Eray Gündoğdu 22101962

EEE431 2024/25 Spring MATLAB Assignment 2

Question 1

a)

$$R_W(k) = E[W_n W_{n+k}] = E[W_n(0.9W_{n+k-1} + V_{n+k})]$$

$$E[W_n V_{n+k}] = 0$$

$$R_W(k) = E[W_n 0.9W_{n+k-1})] = 0.9R_W(k-1)$$

$$R_W(k) = 0.9R_W(k-1) = 0.9^2 R_W(k-2) = \cdots = 0.9^k R_W(0)$$

$$R_W(k) = R_W(-k)$$

Hence it can be concluded that,

$$R_W(k) = 0.9^{|k|} R_W(0)$$

The autocorrelation is oly dependent on 'time' difference (k) and the mean can be simply observed as 0 by taking expectation of the both sides in the first equation of assignemnt (Auto Regressive equation). Due to constant mean and autocorrelation only depending on time difference the random process is WSS.

b) The random seed is fixed to 22101962 and W_n is generated by following the AR equation. V_n is a gauissan i.i.d. process with zero mean and 10^6 samples are generated with randn function and unit variance in MATLAB. Small section of resulting W_n can be observed in Figure 1.

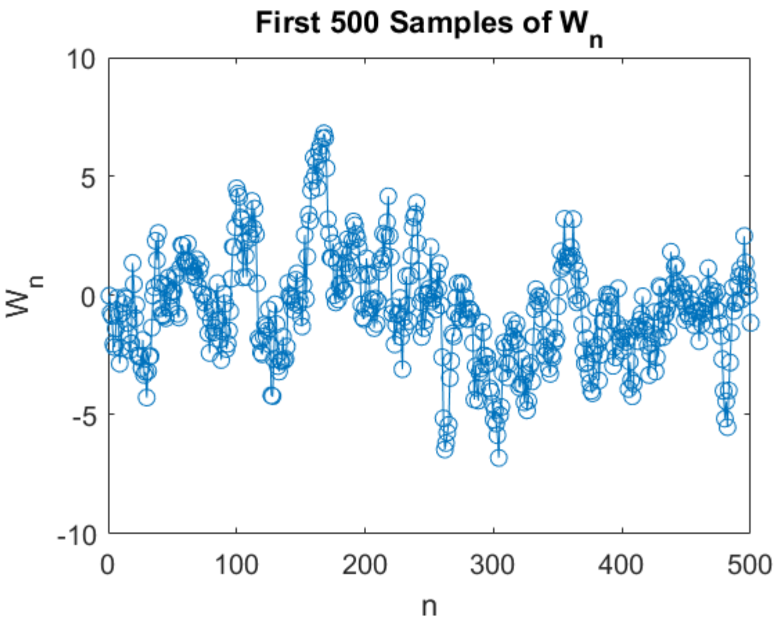

Figure 1 First 500 Samples of W_n

c) The difference equation indicates that Y_n is 20 point average of W_n. This filter is called 'smoothining filter' in DSP and acts as a low pass filter and extracts the trend of the signal. If Figure 2 is observed it is observable that it cancled out the high frequency components of W_n and outputs the trend of the process.

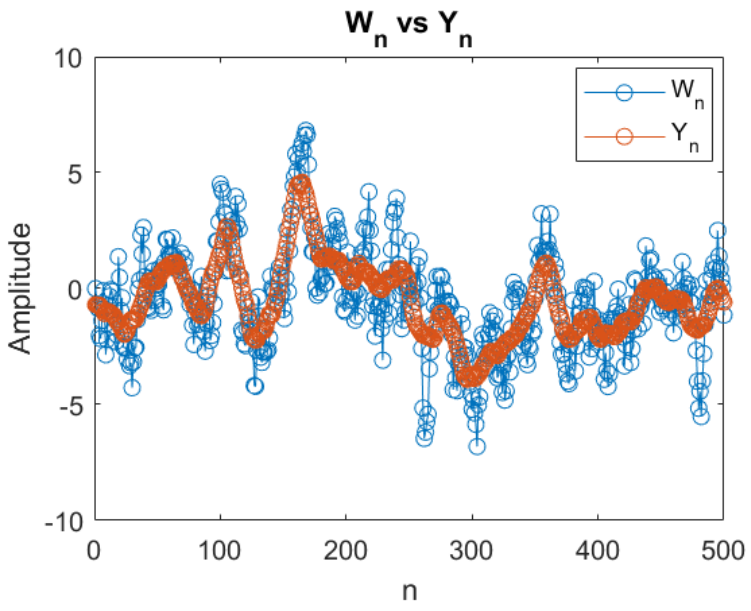

Figure 2 Comparison of W_n and Y_n

d) In this part of the question it is better to start with the analyticsl derivations of impulse respone and magnitude response.

$$h[n] = \frac{1}{20}\sum_{i=0}^{19} \delta[n-i]$$

$$H(e^{j\omega}) = \frac{1}{20}\sum_{i=0}^{19} e^{-j\omega i}$$

$$H(e^{j\omega}) = \frac{1}{20}\frac{1-e^{-j20\omega}}{1-e^{-j\omega}} = \frac{e^{-j10\omega}}{20e^{-j\omega/2}}\frac{\sin(10\omega)}{\sin(\frac{\omega}{2})}$$

$$|H(e^{j\omega})| = \frac{1}{20}\frac{\sin(10\omega)}{\sin(\frac{\omega}{2})}$$

By this way the low pass filter characteristic can also be observed from analytically derived expressions by inspecting frequency near infinity and zero. In this part PSD of W_n and Y_n are estimated with the help of pwelch function in MATLAB and resulting magnitude response, so called estimated magnitude response is found by the following relation.

$$S_Y = |H|^2 S_x$$

$$|H| = \sqrt{\frac{S_Y}{S_X}}$$

The theoratical magnitude response is obtained by taking 512 point FFT of the filter wich is summation of impulses or can be referred as truncated impulse train with length 20. It is very import to mention that frequency axis represents normalized frequency ie 0 to 2π. In Figure 3 and Figure 4 PSD estimate plots of W_n and Y_n can be observed in linear (without taking the log) PSD values and dB respectively.

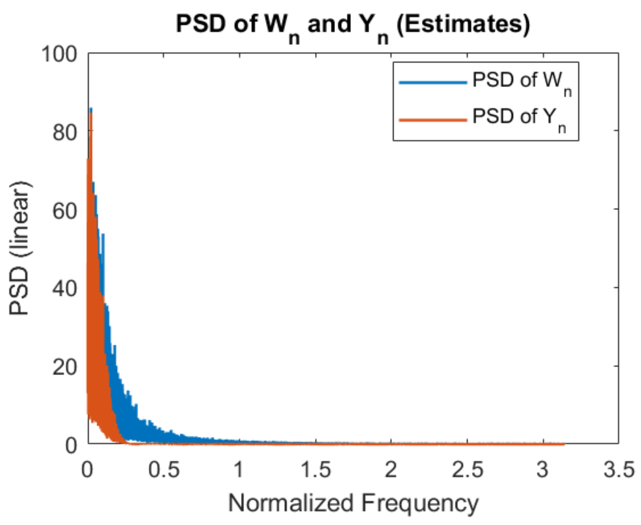

Figure 3 Linear values of estimated PSD of W_n and Y_n

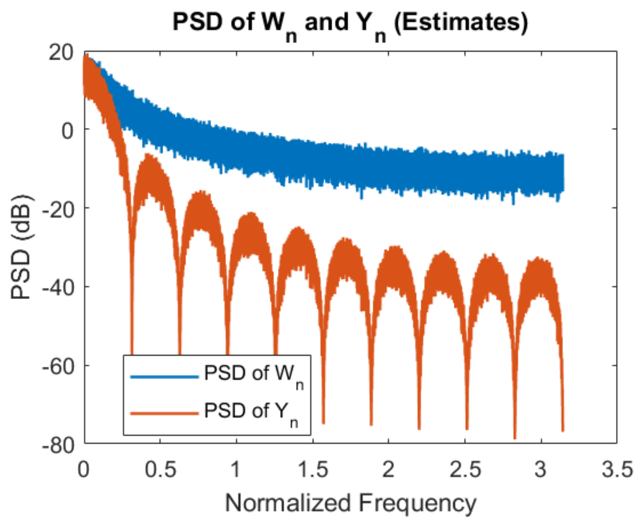

Figure 4 dB values of estimated PSD of W_n and Y_n

These PSD plots are benefitial to understand that the process is low pass filtered since the output power decreases as frequency increases. Also PSD estimates are nonnegative which also an indicator of correct implementation. Moreover to test the accuracy of the PSD estimates the theoratical result of the filter and it's estimate in frequency domain are plotted on the same figure in Figure 5.

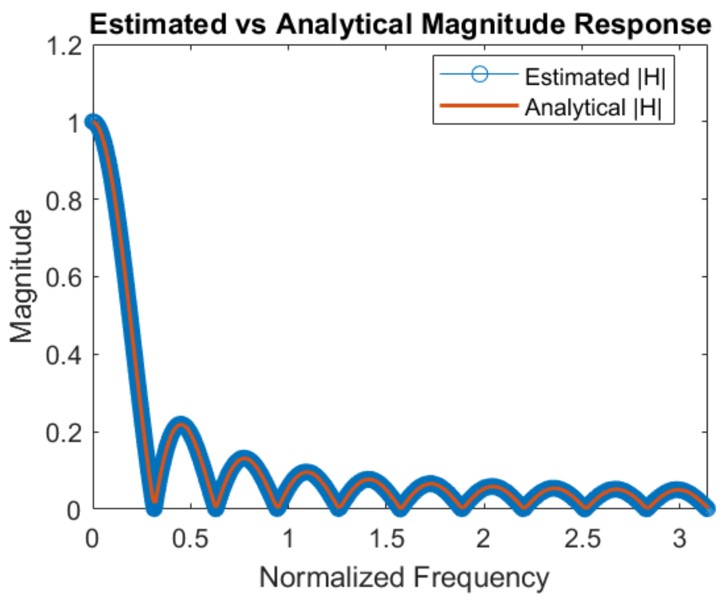

Figure 5 Plots of Estimated and Analytical filter in frequency domain

The estimated magnitude respond closely follows the analytically derived magnitude response which is obtained by 512 point FFT. This also indictes that PSD estimates are accurate since the estimate of the filter is derived from estimates of PSD of W_n and Y_n.

e) In this part awhite Gaussian noise is generated and part c and d is repeated and at last PSD of White gaussian nosie and W_n are plotted together. To start with part c is repeated in which the White noise will we filtered. Filtered White noise output can be observed in Figure 6. The output of filtered White noise is denoted as Z_n and noise is denoted as G_n

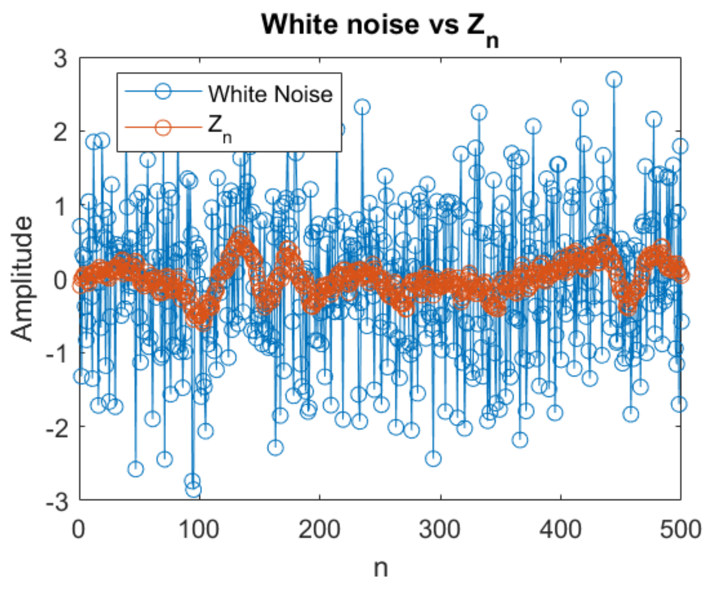

Figure 6 Comparison of G_n and Z_n

After observing the filtered version, it is benefitial to investigate the analytical PSD of White noise:

$$R_G[k] = E[G(n+k)G(n)] = \sigma^2\delta[k] = \delta[k]$$

$$S_G(f) = 1$$

After the analytical inspection observe Figure 7 and Figure 8 which are same procedures as Figure 3 and Figure 4. The PSD values extend up to 1 and the logarithm results in values near zero.

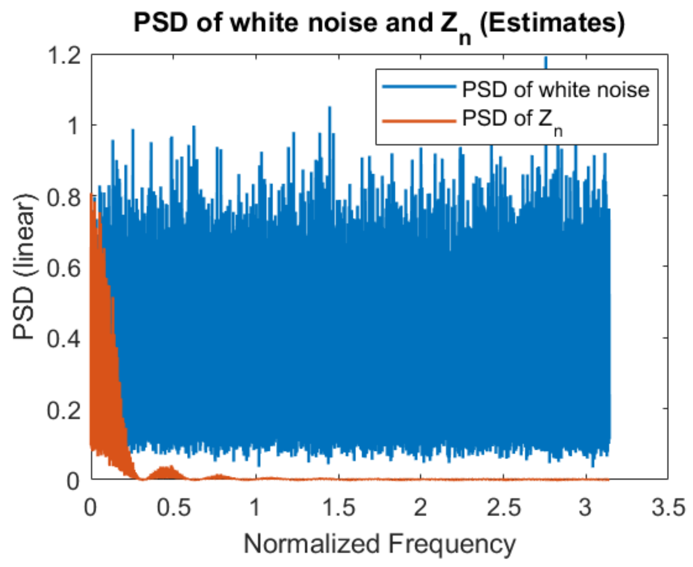

Figure 7 Linear values of estimated PSD of G_n and Z_n

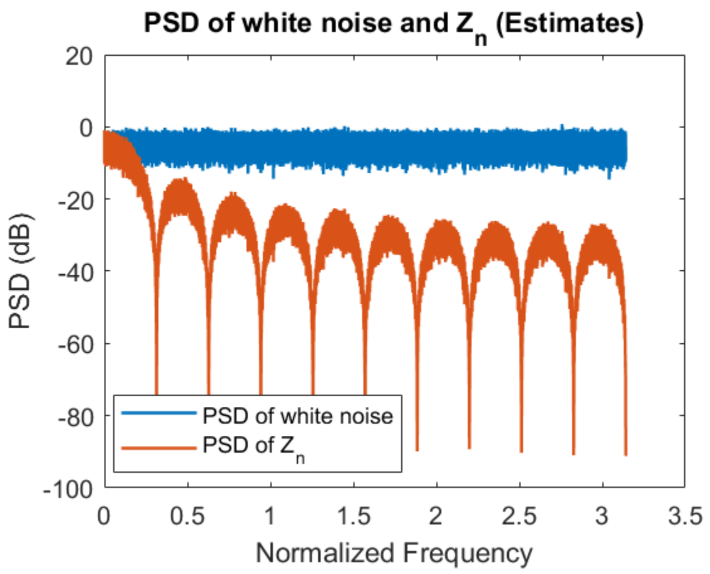

Figure 8 dB values of estimated PSD of G_n and Z_n

Also, again a new estimate of the filter in frequency domain is obtained by taking advantage of the relation between S_G and S_Z (the same operation in part d). The estimate of the filter in frequency domain can be observed in Figure 9 which indicates again the estimates of S_G and S_Z are accurate.

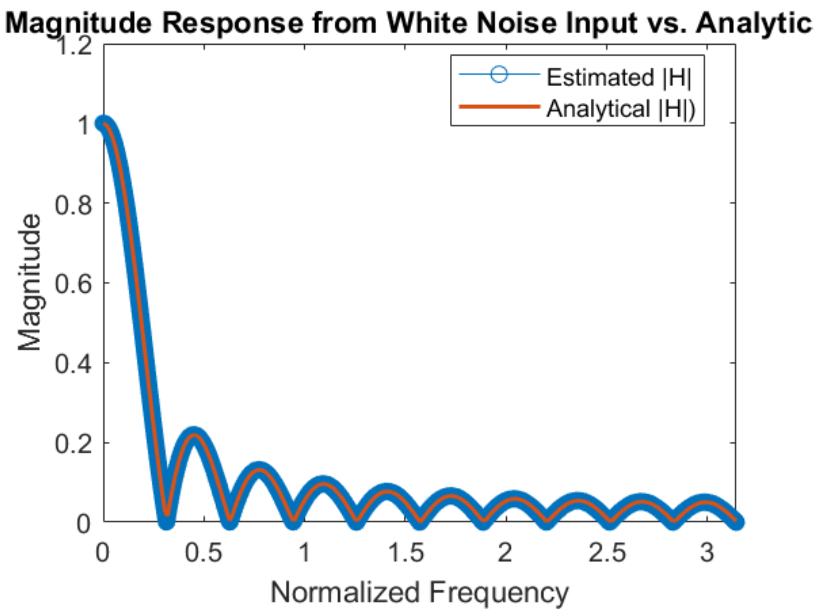

Figure 9 Plots of Estimated and Analytical filter in frequency domain obtained by White gaussian noise input

As the final step it is desired to have PSD plots of White gaussian noise and W_n. Additionally plots of filtered outputs are also included.

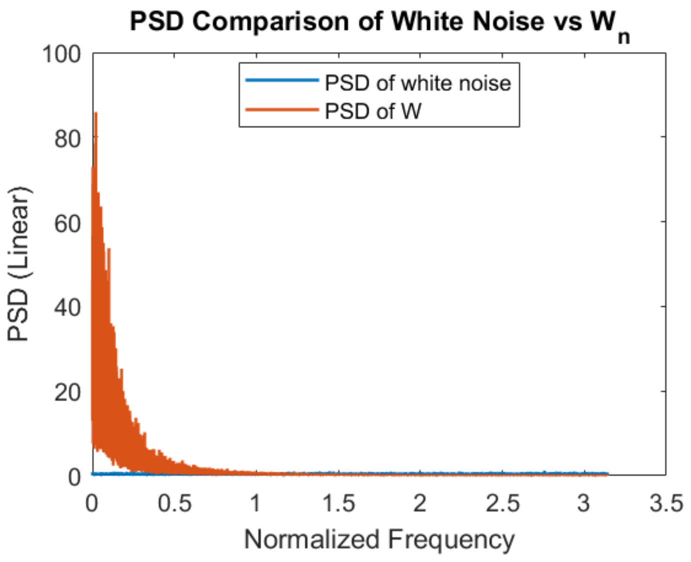

Figure 10 Linear values of estimated PSD of G_n and W_n

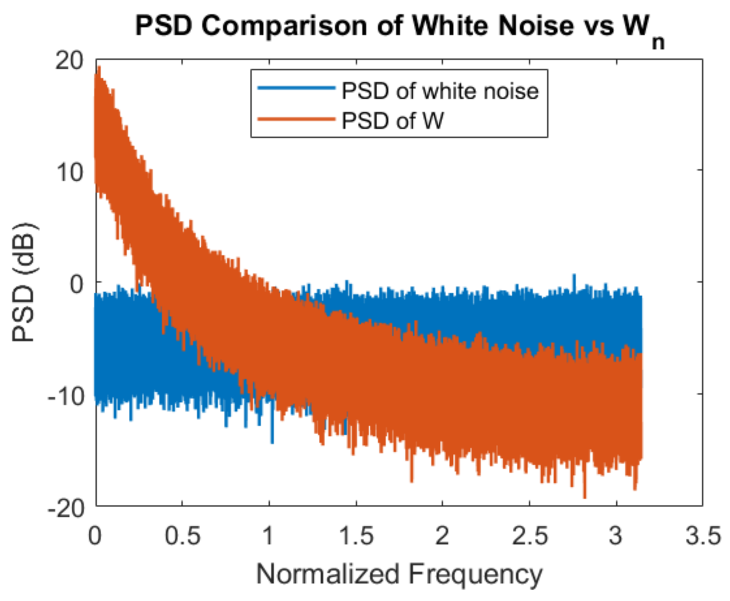

Figure 11 dB values of estimated PSD of G_n and W_n

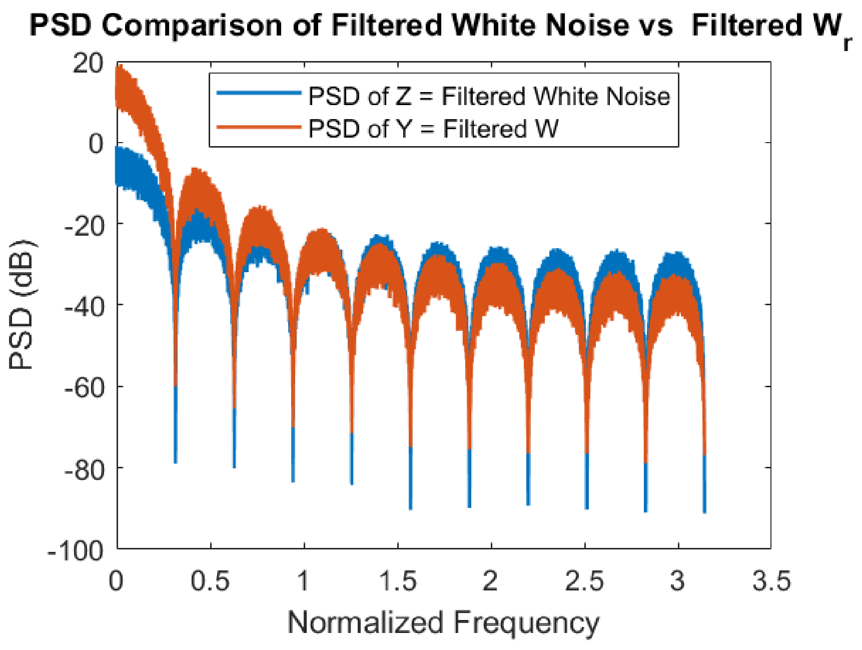

Figure 12 dB values of estimated PSD of Y_n and Z_n

If the PSD of W_n is checked analytically (please check the fourier pair for $0.9^{|k|}$) as frequency increases the -cos term in denominator decreases the overall value as frequency increases which makes the PSD of W_n inversely proportional with frequency. On the other hand, the analytical expression for PSD of White gaussian noise is derived which is constant along the frequency. MATLAB sampling methods ofcourse resulted in some errors compared to analytical expressions however MATLAB experiments highly corralate with the analytical findings.

Question 2

a)In this question a message signal with 100Hz is DSB-SC modulated with a carrier cosine signal which has 5kHz frequency. Initially the message signal is sampled with 100kHz frequency to perfectly define the signal in MATLAB without losing information. Afterwards, the message signal is DSB-SC modulated with the carrier signal which is as decribed previously. The sampled version of the message signal and resulting DSB-SC modulated signal can be observed in Figure 13. Analytic expression for y(t) can be observed as following.

$$y(t) = \sin(2\pi 100t)\cos(2\pi 5000t)$$

$$y(t) = \frac{1}{2}(\sin(2\pi 5100t) + \sin(2\pi 4900t))$$

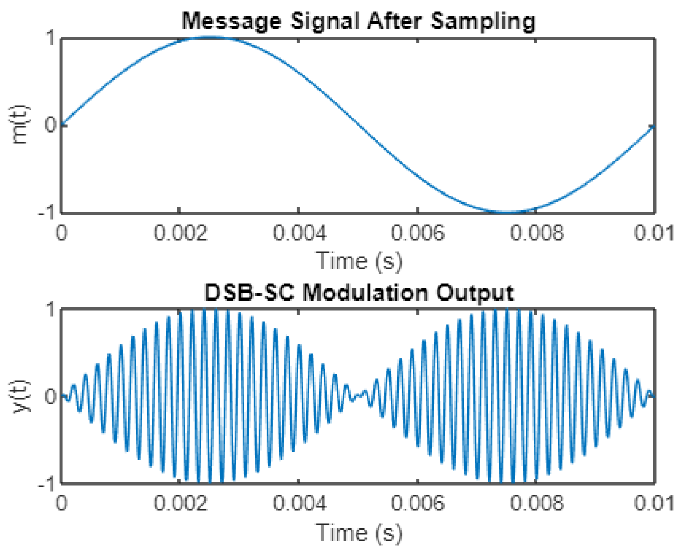

Figure 13 Demosntration of message signal and output of DSB-SC modulation

b) In this part it is desired to obtain m(t) back after demodulation but the local oscillator has frequency offsets values denoted as Δ. The coherent detector design in this section can be described as following. Initially y(t) is is multiplied with tthe local oscillator with offset. Afterwards it is multiplied with 2 to arrange proper magnitude and then low pass filtered to extract the low frequencies (can be clearly observed fro MATLAB codes). As the low pass filter a sixth order butterworth low pass filter is used due to its properties of sharp decay in magnitude response which makes it similar to ideal low pass characteristics in some sense. Before observing the outputs of the coherent detection it is benefitial to derive the coherent detection procedure (x(t)y(t)) analytically.

$$y(t)x(t) = \sin(2\pi 100t)\cos(2\pi f_c t)\cos(2\pi(f_c + \Delta)t)$$

$$y(t)x(t) = \frac{1}{2}\sin(2\pi 100t)(\ \cos(2\pi\Delta t) + \cos(2\pi(2f_c + \Delta)t)$$

After the low pass filtering and multiplying by 2,

$$y(t)x(t) = \sin(2\pi 100t)\cos(2\pi\Delta t)$$

$$\sin(2\pi 100t)\cos(2\pi\Delta t) = \frac{1}{2}(\sin(2\pi(100+\Delta)t) + \sin(2\pi(100-\Delta)t))$$

So according to analytical derivation, when Δ = 0 m(t) can be detected perfectly.The cutoff frequency of the LPF is determined to be 200Hz. If we consider the cutoff frequency, some Δ values will make the output dissappear (such as $\Delta = 1000$) due to LPF in coherent detection. In the following figures different plots are provided for different LPF cutoff frequencies.

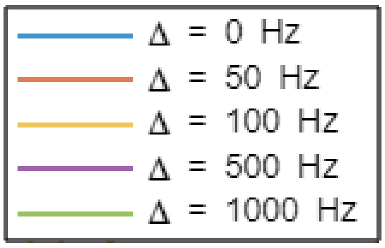

Figure 14 Legend for frequency offset coherent detection output plots

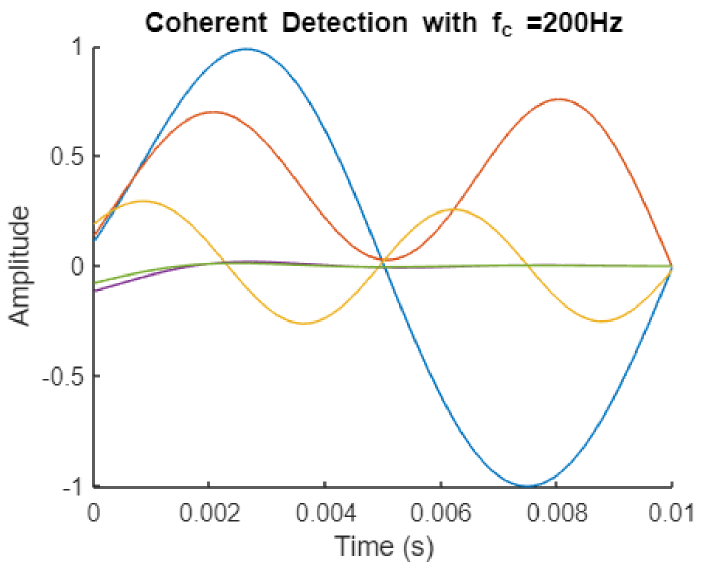

Figure 15 Coherent Detection Outputs with 200Hz cutoff frequency

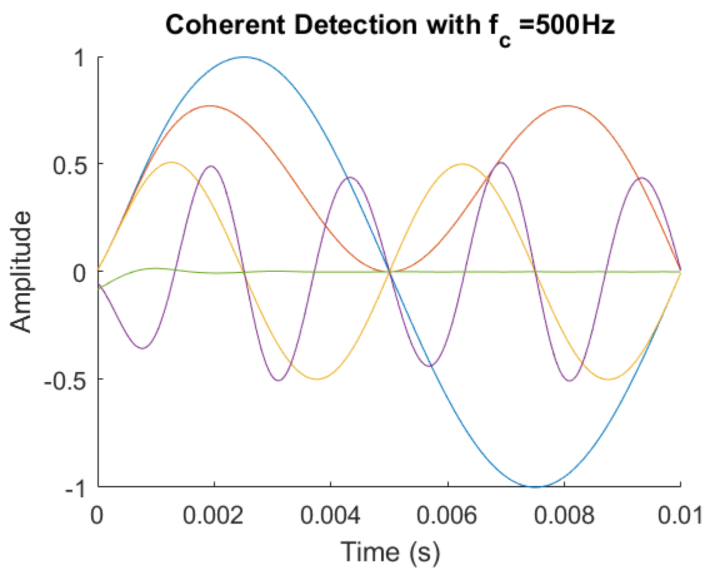

Figure 16 Coherent Detection Outputs with 500Hz cutoff frequency

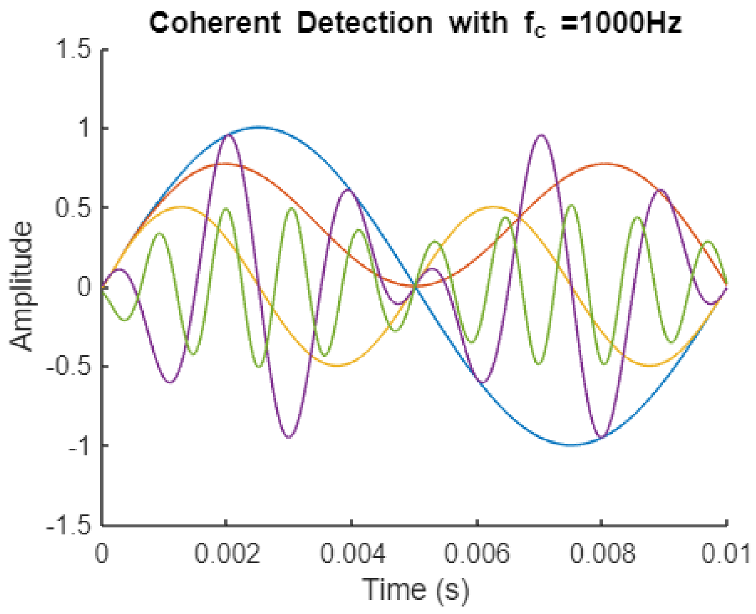

Figure 17 Coherent Detection Outputs with 500Hz cutoff frequency

The coherent detection implementation can be verified by inspecting the Figures. In Figure 15, 0 offset perfectly detects the message signal, if 50Hz offset is inspected from the analytical derivation there should be some of 2 sinusoids 150Hz and 50 Hz which makes the waveform always above zero, if 100Hz offset is inspected a single sinusoid is expected with 200Hz. The offsets 500Hz and 100Hz change the frequency content of the signal such that the content is becomes high with respect to the cutoff of LPF therefore the outputs of 1000Hz and 500Hz offsets are nearly zero. However as the cutoff increases the inclusion of the 500Hz and 1000Hz offsets can also be observed.

c)CFO occurs when the receiver local oscillator is not perfectly match ups with the transmitter's carrier frequency. As a result the reciever demodulates the signal wth a different frequency and some distortions such as frequency and/or amplitude distortions which are already observed in par b. To solve CFO problem, PLL (Phase Locked Loops) can be used. In PPL a refernce signal is synchronized with an oscillator and in such a case the system becomes locked. This model can be implemented for demodulation process to establish synchronization between the local oscillator and carrier frequency phases. Another solution might be implementing software based predictive models to match the carrier frequency and local oscillator.

d) Now phase difference local oscillator is used in coherent detection to obtain original message signal from modulated signal. The analytical expression can be observed as following.

$$\sin(2\pi 100t)\cos(2\pi f_c t)\cos(2\pi f_c t + \phi) = \frac{1}{2}\sin(2\pi 100t)\cos(\phi) + \frac{1}{2}\sin(2\pi 100t)\cos(4\pi f_c t + \phi)$$

After LPF and multiplying by 2 the resulting signal is:

$$\sin(2\pi 100t)\cos(\phi) = \mathrm{m(t)}\cos(\phi)$$

The experimental results for different phase local oscillator coherent detection outputs can be observed in Figure 18.

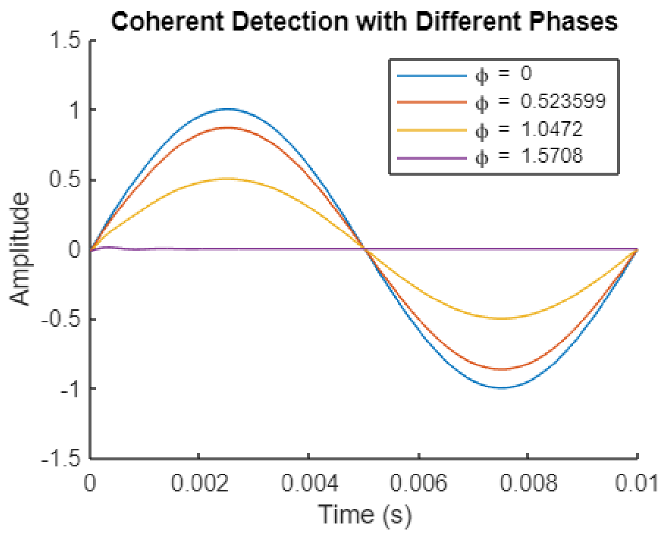

Figure 18 Coherent Detection outputs with different local oscillator phases

The result in Figure 16 also correlates with the analytical findings. The resulting signal in the receiver side is $\mathrm{m(t)}\cos(\phi)$ which means original message is multiplied with a constant $\cos(\phi)$ which means frequency content of the signal is not changed (unlike frequency offset oscillators which change the frequency content) but amplitude of the original message signal is multiplied with $\cos(\phi)$ which is clearly observable in Figure 18.

APPENDIX

Matlab Codes for Question 1)

```
num_sample = 1e6;
alpha = 0.9;
var_V = 1;
rng(22101962);


V = var_V * randn(num_sample,1);
W = zeros(num_sample,1);
for n = 2:num_sample
    W(n) = alpha * W(n-1) + V(n);
end


plot(W(1:500),'o-');
xlabel('n');
ylabel('W_n');
title('First 500 Samples of W_n');


h = ones(1,20)/20;
Y = conv(W,h,'same');


idxPlot = 1:500;
figure;
plot(idxPlot,W(idxPlot),'o-'); hold on;
plot(idxPlot,Y(idxPlot),'o-');
legend('W_n','Y_n','Location','best');
xlabel('n');
ylabel('Amplitude');
title('W_n vs Y_n');


[PSD_W,f_w] = pwelch(W,[],[],[]);
[PSD_Y,f_y] = pwelch(Y,[],[],[]);
H_estimate = sqrt(PSD_Y./PSD_W);




fft_point = 512;
H_fft = fft(h,fft_point);
H_fft = abs(H_fft);




w_theory = linspace(0,2*pi,fft_point);
```

```
figure;


plot(f_w, 10*log10(PSD_W),'LineWidth',1.2);
hold on;
plot(f_y, 10*log10(PSD_Y),'LineWidth',1.2);
xlabel('Normalized Frequency');
ylabel('PSD (dB)');
legend('PSD of W_n','PSD of Y_n','Location','best');
title('PSD of W_n and Y_n (Estimates)');
figure;
plot(f_w, (PSD_W),'LineWidth',1.2);
hold on;
plot(f_y, (PSD_Y),'LineWidth',1.2);
xlabel('Normalized Frequency');
ylabel('PSD (linear)');
legend('PSD of W_n','PSD of Y_n','Location','best');
title('PSD of W_n and Y_n (Estimates)');


figure;
plot(f_w,H_estimate,'o-'); hold on;
plot(w_theory,H_fft,'LineWidth',1.5);
xlabel(' Normalized Frequency');
xlim([0,pi]);
ylabel('Magnitude');
legend('Estimated |H|','Analytical |H|','Location','best');
title('Estimated vs Analytical Magnitude Response');


X_noise = randn(num_sample,1);
Z = conv(X_noise,h,'same');


[PSD_X,fX] = pwelch(X_noise,[],[],[]);
[PSD_Z,fZ] = pwelch(Z,[],[],[]);


figure;
plot(idxPlot,X_noise(idxPlot),'o-'); hold on;
plot(idxPlot,Z(idxPlot),'o-');
legend('White Noise','Z_n','Location','best');
xlabel('n');
ylabel('Amplitude');
title('White noise vs Z_n');
```

```
figure;
plot(f_w, 10*log10(PSD_X),'LineWidth',1.2);
hold on;
plot(f_y, 10*log10(PSD_Z),'LineWidth',1.2);
xlabel('Normalized Frequency');
ylabel('PSD (dB)');
legend('PSD of white noise','PSD of Z_n','Location','best');
title('PSD of white noise and Z_n (Estimates)');
figure;
plot(f_w, (PSD_X),'LineWidth',1.2);
hold on;
plot(f_y, (PSD_Z),'LineWidth',1.2);
xlabel('Normalized Frequency');
ylabel('PSD (linear)');
legend('PSD of white noise','PSD of Z_n','Location','best');
title('PSD of white noise and Z_n (Estimates)');


figure;
plot(fX,PSD_X,'LineWidth',1.2); hold on;
plot(f_w,PSD_W,'LineWidth',1.2);
xlabel('Normalized Frequency');
ylabel('PSD (Linear)');
legend('PSD of white noise','PSD of W ','Location','best');
title('PSD Comparison of White Noise vs W_n');
figure;
plot(fX,10*log10(PSD_X),'LineWidth',1.2); hold on;
plot(f_w,10*log10(PSD_W),'LineWidth',1.2);
xlabel('Normalized Frequency');
ylabel('PSD (dB)');
legend('PSD of white noise','PSD of W ','Location','best');
title('PSD Comparison of White Noise vs W_n');


figure;
plot(fZ,10*log10(PSD_Z),'LineWidth',1.2); hold on;
plot(f_y,10*log10(PSD_Y),'LineWidth',1.2);
xlabel('Normalized Frequency');
ylabel('PSD (dB)');
legend('PSD of Z = Filtered White Noise','PSD of Y = Filtered
W','Location','best');
title('PSD Comparison of Filtered White Noise vs  Filtered W_n');


H_estimate_white = sqrt(PSD_Z./PSD_X);
figure;
plot(fX,H_estimate_white,'o-'); hold on;
plot(w_theory,H_fft,'LineWidth',1.5);
xlabel('Normalized Frequency');
xlim([0,pi]);
ylabel('Magnitude');
legend('Estimated |H|','Analytical |H|)','Location','best');
```

```
title('Magnitude Response from White Noise Input vs. Analytical');
```

Matlab Codes for Question 2)

```


clear; clc; close all;


Fs = 100000;
T = 0.02;
t = (0 : 1/Fs : T)';




mt = sin(200*pi*t);




fc = 5000;




kt = mt .* cos(2*pi*fc*t);




figure;
subplot(2,1,1);
plot(t(1:1000), mt(1:1000));
xlabel('Time (s)'); ylabel('m(t)');
title('Message Signal After Sampling');


subplot(2,1,2);
plot(t(1:1000), kt(1:1000));
xlabel('Time (s)'); ylabel('y(t)');
title('DSB-SC Modulation Output');
Delta_vals = [0 50 100 500 1000];
fcut_vals = [200 300 500 1000 10000];




cutoff = 1000;
Wn = cutoff/(Fs/2);
[b,a] = butter(6,Wn,'low');
figure; hold on;
for d = 1:length(Delta_vals)
```

```
    osc = cos(2*pi*(fc + Delta_vals(d))*t);
    mixed = 2*kt.*osc;
    demod_out = filtfilt(b,a,mixed);
    plot(t(1:1000), demod_out(1:1000), 'DisplayName', ...
         sprintf('\\Delta = %d Hz', Delta_vals(d)));
end
xlabel('Time (s)');
ylabel('Amplitude');
title('Coherent Detection with f_c =1000Hz');
cutoff_vals = [100, 200, 400, 1000];
Delta_vals = [0 50 100 500];




%legend('show');


radians = [0 pi/6 pi/3 pi/2];
figure; hold on;
for p = 1:length(radians)
    x_osc_phi = cos(2*pi*fc*t + radians(p));
    mixed_phi = 2*kt.*x_osc_phi;
    demod_out2 = filtfilt(b,a,mixed_phi);
    plot(t(1:1000), demod_out2(1:1000), 'DisplayName', ...
         sprintf('\\phi = %g', radians(p)));
end
xlabel('Time (s)');
ylabel('Amplitude');
title('Coherent Detection with Different Phases');
legend('show');
```