4-PAM Digital Communication System Simulation

Communications & Networks Assignment

Abstract

This report describes the implementation of a vector model Digital Communication System simulation. It provides an estimation of the error probability of a Four-Level Pulse Amplitude Modulation (4-PAM) system, where the receiver is a single correlation-type detector. Conclusions are obtained from a comparison plot between the expected and obtained data.

30/04/2017

Eva Esteban

Introduction

This project consists of a simulation of a DCS vector model which implements 4-PAM technique. It includes a plot of the simulated probability of bit error as a function of the signal-to-noise ratio per bit compared to the expected theoretical values. The detector is assumed to be a single correlation-type demodulator.

1. Block diagram of the transmitter and receiver

The following diagram illustrates the different parts of a Digital Communications System. It consists of a transmitter, a communications channel and a receiver, and can implement 4-PAM.

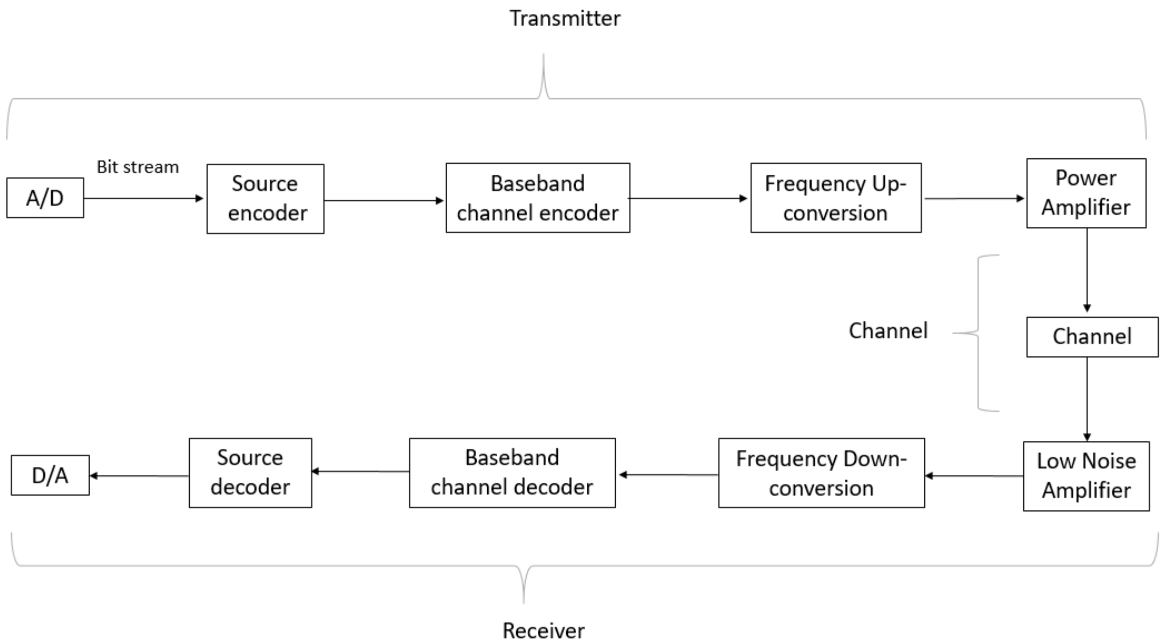

Figure 1: Block diagram of Digital Communications System

The baseband channel, which is capable of transmitting frequencies that are near zero, consists of a set of elements whose function and block diagram is provided below.

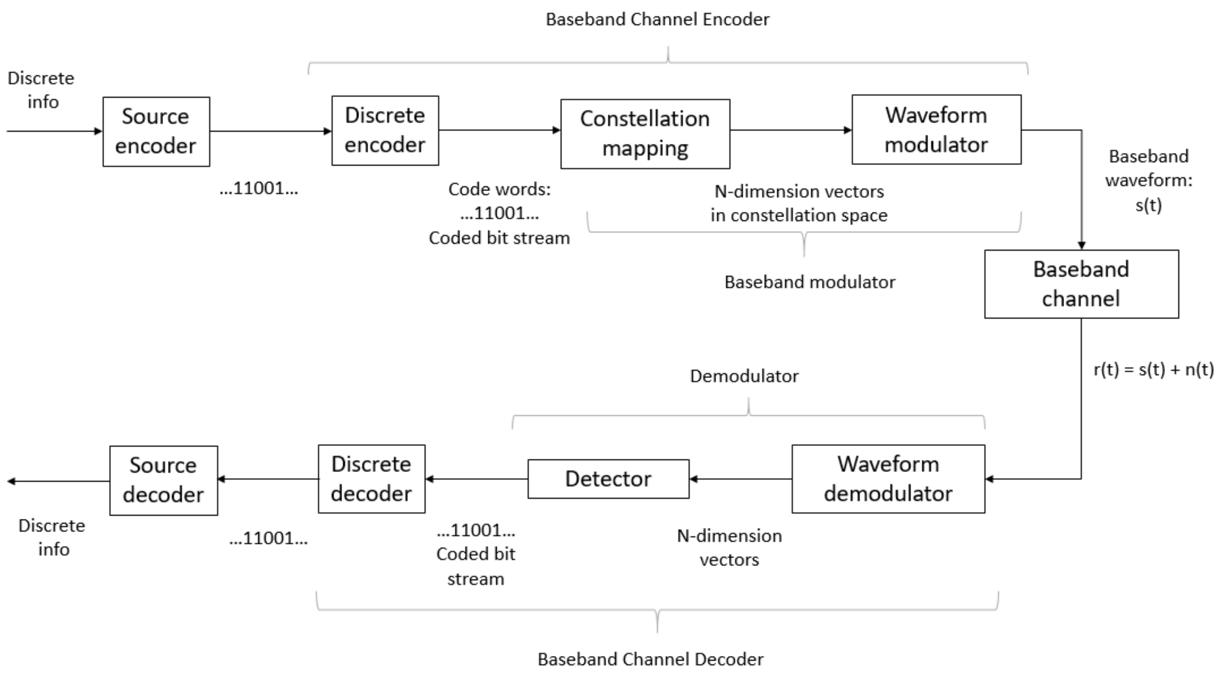

Figure 2: Detailed block diagram of Digital Communications System

The receiver in this DCS system consists of a single correlation-type detector described in **Figure 3**.

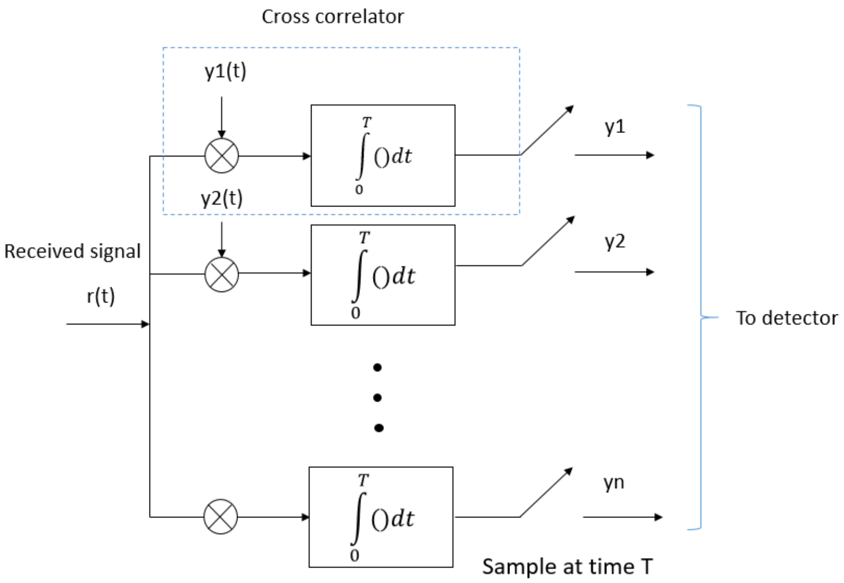

Figure 3: Correlation-type demodulator block diagram

2. Constellation diagram of the system

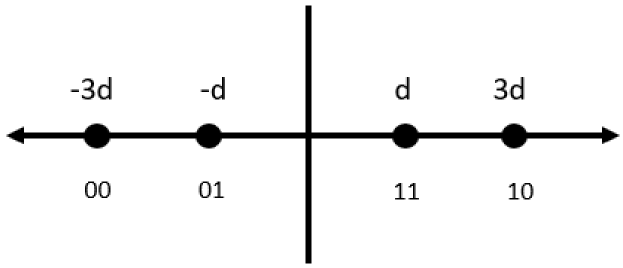

Figure 4: Constellation diagram of 4-PAM proposed system

This constellation diagram represents a 4-PAM digital modulation system. It represents the possible symbols that may be selected in this scheme. Each symbol consists of a pair of bits and an assigned constellation level, which is a function of d and represents the energy of each symbol.

3. Optimum detector function

An optimum detector determines the most likely transmitted signal given the observed vector. The most likely transmitted signal, following maximum a posteriori (MAP) probability detection will be the one that maximises the criterion P(Sm was transmitted | R = r), which represents the probability that the signal Sm was transmitted given that r is the received vector. Using Bayes' theorem we can derive $P(Sm|\mathrm{R}=\mathrm{r})=\frac{f_R(r|\,\mathrm{Sm})P(Sm)}{f_R(r)}$ where $f_R(r|\,\mathrm{Sm})$ is the conditional PDF of the observed vector (likelihood function), $P(Sm)$ is the a-priori probability of the m-th signal and $f_R(r)$ is the PDF of the observed vector. Since $f_R(r)$ is independent of Sm, the decision criteria depends on the a-priori probability and the likelihood function.

In this particular Digital Communication System the information source generates two bits per transmission, with equal probability for bit 0 and bit 1. Each transmission is then mapped to a constellation level specified by the following table:

Bit pair	Constellation level
00	-3d
01	-d
11	d
10	3d

Figure 5: Bit mapping to constellation levels

The symbols are then transmitted through an AWGN (Additive White Gaussian Noise) channel. The detector compares each level of the noisy signal obtained to the middle value between the two constellation levels that are closer to it. It will assign it the level that is closer to it. For instance, if the value of the signal after being transmitted through the channel is 2.1d, the value will be compared with 2d, which is the middle point between the constellation values d and 3d. Since 2.1d is bigger than 2d, the detector will consider 2.1d as 3d.

4. Average SNR per bit as a function of d and variance.

The Signal-to-noise ratio of a signal is a measure that compares the level of the signal to the amount of noise and it is given by: $SNR = \frac{Average\ Signal\ Power}{Noise\ Power} = \frac{Pav}{Pn}$ [1]. The average SNR per bit is, therefore, $SNR\ per\ bit = \frac{\frac{Eb}{T}}{No} = \frac{Eb}{No}$ [2], where *Eb* is the average energy per bit, *T* is the period which equals 1 and *No* is the power spectral density of noise.

The energy per bit to *No* is given by $\frac{Eb}{No}$ and the energy per symbol to *No* is $\frac{Es}{No}$. The relationship between them when using gray code encoding is $\frac{Es}{No} = \frac{Eb}{No}\log_2 M$ with *M* = 4 in this case, which is the number of possible modulation symbols in 4-PAM, so $\frac{Es}{No} = \frac{2Eb}{No}$ [3]. Substituting this result in [2] we obtain $SNR\ per\ bit = \frac{Es}{2No}$ [4]

Since there are four symbols with equal probability, which is 1/4, the average energy per symbol is given by the sum of each symbol's energy by its probability: $Es = \frac{1}{4} \times (3d)^2 + \frac{1}{4} \times (d)^2 + \frac{1}{4} \times (-3d)^2 + \frac{1}{4} \times (-d)^2 = 5d^2$ [5]. In addition, the variance of AWGN noise in the channel is related to the power spectral density of noise as follows: $\sigma^2 = \frac{No}{2} \rightarrow No = 2\sigma^2$ [6]. Substituting this in [4] we obtain the average SNR per bit as: $SNR\ per\ bit = \frac{5d^2}{4\sigma^2}$ [7].

To run the simulation for different SNR values, d is assumed to equal 1 and the formula $SNR\ per\ bit = \frac{5}{4\sigma^2}$ [8] is used to calculate different AWGN variance values.

5. Probability of bit error as a function of average SNR per bit.

The output symbol of a PAM communications system is given by $r = Sm + n$ where *Sm* is the signal transmitted and *n* the added noise with zero mean and variance σ. *Sm* can be expressed as $Sm = \sqrt{Eg}Am$ with *Eg* being the average energy per signal pulse and *Am* the amplitude.

The probability of error is the probability that the detector will make a wrong guess, which occurs when the noisy signal level is higher than the middle point between the two symbols in question. Since all symbols are equally probable, the probability of symbol error is given by $Ps = \frac{3}{4} \times P(|r - Sm| < \sqrt{Eg})$ [9]. The Probability Density Function (PDF) of noise follows a Gaussian distribution: $f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$, so $P\left(|r - Sm| < \sqrt{Eg}\right) = \frac{2}{\sqrt{\pi No}} \int_{\sqrt{Eg}}^{\infty} e^{-\frac{x^2}{No}} dx = \frac{2}{\sqrt{2\pi}} \int_{\sqrt{2Eg/No}}^{\infty} e^{-\frac{x^2}{2}} dx$ [10] . Since Q-function equals $Q(x) = \frac{1}{\sqrt{2\pi}} \int_{x}^{\infty} e^{-\frac{u^2}{2}} dx$, we obtain

$Ps = \frac{6}{4} Q\left(\sqrt{\frac{2Eg}{No}}\right) = \frac{3}{2} Q\left(\sqrt{\frac{2Eg}{No}}\right)$ [11] . The average power is given by $Pav = \frac{Average\ Energy}{T} = \frac{5Eg}{T}$ [12]. Substituting this in [11] we obtain $Ps = \frac{3}{2} Q\left(\sqrt{\frac{2PavT}{5No}}\right) = \frac{3}{2} Q\left(\sqrt{\frac{2Eav}{5No}}\right)$ [13] as the probability of symbol error, where $Eav = PavT$ is the average energy. This can be expressed in terms of the average energy per bit *Eb using Eav =* *Eb* $\log_2 M$ with *M = 4,* giving $Ps = \frac{3}{2} Q\left(\sqrt{\frac{4Eb}{5No}}\right)$ [14]. The probability of bit error *Pb* is related to *Ps* by $Ps = Pb \log_2 M$ with *M* = 4 so $Pb = \frac{Ps}{2} = \frac{3}{4} Q\left(\sqrt{\frac{4Eb}{5No}}\right)$ [15] .

Thus, the probability of bit error as a function of average SNR per bit is $Pb = \frac{3}{4} Q\left(\sqrt{\frac{4}{5}\ SNR\ per\ bit}\right)$ [16].

6. Simulation and theoretical plots

A 4-PAM digital communications system subject to the above-mentioned derivations was implemented using MATLAB software. The simulation was run for 10,000 transmissions each time, for the average SNR per bit equal to 0, 2, 4, 6, 8, 10, 12 and 15 dB. The SNR value used for the simulations is linear, obtained from the formula $SNRLinear = 10SNR^{10}$, and then plotted in a logarithmic plot. The following table shows the theoretical values for the probability of bit error Pb compared to the simulated ones.

SNR (dB)	Theoretical Pb	Simulated Pb
0	0.139200	0.139500
2	0.097600	0.098200
4	0.058600	0.058900
6	0.027900	0.028600
8	0.009200	0.009500
10	0.001800	0.001500
12	0.000130	0.000130
15	0.000010	0.0000001

Figure 6: Bit error probability for 4-PAM

The values obtained in the simulation are very close to the theoretical ones, which can be observed on the graph below.

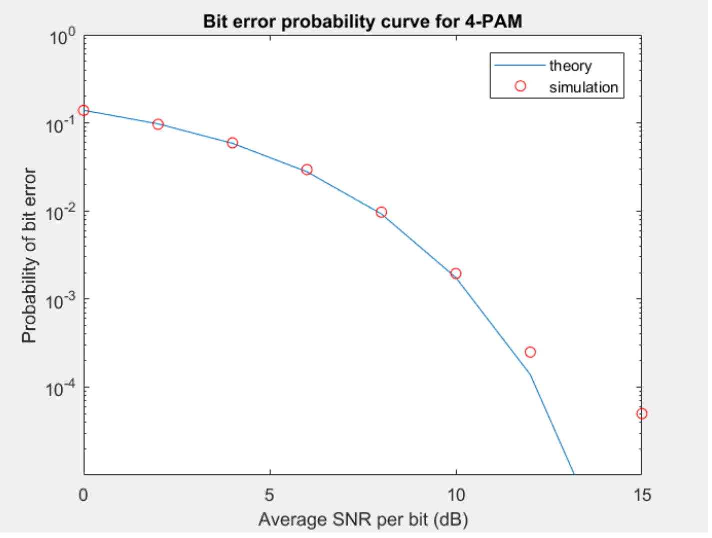

Figure 7: Bit error probability curve for 4-PAM plot

7. MATLAB 4-PAM system simulation code.

```
  % 4-PAM Digital Communication System Modelling

numTrans = 10000; % Number of transmissions
input = zeros(1, 2*numTrans); % Matrix to store, in this case, 10,000 transmissions of 2
random bits each
encodedInput = zeros(1, numTrans); % Matrix to store contellation mapping values of
10,000 pair of input bits
encodedOutput = zeros(1, numTrans); % Matrix to store values after transmission through
noisy channel
output = zeros(1, 2*numTrans); % Matrix to store pairs of bits obtained after decoding
snr = [0 2 4 6 8 10 12 15]; % 8 different values of SNR in dB
pBitError = zeros(1,8); % Probability of bit error for each value of SNR
d = 1; % Assume d = 1
simulatedSER = zeros(1,8); % Simulated probability of bit error for each value of SNR
simulatedBER = zeros(1,8); % Simulated probability of symbol error for each value of SNR
theorySER = zeros(1,8); % Derived probability of bit error for each value of SNR
theoryBER = zeros(1,8); % Derived probability of symbol error for each value of SNR

for s = 1:8

  i = 1; % Reset variables to iterate through the matrices
  j = 1;
  k = 1;
  n = 1;
  q = 1;
  errorBits = 0; % Variable to count the number of error bits after decoding
  errorSymbols = 0; % Variable to count the number of symbol errors

  snrLinear = 10^(snr(s)/10); % Linear value of each SNR value

  % Information source modelling

  while i <= 2*numTrans % Input random bit combinations in matrix

    twoBits = round(rand(1, 2)); % Random bit generator for a pair of bits with equal
probability for 0 and 1
    input(i:i+1) = twoBits;
    i = i + 2;
  end

  % Encoding

  while k < numTrans % Iterate through input sequence of transmissions
```

```
        if input(j:j+1) == [0 0] % Assign appropriate constellation value to pair of bits
           inputSymbol = -3*d;
        elseif input(j:j+1) == [0 1]
           inputSymbol = -d;
        elseif input(j:j+1) == [1 1]
           inputSymbol = d;
        elseif input(j:j+1) == [1 0]
           inputSymbol = 3*d;
        end

        j = j + 2;
        encodedInput(k) = inputSymbol; % Store obtained values in appropriate matrix
        k = k + 1;

    end

    v = 5 / (4*snrLinear);

    % Transmission over AWGN channel

    noiseSamples = sqrt(v)*randn(1,numTrans); % Generate as many samples of real (I-axis)
AWGN as the number of transmissions
    noisyOutput = encodedInput + noiseSamples; % Add noise to the signal transmitted over
the channel

    % Detector modelling

    for m = 1:numTrans % Iterate through noisy signal values and assign appropriate level to
noisy constellation value

        if noisyOutput(m)>0 % Positive values decoding

           if noisyOutput(m)>3*d
             outputSymbol = 3*d;
           elseif(noisyOutput(m)<3*d) && (noisyOutput(m)>2*d)
              outputSymbol = 3*d;
           else
              outputSymbol = d;
           end

        elseif noisyOutput(m)<0 % Negative values decoding
```

```
        if noisyOutput(m)<-3*d
            outputSymbol = -3*d;
        elseif (noisyOutput(m)>-3*d) && (noisyOutput(m)<-2*d)
            outputSymbol = -3*d;
        else
            outputSymbol = -d;
        end
    end

    encodedOutput(m) = outputSymbol; % Store obtained value in appropriate matrix
end

% Decoding

while n < numTrans % Iterate through noisy output constellation values

    if encodedOutput(n) == 3*d % Assign appropriate pair of bits to each value
        output(q:q+1) = [1 0];
    elseif encodedOutput(n) == d
        output(q:q+1) = [1 1];
    elseif encodedOutput(n) == -d
        output(q:q+1) = [0 1];
    elseif encodedOutput(n) == -3*d
        output(q:q+1) = [0 0];
    end

    q = q + 2;
    n = n + 1;

end

% Determine symbol error

for l = 1: numTrans

    if encodedOutput(l) ~= encodedInput(l)
        errorSymbols = errorSymbols + 1;
    end
end

% Determine bit error
```

```
  for r = 1:2*numTrans % Iterate through input and output combination of bits and record the differences between them
      if output(r) ~= input(r)
        errorBits = errorBits + 1;
      end
  end

  simulatedBER(s) = errorBits / (2*numTrans);

  simulatedSER(s) = errorSymbols / numTrans;

  theorySER(s) = (3/2)* qfunc(sqrt((4/5)*snrLinear)); % Theoretical probability of symbol error value
  theoryBER(s) = theorySER(s) / log2(4); % BER is SER divided by log in base 2 of the dimensions of the system due to gray code encoding of input data

  disp(theoryBER(s));
end

% Plot of graph

figure;
semilogy(snr, theoryBER);
hold on;
semilogy(snr, simulatedBER, 'ro');
axis([0 15 1e-5 1]);
xlabel('Average SNR per bit (dB)');
ylabel('Probability of bit error');
title('Bit error probability curve for 4-PAM');
legend('theory', 'simulation');

% End of code
```

7. Conclusion.

Personally, during the implementation of this project I broadened my Digital Communications knowledge, whilst improving my coding skills and learning how to use MATLAB. In addition, it provides a good insight into 4-PAM modulation and the algorithms the information source and detector might implement, as well as into how AWGN can be created using its Probability Density Function.

8. References

- Dianati, Mehrdad (2017) ‘Communications & Networks’. University of Surrey.
- Proakis, G. Salehi, M. (2005) ‘Communication Systems Engineering’, Prentice Hall.
- Proakis, G. Salehi, M. (2005) ‘Fundamentals of Communication Systems’, Prentice Hall.
- http://mathworks.com
- http://embedded.com