BÁO CÁO PROJECT 2

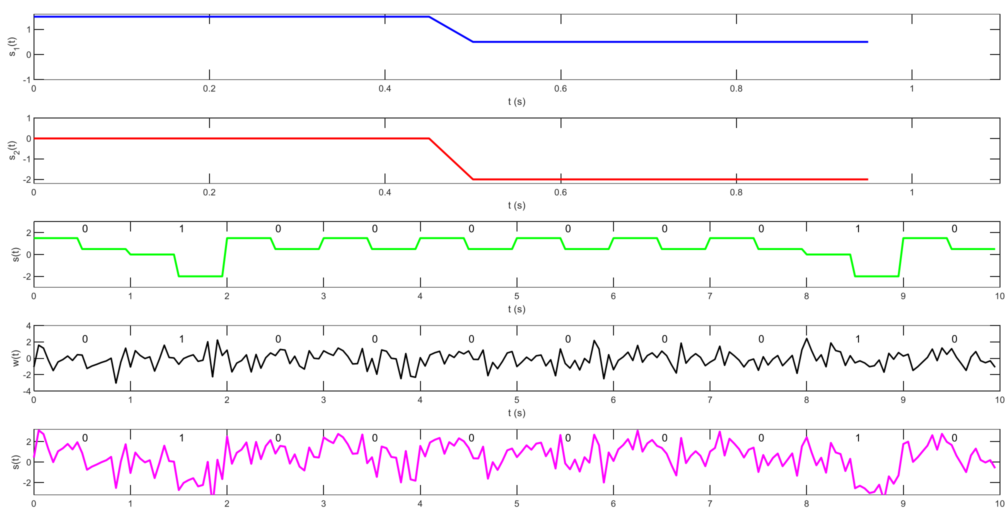
Môn học: Truyền thông số

Họ và tên: Lê Hoàng Nam – MSSV: 21207246

Question 1: Using the hyperthesis in Example 3, complete the following code to present 10 bits of the signal, which is transmitted with P1 = P2 = 0.5.

```
clc; clear; close all;
%% ================ Represent s1(t) and s2(t)
ts = 0.05; % The sample time
t1 = 0: ts: 0.5 - 0.05; % 0 - 0.45
t2 = 0.5: ts: 1 - 0.05; % 0.5 - 0.95
t_1bit = [t1 t2]; % Time of 1 bit
L = length(t_1bit); % The number of samples of 1 bit

% Value of s1(t), s2(t) in Tb
s1_t1 = 1.5*ones(1,length(t1));
s1_t2 = 0.5*ones(1, length(t2));
s1 = [s1_t1 s1_t2]; % s1(t)

s2_t1 = 0*ones(1, length(t1));
s2_t2 = -2*ones(1, length(t2));
s2 = [s2_t1 s2_t2]; % s2(t)

%% ================ The transmitted signal
Ntry = 10^1; % The total transmitted bits
Bit = randsrc(1, Ntry, [0 1]) ;% Transmission with P1 = P2 = 0.5
s = []; % The transmitted signal s(t)
t = []; % The time of s(t)
for i = 1:Ntry
    if Bit(i) == 0
        s = [s s1];
    else
        s = [s s2];
    end

    t_ibit = t_1bit + i - 1; % Time of i-bit
    t = [t t_ibit];
end

%% ================= The AWGN channel
N0_2 = 0.05; % The noise power spectrum desity (W/Hz) N0/2
B = 1/ts; % Bandwidth of signals
Power_noise = N0_2*B; % The power of noise
w = sqrt(Power_noise)*randn(1, length(s));
%% ================= The received signal
r = s+w;

%% ================= Plot
figure(1)
subplot(5,1,1)
plot(t_1bit,s1,'b-','linewidth',1.8); hold on;
```

```
xlabel('t (s)'); ylabel('s_1(t)');
axis([0 1.1 -1 1.6])
subplot(5,1,2)
plot(t_1bit,s2,'r-','linewidth',1.8);
xlabel('t (s)'); ylabel('s_2(t)')
axis([0 1.1 -2.2 1])
x_note = 0.5 :1 :Ntry - 0.5;
y_note = 2.4 *ones(1,Ntry);
Text = string(Bit);
subplot(5,1,3)
plot(t,s,'g-','linewidth',1.8);
text(x_note, y_note, Text);
xlabel('t (s)'); ylabel('s(t)')
axis([0 Ntry -3 3])
subplot(5,1,4)
plot(t,w,'k-','linewidth',1.4);
text(x_note, y_note, Text);
xlabel('t (s)'); ylabel('w(t)')
axis([0 Ntry -4 4])
subplot(5,1,5)
plot(t,r,'m-','linewidth',1.8);
text(x_note, y_note, Text);
xlabel('t (s)'); ylabel('s(t)')
axis([0 Ntry -3.2 3.2])
```



Question 2: Based on the receiver implementation in Example 4, complete the following code to evaluate the system performance via the bit error probability.

```
clear;clc
%% ================ Represent s1(t) and s2(t)
ts = 0.1; % The sample time
t1 = 0: ts: 0.5 - 0.05;
t2 = 0.5: ts: 1 - 0.05;
t_1bit = [t1 t2]; % Time of 1 bit
L = length(t_1bit); % The number of samples of 1 bit

s1_t1 = 1.5*ones(1, length(t1));
s1_t2 = 0.5*ones(1, length(t2));
s1 = [s1_t1 s1_t2]; % s1(t)
s2_t1 = 0*ones(1, length(t1));
s2_t2 = -2*ones(1, length(t2));
s2 = [s2_t1 s2_t2]; % s2(t)
%% ================ The transmitted signal
Ntry = 10^4; % The total transmitted bits
N0_2 =0.2:0.2:1.2; % The noise power spectrum desity (W/Hz) N0/2
P_error_simul = zeros(1,length(N0_2));
P_error_theo = zeros(1,length(N0_2));
for j = 1: length(N0_2)
    Bit = randsrc(1, Ntry, [0 1]); % Transmission with P1 = P2;
    s = []; % The transmitted signal s(t)
    t = []; % The time of s(t)
    for i = 1:Ntry
        if Bit(i) == 0
            s = [s s1];
        else
            s = [s s2];
        end
        t_ibit = t_1bit + i -1; % Time of i-bit
        t = [t t_ibit];
    end
    %% ================= The AWGN channel
    B = 1/ts; % Bandwidth of signals
    Power_noise = N0_2(j)*B; % The power of noise
    w = sqrt(Power_noise)*randn(1, length(s));
    %% ================= The received signal
    r = s+w;

    %% ================= The recovered signal
    h_t1 = -5/sqrt(17)*ones(1, length(t1));
    h_t2 = -3/sqrt(17)*ones(1, length(t2));
    h = [h_t1 h_t2]; % The impulse response of the matched filter
    T = 3/(4*sqrt(17)); % The decision threshold
    Bit_rec = zeros(1,Ntry);
    for i = 1:Ntry
        Frame = r((i-1)*L+1 : i*L); % Construct 1 Frame with L samples
        y = conv(Frame,h)*ts; % The signals pass through the matched filter
        r2_mu = y(L);
        % --------- Comparator for decision
```

```
        if r2_mu >= T
            Bit_rec(i) = 1;
        else
            Bit_rec(i) = 0;
        end
    end
    Bit_rec;

    %% ================== The bit error probability
    % ------------- Simulation
    [Num, rate] = biterr(Bit, Bit_rec);
    P_error_simul(j) = rate;
    % ------------- Theory
    s12_mu = -7/(2*sqrt(17));
    s22_mu = 5/sqrt(17);
    P_error_theo(j) = qfunc((s22_mu - s12_mu)/(2*sqrt(N0_2(j))));
end

%{
%% ==============Plot
figure(1)
subplot(5,1,1)
plot(t_1bit,s1,'b-','linewidth',1.8); hold on;
xlabel('t (s)'); ylabel('s_1(t)');
axis([0 1.1 -1 1.6])
subplot(5,1,2)
plot(t_1bit,s2,'r-','linewidth',1.8);
xlabel('t (s)'); ylabel('s_2(t)')
axis([0 1.1 -2.2 1])
x_note = 0.5 :1 :Ntry - 0.5;
y_note = 2.4 *ones(1,Ntry);

Text = string(Bit);
subplot(5,1,3)
plot(t,s,'g-','linewidth',1.8);
text(x_note, y_note, Text);
xlabel('t (s)'); ylabel('s(t)')
axis([0 Ntry -3 3])
subplot(5,1,4)
plot(t,w,'k-','linewidth',1.4);
text(x_note, y_note, Text);
xlabel('t (s)'); ylabel('w(t)')
axis([0 Ntry -4 4])

subplot(5,1,5)
plot(t,r,'m-','linewidth',1.8);
text(x_note, y_note, Text);
xlabel('t (s)'); ylabel('s(t)')
axis([0 Ntry -3.2 3.2])
%}

figure(1)
plot(N0_2,P_error_simul,'ko','linewidth',1.6,'markersize',6);
hold on;
plot(N0_2,P_error_theo,'r-','linewidth',1.8,'markersize',6);
```

```
xlabel('N_0/2'); ylabel('The bit error probability');
legend('Simulation','Theory')
```