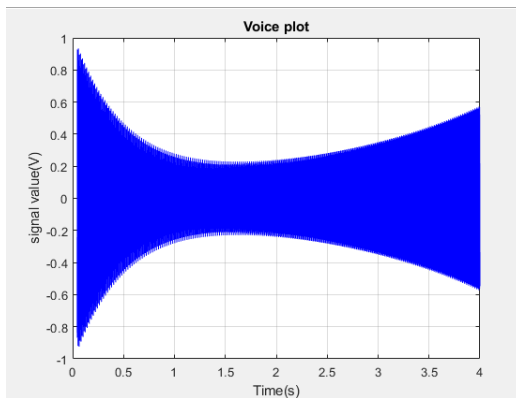
Interference is controlled by using a RAKE receiver which combines multipath signals constructively; this then helps improve signal quality and recover signal paths that would cause interference.

(C.Y. Lee, Frequency Reuse 2002)

Evaluation and Filtering of a Telephone Mobile Power Ringing System (TMPRS)

```
newfilename='josh_wave_output.wav';  
d=4.0;  
  
[s,fs]=audioread('wav_Josh.wav');  
fs  
t = (0:length(s)-1)/fs;  
  
figure(1);  
w=length(t);  
plot(t(1000:w-1),s(1000:w-1),'b');  
grid  
xlabel('Time(s)')  
ylabel('signal value(V)')  
title('Voice plot')  
  
figure(2);  
n = 2000;  
plot(t(1:n),s(1:n));  
xlabel('Time')  
ylabel('S[N]')  
  
figure(3);  
plot(t,s,'r');  
grid  
xlabel('Time(s)')  
ylabel('Signal Value(V)')  
title('Simple Voice Signal')
```

The first step to purify the sound file that was given is to set a new file name so the code can save the new sound and tell the code this was done by using the line of code 'newfilename=' to read the sound file that i was given by using the line of code '[s,fs]=audioread'
After that 'fs' is used to create a display then plot the first three figures that are shown below
Figure one shows the voice plot figure two shows part of a voice plot and figure three shows a simple vice single



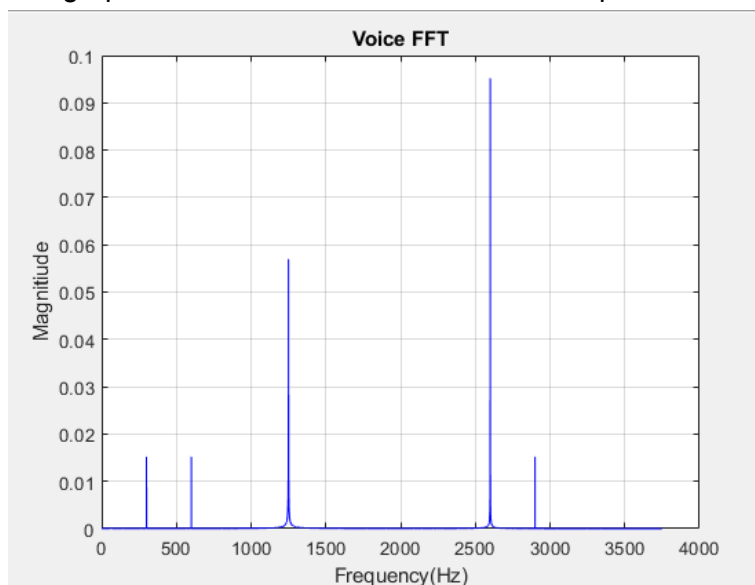
```

33     sound(s,fs);
34     pause(d + 0.5);
35
36     p=fft(s);
37     L=length(s);
38     p2=abs(p/L);
39     p1=p2(1:L/2+1);
40     k=fs*(0:(L/2))/L;
41
42     figure (4)
43     plot(k,p1)
44     plot(k(1:15000),p1(1:15000),'b')
45     grid
46     xlabel('Frequency(Hz)')
47     ylabel('Magnitiude')
48     title('Voice FFT')

```

The next step is to process f1-5 and identify them. In order to do this the line of code 'p=fft(s)' will be necessary. What that does is tell matlab to perform a fourier transform and changes the signal from the time domain into the frequency domain. The line of code 'p2=abs(p/L)' computes the magnitude of the FFT and normalizes it by the length of the signal it also gets rid of the imaginary part giving the signal actual magnitude and the line of code 'p1 = p2(1:L/2+1)' only keeps the first half of the FFT result.

The graph below shows this once it has been plotted.



```

LCoff1 = 250/(fs/2);
HCoff1= 1250/(fs/2);
ORD1=500;
filtcoeff1 = fir1(ORD1,[LCoff1, HCoff1],'stop');
voic1=filter(filtcoeff1,1,s);
sound(voic1,fs);
pause(d+0.5);

LCoff2 = 500/(fs/2);
HCoff2 = 2800/(fs/2);
ORD2 = 500;
filtcoeff2=fir1(ORD2,[LCoff2,HCoff2],'stop');
voic2=filter(filtcoeff2,1,voic1);
sound (voic2,fs);

```

The third step is to use a bandstop filter which filters the voice signal twice and plays the result at each stage. The lines of code 'LCoff1' and 'HCoff1' are the normal cutoff frequency for the filter in this case the cut offs are set at 250Hz and 1250Hz. The line of code 'ORD1=' is what orders the filter the higher the number next to it the sharper the cutoff however that requires more processing time.

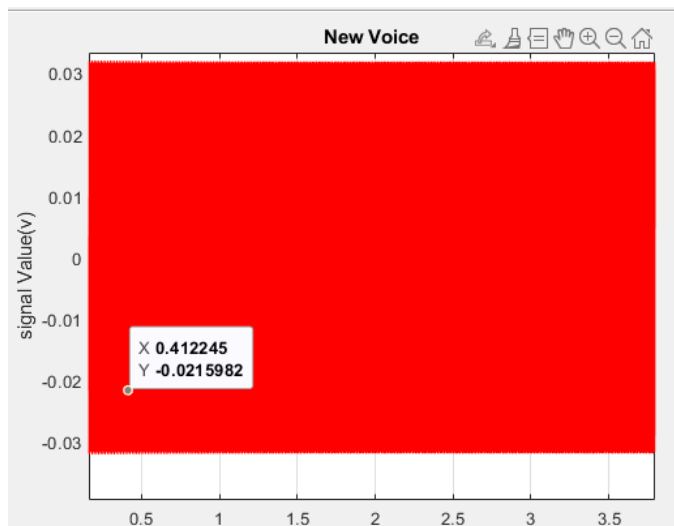
The line of code 'filtcoeff1 = fir1(ORD1,[LCoff1, HCoff1], 'stop')' is what designs the band stop filter then removes the frequencies stated above the two lines of code below that on the image above is what applies the filter then plays the new filtered voice, this step is repeated but removing different frequencies from the first one

```

65 figure (5)
66 plot(t(1000:w-1),voic2 (1000:w-1),'r')
67 grid
68 xlabel('time(s)')
69 ylabel ('signal Value(v)')
70 title ('New Voice')
71
72 pause(d+ 0.5);
73

```

The fourth and final step is to plot the new sound. Figure 5 plots a portion of the filtered voic2 signal which will show the new filtered waveform over time; this is shown in the image below.

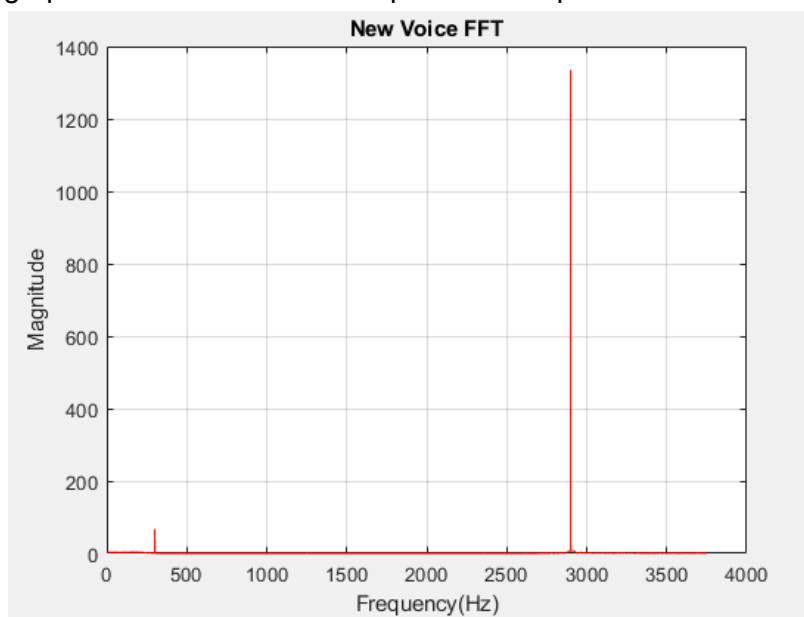


```

74     figure(6)
75     L1=length(voic2);
76     mag1=abs(fft(voic2));
77     kk=fs*((0:(L1-1))/L1);
78     plot (kk(1:15000),mag1(1:15000),'r')
79     grid
80     xlabel('Frequency(Hz)')
81     ylabel('Magnitude')
82     title('New Voice FFT')
83
84     audiowrite(newfilename,voic2,fs);

```

What figure six dose is computes the fft of the signal of voic2. Using the line of code 'mag1=abs' to obtain the magnitude spectrum. Then only the first 15000Hz are plotted. The graph below shows which frequencies are present once both of the filters have been applied.



Biomedical Signal Processing Using Moving Average Filtering

```
1  clc;
2  r=80;
3  n=0:r-1;
4  s=3*n.*(0.9.^n);
5  noi=1.5*(rand(r,1)-0.5);
6  cor=s+noi';
7  subplot(2,2,1);
8  plot (n,s);
9  title('Uncorrupt')
10
11  subplot(2,2,2);
12  plot (n,noi);
13  title ('Noise')
14
15  subplot(2,2,3);
16  plot(n,s+noi');
17  title('corrupt');
18
19  M = 20;
20  num = ones(1,M);
21  y=filter(num,1,cor)/M;
22  subplot(2,2,4);
23  plot(n,y);
24  title('filter')
```

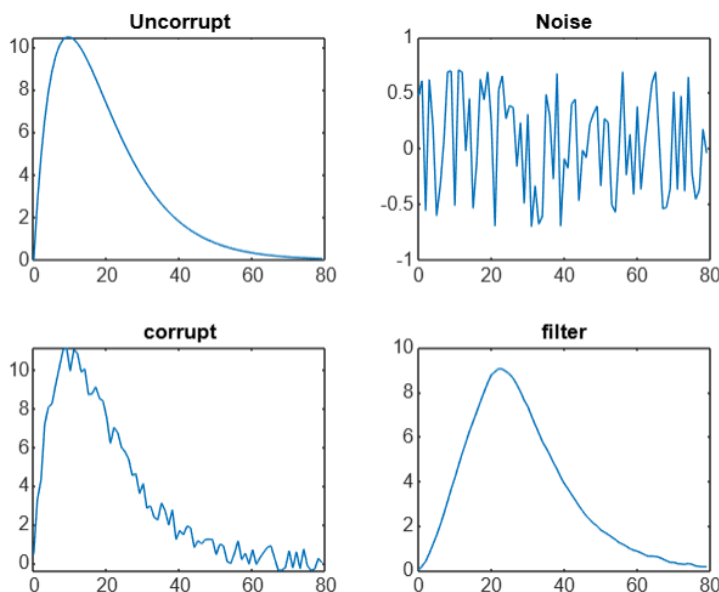
The first step is to add the original signal to the code to do this you need to add the line of code 's=3*n.*(0.9.^n)' the signal is increasing laterally due to the 3*n in the code, next the noise is added by using the line of code 'noi=1.5*(rand(r,1)-0.5)', the noi is a vector that uses a random number in this case 1 to -0.5 to simulate static disruption in the signal .

The next step is to then combine the original signal and then add the noise. This is done by using the line of code 'cor=s+noi'.

The third step is to plot the uncorrupted, noise and the corrupted signals all the plots are shown in the image below

After this, the moving filter has to be added, 'M=' is an average over a number of samples, in this case it's 20. 'num =' defines the shape of the filter, in this case it's a rectangular filter.

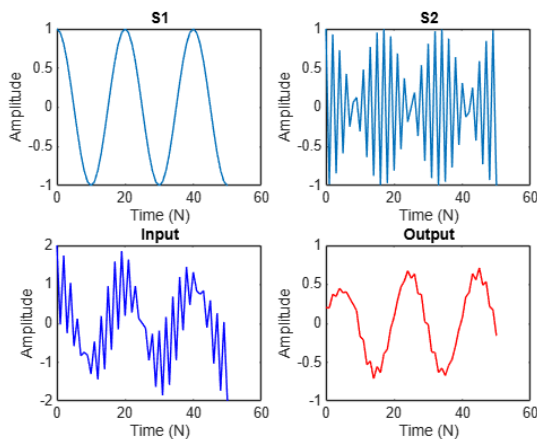
The line of code 'y=filter(num,1,cor)/M' is what applies the FIR filter to the corrupted signal



Generating and Filtering a Complex Signal Using MATLAB

```
2
3     n=0:50;
4     s1=cos(2*pi*0.05*n);
5     s2=cos(2*pi*0.47*n);
6     x=s1+s1;
7
8     M=input ('length of Filter');
9     y=filter(num,1,x);
10    num=ones(1,M);
11    y=filter(num,1,x)/M;
12
13    subplot(2,2,1);
14    plot(n,s1);
15    xlabel('Time (N)');
16    ylabel('Amplitude');
17    title ('S1');
18
19    subplot(2,2,2);
20    plot(n,s2);
21    xlabel('Time (N)');
22    ylabel('Amplitude');
23    title ('S2');
24
25
26    subplot(2,2,3);
27    plot(n,x,'b');
28    xlabel('Time (N)');
29    ylabel('Amplitude');
30    title ('Input');
31
32
33    subplot(2,2,4);
34    plot(n,y,'r');
35    xlabel('Time (N)');
36    ylabel('Amplitude');
37    title ('Output');
```

The way the moving filter works is by adding 's1' which is the low frequency cosine wave and 's2' the high frequency cosine wave creating a signal containing both components what the moving average filter dose is reduce the high frequency parts by smoothing the signal 'm=input' allows you to choose the strength of the filtering the higher the number the smother it gets, in the graph below 10 was used as the signal strength

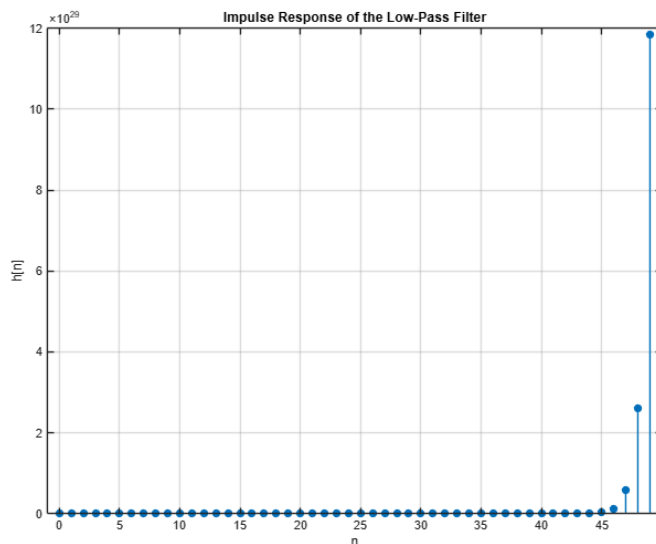


Signal Analysis and Filtering in a Communication System

Low-pass Filter

```
1  N = 50;
2
3  x = zeros(1, N);
4  x(1) = 1;
5
6  y = zeros(1, N);
7
8  b = [0.013, -0.032, 0.044, -0.033, 0.013];
9  a = [1, -3.6, -5.0, 3.2, -0.82];
10
11 h = filter(b, a, x);
12
13 stem(0:N-1, h, 'filled')
14 xlabel('n')
15 ylabel('h[n]')
16 title('Impulse Response of the Low-Pass Filter')
17 grid on
18
```

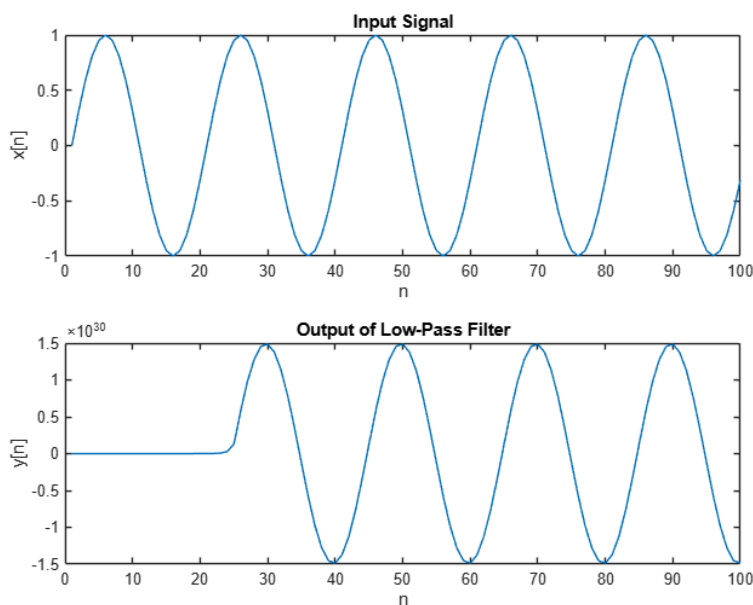
The first step is to set the number of samples to compute. To do this you need to input 'N=' then the number of samples you want to compute for this example 50 has been used. The next line of code is what creates the input signal. The line of code 'y = zeros(1, N)' is what initializes the output signal, the two lines of code underneath 'b=' and 'a=' corresponds do the input and output terms 'b=' corresponds to the $x[n]$ term and 'a=' corresponds to the $y[n]$ term then by using 'h = filter()' matlab simulates the filters output as shown below.



Low-Pass Filter Output

```
1  clc
2  N = 50;
3  N_input=100;
4  x_input=sin(2*pi*0.05*(0:N_input-1));
5  y_output=conv(x_input, h, 'same');
6  x = zeros(1, N);
7  x(1) = 1;
8  y = zeros(1, N);
9
10 b = [0.013, -0.032, 0.044, -0.033, 0.013];
11 a = [1, -3.6, -5.0, 3.2, -0.82];
12 h = filter(b, a, x);
13
14 stem(0:N-1, h, 'filled')
15 xlabel('n')
16 ylabel('h[n]')
17 title('Impulse Response of the Low-Pass Filter')
18 grid on
19
20 subplot(2,1,1);
21 plot(x_input);
22 title('Input Signal');
23 xlabel('n');
24 ylabel('x[n]');
25
26 subplot(2,1,2);
27 plot(y_output);
28 title('Output of Low-Pass Filter');
29 xlabel('n');
30 ylabel('y[n]');
```

In order to get the output of the low-pass filter a few new lines of code were added to the previous code, at first an example signal is defined by using the lines of code 'N_input' and 'x_input' after that the line of code 'y_output=conv()' is input into matlab in order to calculate the output. All that's left to do then is plot a graph for the input and output which are shown in the image below

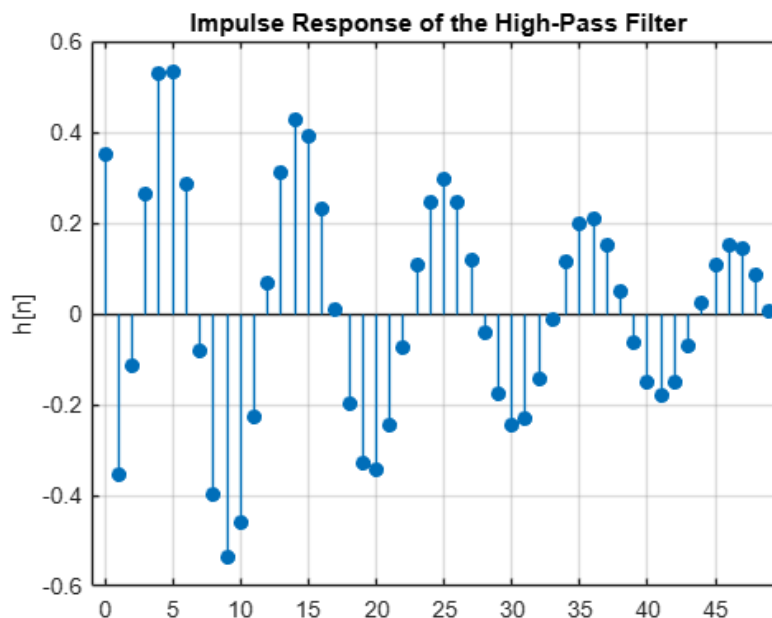


Hi-Pass Filter

```
N = 50;  
  
x = zeros(1, N);  
x(1) = 1;  
y = zeros(1, N);  
  
b = [0.35, -1.3, 2.0, -1.3, 0.35];  
a = [1, -2.7, 3.3, -2.0, 0.57];  
h = filter(b, a, x);  
  
stem(0:N-1, h, 'filled');  
xlabel('n');  
ylabel('h[n]');  
title('Impulse Response of the High-Pass Filter');  
grid on;
```

The first step is to set the number of samples to compute. To do this you need to input 'N=' then the number of samples you want to compute for this example 50 has been used.

The next line of code is what creates the input signal. The line of code 'y = zeros(1, N)' is what initializes the output signal, the two lines of code underneath 'b=' and 'a=' corresponds do the input and output terms 'b=' corresponds to the $(x[n])$ term and 'a=' corresponds to the $(y[n])$ term then by using 'h = filter()' matlab simulates the filters output as shown below.



High-pass Filter Output

```
N = 50;

x = zeros(1, N);
x(1) = 1;
y = zeros(1, N);
N_input = 100;
x_input = sin(2*pi*0.1*(0:N_input-1));
y_output = conv(x_input, h, 'same');

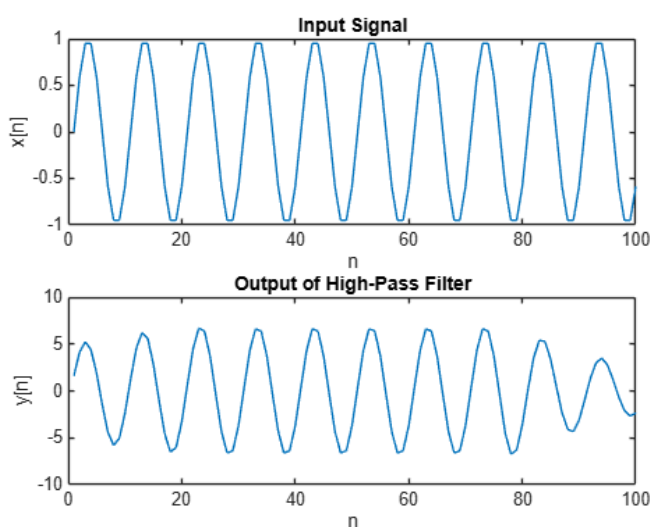
b = [0.35, -1.3, 2.0, -1.3, 0.35];
a = [1, -2.7, 3.3, -2.0, 0.57];
h = filter(b, a, x);

stem(0:N-1, h, 'filled');
xlabel('n');
ylabel('h[n]');
title('Impulse Response of the High-Pass Filter');
grid on;

subplot(2,1,1);
plot(x_input);
title('Input Signal');
xlabel('n');
ylabel('x[n]');

subplot(2,1,2);
plot(y_output);
title('Output of High-Pass Filter');
xlabel('n');
ylabel('y[n]');
```

In order to get the output of the high-pass filter a few new lines of code were added to the previous code, at first an example signal is defined by using the lines of code 'N_input' and 'x_input' after that the line of code 'y_output=conv()' is input into matlab in order to calculate the output. All that's left to do then is plot a graph for the input and output which are shown in the image below



References

- Bigelow, S.J. (2021) *What is edge computing? everything you need to know*, Search Data Center. Available at: <https://www.techtarget.com/searchdatacenter/definition/edge-computing> (Accessed: 24 April 2025).
- C.Y. Lee, W. (2002) *Frequency Reuse, Frequency Reuse - an overview* | ScienceDirect Topics. Available at: <https://www.sciencedirect.com/topics/engineering/frequency-reuse#:~:text=FREQUENCY%20REUSE%20AND%20DESIGN%20ASPECT,12>. (Accessed: 24 April 2025).
- Cloudflare (2022) *Caching static and dynamic content | how does it work?* | cloudflare. Available at: <https://www.cloudflare.com/en-gb/learning/cdn/caching-static-and-dynamic-content/> (Accessed: 20 March 2025).
- Gbenga-Ilori, A. (2020) *Energy efficiency techniques in ultra-dense wireless heterogeneous networks: An overview and outlook*, Engineering Science and Technology, an International Journal. Available at: <https://www.sciencedirect.com/science/article/pii/S2215098619328745#:~:text=Abstract,EE%20in%20Ultra%20Ddense%20HetNets>. (Accessed: 24 March 2025).
- Group, E. (2019) *Network slicing – enabling differentiated connectivity - ericsson, Network slicing*. Available at: <https://www.ericsson.com/en/network-slicing> (Accessed: 24 February 2025).
- Microamp (2024) *The most powerful 5G mmwave backhaul, 5G mmWave Backhaul*. Available at: <https://microamp-solutions.com/5g-mmwave-backhaul#:~:text=5G%20mmWave%20wireless%20backhaul%2C%20or,system%20in%20a%20wireless%20manner>. (Accessed: 10 March 2025).
- Neos Networks (2024) *What is backhaul in networking?* Available at: <https://neosnetworks.com/resources/blog/what-is-backhaul-in-networking/#:~:text=By%20deploying%20high%20capacity%20fibre,than%20building%20your%20own%20infrastructure>. (Accessed: 24 March 2025).
- Olamide, O. (2014) *The comparison of Time Division Multiple Access (TDMA) ..., The comparison of time division multiple access (TDMA) (global system for mobile communication, GSM) and wideband-code division multiple access (W-CDMA) (third generation, 3G) system based on their modulation techniques*. Available at: <https://www.semanticscholar.org/paper/The-comparison-of-time-division-multiple-access-for-Olamide-Abiodun/6c9f83a19b1fd9a580d235ee07e89cac25fbafd0> (Accessed: 24 March 2025).
- Perez, R. (1998) *Time Division Multiple Access, Time Division Multiple Access - an overview* | ScienceDirect Topics. Available at: <https://www.sciencedirect.com/topics/engineering/time-division-multiple-access#:~:text=Time%20division%20multiple%20access%20is,in%20its%20allocated%20time%20slot>. (Accessed: 24 March 2025).

