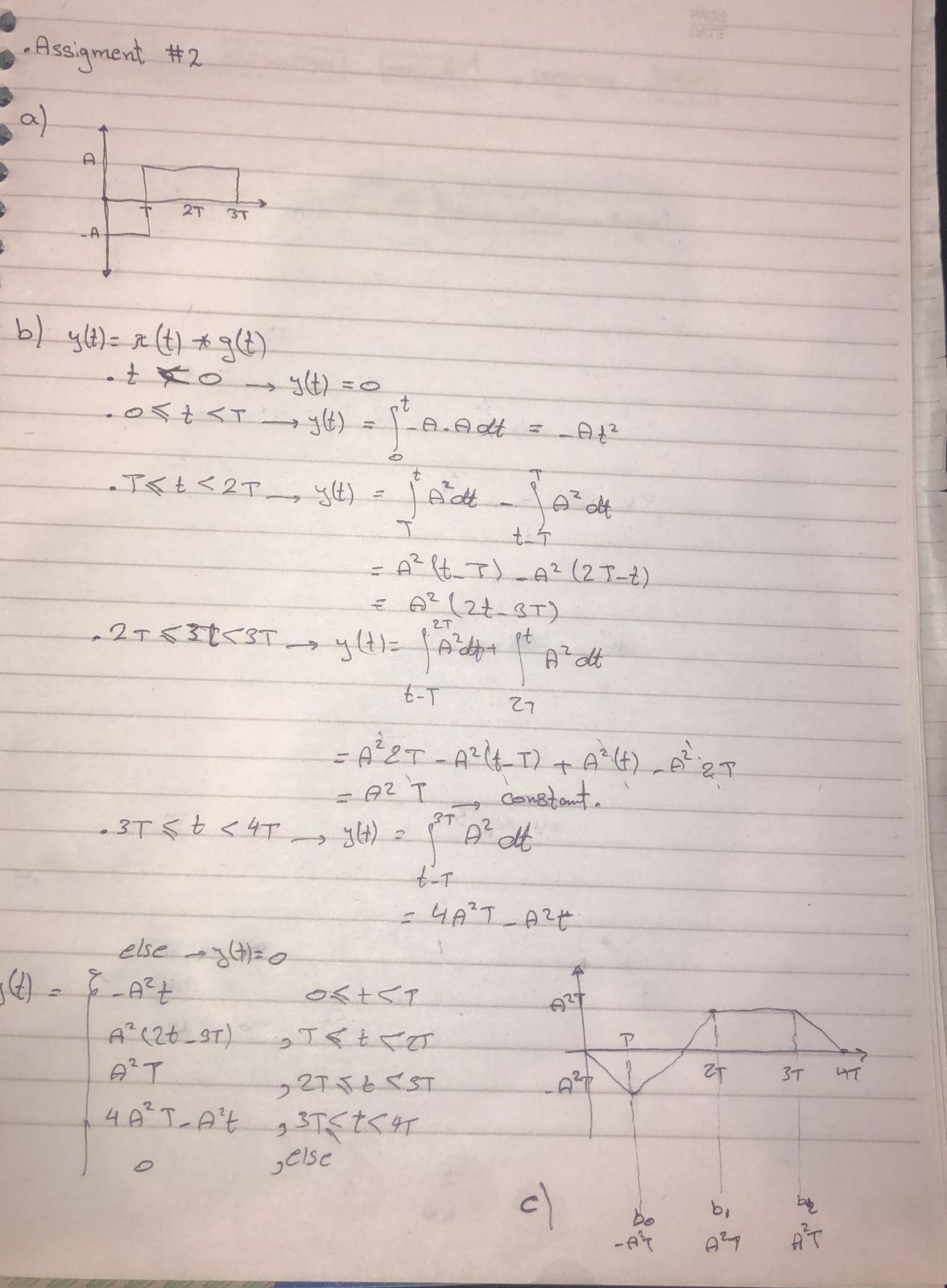
Cairo University

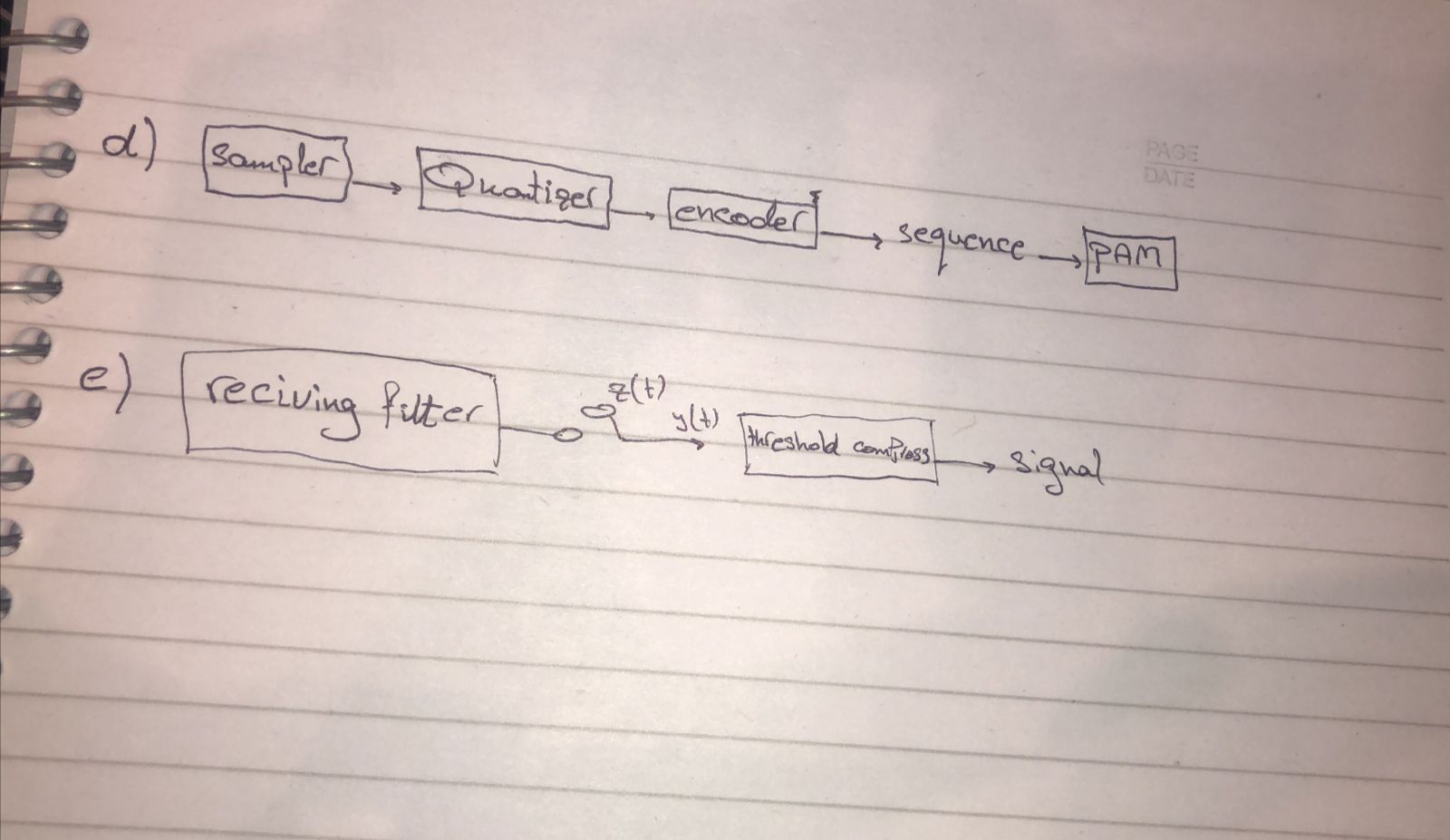
Faculty of Engineering

**Assignment #2 Digital Communication**

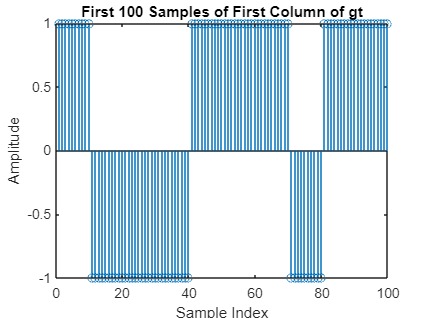
|  |  |
| --- | --- |
| **9204096** | **كريم محمود صقر** |
| **9202758** | **عبد الحميد عماد** |

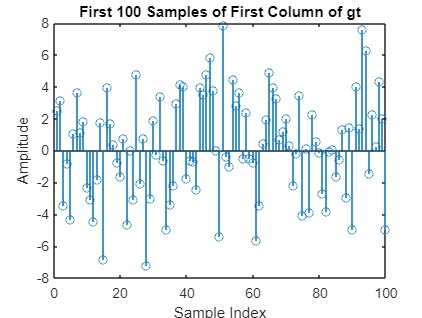
**Problem 1 :**

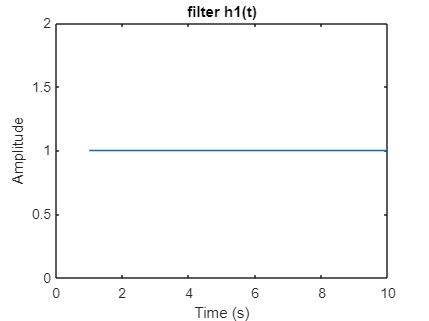


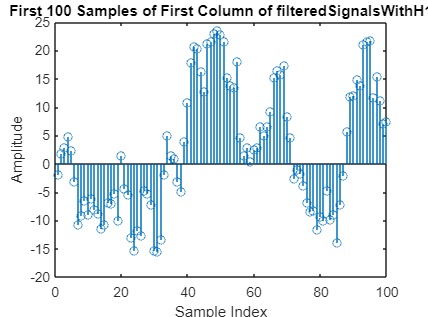


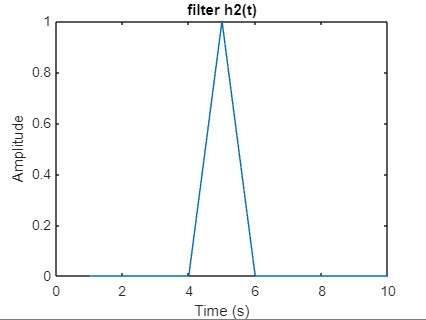
**Problem 2:**

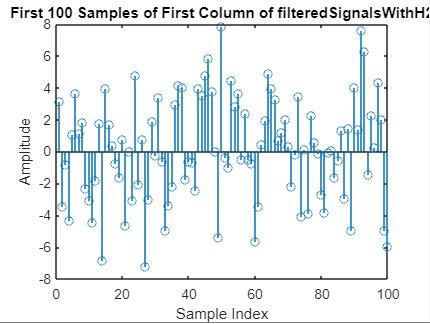


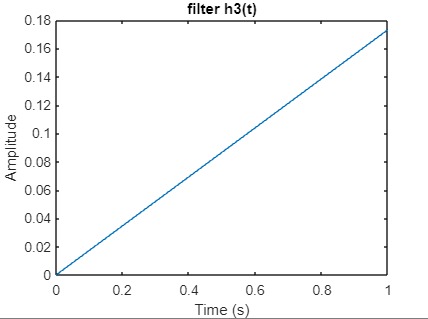


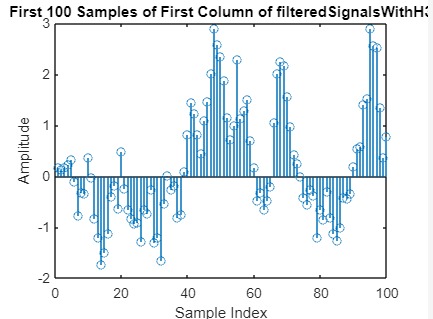


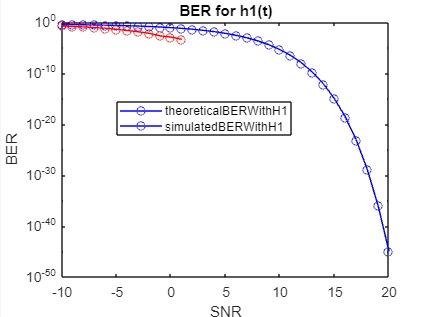


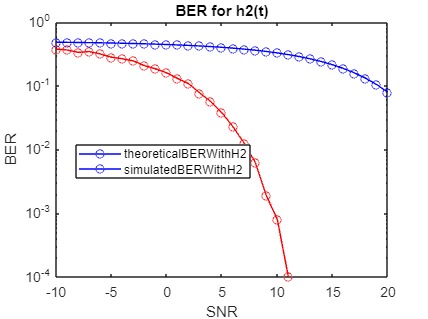


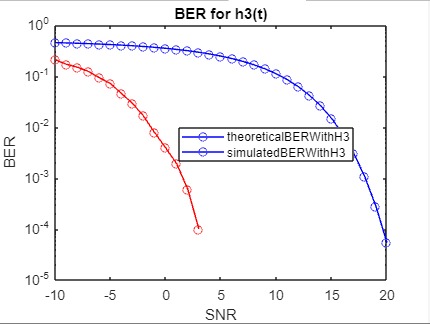


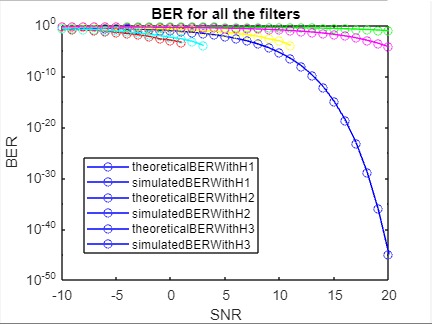


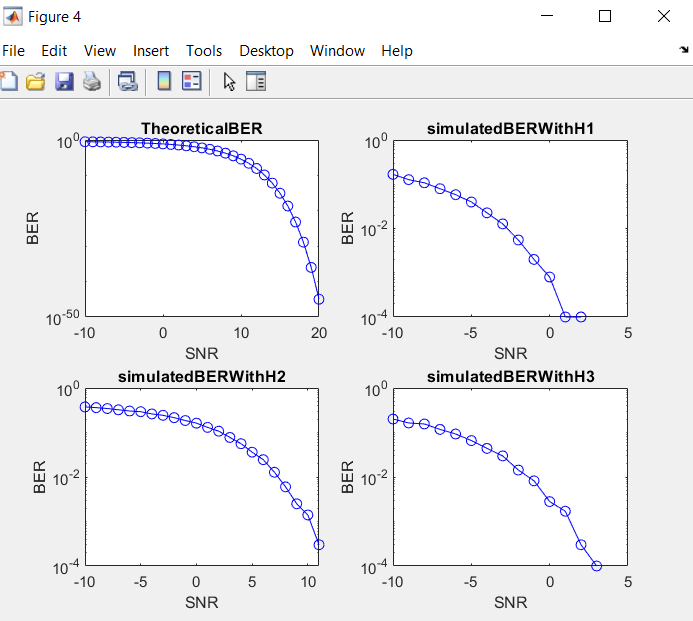












5-Generally, the BER decreases as the Energy per bit to Noise power Spectral Density ratio (E/No) increases. This is because increasing the E/No ratio reduces the effect of noise on the received signal, making it easier to distinguish between the transmitted bits.

i.e, as the E/No ratio increases, the signal becomes stronger relative to the noise, and the probability of errors decreases.

6-The matched filter with unit energy typically results in the lowest BER. This is because the matched filter is designed to maximize the signal-to-noise ratio (SNR) at the receiver, which is equivalent to minimizing the noise power while maximizing the signal power.

**%CODE**

v1 = zeros(1, 10000);

idx = randperm(length(v1), 5000);

v1(idx) = 1;

gt = zeros(1, 100000);

for i = 1:length(v1)

    if v1(i) == 1

        gt(:, (i-1)\*10+1:i\*10) = 1;

    end

    if v1(i) == 0

        gt(:, (i-1)\*10+1:i\*10) = -1;

    end

end

figure;

stem(1:100, gt(1:100));

xlabel('Sample Index');

ylabel('Amplitude');

title('First 100 Samples of First Column of gt');

noisy\_gt = zeros(length(gt),31);

i=1;

for snr=-10:1:20

     noisy\_gt(:,i) = awgn(gt, snr, 'measured', 'dB');

     i = i +1 ;

end

figure;

stem(1:100, noisy\_gt(1:100));

xlabel('Sample Index');

ylabel('Amplitude');

title('First 100 Samples of First Column of gt');

%% h1(t)

h\_t\_1 = ones(10,1);

% ploting the filter h(t)

figure;

plot(h\_t\_1);

xlabel('Time (s)');

ylabel('Amplitude');

title('filter h1(t)');

filteredSignalsWithH1 = zeros(length(gt), 31);

i=1;

for snr=-10:1:20

      filteredSignalsWithH1(:,i) =conv(noisy\_gt(:,i),h\_t\_1, 'same');

      i=i+1;

end

% ploting the filtered signal with h1(t)

figure;

stem(1:100, filteredSignalsWithH1(1:100,1));

xlabel('Sample Index');

ylabel('Amplitude');

title('First 100 Samples of First Column of filteredSignalsWithH1');

%% h2(t)

h\_t\_2 = zeros(10,1);

h\_t\_2(5) =1;

% ploting the filter h2(t)

figure;

plot(h\_t\_2);

xlabel('Time (s)');

ylabel('Amplitude');

title('filter h2(t)');

filteredSignalsWithH2 = zeros(length(gt), 31);

i=1;

for snr=-10:1:20

     filteredSignalsWithH2(:,i) =conv(noisy\_gt(:,i),h\_t\_2, 'same');

     i=i+1;

end

% ploting the filtered signal with h2(t)

figure;

stem(1:100, filteredSignalsWithH2(1:100,1));

xlabel('Sample Index');

ylabel('Amplitude');

title('First 100 Samples of First Column of filteredSignalsWithH2');

T = 10;

A=1/10;

t = linspace(0, 1, 10);

%% h3(t)

h\_t\_3 = zeros(1,10);

% y = mx +c

% slope r(3)/10

for i = 0:9

    h\_t\_3(i+1) = sqrt(3)\*i/T;

end

figure;

plot(t,h\_t\_3)

xlabel('Time (s)')

ylabel('Amplitude')

title('filter h3(t)')

%b\_3 =conv(g,noisy\_gt(:,1));

filteredSignalsWithH3 = zeros(length(gt), 31);

i=1;

for snr=-10:1:20

      filteredSignalsWithH3(:,i) =conv(noisy\_gt(:,i),h\_t\_3, 'same');

      i=i+1;

end

% ploting the filtered signal with h3(t)

figure;

stem(1:100, filteredSignalsWithH3(1:100,1));

xlabel('Sample Index');

ylabel('Amplitude');

title('First 100 Samples of First Column of filteredSignalsWithH3');

%%sampling

%fs = 10; % sampling frequency of x

%T = 1/fs; % sampling period of x

%n = fs/100; % downsampling factor

%y = downsample(x, n); % sample x at T

sampledSignalWithH1 = zeros(length(gt)/10, 31);

i=1;

for snr=-10:1:20

      ss =filteredSignalsWithH1(:,i);

      sampledSignalWithH1(:,i) =ss(5:10:end);

      i=i+1;

end

sampledSignalWithH2 = zeros(length(gt)/10, 31);

i=1;

for snr=-10:1:20

      ss =filteredSignalsWithH2(:,i);

      sampledSignalWithH2(:,i) =ss(5:10:end);

      i=i+1;

end

sampledSignalWithH3 = zeros(length(gt)/10, 31);

i=1;

for snr=-10:1:20

     ss =filteredSignalsWithH3(:,i);

     sampledSignalWithH3(:,i) =ss(5:10:end);

     i=i+1;

end

%%

threshold=0;

snr = -10:1:20;

decoded\_signalWithH1 = zeros(length(gt)/10, length(snr));

decoded\_signalWithH2 = zeros(length(gt)/10, length(snr));

decoded\_signalWithH3 = zeros(length(gt)/10, length(snr));

for i = 1:length(snr)

     decoded\_signalWithH1(:,i) = (sampledSignalWithH1(:,i) > threshold);

     decoded\_signalWithH2(:,i) = (sampledSignalWithH2(:,i) > threshold);

     decoded\_signalWithH3(:,i) = (sampledSignalWithH3(:,i) > threshold);

end

n = length(gt)/10;

T = 10;

A=1/10;

theoreticalBER = zeros(31);

for i = 1:length(snr)

    s = 10^(snr(i)/10);

    theoreticalBER(i) = 0.5\*erfc(sqrt(s));

end

simulatedBERWithH1 = zeros(length(snr),1);

simulatedBERWithH2 = zeros(length(snr),1);

simulatedBERWithH3 = zeros(length(snr),1);

for i = 1:length(snr)

        decodedSignal = decoded\_signalWithH1(:,i);

        errors = sum(xor(decodedSignal', v1));

        simulatedBERWithH1(i,1) = errors/n;

        decodedSignal = decoded\_signalWithH2(:,i);

        errors = sum(xor(decodedSignal', v1));

        simulatedBERWithH2(i,1) = errors/n;

        decodedSignal = decoded\_signalWithH3(:,i);

        errors = sum(xor(decodedSignal', v1));

        simulatedBERWithH3(i,1) = errors/n;

end

% calculate the theoritical BER

% A=1;

% T=1;

% clculate the BER for h1

theoreticalBERWithH1 = zeros(31);

for i = 1:length(snr)

    s = 10^(snr(i)/10);

    theoreticalBERWithH1(i) = 0.5\*erfc(sqrt(s));

end

% plot the BER

figure;

semilogy(snr, theoreticalBERWithH1, 'b-o')

hold on

semilogy(snr, simulatedBERWithH1(:,1), 'r-o')

hold off

xlabel('SNR')

ylabel('BER')

legend('theoreticalBERWithH1','simulatedBERWithH1')

title('BER for h1(t)')

% clculate the BER for h2

theoreticalBERWithH2 = zeros(31);

for i = 1:length(snr)

    s = 10^(snr(i)/10);

    theoreticalBERWithH2(i) = 0.5\*erfc(A\*sqrt(s));

end

% plot the BER

figure;

semilogy(snr, theoreticalBERWithH2, 'b-o')

hold on

semilogy(snr, simulatedBERWithH2(:,1), 'r-o')

hold off

xlabel('SNR')

ylabel('BER')

legend('theoreticalBERWithH2','simulatedBERWithH2')

title('BER for h2(t)')

% clculate the BER for h3

theoreticalBERWithH3 = zeros(31);

for i = 1:length(snr)

    s = 10^(snr(i)/10);

    theoreticalBERWithH3(i) = 0.5\*erfc((sqrt(3)/2)\*A\*(T^2)\*sqrt(s)/sqrt(T^3));

end

% plot the BER

figure;

semilogy(snr, theoreticalBERWithH3, 'b-o')

hold on

semilogy(snr, simulatedBERWithH3(:,1), 'r-o')

hold off

xlabel('SNR')

ylabel('BER')

legend('theoreticalBERWithH3','simulatedBERWithH3')

title('BER for h3(t)')

figure;

subplot(2,2,1)

semilogy(snr, theoreticalBER, 'b-o')

xlabel('SNR')

ylabel('BER')

title('TheoreticalBER')

subplot(2,2,2)

semilogy(snr, simulatedBERWithH1(:,1), 'r-o')

xlabel('SNR')

ylabel('BER')

title('simulatedBERWithH1')

subplot(2,2,3)

semilogy(snr, simulatedBERWithH2(:,1), 'g-o')

xlabel('SNR')

ylabel('BER')

title('simulatedBERWithH2')

subplot(2,2,4)

semilogy(snr, simulatedBERWithH3(:,1), 'y-o')

xlabel('SNR')

ylabel('BER')

title('simulatedBERWithH3')

% plot the theoretical BER vs simulated BER for all the filters

figure

semilogy(snr, theoreticalBERWithH1, 'b-o')

hold on

semilogy(snr, simulatedBERWithH1(:,1), 'r-o');

semilogy(snr, theoreticalBERWithH2, 'g-o');

semilogy(snr, simulatedBERWithH2(:,1), 'y-o');

semilogy(snr, theoreticalBERWithH3, 'm-o');

semilogy(snr, simulatedBERWithH3(:,1), 'c-o');

hold off

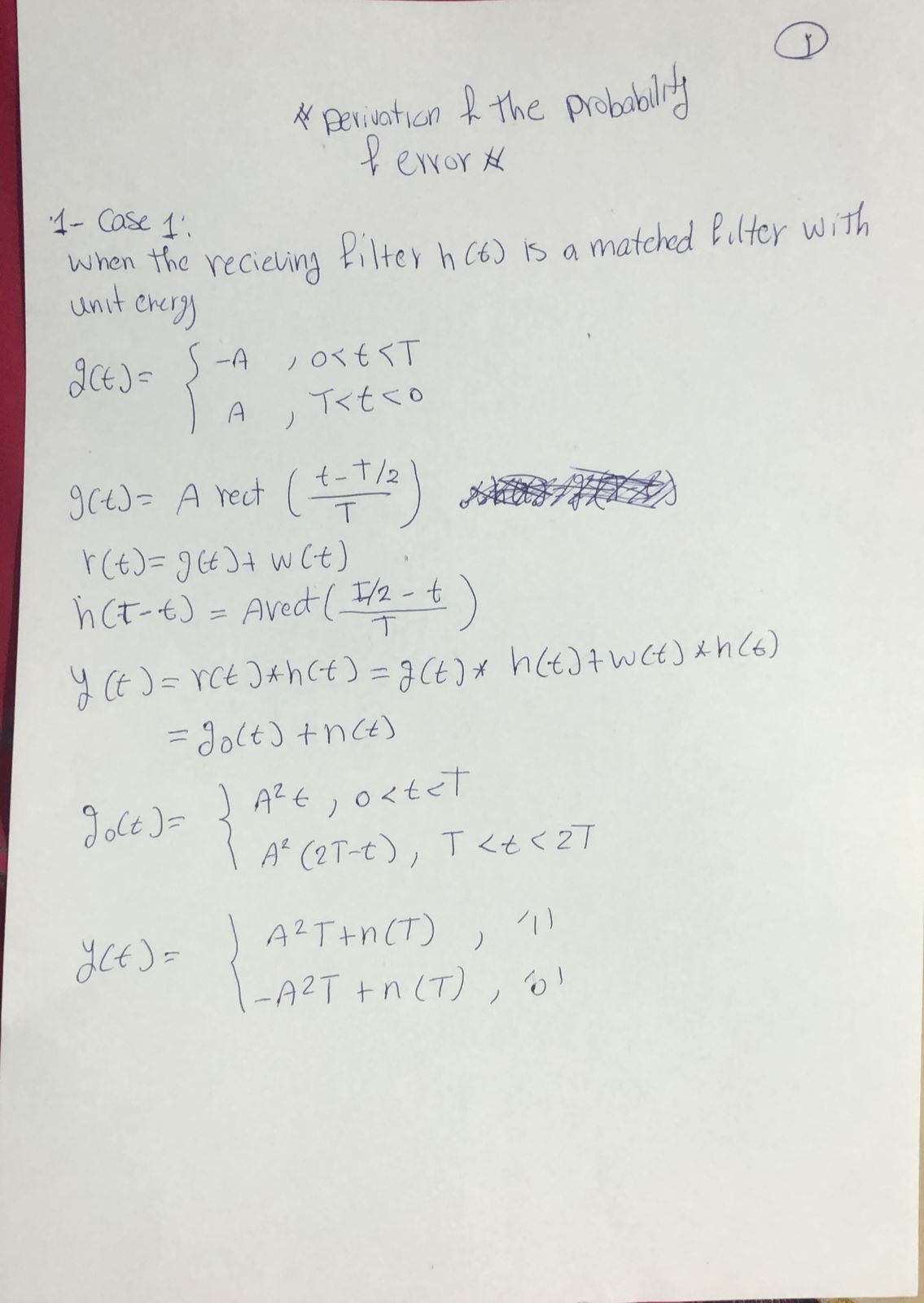
xlabel('SNR')

ylabel('BER')

legend('theoreticalBERWithH1','simulatedBERWithH1','theoreticalBERWithH2','simulatedBERWithH2','theoreticalBERWithH3','simulatedBERWithH3')

title('BER for all the filters')

**Derivations:**

****

