Eray Gündoğdu 22101962

EEE431 Spring 2024/25 MATLAB Assignment 3

Question 1

In this question Frequency Shift Keying (FSK) is observed by using signal space concept, probability of error analysis and decision boundries. In this question there are two defined signals which are modulated h(t) signals with cosines. The bit assignemnt of four different symbols is given in the assignemnt.

a) In this part it is desired to obtain a x(t) which is random combination of the symbols. The decimal values for the randomly generated bits are 2, 0, 3, 0, 2. The duration of x(t) is 0.5s since x(t) is the random combination of 5 symbols each has duration of 0.1s. In Figure 1 x(t) can be observed.

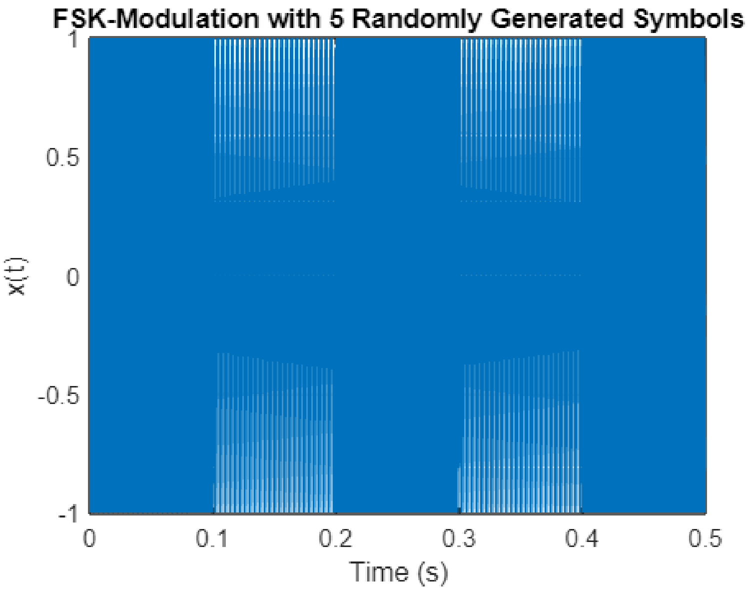


Figure 1 Demonstration of x(t) in time

b)s1(t) and s2(t) are obtained with FSK modulation meaning that baseeband signal is multiplied with cosines with different frequencies. From the fourier transformation it is already known that cosines with different frequencies form an orthagonal basis. In this case it is only desired to normalize the cosine signals to find the orthagonal space. The inner product of a basis function with itself is expected to be 1 and cross inner poduct is expected to be 0. The found bais functions are verified with the previosly mentioned inner product properties by using MATLAB.

$$E_i = \int_0^T \cos^2(2\pi f_i t)\, dt = \frac{T}{2}$$

$$\phi_1(\mathrm{t}) = \mathrm{h(t)cos}(2\pi 250t)\sqrt{\frac{2}{T}} \quad , \phi_2(\mathrm{t}) = \mathrm{h(t)cos}(2\pi 500t)\sqrt{\frac{2}{T}} \; ,$$

```
⟨phi1,phi1⟩=0.996, ⟨phi2,phi2⟩=0.997, ⟨phi1,phi2⟩=-3.539e-03
```

Figure 2 Inner product verification of obtained basis functions

Since there are 2 basis functions the signal space is a 2D space. The inner product of basis functions and symbols gives the constellation points (s_i) in the signal space in which every s_i has a magnitude of $\sqrt{\frac{T}{2}}$. The basis functions can be observed in Figure 3 and s_i's can be observed in Figure 4.

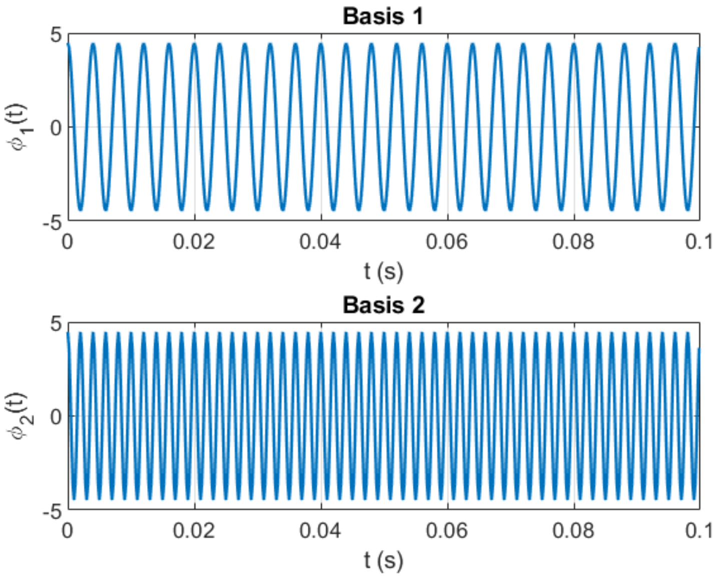


Figure 3 Basis functions in time

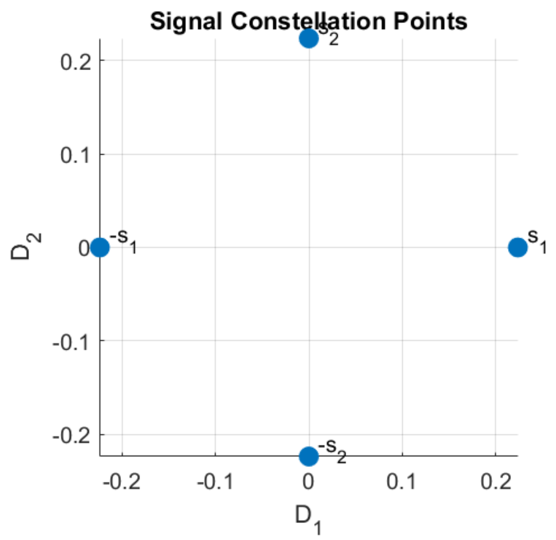


Figure 4 Demonstration of signal constellation points

c)In this part noise is added to x(t) with variances s 10^{-4}, 10^{-2} and, 1. Before plotting the noise added signals influence of the samples per symbol on the SNR will be discussed. The number of samples per sample is related to the sampling frequency hence the relation between sampling frequency and SNR is investigated.

$$SNR = \frac{E_s}{N_0}$$

$$E_s = \frac{1}{M}\sum_{i=0}^{M-1} |s_i|^2$$

In this case variance of noise does not change, and E_s is also contaant since the inner product between the symbols and basis functions does not change hence SNR is not inluenced from changes in sampling frequency in other words, the number of samples forr each symbol. The noise added sginals can be observed in the following figure.

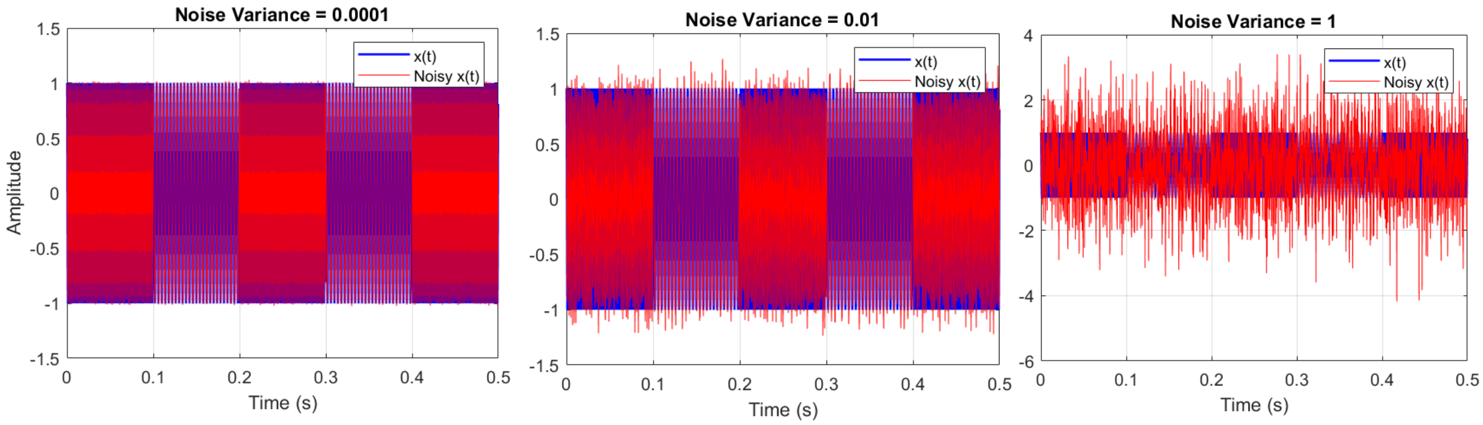
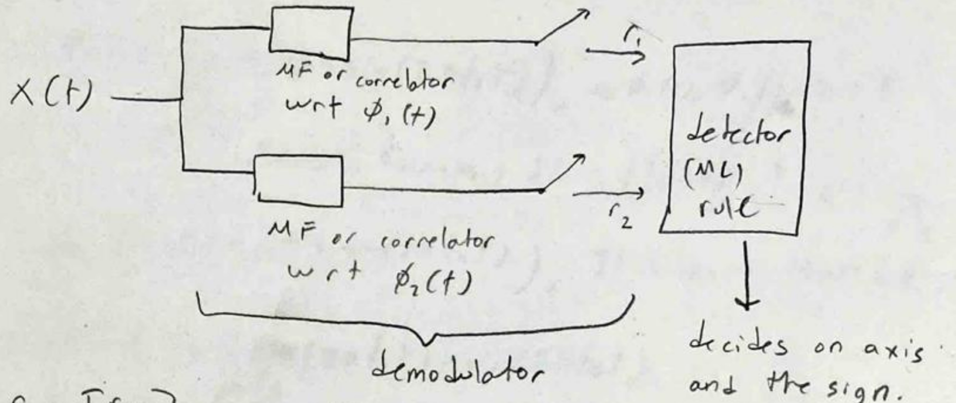


Figure 5 Noise added x(t) signals with different noise variances

(d)



$$\underline{r} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = \underline{s}_i + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}$$

ML rule $\rightarrow \underset{i \in \{1,2,3,4\}}{\operatorname{argmin}} \|\underline{r} - \underline{s}_i\|^2 = \underset{i \in \{1,2,3,4\}}{\operatorname{argmin}} \; -2\underline{r}\underline{s}_i = \underset{i \in \{1,2,3,4\}}{\operatorname{argmax}} \; \underline{r}\underline{s}_i$

For Probability of Error analysis let $T/2 = A$, assume s_1 is sent.

$$\underline{s}_1 = \begin{bmatrix} \sqrt{A} \\ 0 \end{bmatrix}, \quad \underline{s}_2 \begin{bmatrix} 0 \\ \sqrt{A} \end{bmatrix}, \quad \underline{s}_3 = \begin{bmatrix} -\sqrt{A} \\ 0 \end{bmatrix}, \quad \underline{s}_4 = \begin{bmatrix} 0 \\ -\sqrt{A} \end{bmatrix}, \quad \underline{r} = \begin{bmatrix} \sqrt{A} + n_1 \\ n_2 \end{bmatrix}$$

$$P_{e,2} = P(\underbrace{\underline{r} \cdot \underline{s}_2 > \underline{r} \cdot \underline{s}_1}_{r_2 > r_1}) = P(\sqrt{A}\, n_2 > A + n_1\sqrt{A}) = P(n_2 - n_1 > \sqrt{A})$$

$$P_{e,3} = P(\underbrace{\underline{r}\,\underline{s}_3 > \underline{r} \cdot \underline{s}_1}_{0 > r_1}) = P(-A - n_1\sqrt{A} > A + n_1\sqrt{A}) = P(n_1 < -\sqrt{A})$$

$$P_{e,4} = P(\underbrace{\underline{r}\,\underline{s}_4 > \underline{r}\,\underline{s}_1}_{r_2 < -r_1}) = P(-\sqrt{A}\, n_2 > A + n_1\sqrt{A}) = P(n_1 + n_2 < -\sqrt{A})$$

Decision Reigon $|r_2| > r_1 \cup r_1 < 0$

$$P_e = 1 - P_c, \quad P_c = P(r_1 > |r_2|)$$

$$P_c = \int_0^{\infty} P(n_1 > x - \sqrt{A}) \cdot f_{|n_2|}(x) \cdot dx$$

$$P_c = \int_0^{\infty} Q\left(\frac{x - \sqrt{A}}{\sqrt{N_0/2}}\right) \cdot \frac{\sqrt{2}}{\sqrt{\cancel{2}\pi \frac{N_0}{\cancel{2}}}} \cdot e^{-\frac{x^2}{\cancel{2} \frac{N_0}{\cancel{2}}}} \rightarrow \text{zero mean } n_2$$

$$P_e = 1 - P_c$$

e)In this part the relation between SNR and probability of symbol error is observed. Both theoratical curve and simulation curve is observed. Theoratical curve is obtained by the closed form expression of 4-ary orthagonal FSK. The probabilaty of error is implemented according to the found decision boundries in part d). Figure 6 demonstrates the Probability of symbol error vs SNR plots.

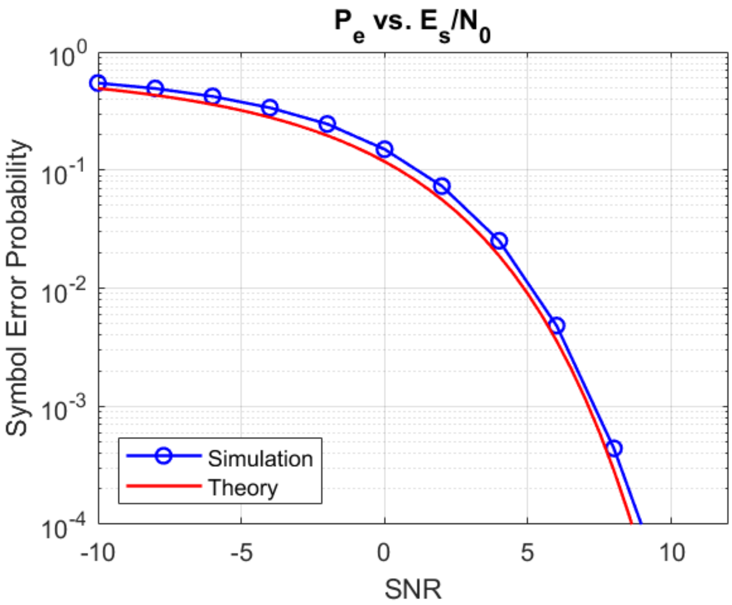


Figure 6 Probability of symbol error vs SNR plots regarding simulation and theory

The behaviour of the plot does fit to logical reasoning. As signal to noise ratio increases, signal suprasses the noise hence chances of making an error diminishes. Therefore in the plot it can be observed that as SNR increases symbol error probability decreases. Also plot verifies that the decision boundries found in part d) are precise since simulation curve tightly follows the theory curve with insignificant error.

Question 2

In this question, upto part f same analysis in Question 1is performed for BPSK modulation. Afterwards the case in which probability of transmitted signals are different is investigated

a)The randomly selected bit sequence is 1, 1, 1,0,0 and the duration of the signal is 0.05s. The sampling frequency is given as 1kHz in this question. The period of the sine wave is 0.01s meaning that for each period of a sine wave 10 samples are acquired hence a perfect smooth waveform can not be obtained. In Figure 7 x(t) can be observed.

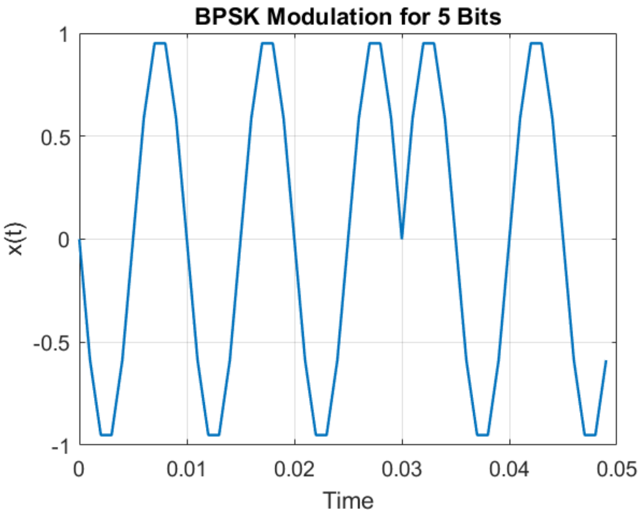


Figure 7 Demonstration of x(t) in time

b)The orthanormal basis function is derived as following. Since there is one basis function the signal lives in 1D space. Baisi function can be observed from Figure 8 and the constellation points can be observed in Figure 9.

$$E = \int_0^T \sin^2(2\pi f t)\, dt = \frac{T}{2}$$

$$\phi(t) = h(t)\sin(2\pi 100 t)\sqrt{\frac{2}{T}}$$

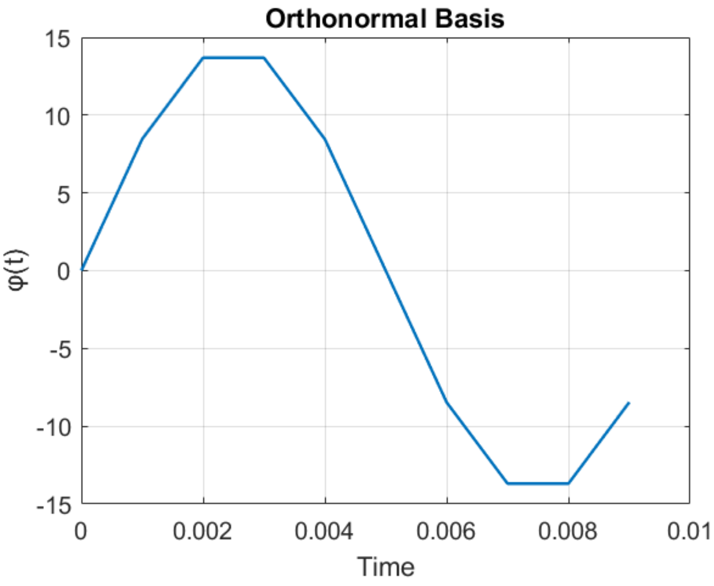


Figure 8 Basis function in time

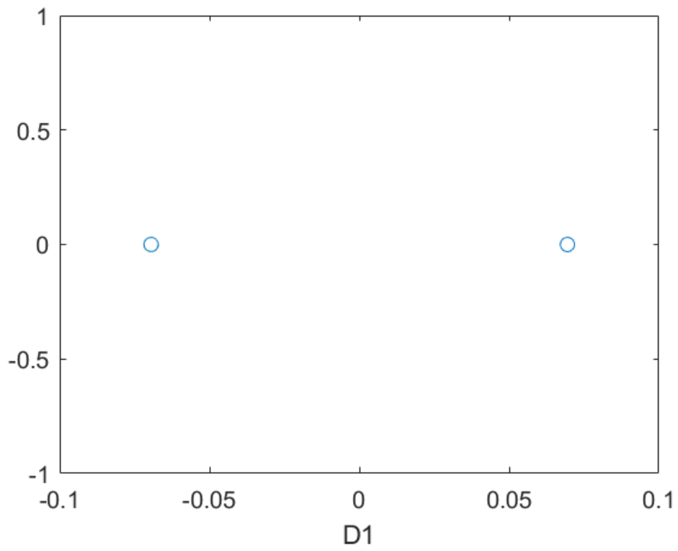


Figure 9 Demonstration of signal constellation points

c)In this part noise is added to x(t) with variances 10^{-4}, 10^{-2} and, 1. Before plotting the noise added signals influence of the samples per symbol on the SNR will be discussed. The definition of SNR does not change for different modulation schemes hence the previous argument in question 1 part c also holds for this part. The noise added signals can be observed in the following figure.

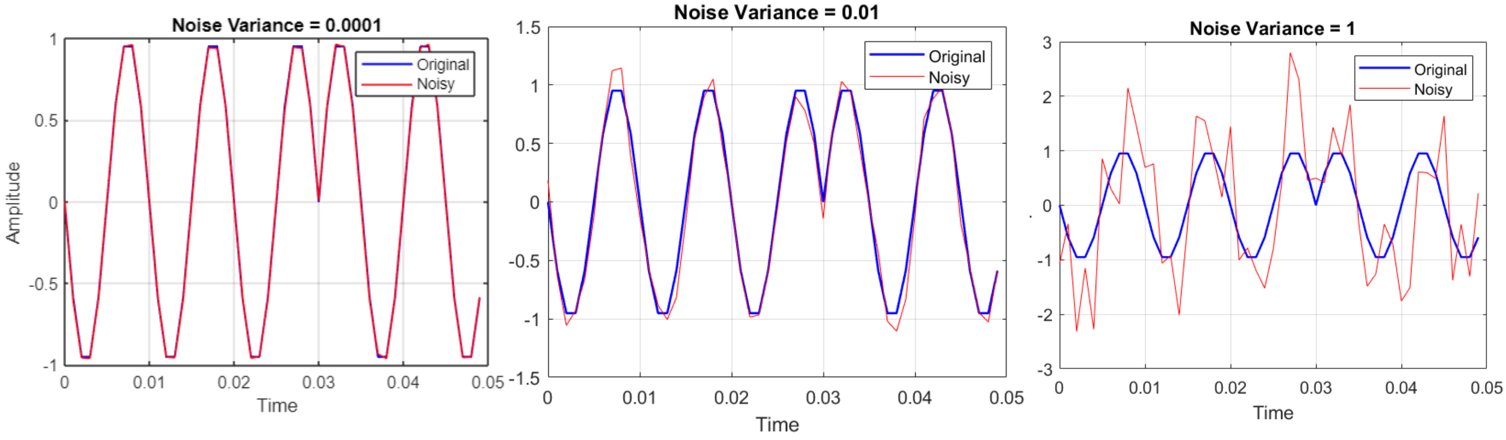
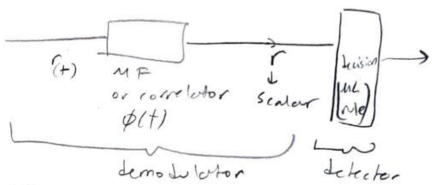


Figure 11 Noise added x(t) signals with different noise variances

d)



ML rule → $\underset{i\in\{1,2\}}{\arg\min} \|r - s_i\|^2$, $s_1 = \left[\sqrt{\frac{T}{2}}\right]$ $s_2 = \left[-\frac{\sqrt{T}}{\sqrt{2}}\right]$

Probability of error analysis, assume s_1 is sent, $r = \frac{\sqrt{T}}{\sqrt{2}} + n$

$P_e = P\left(C(r, s_2) > C(r, s_1) \right) = P\left(-\frac{T}{2} - n\sqrt{\frac{T}{2}} > \frac{T}{2} + n\sqrt{\frac{T}{2}} \right)$

or $P_e = P\left(r\cdot\left(-\frac{\sqrt{T}}{\sqrt{2}}\right) > r\cdot\sqrt{\frac{T}{2}} \right) = P(r < 0)$

$P(r<0) = P\left(z < -\frac{\sqrt{T}/\sqrt{2}}{\sqrt{N_0}}\right) = Q\left(\sqrt{2T/N_0}\right) = Q\left(\frac{\sqrt{2E_s}}{\sqrt{N_0}}\right)$

e) In this part the relation between SNR and probability of symbol error is observed. Both theoratical curve and simulation curve is observed. Theoratical curve is obtained by the closed form expression of BPSK modulation which is given below. The probabilaty of error (simualtion) is implemented according to the found decision boundries in part d). Figure 12 demonstrates the Probability of symbol error vs SNR plots.

$$P_e = Q\left(\sqrt{\frac{2E_s}{N_0}}\right)$$

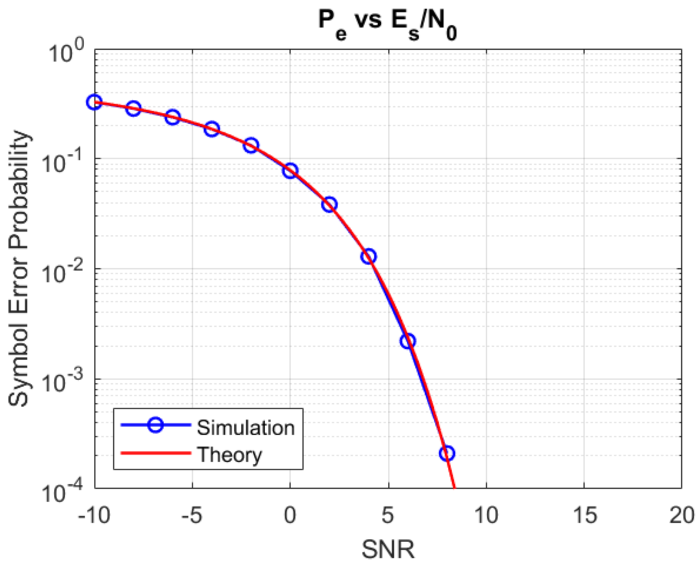


Figure 12 Probability of symbol error vs SNR plots regarding simulation and theory

The behaviour of the curve again suits the logic described in question 1 part 2. The plot also verifies that obtained decision boundry is accurate since the simulation curve tightly follow the theory curve. The 10^{-4} probability crossing of BPSK and FSK require very close SNR. Meaning that both modulation scheme gves approximately same probability of error at fixed SNR value. Check Figure 6 and Figure 12

f)Now the transmission probability of bits differ. In this case MAP rule is observed. The expressions and final decision value is obtained as following. The reciever in part d is not optimal any more since bit transmission probabilities are considered.

$$\frac{p(r|0)}{p(r|1)} > \frac{P(1)}{P(0)}$$

$$\exp\left(-\frac{\left(r-\sqrt{E}\right)^2}{2\sigma^2}+\frac{\left(r+\sqrt{E}\right)^2}{2\sigma^2}\right) > \frac{P(1)}{P(0)}$$

$$r > \frac{\sigma^2}{2\sqrt{E}} ln\left(\frac{P(1)}{P(0)}\right)$$

g)In this part various probability of error as a function of different values of bit transmission probability is observed. The relation can be observed from the Figure 13.

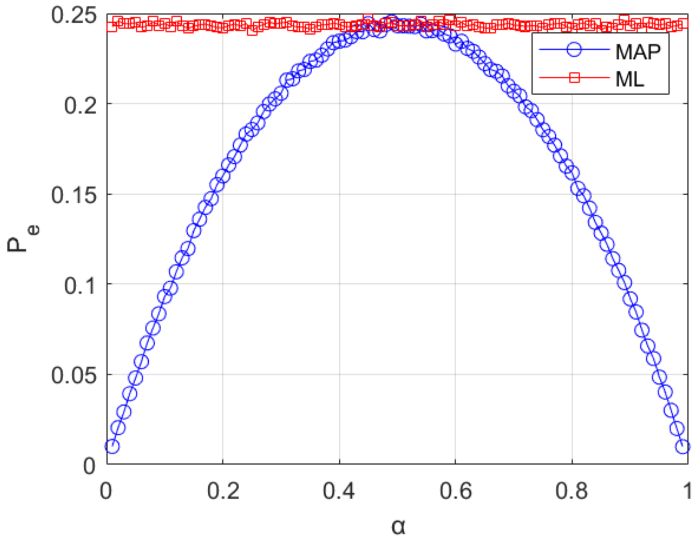


Figure 13 Probability of error as a function of bit transmission probability for ML and MAP receiver

As it can be observed the ML receiver is not able to capture the differences in probabilities since it takes only L2 norm into consideration by assuming equal probabilities. However MAP reciever outputs a new decision boundry by taking bit transmission probabilites into consideration. The MAP receiver converges to ML receiver when α=0.5 which is the case where transmission probability is the same for 2 bit modulation. Therefore the experimntaion supports the argument.

h)As it can be observed from Figure 13 the MAP receiver probability error converges to ML receiver probability error when probability of the two bits are equal. Meaning that MAP adapts to different transmission probabilities and at the worst case converges to ML receiver. So MAP receiver is more preferable. However in a case where transmission probabilities can not be defined strictly then using ML receiver is more appropriate to overcome any unexpected behaviour.

Eray Gündoğdu 22101962

Question 3

(a) $\phi_1(t) = c \cdot (\cos(2\pi f_0 t) + \sin(2\pi f_0 t))$, $\langle \phi_1(t), \phi_1(t) \rangle = 1$

$$\int_0^{T_s} \phi_1(t)^2 \, dt = c^2 \int_0^{T_s} (\cos^2 + \sin^2 + 2\sin\cos) \, dt = (T_s + 0)c^2, \quad c = \frac{1}{\sqrt{T_s}}$$

$\phi_1(t) = \frac{1}{\sqrt{T_s}} \cdot (\cos(2\pi f_0 t + \sin(2\pi f_0 t))$, It is desired that $\langle \phi_1, \phi_2 \rangle = 0$

Let $\phi_2(t) = c_2 \cdot (\cos(2\pi f_0 t) - \sin(2\pi f_0 t))$, $c_2 = \frac{1}{\sqrt{T_s}}$

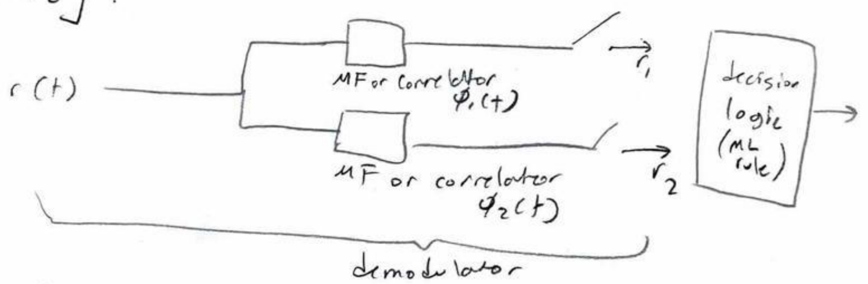
$$\langle \phi_1(t), \phi_2(t) \rangle = \frac{1}{T_s} \int_0^{T_s} (\cos^2 - \sin^2) \, dt = \frac{1}{T_s} \int_0^{T_s} \cos(2\pi . 2 f_0 t) \, dt = 0$$

Hence $\phi_2(t) = \frac{1}{\sqrt{T_s}} \cdot (\cos(2\pi f_0 t) - \sin 2\pi f_0 t))$

$\underline{s}_i = \langle s_i(t), \phi(t) \rangle$, $s_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $s_2 = \begin{bmatrix} -\sqrt{\frac{T}{2}} \\ -\sqrt{T/2} \end{bmatrix}$, $s_3 = \begin{bmatrix} \sqrt{T}/2 \\ -\sqrt{T}/2 \end{bmatrix}$

$s_4 = \begin{bmatrix} \sqrt{T}/2 \\ \sqrt{T}/2 \end{bmatrix}$.

(b)



ML rule $\rightarrow \underset{i \in \{1,2,3,4\}}{\arg\min} \| r - s_i \|^2$. ML rule is determined so that L2 norm, euclidian distance is minimized in signal space.

(c) $d_{min} = \sqrt{T_s}$, the loose bound $\rightarrow P_e \le \underbrace{(M-1)}_{M-1=3} \cdot Q\left(\frac{\sqrt{T_s}}{\sqrt{2}\sqrt{N_0}}\right)$.

The exact union bound is $\longrightarrow P_e \le \frac{1}{4} \cdot \sum_{i=1}^{4} \sum_{\substack{j=1 \\ i \ne j}}^{4} Q\left(\frac{d_{ij}}{\sqrt{2N_0}}\right)$

$$P_e \le \frac{1}{4}\left(\underbrace{3Q\left(\frac{\sqrt{T_s}}{\sqrt{2N_0}}\right)}_{\text{from } s_1} + \underbrace{2\left(2Q\left(\frac{\sqrt{T_s}}{\sqrt{2N_0}}\right) + Q\left(\frac{\sqrt{T_s}}{2\sqrt{N_0}}\right)\right)}_{\text{from } s_2 \text{ and } s_3} + \underbrace{3Q\left(\frac{\sqrt{T_s}}{\sqrt{2N_0}}\right)}_{\text{from } s_4}\right)$$

(d) $s_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $s_2 = \begin{bmatrix} -\sqrt{T}/2 \\ -\sqrt{T}/2 \end{bmatrix}$, $s_3 = \begin{bmatrix} \sqrt{T}/2 \\ -\sqrt{T}/2 \end{bmatrix}$, $s_4 = \begin{bmatrix} \sqrt{T}/2 \\ \sqrt{T}/2 \end{bmatrix}$

$\underline{r} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} n_1 + 0 \\ n_2 + 0 \end{bmatrix}$, $\quad \|r - s_1\|^2 < \|r - s_2\|^2$ (1)

$-\frac{\sqrt{T}}{2} r_1 - \frac{\sqrt{T} r_2}{2} > 0$

$0 > r_1 + r_2$

$\|r - s_1\|^2 < \|r - s_3\|^2$ (2) $\quad$, $\quad \|r - s_4\|^2 > \|r - s_1\|^2$ (3)

$r_1 - r_2 > 0 \qquad\qquad r_1 + r_2 > 0$

Hence, $P_{e|s_1} = P(\{r_1 + r_2 > 0\} \cup \{r_1 - r_2 > 0\} \cup \{r_1 + r_2 < 0\})$

Note that $\{r_1 + r_2 > 0\} \cup \{r_1 + r_2 < 0\} = \mathbb{R}^2$ and $P(r_1 + r_2 > 0) = \frac{1}{2}$

$P(r_1 + r_2 < 0) = \frac{1}{2}$. As a result, $P(\text{error} | s_1) = 1$.

APPENDIX

MATLAB Codes for Q1

```
Fs = 5000;
T  = 0.1;
Ts = 1/Fs;
N  = T*Fs
time_vect = (0:N-1)*Ts;
s1 =  cos(2*pi*250*time_vect);
s2 =  cos(2*pi*500*time_vect);
symbols = [2, 0 ,3 ,0,2];
x_t= zeros(1,5*N);
%disp(x_t);
disp(symbols)
for k = 1:5
    idx = symbols(k);
    switch idx
        case 0
            segment =  s1;
        case 1
            segment = -s1;
        case 2
            segment =  s2;
        case 3
            segment = -s2;
    end
    first_idx=(k-1)*N + 1;
    last_idx =k*N;
    x(first_idx:last_idx)=segment;
end
t_plot = (0:length(x_t)-1)*Ts;
figure;
plot(t_plot, x, 'LineWidth', 1.2);
xlabel('Time (s)');
ylabel('x(t)');
title('FSK-Modulation with 5 Randomly Generated Symbols');
grid on;
%%part b
t_demo = (0:N-1)*Ts;
basis1 = sqrt(2/T)*cos(2*pi*250*t_demo);
basis2 = sqrt(2/T)*cos(2*pi*500*t_demo);
inner1 = trapz(t_demo, basis1.*basis1);
inner2 = trapz(t_demo, basis2.*basis2);
inner3 = trapz(t_demo, basis1.*basis2);
fprintf('⟨phi1,phi1⟩=%.3f, ⟨phi2,phi2⟩=%.3f, ⟨phi1,phi2⟩=%.3e\n', inner1,
inner2, inner3);
figure;
subplot(2,1,1);
plot(t_demo, phi1, 'LineWidth',1.2);
xlabel('t (s)'); ylabel('\phi_1(t)');
title('Basis 1');
```

```
grid on;
subplot(2,1,2);
plot(t_demo, phi2, 'LineWidth',1.2);
xlabel('t (s)'); ylabel('\phi_2(t)');
title('Basis 2');
grid on;
%%consellation points in 2D space
E = sqrt(T/2);
constellation = [ E,  0;
          -E,  0;
           0,  E;
           0, -E ];
labels = {'s_1','-s_1','s_2','-s_2'};


figure; hold on;
scatter(constellation(:,1), constellation(:,2), 80, 'filled');
for i=1:4
    text(constellation(i,1)+0.01,constellation(i,2)+0.01, labels{i});
end
xlabel('D_1'); ylabel('D_2');
title('Signal Constellation Points');
axis equal tight; grid on;
%%part c
noise_vars = [1e-4, 1e-2, 1];


for i = 1:length(noise_vars)
    var_n = noise_vars(i);
    noise_gauss = sqrt(var_n)*randn(size(x));
    x_noisy = x + noise_gauss;
    figure;
    plot(t_plot, x,    'b', 'LineWidth',1.2); hold on;
    plot(t_plot, x_noisy, 'r');
    xlabel('Time (s)');
    ylabel('Amplitude');
    title(sprintf('Noise Variance = %g', var_n));
    legend('x(t)','Noisy x(t)');
    grid on;
end


SNR_db = -10:2:12;
SNR_sim    = 10.^(SNR_db/10);
SNR_db_theo  = -10:0.5:20;
SNR_th     = 10.^(SNR_db_theo/10);
number_symbols = 1e5;
Es      = T/2;
Eb      = Es/2;


sig_space = [ E,  0;
```

```
           -E,  0;
            0,  E;
            0, -E];


sim_Pe     = zeros(size(SNR_sim));
theory_Pe = (4-1)/4 * erfc(sqrt(SNR_th));
for ii = 1:length(SNR_sim)
    N0     = Eb / SNR_sim(ii);
    sigma = sqrt(N0/2);
    tx = randi([0 3],1,number_symbols);
    rx = zeros(size(tx));
    for k = 1:number_symbols
        a = sig_space(tx(k)+1,:);
        r_vect = a + sigma*randn(1,2);
        if abs(r_vect(2)) < abs(r_vect(1))
            if 0<r_vect(1), rx(k)=0; else rx(k)=1; end
        else
            if r_vect(2)>0, rx(k)=2; else rx(k)=3; end
        end
    end
    sim_Pe(ii) = mean(rx~=tx);
end
sim_Pe(sim_Pe==0) = 1/number_symbols;
figure;
semilogy(SNR_db, sim_Pe,    'bo-','LineWidth',1.2); hold on;
semilogy(SNR_db_theo,  theory_Pe, 'r-','LineWidth',1.2);
xlabel('SNR');
ylabel('Symbol Error Probability');
legend('Simulation','Theory','Location','southwest');
grid on;
title('P_e vs. E_s/N_0');
ylim([1e-4 1]);
xlim([SNR_db(1), SNR_db(end)]);
```

MATLAB Codes for Q2

```
Fs=1000;
T=0.01;
Ts=1/Fs;
N=T*Fs;
time_vect=(0:N-1)*Ts;
signal1=sin(2*pi*100*time_vect);
bits=randi([0 1],1,5);
disp(bits)
%bits=[1,1,1,0,0];
x=zeros(1,5*N);
for k=1:5
    if bits(k)==0
        segment=signal1;
    else
```

```
        segment=-signal1;
    end
    start_idx=(k-1)*N+1;
    end_idx=k*N;
    x(start_idx:end_idx)=segment;
end


t_plot=(0:length(x)-1)*Ts;
figure;
plot(t_plot,x,'LineWidth',1.2);
xlabel('Time');
ylabel('x(t)');
title('BPSK Modulation for 5 Bits');
grid on;
E=trapz(time_vect,s.^2);
basis_func=signal1/sqrt(E);
figure;
plot(time_vect,phi,'LineWidth',1.2);
xlabel('Time');
ylabel('ϕ(t)');
title('Orthonormal Basis');
grid on;
constellation=[sqrt(E), -sqrt(E)];
figure;
plot(constellation,0*constellation,'o');
xlabel('D1');
ylabel('');
noise_vars=[1e-4,1e-2,1];
for i=1:length(noise_vars)
    current_var=noise_vars(i);
    noise_gaussian=sqrt(current_var)*randn(size(x));
    x_noisy=x+noise_gaussian;
    figure;
    plot(t_plot,x,'b','LineWidth',1.2);
    hold on;
    plot(t_plot,x_noisy,'r');
    xlabel('Time');
    ylabel('Amplitude');
    title(sprintf('Noise Variance = %g',current_var));
    legend('Original','Noisy');
    grid on;
end
SNRdb_sim=-10:2:20;
SNR_sim=10.^(EbN0_dB_sim/10);
SNRdb_th=-10:0.5:20;
SNR_th=10.^(EbN0_dB_th/10);
numBits=1e5;
simulation_error=zeros(size(SNR_sim));
for ii=1:length(SNR_sim)
    N0=E/SNR_sim(ii);
    var_noise=sqrt(N0/2);
```

```
    tx=randi([0 1],1,numBits);
    a=(1-2*tx)*sqrt(E);
    received_vect=a+var_noise*randn(size(a));
    rx=received_vect<0;
    simulation_error(ii)=mean(rx~=tx);
end
theoratical_error=0.5 * erfc( sqrt(SNR_th) );
figure;
semilogy(SNRdb_sim,simulation_error,'bo-','LineWidth',1.2);
hold on;
semilogy(SNRdb_th,theoratical_error,'r-','LineWidth',1.2);
xlabel('SNR');
ylabel('Symbol Error Probability');
legend('Simulation','Theory','Location','southwest');
grid on;
title('P_e vs E_s/N_0');
ylim([1e-4 1]);
xlim([SNRdb_sim(1),SNRdb_sim(end)]);
E=trapz(t,s.^2);
var_n=1e-2;std=sqrt(var_n);
alpha=0.01:0.01:0.99;
symbol_num=1e5;
map_rule=zeros(size(alpha));
ml_rule=zeros(size(alpha));
for i=1:length(alpha)
    prob=rand(1,symbol_num)<alpha(i);
    tx=(1-2*prob)*sqrt(E);
    received_vector=tx+std*randn(1,symbol_num);
    boundry=(var_n/(2*sqrt(E)))*log(alpha(i)/(1-alpha(i)));
    map_received=received_vector<=boundry;
    map_rule(i)=mean(map_received~=prob);
    ml_received=received_vector<0;
    ml_rule(i)=mean(ml_received~=prob);
end
figure
plot(alpha,map_rule,'b-o',alpha,ml_rule,'r-s')
xlabel('α');ylabel('P_e');legend('MAP','ML')
grid on
```