

# The Q Function and Baseband Data Communication

Eren Can Gungor

Riccardo Miccini

Technical University of Denmark - DTU

December 9, 2016

## Contents

<b>1</b>	<b>Eye Diagram for a Digital Communication Channel</b>	<b>2</b>
1.1	Eye diagram . . . . .	2
1.2	c5ce2.m: explanation . . . . .	2
1.3	Channel model . . . . .	3
1.4	c5ce2.m: different bandwidths . . . . .	3
1.5	c5ce2.m: plots . . . . .	3
<b>2</b>	<b>The Q function</b>	<b>4</b>
2.1	Normal probability density function . . . . .	4
2.2	Explanation . . . . .	4
2.2.1	Inverse Q function . . . . .	4
2.2.2	Complementary error function . . . . .	4
2.3	Plots . . . . .	4
<b>3</b>	<b>The Matched Filter Base Band Receiver</b>	<b>5</b>
3.1	Additive white gaussian noise model . . . . .	5
3.2	c8ce1a.m: explanation . . . . .	5
3.3	. . . . .	5

# 1 Eye Diagram for a Digital Communication Channel

## 1.1 Eye diagram

## 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.

```
% 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 = 100;
nsamp = 50;
bw = [0.4 0.6 1 2];

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

    l = nsym*nsamp;

    % Total sequence length
    y = zeros(1,l-nsamp+1);

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

    % for each overlap..
    for i = 1:nsym
        % place symbols into vector y
        kk = (i-1)*nsamp+1;
```

```

        y(kk) = x(i);
    end
    % zero-order hold
    datavector = conv(y, ones(1, nsamp));

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

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

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

    % plot and format
    subplot(length(bw), 1, k)
    plot(datamatrix1, 'k')
    ylabel('Amplitude')
    axis([0 200 -1.4 1.4])
    legend(['Bn= ', num2str(lambda)])
    if k == 4
        xlabel('t/Tsamp')
    end
end

```

### 1.3 Channel model

#### 1.4 c5ce2.m: different bandwidths

#### 1.5 c5ce2.m: plots

This section will elaborate on the results and implications of the plots generated by the two scripts.

## **2 The Q function**

### **2.1 Normal probability density function**

### **2.2 Explanation**

#### **2.2.1 Inverse Q function**

#### **2.2.2 Complementary error function**

### **2.3 Plots**

### **3 The Matched Filter Base Band Receiver**

#### **3.1 Additive white gaussian noise model**

#### **3.2 c8ce1a.m: explanation**

#### **3.3**