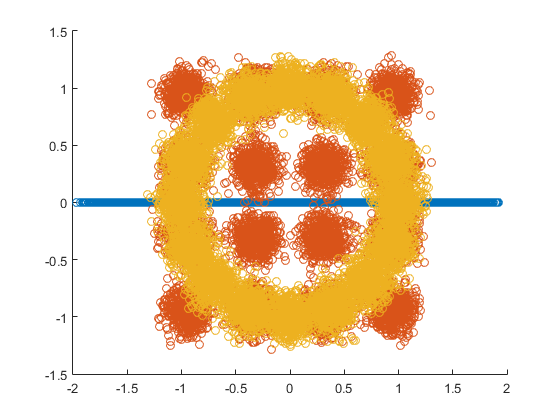
Homework 3 Jeffrey Jiang

EE 132A ID: 904255069

Problem 5:

For the code of this problem, we mainly deal with 5 main parts.

1. Randomly choose message to send in a random vector, which we call H. Since we assume that each of the 16 messages is equiprobable, we can just use the uniform random variable to determine which of the messages to send. This procedure is the same for each of the three encoding schemes.
2. Encode the chosen message to the point symbol that is represented by the message number chosen. Each of these is different for the different encoding schemes:
   1. For the PAM, just shift the message to be centered around 0 and scale to fit the energy profile:
   2. For the SPK, the message determines the angle .
   3. For the QAM, the symbols will just be arranged from left to right, then from top to bottom. The first symbol is the top left and the second would be to the right of the first point.
3. Generate the normal distributions and add it to the symbol we chose. This will be the received signal to the decoder.
4. Decode the received signal by using the ML rule. For this part, I loop through every point to determine the closest point and the corresponding message that would be decoded from the point.
5. Determine the probability of error by comparing the difference between the original messages, H, and the newly generated decoded messages. Note this value.

The entirety of the code will be shown at the end of this report. The errors in one instance of the code would result in error of the following.

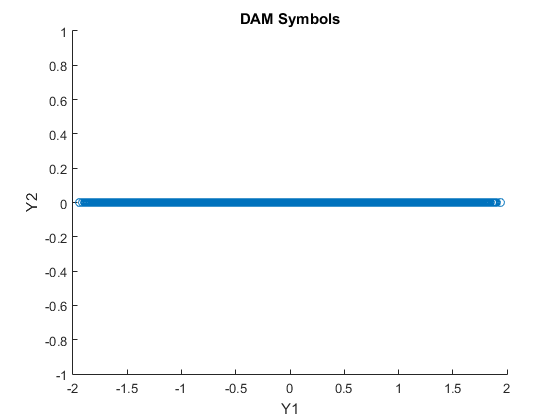
The probability of error in PAM is 0.258900

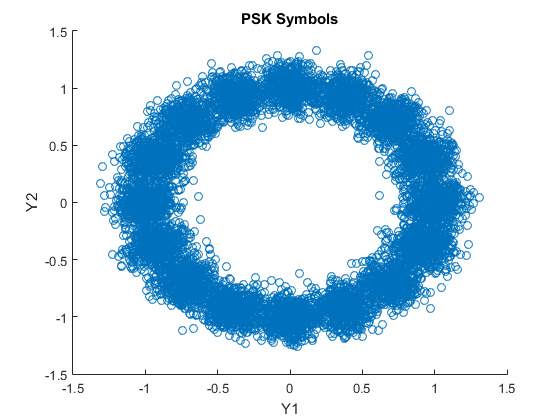
The probability of error in QAM is 0.001900

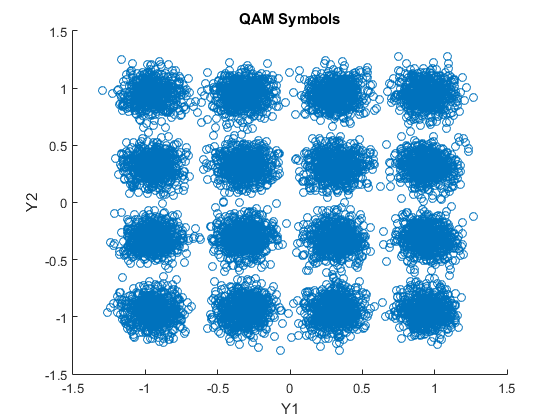
The probability of error in PSK is 0.053200

Notice that the QAM has the lowest probability of error. This is largely because the distance between points is very large, even though the average number of neighbors is higher. The Q function falls off close to exponentially, and the number of neighbors is only larger by about double. In another word, the area of possible points is a lot higher in the QAM and very small for the PAM, which shows in the error probability. This can be shown in the graph that shows all three encoding schemes at the same time shown above (where blue represents PAM, orange represents QAM, and yellow represents PSK).

The three distributions are shown as follows.







**CODE**

E = 1;

DPAM = 0.21963;

DQAM = sqrt(0.4);

M = 16;

% random PAM variable.

sig = .1;

x\_pts\_PAM = -7.5\*DPAM:DPAM:7.5\*DPAM;

N = 10000;

H = ceil(16\*rand(1, N));

V1 = (H - 8.5) \* DPAM;

V2 = zeros(1, N);

Z = randn(1, N) \* sig;

Y1 = V1 + Z;

%plot scatterplot.

figure;

scatter(Y1, V2);

title('DAM Symbols');

xlabel('Y1');

ylabel('Y2');

%hold on;

decode = H;

%determine each decoded value using ML rule.

for i = 1:N

min = 1;

dist = abs(Y1(i) - x\_pts\_PAM(1));

for j = 2:M

if abs(Y1(i) - x\_pts\_PAM(j)) < dist

min = j;

dist = abs(Y1(i) - x\_pts\_PAM(j));

end

end

decode(i) = min;

end

%finding the probability of error.

p\_error = mean(decode ~= H);

PAM\_msg = sprintf('The probability of error in PAM is %f', p\_error);

disp(PAM\_msg);

%random QAM variable.

%since it doesn't actually matter too much,

%we will use the same H vector.

x\_pts\_QAM = -1.5\*DQAM:DQAM:1.5\*DQAM;

y\_pts\_QAM = x\_pts\_QAM;

V1\_QAM = (mod(H - 1, 4) - 1.5) \* DQAM;

V2\_QAM = (floor((H - 1)/ 4) - 1.5) \* DQAM;

Z1 = randn(1, N) \* sig;

Z2 = randn(1, N) \* sig;

Y1\_QAM = V1\_QAM + Z1;

Y2\_QAM = V2\_QAM + Z2;

figure;

scatter(Y2\_QAM, Y1\_QAM);

title('QAM Symbols');

xlabel('Y1');

ylabel('Y2');

%hold on;

decode\_QAM = H;

%determine each decoded value using ML rule.

for i = 1:N

min\_x = 1;

min\_y = 1;

dist = (Y1\_QAM(i) - x\_pts\_QAM(1))^2 + (Y2\_QAM(i) - y\_pts\_QAM(1))^2;

for j = 1:4

for k = 1:4

if (Y1\_QAM(i) - x\_pts\_QAM(j))^2 + (Y2\_QAM(i) - y\_pts\_QAM(k))^2 < dist

min\_x = j;

min\_y = k;

dist = (Y1\_QAM(i) - x\_pts\_QAM(j))^2 + (Y2\_QAM(i) - y\_pts\_QAM(k))^2;

end

end

end

decode\_QAM(i) = 4\*(min\_y-1) + min\_x;

end

%finding the probability of error.

p\_error\_QAM = mean(decode\_QAM ~= H);

QAM\_msg = sprintf('The probability of error in QAM is %f', p\_error\_QAM);

disp(QAM\_msg);

%random PSK variable.

angles = 0:2\*pi/M:(M-1)\*2\*pi/M;

x\_pts\_PSK = sqrt(E) \* cos(angles);

y\_pts\_PSK = sqrt(E) \* sin(angles);

V1\_PSK = sqrt(E) \* cos((H-1) \* 2\*pi / M);

V2\_PSK = sqrt(E) \* sin((H-1) \* 2\*pi / M);

Z1 = randn(1, N) \* sig;

Z2 = randn(1, N) \* sig;

Y1\_PSK = V1\_PSK + Z1;

Y2\_PSK = V2\_PSK + Z2;

figure;

scatter(Y2\_PSK, Y1\_PSK);

title('PSK Symbols');

xlabel('Y1');

ylabel('Y2');

decode\_PSK = H;

%determine each decoded value using ML rule.

for i = 1:N

min = 1;

dist = (Y1\_PSK(i) - x\_pts\_PSK(1))^2 + (Y2\_PSK(i) - y\_pts\_PSK(1))^2;

for j = 1:M

if (Y1\_PSK(i) - x\_pts\_PSK(j))^2 + (Y2\_PSK(i) - y\_pts\_PSK(j))^2 < dist

min = j;

dist = (Y1\_PSK(i) - x\_pts\_PSK(j))^2 + (Y2\_PSK(i) - y\_pts\_PSK(j))^2;

end

end

decode\_PSK(i) = min;

end

%finding the probability of error.

p\_error = mean(decode\_PSK ~= H);

PSK\_msg = sprintf('The probability of error in PSK is %f', p\_error);

disp(PSK\_msg);