The Q Function and Baseband Data Communication

Eren Can Gungor
Riccardo Miccini
Technical University of Denmark - DTU

December 13, 2016

Contents

1 Eye Diagram for a Digital Communication Channel

1.1 Eye diagram

A so-called *eye diagram* is the pattern that originates from overlapping a digital signal over the length of one or more transmitted bits. For several types of digital modulation the plot would show a series of round shapes (eyes) delimited by two rails, hence the name. This tool is commonly used to investigate several performance measures of a transmission channel, including noise level, distortion, inter-symbol interference, and synchronization errors.

An eye diagram can be generated with an oscilloscope by enabling infinite persistence, and setting the trigger to react on a separate clock signal.

1.2 `c5ce2.m`: explanation

Here follows a thoroughly commented version of the provided `c5ce2.m` MATLAB script. The code below generates and plots the eye diagrams of four band-limited signals composed of random sequences of bits.

Listing 1: ../scripts/1/c5ce2.m

```
% clean figure and load signal package (only for Octave)
clf
pkg load signal

% simulation parameters:
%  - nr of symbols (must be divisible by 4)
%  - nr of samples per symbol
%  - filter cutoff values (normalized values)
nsym = 1000;
nsamp = 50;
bw = [0.4 0.6 1 2];

% for each filter cutoff value..
for k = 1:length(bw)
  % generate filter coefficients using one of the cutoff values
  lambda = bw(k);
  [b,a] = butter(3,2*lambda/nsamp);

  % allocate space for total bit sequence
  l = nsym*nsamp;
  y = zeros(1,l-nsamp+1);

  % initalize random output vector with +1 and -1
  x = 2*round(rand(1,nsym))-1;
```

```
  % for each overlap..
  for i = 1:nsym
    % place one symbol into vector y, leaving nsamp samples of
       spacing
    kk = (i-1)*nsamp+1;
    y(kk) = x(i);
  end
  % zero-order hold
  datavector = conv(y,ones(1,nsamp));

  % apply filter to the total sequence
  filtout = filter(b, a, datavector);

  % splice sequence into sub-sequences of 4 symbols
  datamatrix = reshape(filtout, 4*nsamp, nsym/4);

  % discart the first 6 sub-sequences
  datamatrix1 = datamatrix(:, 6:(nsym/4));

  % plot eye diagram
  subplot(length(bw),1,k)
  plot(datamatrix1, 'k')

  % format: print axis labels, legend, and set plot range
  ylabel('Amplitude')
  axis([0 4*nsamp -1.4 1.4])
  legend(['Bn = ', num2str(lambda)])
  if k == 4
    xlabel('t/Tsamp')
  end
end
```

1.3 Channel model

In order to analyze the performances of a given transmission system, a model of the adopted transmission channel has to be devised.

For the purpose of this simulation, the most interesting characteristic of the channel is its bandwidth, and therefore it suitability for transmitting information at a specific rate. The limitation in bandwidth is obtained by feeding the signal into a third-order Butterworth low-pass filter.

The channel is therefore characterized by its *normalized bandwidth*, which represents the bandwidth as a percentage of the data bit rate; e.g if the bit rate is 8000Hz and the normalized bandwidth is 0.5, the channel will have a cutoff frequency of 4000Hz.

The MATLAB tools for generating digital filters (`butter`, `ellip`, `cheby1`...)

accepts the normalized frequency as input argument, which is the cutoff frequency in terms of the *Nyquist frequency* (half of the sampling frequency). The normalized bandwidth is proportional to the normalized frequency by a factor equivalent to the number of samples per bit.

1.4 `c5ce2.m`: different bandwidths

The following section will present a modified version of the previously introduced script, which can plot the eye diagrams for the normalized bandwidth 0.15, 0.3, 0.7, 1.2, and 4.

1.5 `c5ce2.m`: plots

This section will elaborate on the structure and implications of the eye diagrams generated using the scripts above.

The plots are generated by the original and augmented versions of the script respectively, and show the eye diagram for an antipodal baseband transmission at various bandwidths.

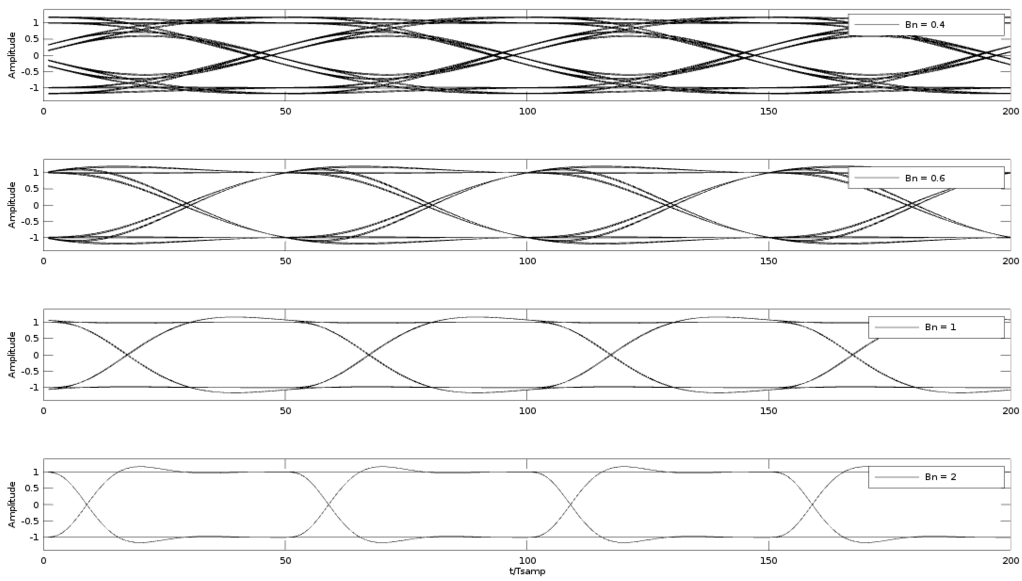

Figure 1: Eye diagrams of signals with filter coefficients 0.4, 0.6, 1, and 2

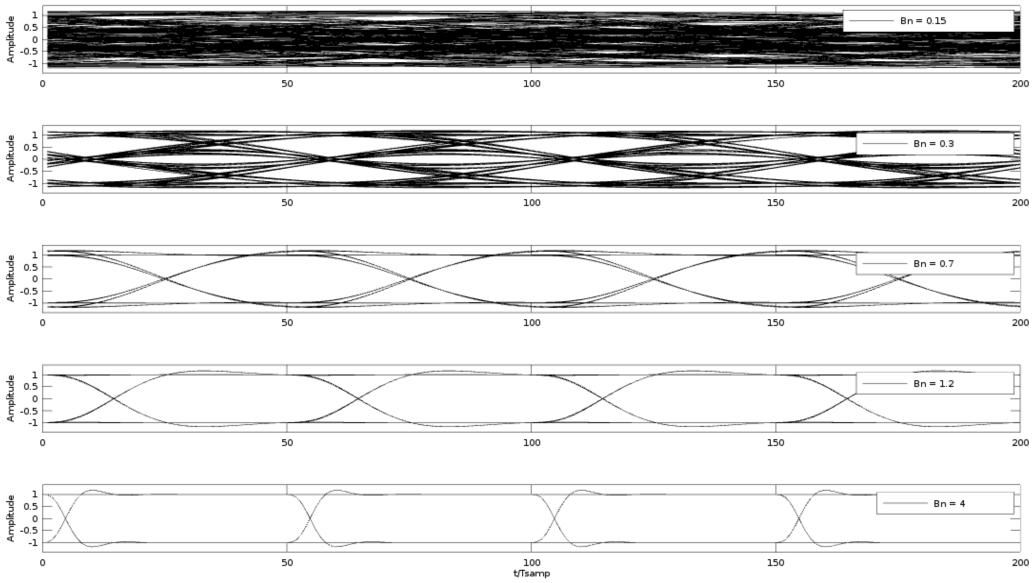

Figure 2: Eye diagrams of signals with filter coefficients 0.15, 0.3, 0.7, 1.2, and 4

2 The Q function

2.1 Normal probability density function 2.1

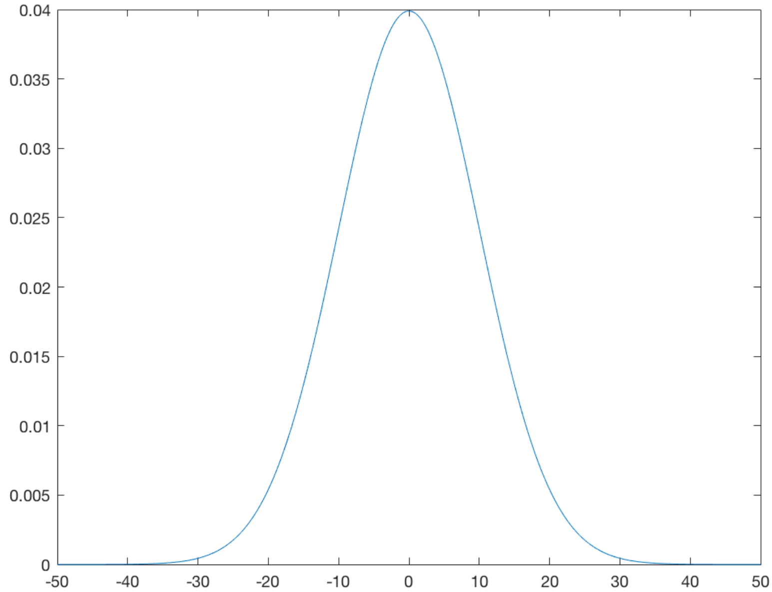

Figure 3: Normal Gaussian pdf graph with defined intervals

Normal gaussian distribution is one of the most important concept in Communication system and in statistics. It is used as a powerful tool when investigating random signals in communication systems. Such as investigating behaviour and application of noise signals.

The other reason is that because it is characterised by the limited variables. Such as mean μ and σ. It is easier compute and understand when we apply on communication systems.

All the variables that has been used in graphingpdf.m has been explained below;

mu . This is the mean value (μ) for the normal probability density function.

sigma This the sensible standard deviation number. (σ).

MAX 50; Maximum x value that x vector will get

MIN -50; Minimum x value that x vector will get

Also general formula for gaussian pdf is ; y=$f(x|\mu,\sigma) = (\frac{1}{\sigma\sqrt{2\pi}})e^{\frac{-(x-\mu)^2}{2(\mu)^2}}$ As we can see and understand from the variables above and the formula that all variables has been specified and we only need σ, μ and range.

2.2 Explanation of Q(u) function in relation to the normal probability density function 2.2

As we have explained and investigated probability density density function above . We can easily link $Q(u)$ function by exploiting the properties of cumulative distribution function and Probability density function.

- Q function is the 1- minus the cumulative distribution function of the standardised normal variable.
- Gaussian pdf with unit variance and zero mean is R=$(\frac{1}{\sqrt{2\pi}})e^{\frac{-(x)^2}{2}}$
- And corresponding cumulative distribution function become; $P = \int_{-\inf}^{x} Z$
- In last step Gaussian Q function defined as $Q(x) = 1 - P(x) = \int_{x}^{-\inf} Z$

2.3 Preparing a script file for plotting the $Q(u)$ function for argument values of relevance to the detection problems for digital communication receivers and inserting them in Appendix 2.3 and 2.4

Script file that has been created for the assignment 2.3 and 2.4 has been added to the Appendix. Important concept has been explained and relevant digital communication input values specified.

2.4 2.5

2.4.1 Complementary error function

2.5 Plots

Plotting for the questions 2.3 and 2.4

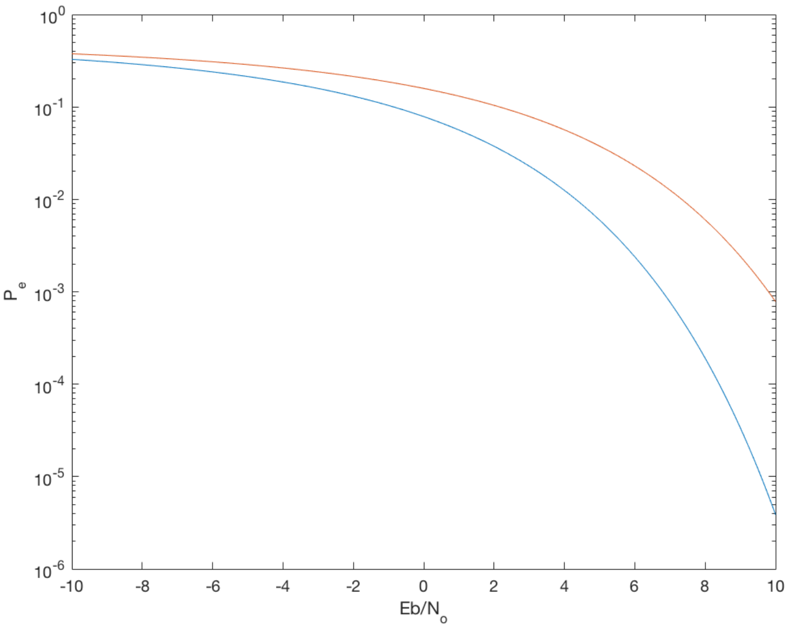

Figure 4: Plot for the $Q(u)$ function relevant to the argument values for the detection problems for digital communication receivers

3 Source Code

Here we have the properly prepared MATLAB codes for the second part of the second project. It has been used for observations, calculations and comparing with specified commands that given in this project.

The code belows computes and graphs normal(Gaussian) probability density function (pdf) in an appropriate intervals

Listing 2: ../scripts/2/graphingpdf.m

```
%graphing PDF function with random variables
mu=0;
sigma=10;
MAX = 50;
MIN = -50;
STEP = (MAX - MIN) / 1000;
PDF = normpdf(MIN:STEP:MAX, mu, sigma);
plot(MIN:STEP:MAX, PDF)
```

For the question 2.3 and 2.4 we have created the following code below. This code will plot the $Q(u)$ function for argument values relevant to the digital communication receivers. As we know from previous chapters that there are two common argument values :

- $R_12 = 0$ (Orthogonal Signals)
- $R_12 = -1$ (Antipodal Signals)

We also used z_db for the ratio for E_b/N_o. We should also notice that z_db is dimensionless ratio.

After that we have investigated the graph for further investigation to see whether we have achieved a satisfactory results for the assignment 2.3 and 2.4. Mathematical calculations are matching up with MATLAB simulation results.

Listing 3: ../scripts/2/qfunction.m

```
z_db = -10:.1:10;
r12 = [-1 0];
z = db2pow(z_db);
p_e(1,:) = qfunc(sqrt((1 - r12(1))*z));
p_e(2,:) = qfunc(sqrt((1 - r12(2))*z));
semilogy(z_db, p_e(1,:))
hold on
semilogy(z_db, p_e(2,:))
xlabel('Eb/N_o')
ylabel('P_e')
```

4 The Matched Filter Base Band Receiver

4.1 Additive white gaussian noise model

4.2 `c8ce1a.m`: explanation

4.3