# C:\Users\a1709295\Downloads\img-514164542-0001.tif

# Introduction

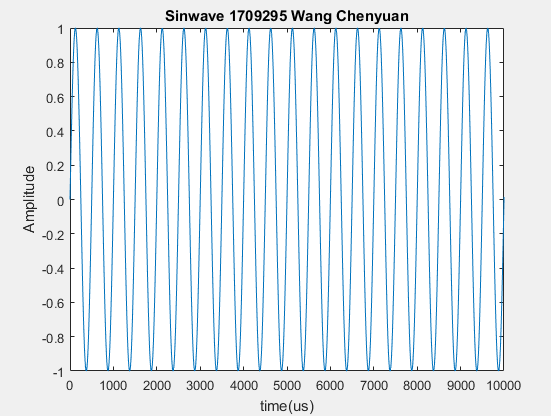
In this MTLAB assignment, the main mission is to generate a sinewave signal, and frequency modulate it into a phasor form. In addition, add additional noise to it and transmit it as FM signal with white noise. In the receiver part, FM signal with white noise is filtered with H3 filter. By calculating its phase, this signal could be recovered into time plane. After filtered by another filter H4, sinewave signal v0 is recovered with much lower noise.

To verify the SNR of the system, random signal (scaled) has also be filtered, and recovered in the receiver system. As a result, additive noise has also be filtered.

The entire MATLAB assignment is to modulate input signal in transmitter and demodulate FM signal with additive noise and filtering it to reduce the noise. The whole system has a theoretical SNR. And the theoretical SNR is also verified and discussed in this assignment.

# Sinewave generation

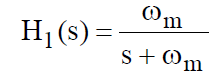
Since input sinewave is m(t) = sin(2\*pi\*2000t), the input sinewave is generated as following graph:

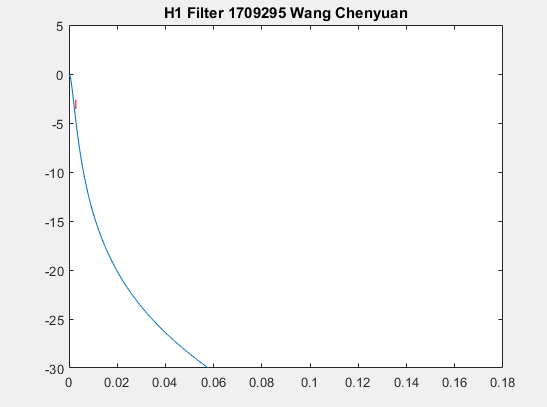


Graph 1 Sinewave m(t)

# Filter

## H1 filter

Required filter in this assignment is that

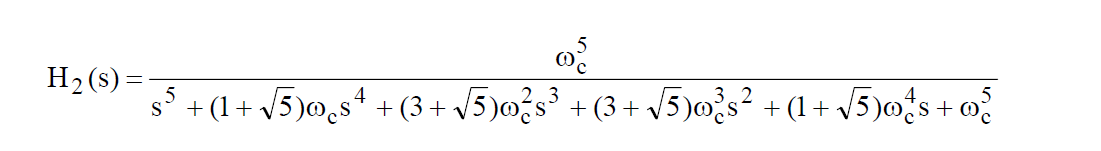


Graph H1frequency response

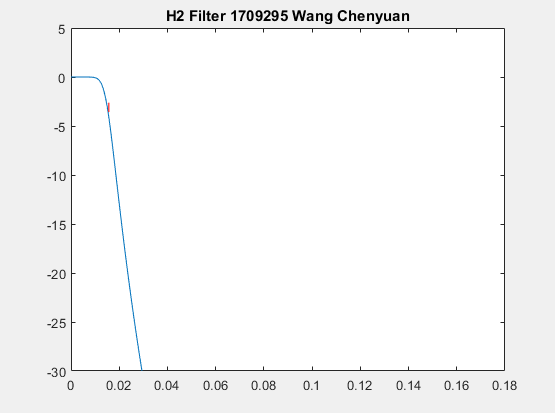
And the frequency response of H1 is following:

It clearly shows the frequency response of the H1 filter, and it’s significant that the 3dB frequency of H1 filter is around 2kHz.

## H2 filter



The frequency response of H2 filter is:

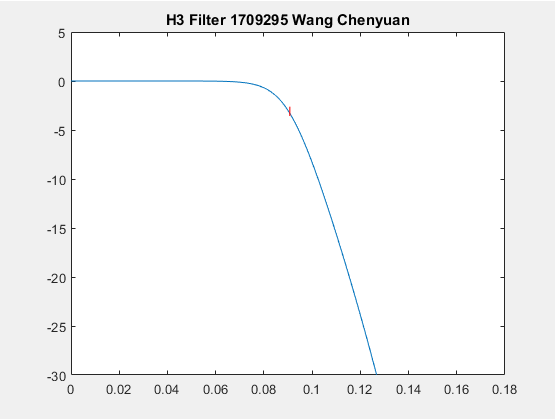


Graph H2 frequency response

It shows that the 3dB frequency is at around 15kHz.

## H3 filter

Since H3 filter has the similar form as H2 filter, the difference is that, the frequency of H3 filter is 90kHz. Therefore, the frequency response of H3 filter is:

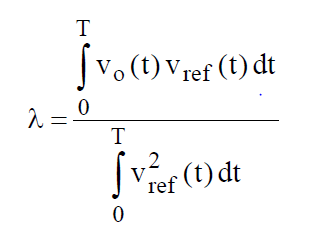


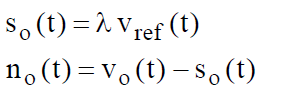
Graph H3 filter frequency response

It shows that the 3dB frequency is at around 90kHz.

# Reference signal

Since reference signal is modulated signal directly received and demodulated by the receiver without adding additive noise, it is a valuable value to verify the separation of input signal and noise.

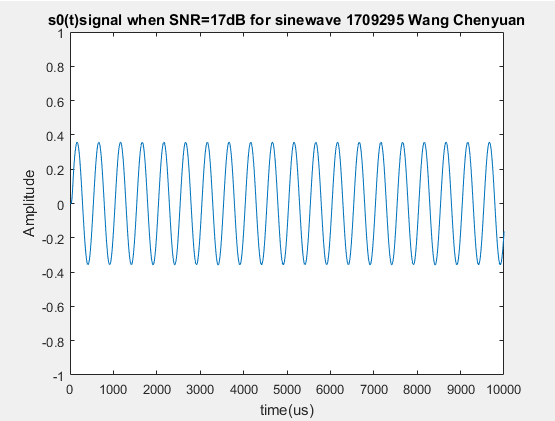
The separated signal could be calculated by following formula:



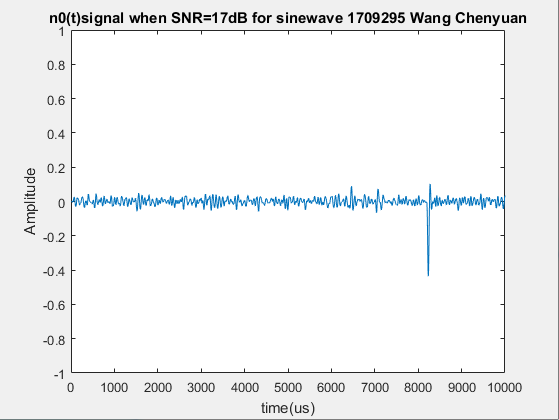
# n0 (t) and s0 (t) signal

In this assignment, s0 signal is the input signal being filtered and demodulated, and n0 signal is separated noise signal.

In specific case that Pr/N0W = 17dB, the n0(t) and s0(t) is shown in following graph:



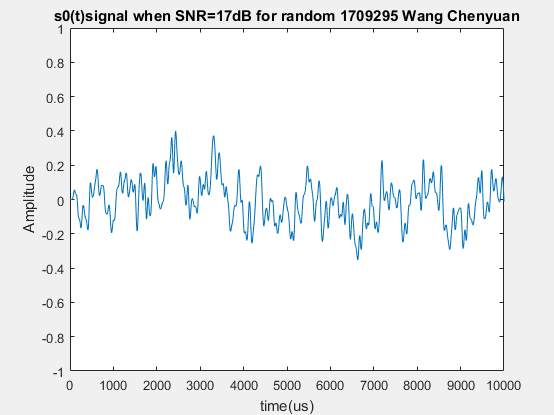
Graph S0(t) when 17dB



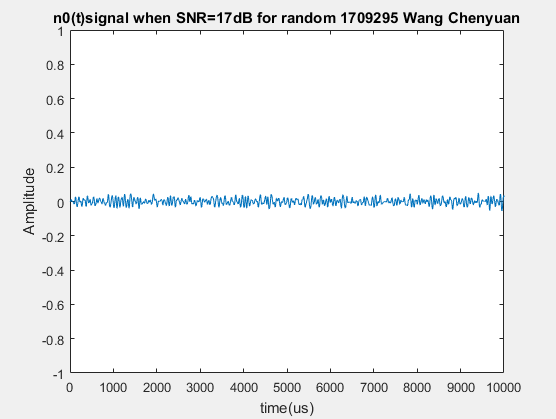
Graph n0(t) for 17dB

Similar consequence also eligible for random modulated signal:

It’s significant that the noise signals have very low magnitude after filtered as the demodulated input signal is strong.



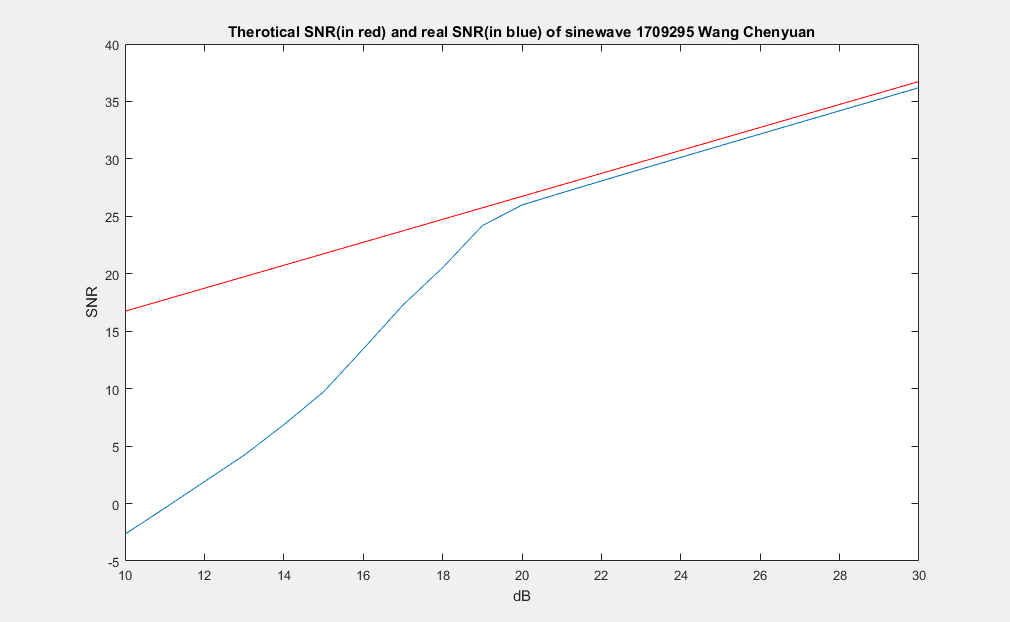
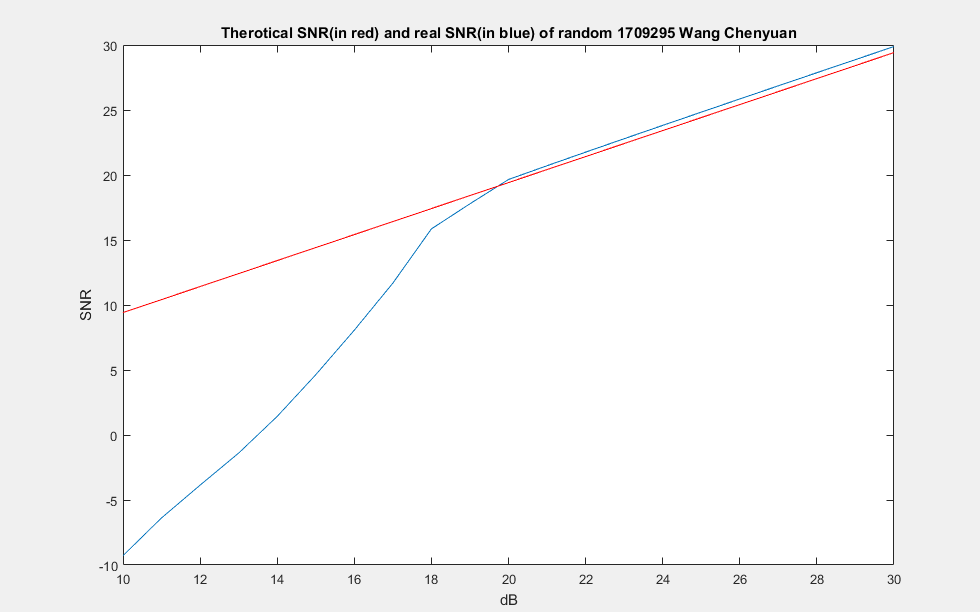
Graph s0(t) for random when 17dB



Graph 8 n0(t) for random when17dB

# Signal to noise ratio

The theoretical SNR and the real SNR of the entire system is in following graph:



It shows that for both sinewave and random modulated signal, the real SNR is significantly lower than the theoretical one when Pr/N0W is low. But getting closer to the theoretical SNR when the value of Pr/N0W is getting larger.

When calculating the real and theoretical SNR of the output system, power (value of <m2>) of both output signal and noise signal is required. Because of the inconstant digital signal, square pf each amplitude of both signals is used to calculate power.

When Pr/N0W is at around 19dB, the output SNR is about 1dB than the theoretical one.

Therefore, the definition of threshold point should be that the noise spectral density is low enough for the system and filter work as the theoretical value. Too strong and dense noise would influence the work of the filter. Which will conclude a lower SNR than required.

# Discussion

## The difference between real and theoretical SNR

When Pr/N0W is small, for Pr/N0W = A2/2N0W, No is very large. For N0 is noise spectral density, a larger N0 means that the dense of noise is high. Which means the power of noise is larger. Therefore, for a large N0, filter can reduce the noise, but could not reduce it to 3dB less than the input signal in the bandwidth of 15kHz.

When N0 is getting larger, the output SNR is significantly getting closer to the theoretical one. But still not exactly the same. That is because, the role of filters are not to delete the noise, but reduce it. So the error exits, and for different noise, the error is different.

## The SNR graph of random modulated signal

After numbers time of run the MATLAB code. It’s found that the error between real SNR and Theoretical SNR is larger than sinewave, and the difference is more significant among different experiment. This may because that, the random modulated signal has some frequencies that can also be filtered by the filters.

## The value of λ

It's found that, the λ of both sinewave and random formulated signals are 1 or pretty near 1. It may because that after filters, the noise has already extremely low.

# Comment

The consequence of the entire assignment turns out to be successful, because noise and input signals are successfully separated. In addition, the real SNR and theoretical SNR of both sinewave and random modulated signal are both logical.

## Difficulties

The difficulties I met is firstly, not very familiar with MATLAB and struggled when want to achieve some aim. I searched on internet and ask for help to achieve the goal. In the procedure of doing the assignment, new skills for me is accepted and learned.

Another difficulty occurs when calculating the output SNR and theoretical SNR of the system, the power of the signals are hard to define. Finally I choose use using getting sum of magnitude instead of doing integrator for both signals are digital signals.

## Process

By doing this assignment, I handle more knowledge of MATLAB, in addition, know better about filters and transmission system.

# Appendix

## MATLAB code towards sinewave:

W = 0.015;

A = 1;

fd = 0.075;

fs = 0.36;

N = 100000;

n = 0: N-1;

t = n/fs ;

delta = 1/fs;

mt = sin(2\*0.002\*pi\*t); %Assuming Sinwave

figure(1);

plot (t, mt);

title ('Sinwave 1709295 Wang Chenyuan');

xlabel ('time(us)');

ylabel ('Amplitude');

axis([0 10000 -1 1]);

hold on;

grng = randn(1,100000); %Making random noise

figure(2);

plot (t, grng);

title ('GRNG 1709295 Wang Chenyuan');

xlabel ('time(us)');

ylabel ('Amplitude');

axis([0 10000 -5 5]);

hold on;

wm = (2\*tan((2\*pi\*0.002\*delta)/2))/delta; %Using fomula to calculate wm,wc and wa

wc = (2\*tan((2\*pi\*0.015\*delta)/2))/delta;

wa = (2\*tan((2\*pi\*0.09\*delta)/2))/delta;

h1num = wm;

h1den = [1, wm];

h2num = wc^5;

h2den = [1, (1+sqrt(5))\*wc, (3+sqrt(5))\*wc^2, (3+sqrt(5))\*wc^3, (1+sqrt(5))\*wc^4, wc^5];

h3num = wa^5;

h3den = [1, (1+sqrt(5))\*wa, (3+sqrt(5))\*wa^2, (3+sqrt(5))\*wa^3, (1+sqrt(5))\*wa^4, wa^5];

[zh1num, zh1den] = bilinear(h1num, h1den, fs); %Transform H1 from S plane to Z plane

[zh2num, zh2den] = bilinear(h2num, h2den, fs); %Transform H2 from S plane to Z plane

[zh3num, zh3den] = bilinear(h3num, h3den, fs); %Transform H2 from S plane to Z plane

grngh1 = filter(zh1num, zh1den, grng); %Use h1 filter to deal the noise

grngh2 = filter(zh2num, zh2den, grngh1); %use h2 filter to deal the noise

figure(4);

freqz(zh1num, zh1den);

hold on;

title ('H1 Filter 1709295 Wang Chenyuan');

axis([0 0.18 -30 5]);

figure(8);

freqz(zh2num, zh2den);

hold on;

title ('H2 Filter 1709295 Wang Chenyuan');

axis([0 0.18 -30 5]);

figure(9);

freqz(zh3num, zh3den);

title ('H3 Filter 1709295 Wang Chenyuan');

axis([0 0.18 -30 5]);%Plot the charactoristic of h1&h2,h3

hold on;

figure(3);

plot (t,grngh2);

title ('GRNG (After H1 & H2 Filter) 1709295 Wang Chenyuan')

xlabel ('time(us)');

ylabel ('Amplitude');

axis([0 10000 -1 1]);

hold on;

% Modulation

i = 1:N-1;

md = zeros(1,N);

mid = sin(2\*pi\*0.002\*i\*delta);

md(1) = 0;

for k = 1:N-1

md(k+1) = md(k) + mid(k);

end

Md = 2\*pi\*fd\*md\*delta; %Md is the phase form of modulation signal

vt = A\*exp(1i\*Md); %Modulation

%Additive Noise genertation

% System

realsnr = zeros(1,21);

hopesnr = zeros(1,21);

ntr = randn(1,100000);

ncr = randn(1,100000);

for SNR = 10:30

snr = 10^(SNR/10);

N0 = A^2/(2\*W\*snr);

P = sum(vt.^2)/N;

row = sqrt(N0\*fs);

noise = ntr\*row+ncr\*1i\*row; % Additive Noise

vtadd = vt + noise; % Transmission signal with noise

% Demodulation

vth3 = filter(zh3num, zh3den, vtadd);

setra = angle(vth3);

setrau = unwrap(setra);

vdkdelta = zeros(1,N);

vdkdelta(1) = setrau(1)/(2\*pi\*fd\*delta); % Mannually announce Vdkdelta(1)

for k = 2:N-1

vdkdelta(k) = (setrau(k)-setrau(k-1))/(2\*pi\*fd\*delta);

end

v0t = filter(zh2num, zh2den, vdkdelta); %Because H4 is the same with H2

% Generate vtref signal

vth3ref = filter(zh3num, zh3den, vt);

setraref = angle(vth3ref);

setrauref = unwrap(setraref);

vdkdeltaref = zeros(1,N);

vdkdeltaref(1) = setrauref(1)/(2\*pi\*fd\*delta); % Mannually announce Vdkdelta(1)

for k = 2:N-1

vdkdeltaref(k) = (setrauref(k)-setrauref(k-1))/(2\*pi\*fd\*delta);

end

vtref = filter(zh2num,zh2den, vdkdeltaref); %This is orifinal vt signal being modulated and demodulated which is vtref

vmix = v0t.\*vtref;

vmix2 = vtref.\*vtref;

lambudanum = sum(vmix);

lambudaden = sum(vmix2);

lambuda = lambudanum/lambudaden; %Calculating lambuda

s0t = lambuda\*vtref;

n0t = v0t-s0t;

s0t2 = s0t.\*s0t;

n0t2 = n0t.\*n0t;

ps0t = sum(s0t2); % power of S0(t)

pn0t = sum(n0t2); % power of n0(t)

realsnr(SNR-9) = 10\*log10(ps0t/pn0t);

hopesnr(SNR-9) = 10\*log10(3\*25\*0.5\*P/(N0\*0.015\*1.153));

switch SNR % Switch the case witch SNR=17 and plot s0(t) and n0(t)

case 17

figure(5);

plot (t, s0t);

title ('s0(t)signal when SNR=17dB for sinewave 1709295 Wang Chenyuan')

axis([0 10000 -1 1]);

xlabel ('time(us)');

ylabel ('Amplitude');

figure(6);

plot (t, n0t);

title ('n0(t)signal when SNR=17dB for sinewave 1709295 Wang Chenyuan')

xlabel ('time(us)');

ylabel ('Amplitude');

axis([0 10000 -1 1]);

hold on;

end

end

snrnum = zeros(1,21);

for i = 10:30

snrnum(i-9) = i;

end

figure(7);

plot(snrnum, realsnr);

hold on;

plot(snrnum, hopesnr,'r');

title ('Therotical SNR(in red) and real SNR(in blue) of sinewave 1709295 Wang Chenyuan')

xlabel ('dB');

ylabel ('SNR');

## MATLAB code towards random modulated signal

W = 0.015;

A = 1;

fd = 0.075;

fs = 0.36;

N = 100000;

n = 0: N-1;

t = n/fs ;

delta = 1/fs;

mt = sin(2\*0.002\*pi\*t); %Assuming Sinwave

figure(1);

plot (t, mt);

title ('Sinwave 1709295 Wang Chenyuan');

xlabel ('time(us)');

ylabel ('Amplitude');

axis([0 10000 -1 1]);

hold on;

grng = randn(1,100000); %Making random noise

figure(2);

plot (t, grng);

title ('GRNG 1709295 Wang Chenyuan');

xlabel ('time(us)');

ylabel ('Amplitude');

axis([0 10000 -5 5]);

hold on;

wm = (2\*tan((2\*pi\*0.002\*delta)/2))/delta; %Using fomula to calculate wm,wc and wa

wc = (2\*tan((2\*pi\*0.015\*delta)/2))/delta;

wa = (2\*tan((2\*pi\*0.09\*delta)/2))/delta;

h1num = wm;

h1den = [1, wm];

h2num = wc^5;

h2den = [1, (1+sqrt(5))\*wc, (3+sqrt(5))\*wc^2, (3+sqrt(5))\*wc^3, (1+sqrt(5))\*wc^4, wc^5];

h3num = wa^5;

h3den = [1, (1+sqrt(5))\*wa, (3+sqrt(5))\*wa^2, (3+sqrt(5))\*wa^3, (1+sqrt(5))\*wa^4, wa^5];

[zh1num, zh1den] = bilinear(h1num, h1den, fs); %Transform H1 from S plane to Z plane

[zh2num, zh2den] = bilinear(h2num, h2den, fs); %Transform H2 from S plane to Z plane

[zh3num, zh3den] = bilinear(h3num, h3den, fs); %Transform H2 from S plane to Z plane

grngh1 = filter(zh1num, zh1den, grng); %Use h1 filter to deal the noise

grngh2 = filter(zh2num, zh2den, grngh1); %use h2 filter to deal the noise

figure(4);

freqz(zh1num, zh1den);

hold on;

title ('H1 Filter 1709295 Wang Chenyuan');

axis([0 0.18 -30 5]);

figure(8);

freqz(zh2num, zh2den);

hold on;

title ('H2 Filter 1709295 Wang Chenyuan');

axis([0 0.18 -30 5]);

figure(9);

freqz(zh3num, zh3den);

title ('H3 Filter 1709295 Wang Chenyuan');

axis([0 0.18 -30 5]);%Plot the charactoristic of h1&h2,h3

hold on;

figure(3);

plot (t,grngh2);

title ('GRNG (After H1 & H2 Filter) 1709295 Wang Chenyuan')

xlabel ('time(us)');

ylabel ('Amplitude');

axis([0 10000 -1 1]);

hold on;

% Modulation

i = 1:N-1;

md = zeros(1,N);

mid = sin(2\*pi\*0.002\*i\*delta);

md(1) = 0;

for k = 1:N-1

md(k+1) = md(k) + mid(k);

end

Md = 2\*pi\*fd\*md; %Md is the phase form of modulation signal

vt = A\*exp(1i\*Md); %Modulation

%Additive Noise genertation

% System

realsnr = zeros(1,21);

hopesnr = zeros(1,21);

ntr = randn(1,100000);

ncr = randn(1,100000);

for SNR = 10:30

snr = 10^(SNR/10);

N0 = A^2/(2\*W\*snr);

P = sum(vt.^2)/N;

row = sqrt(N0\*fs);

noise = ntr\*row+ncr\*1i\*row; % Additive Noise

vtadd = vt + noise; % Transmission signal with noise

% Demodulation

vth3 = filter(zh3num, zh3den, vtadd);

setra = angle(vth3);

setrau = unwrap(setra);

vdkdelta = zeros(1,N);

vdkdelta(1) = setrau(1)/(2\*pi\*fd\*delta); % Mannually announce Vdkdelta(1)

for k = 2:N-1

vdkdelta(k) = (setrau(k)-setrau(k-1))/(2\*pi\*fd\*delta);

end

v0t = filter(zh2num, zh2den, vdkdelta); %Because H4 is the same with H2

% Generate vtref signal

vth3ref = filter(zh3num, zh3den, vt);

setraref = angle(vth3ref);

setrauref = unwrap(setraref);

vdkdeltaref = zeros(1,N);

vdkdeltaref(1) = setrauref(1)/(2\*pi\*fd\*delta); % Mannually announce Vdkdelta(1)

for k = 2:N-1

vdkdeltaref(k) = (setrauref(k)-setrauref(k-1))/(2\*pi\*fd\*delta);

end

vtref = filter(zh2num,zh2den, vdkdeltaref); %This is orifinal vt signal being modulated and demodulated which is vtref

vmix = v0t.\*vtref;

vmix2 = vtref.\*vtref;

lambudanum = sum(vmix);

lambudaden = sum(vmix2);

lambuda = lambudanum/lambudaden; %Calculating lambuda

s0t = lambuda\*vtref;

n0t = v0t-s0t;

s0t2 = s0t.\*s0t;

n0t2 = n0t.\*n0t;

ps0t = sum(s0t2); % power of S0(t)

pn0t = sum(n0t2); % power of n0(t)

realsnr(SNR-9) = 10\*log10(ps0t/pn0t);

hopesnr(SNR-9) = 10\*log10(3\*25\*0.5\*P/(N0\*0.015\*1.153));

switch SNR % Switch the case witch SNR=17 and plot s0(t) and n0(t)

case 17

figure(5);

plot (t, s0t);

title ('s0(t)signal when SNR=17dB for sinewave 1709295 Wang Chenyuan')

axis([0 10000 -1 1]);

xlabel ('time(us)');

ylabel ('Amplitude');

figure(6);

plot (t, n0t);

title ('n0(t)signal when SNR=17dB for sinewave 1709295 Wang Chenyuan')

xlabel ('time(us)');

ylabel ('Amplitude');

axis([0 10000 -1 1]);

hold on;

end

end

snrnum = zeros(1,21);

for i = 10:30

snrnum(i-9) = i;

end

figure(7);

plot(snrnum, realsnr);

hold on;

plot(snrnum, hopesnr,'r');

title ('Therotical SNR(in red) and real SNR(in blue) of sinewave 1709295 Wang Chenyuan')

xlabel ('dB');

ylabel ('SNR');