

Article

Not peer-reviewed version

Enhanced Matched Filtering for Investigating Eye Diagrams in Baseband Binary Transmission

[Georgios Giannakopoulos](#)*, [Peter Adegbenro](#), [Maria Antonnette Perez](#)

Posted Date: 16 January 2025

doi: 10.20944/preprints202501.1252.v1

Keywords: Matched Filter; Baseband Binary Transmission; Gaussian Noise; BER Performance; BPSK Encoder; BPSK Decoder; SQRC Filter; Signal-to-Noise Ratio (SNR); Monte Carlo Simulation; MATLAB Simulation; MATLAB SIMULINK; Statistical Predictions



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Enhanced Matched Filtering for Investigating Eye Diagrams in Baseband Binary Transmission

Georgios Giannakopoulos ^{1,*}, Peter Adegbenro ² and Maria Antonnette Perez ³

¹ Independent Researcher, The Hague, The Netherlands

² Independent Researcher, Ilorin, Nigeria

³ Independent Researcher, Manila, Philippines

* Correspondence: georgios@progenesis.nl

Abstract: The aim of this paper is to study the performance of a matched filter for baseband binary transmission when subjected to Gaussian noise. A BER performance simulation model is created which consists of a BPSK encoder, a transmission SQRC filter, a Gaussian noise generator, a receive SQRT filter and a BPSK decoder. A Monte Carlo simulation technique is used to determine the dependency of BER performance on the signal to noise ratio. A MATLAB program and a MATLAB SIMULINK program was written to generate the required data. The MATLAB results show good agreement with statistical theoretical predictions.

Keywords: matched filter; baseband binary transmission; gaussian noise; BER performance; BPSK encoder; BPSK decoder; SQRC filter; signal-to-noise ratio (SNR); Monte Carlo simulation; MATLAB simulation; MATLAB SIMULINK; statistical predictions

1. Introduction

High-speed digital transmission of baseband data has become an important part of communication systems. However, when data are transmitted at high speeds over long distances, the signal is subjected to noise levels, which can affect the signal output and cause errors in the received data. To reduce errors, matched transmission filters and reception filters are commonly used in high-speed transmissions such as a matched SQRC filter [1].

The purpose of this paper is to investigate the effect of noise on a baseband transmission line that is filtered at both ends. This will involve the simulation of a binary transmission, which is represented by the block diagram shown in Figure 1 [1].

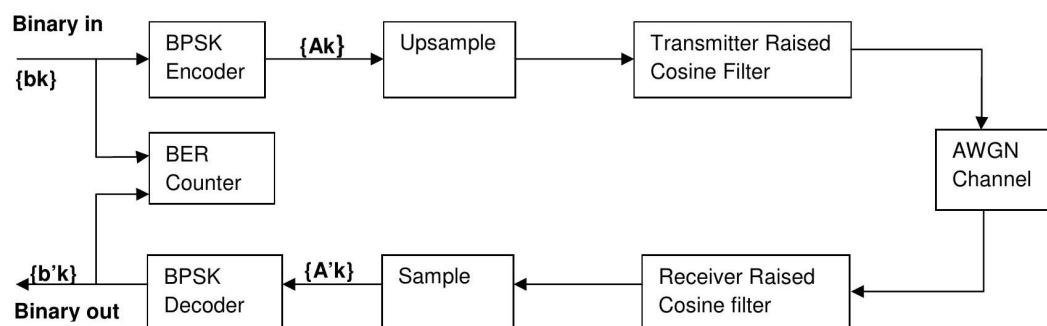


Figure 1. Baseband binary simulation.

The simulation will involve passing a binary signal through a BPSK encoder which converts the signal into polar data. The signal is then upsampled and passed through a matched transmission SQRC filter. An AWGN noise signal is added to the transmitted data, using various values of SNR, and is received by a matching SQRC filter. The received filtered data is then sampled and converted to

binary data a Maximum Likely rule. To determine the BER value for various values of SNR, a Monte Carlo method is used in which a large random sample of input data is generated [1,2].

This paper considers two simulations. The first baseband binary simulation will be carried out by creating a MATLAB [3] program to produce BER data. The second simulation will be carried out using SIMULINK to produce BER data.

Both experimental simulations show good agreement with each other and the theoretical prediction. The theoretical prediction is discussed in the following section.

2. Theoretical Prediction of BER

The simulation of matched filtering for baseband binary transmission initially involves the conversion of unipolar binary data into bipolar data [4]. A sequence of unipolar data b is converted into a polar data sequence A by the formula:

$$A_i = 2b_i - 1, \quad i = 1 \dots N$$

where i is the i^{th} element of the sequence and N is the length of the sequence.

For example the sequence $b = [101101]$ is converted to the sequence $A = [1 -1 \ 1 \ 1 \ -1 \ 1]$. The data is then passed through a transmission raised cosine filter, which is used to pulse shape the signal and minimize InterSymbol Interference (ISI) [5]. The theory involved in this type of filter is described in the following section.

2.1. Raised Cosine Filter Theory

A raised cosine filter has a frequency spectrum given by the formula [6]:

$$H(f) = \begin{cases} T & , |f| \leq \frac{1-\beta}{2T} \\ \frac{T}{2} \left[1 + \cos\left(\frac{\pi T}{\beta} \left[|f| - \frac{1-\beta}{2T} \right] \right) \right] & , \frac{1-\beta}{2T} < |f| \leq \frac{1+\beta}{2T} \\ 0 & , \text{otherwise} \end{cases} \quad (1)$$

where β is a roll-off factor in the range $0 \leq \beta \leq 1$, and T is the reciprocal of the symbol rate.

The impulse response $P(t) = F^{-1}[H(f)]$ of the raised cosine function for values of $\beta = 0.0, 0.5$ and 1.0 is shown in Figure 2 (as produced on a Maple package).

It can be seen in Figure 2, $P(t)$ has zero values at the normalized sampling times $T = 1, 2, 3, \dots$, which leads to zero ISI values being zero at all sampling points provided delay is taken into account [7,8].

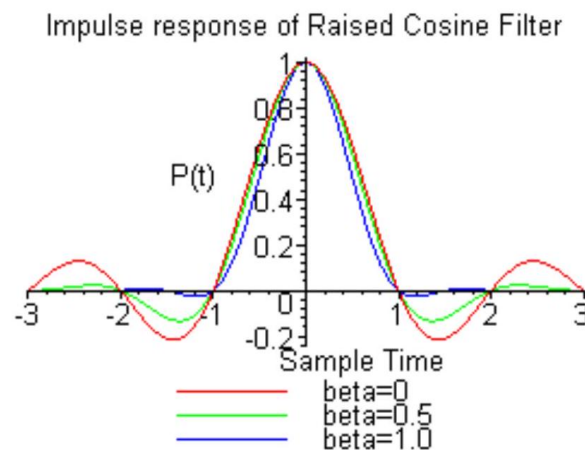


Figure 2. Impulse response of a raised cosine filter.

A raised cosine filter is commonly used at both ends of a transmission line. In practice a matched filter system is used at both ends in order to maximize signal-to-noise ratios and hence minimize the bit error rate (BER) [9]. In this paper a pair of square-root raised cosine filters is chosen at both ends, which is equivalent to the signal being subjected to a single raised cosine filter [1].

To demonstrate the effect of an SQRT filter on input polar data, a 30 symbol random sample is selected, as shown as message data in Figure 3. The filtered output of an SQRT filter is also presented in Figure 2 which has been upsampled by a factor of 16 and is compared with the message data, which have been shifted by a delay factor of 4 [10]. The output data from the transmit SQRT filter (without further upsampling) are passed through the receive SQRT filter, which has an output shown in Figure 4. It can be seen that zero ISI values occur when the input message is delayed by 8 [11].

The MATLAB program presented in Appendix 1 was used to generate the data presented in Figures 3 and 4 respectively [12].

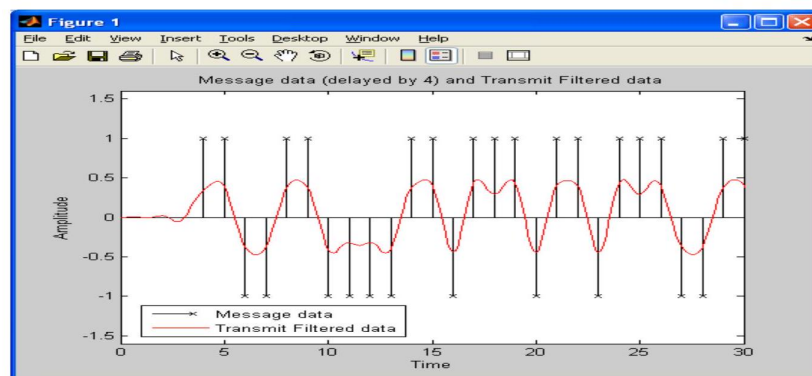


Figure 3. Square root raised cosine transmitted data (roll off factor $\beta = 0.5$) compared with message data (shifted by a delay factor 4 symbols).

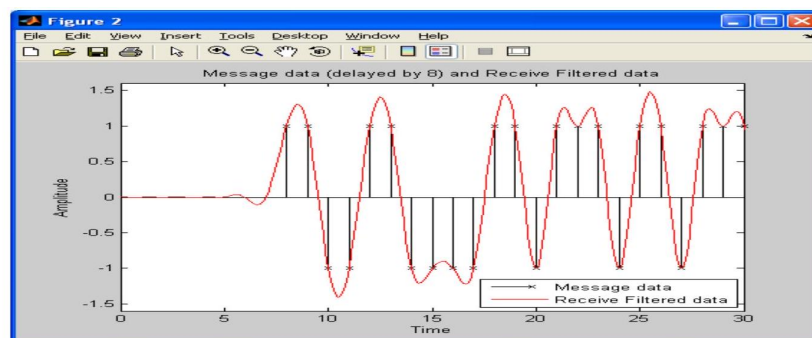


Figure 4. Square root raised cosine received data (roll off factor $\beta = 0.5$) compared with message data (shifted by a delay factor of 8 symbols).

2.2. Gaussian Noise

When noise is introduced in the transmission line, the ISI will no longer be zero at the sample points, and depending on the magnitude of the signal-to-noise ratio can lead to bit errors at the receiving end [13]. The theory for predicting BER rates as a function of SNR is as follows.

Consider an input signal $S_i(t)$ to a matched filter system over an AWGN channel with an output signal $S_o(t) + N_o(t)$, where $N_o(t)$ is the noise function as shown in Figure 5 [1,14].

To simulate noise, a Gaussian distribution is used to model the noise level in the transmission line, which is assumed to have a zero mean. A Gaussian function with a mean of $+A$ volts and a Gaussian function of mean $-A$ volts are shown in Figure 6 [15,16].



Figure 5. Diagram of a Matched filter.

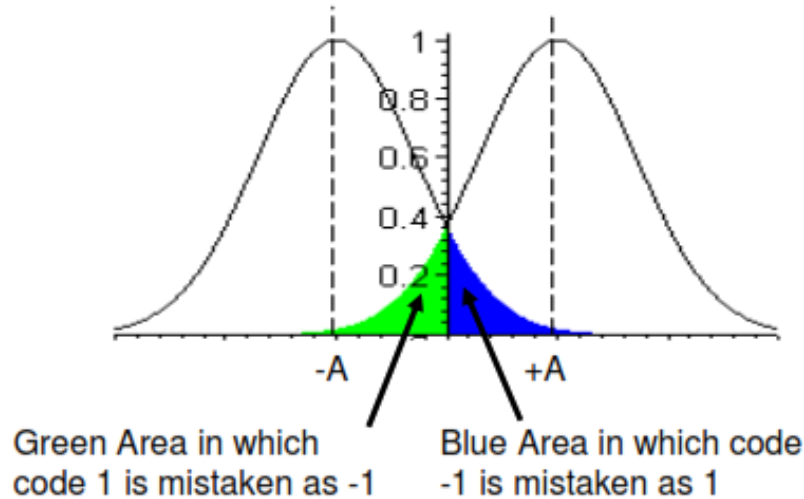


Figure 6. Gaussian distribution.

As shown in Figure 6, the probability that code 1 is mistaken as code -1 corresponds to the area of the green section and the probability that code -1 is mistaken as code 1 corresponds to the area of the blue section [13,17].

From the Gaussian distribution, these probabilities are represented by [13,16]:

$$\text{(Probability of bit 1 mistaken as -1)} = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^0 e^{-(x-A)^2/2\sigma^2} dx \quad (2)$$

$$\text{and (Probability of bit -1 mistaken as 1)} = \frac{1}{\sigma\sqrt{2\pi}} \int_0^{\infty} e^{-(x+A)^2/2\sigma^2} dx \quad (3)$$

The probability of a bit error occurring in a transmission line subjected to Gaussian noise is given by:

$$P_e = (\text{Probability of bit =1}) \times (\text{Probability of bit 1 mistaken as -1}) + (\text{Probability of bit =-1}) \times (\text{Probability of bit -1 mistaken as 1}) \quad (4)$$

Since $(\text{Probability of bit = 1}) = (\text{Probability of bit = -1}) = 0.5$ then probability of a bit error is:

$$P_e = 0.5 \times \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^0 e^{-(x-A)^2/2\sigma^2} dx + 0.5 \times \frac{1}{\sigma\sqrt{2\pi}} \int_0^{\infty} e^{-(x+A)^2/2\sigma^2} dx \quad (5)$$

Using symmetry:

$$P_e = \frac{1}{\sigma\sqrt{2\pi}} \int_0^{\infty} e^{-(x+A)^2/2\sigma^2} dx \quad (6)$$

On making the substitution: $u = \frac{x+A}{\sigma}$, then

$$P_e = \frac{1}{\sqrt{2\pi}} \int_{A/\sigma}^{\infty} e^{-u^2/2} du \quad (7)$$

i.e., $P_e = Q\left(\frac{A}{\sigma}\right)$ where

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-u^2/2} du \quad (8)$$

For a polar signal of $\pm A$ volts, the mean signal power $S = A^2$.

For a Gaussian distribution, normalized Gaussian noise, for a zero mean, is equal to the expected value:

$$N = E(x^2) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\infty} x^2 e^{-x^2/2\sigma^2} dx = \sigma^2 \quad (9)$$

Hence on substituting $A = \sqrt{S}$ and $\sigma = \sqrt{N}$ into Equation 8 give the probability of a bit error as [17]:
For a polar signal,

$$P_e = Q\left(\sqrt{\frac{S}{N}}\right) = Q(\sqrt{SNR}) \quad (10)$$

where SNR is the ratio of signal to noise.

3. Simulation Model Using MATLAB Programming

A random number generator is used to generate a source of binary data for transmission through a filtering system. Before filtering, the binary data are first converted to Non-Return to Zero (NRZ) bipolar data using a BPSK decoder [1,14]. A raised cosine-matched SQRC filter is used at the transmission end and at the receiver end in the simulation model. This type of matched filter has the advantage that the shape of the waveform does not cause Inter-Symbol Interference (ISI) of the signal between the transmitter and receiver [7,8,13].

3.1. Simulation Using Programming

A MATLAB program was written to generate BER vs. SNR data for a matched filtering of baseband binary transmission using the Monte Carlo method as shown in Appendix 2 [1,11,18].

A sequence of 50,000 unipolar data points, denoted by b , was created using a random number generator. The data is then converted to polar data, denoted by A , using the formula $A_i = 2 \times b_i - 1$, where b_i and A_i are the i^{th} elements of the sequences b and A respectively [2,4].

The MATLAB command RCOSFLT was used to generate transmission output data from a square root raised output filter using an upsampling value of 16, a delay of 4 and a roll-off factor of 0.5. The MATLAB RCOSINE command was not used in this program, since this function is incorporated into the MATLAB RCOSFLT function [1,8]. To generate Gaussian noise data from the transmission output, the command AWGN was used, which was set to SNR values ranging from 1 to 12 [1,2].

The received square-root raised cosine filter used in the program was also generated from the MATLAB RCOSFLT function, which was set without further upsampling [6,11]. The received filtered data was converted to unipolar data by using the Maximum Likelihood rule:

$$b'_k = \begin{cases} 1 & \text{if } A' \geq 0 \\ 0 & \text{if } A' < 0 \end{cases}, \text{ involving the MATLAB if statements.} \quad (11)$$

In order to computerize the BER values, the first 8 symbols of the output sequence were removed from the output data to compensate for the delay of 4 in each of the two raised cosine filters [18].

The BER value was obtained by comparing the truncated output sequence with the input sequence using the MATLAB sumerr function which counts the number of non-matching bits occurring between the input and output sequence [4-6].

To ensure an accurate estimation of the bit error, a random sample of 50,000 symbols was generated for the input sequence. A loop was used to generate BER data for SNR values ranging from 1 to 12 [9,10].

Theoretical BER values were also computed in the MATLAB program for comparison purposes. The results of this program are discussed in the results and discussion in Section 4.

3.2. Simulation Using SIMULINK

A SIMULINK program [3,18] was developed to simulate the BER vs SNR values for matched filtered baseband transmission as shown in Figure 7.

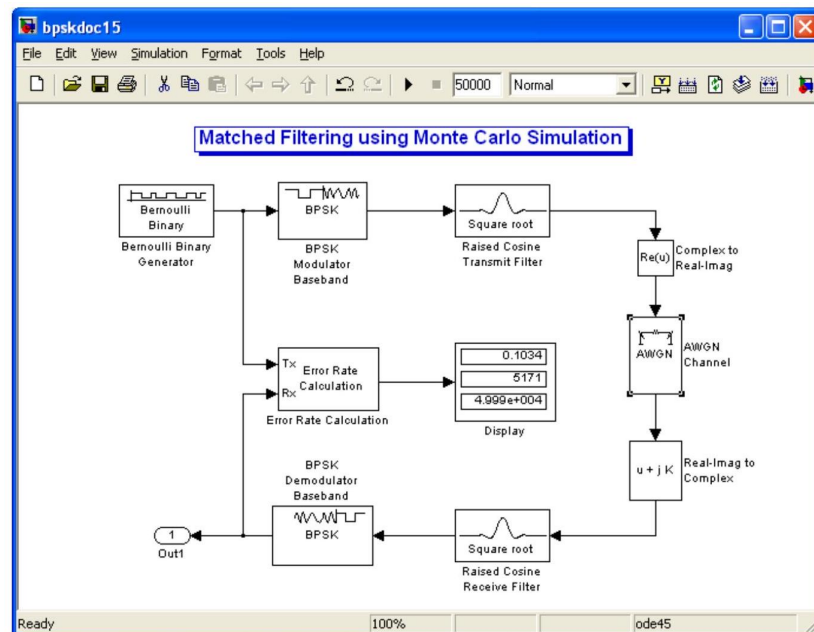


Figure 7. Matched Filtering using Monte Carlo Simulation Demonstration for a 50000 random bit sequence, with AWGN set to $SNR = 2$ dB.

In the SIMULINK block diagram, the Bernoulli binary block was used to generate binary data using random seed of 61. To generate polar data from the binary data, a BPSK modulator block was used in which the phase offset was set to π [6,8]. This ensured that the binary code 1 was encoded as 1 and the binary code 0 was encoded as -1 . The upsampling of the data was incorporated into the SQRT transmit filter using an upsampling ratio of $16 : 1$. The roll-off factor was set to 0.5 with a chosen delay of 4 [5,6].

It should be noted that the output from the BPSK encoder and the SQRT filter is represented as a complex number in which the imaginary part is zero [2,18]. In order to avoid the AWGN channel creating noise on the real part and the imaginary part of the complex number output, a real part converter was placed before the AWGN block channel as shown in Figure 7. A real number conversion to a complex number block was required after the AWGN channel for consistent matching of the data in the matched receive filter [5,8].

The receive cosine filter was set to 16 input symbols per second with group delay of 4 and roll-off factor of 0.5.

The BPSK demodulator was set with phase offset of π to decode the polar bit -1 as 0 and to decode the polar bit 1 as 1.

The Error rate calculation block, was set with a received delay of 8 (since each of the two matched filters were each set with a delay of 4). The computation delay was set to 0. The target number of errors and the maximum number of symbols in the Error rate calculation block was set to infinity [7,8].

4. Discussion and Results

A MATLAB program was written to generate BER data vs SNR values ranging from 1 to 12 as shown in Appendix 2. A sequence of 50,000 random data bits was generated to ensure sufficient accuracy with the theoretical data, which is also produced in the same program [1]. The graphical

output and tabulated output generated by the computer program are shown in Figures 8 and 9, respectively. 143
144

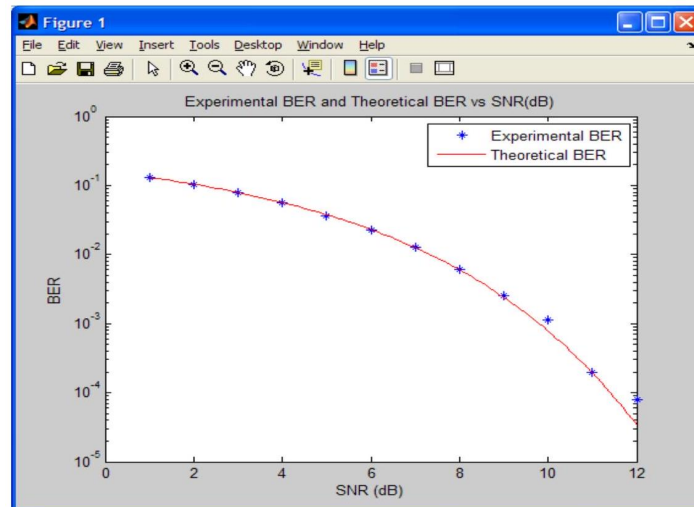


Figure 8. Experimental and Theoretical BER versus SNR (dB).

SNR (dB)	BER (Exp)	BER (Theory)
1.0	1.315e-001	1.309e-001
2.0	1.038e-001	1.040e-001
3.0	7.882e-002	7.890e-002
4.0	5.570e-002	5.650e-002
5.0	3.628e-002	3.768e-002
6.0	2.278e-002	2.301e-002
7.0	1.328e-002	1.259e-002
8.0	5.800e-003	6.004e-003
9.0	2.760e-003	2.413e-003
10.0	9.200e-004	7.827e-004
11.0	2.200e-004	1.940e-004
12.0	6.000e-005	3.430e-005

Figure 9. Tabulated Experimental and Theoretical BER versus SNR (dB).

Figure 8 shows good agreement between the theoretical and experimental data. Some discrepancy in the experimental data occurs at high values of SNR which is to be expected since the BER values are in the region of 10^{-4} . To improve accuracy for higher values of SNR, a simulation run of at least $10/\text{BER} = 10/10^{-4} = 10^5$ should be used in the program. 145
146
147
148

A MATLAB SIMULINK program was also run with a symbol size of 50,000 random binary data. As seen in Figure 7, a bit error rate of $\text{BER} = 0.1034$ was obtained when $\text{SNR} = 2$ dB. This value is very close to the tabulated theoretical value of 0.1040 shown in Figure 9, which confirms the accuracy of the SIMULINK program [17,19]. 149
150
151
152

In order to visualize the bit error dependence of the matched filter baseband transmission on SNR values, eye diagrams were constructed using a computer-generated program (see Appendix 2). A SIMULINK program was also used to generate an eye diagram by including a Discrete-Time Eye Diagram Scope block in the program (see Appendix 5) [18]. 153
154
155
156

Figure 10 shows the eye diagram of the signal after the transmit filter and also the eye diagrams of the signal after the receive filter when subjected to values of $\text{SNR} = 20$ dB and $\text{SNR} = 8$ dB [20]. The eye diagrams for after the transmit filter and receive filter show minimal time variation and low 157
158
159

signal distortion when the signal to noise ratio is large. Time variation and noise levels after the receive filter are larger than those of the receive filter (with $SNR = 20$ dB) since the signal is passed through an extra filter. No bit errors occurred in the SIMULINK program when $SNR = 20$ dB [21,22]. This can be confirmed by inspecting the eye diagram, as the eye has a wide opening at the sampling points of $t = 0, 1, 2$ seconds when $SNR = 20$ dB.

In the case when $SNR = 8$ dB, a bit error rate of 0.5% was recorded in the SIMULINK program. This result is consistent with the eye diagram representation shown in Figure 10 when $SNR = 8$ dB. A bit error rate of 0.05% corresponds to one trace error in 200 traces occurring at a sampling point. The eye diagram clearly shows one or two traces closing the eye near a sampling point.

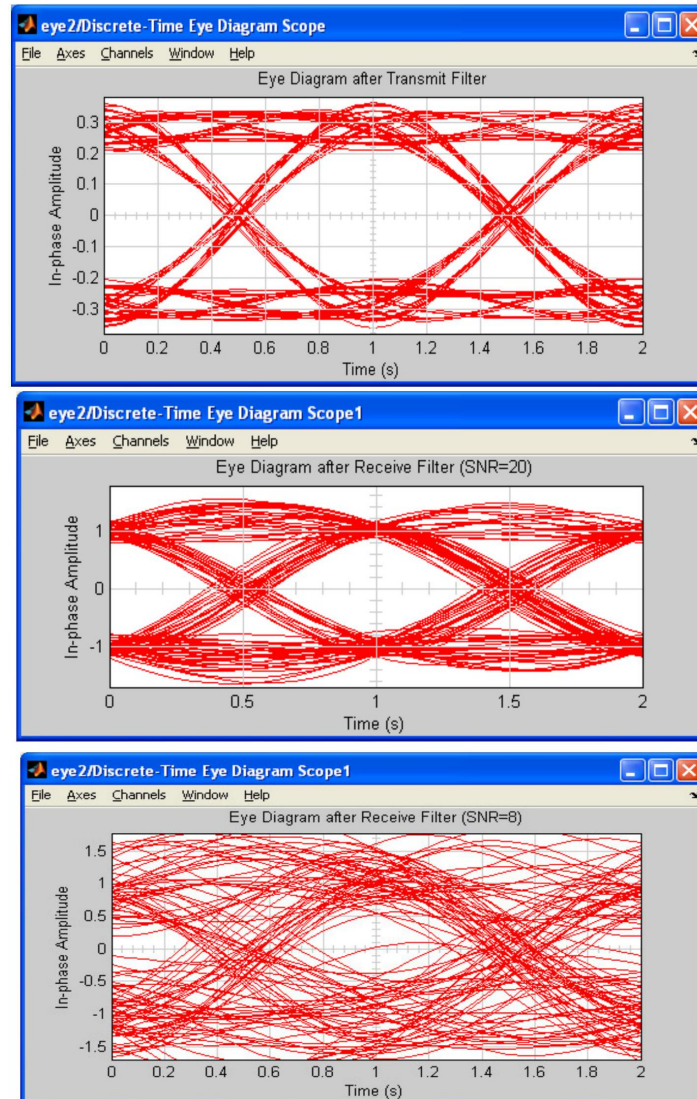


Figure 10. Transmit Filter and Receive Filter Eye Diagrams ($SNR = 8, 20$ dB).

For completeness and understanding of the simulation process, a SIMULINK program was generated with time scopes positioned at 6 points of the simulation loop as shown in [Appendix 3](#). A random 30 symbol sequence was generated and the time scope displays at each of the 6 positions is shown in [Appendix 4](#) using a SNR value of zero. [Appendix 6](#) also shows eye diagram outputs generated by the MATLAB program presented in [Appendix 2](#).

In conclusion of the paper, the performance of Matched filtering baseband binary transmission using a Monte Carlo simulation accurately predicts the theoretical values of BER over the range of SNR from 0 dB to 12 dB.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix 1

```

1  clear all;clc;
2  % Program to create Cosine Filter data
3  Delay = 4; DataL = 30; R = .5; Fs = 8; Fd = 1; PropD = 0;
4  % Generate random data.
5  x = randint(DataL, 1, [], 1234);x=2*x-1;
6  % at time 0, 1/Fd, 2/Fd, ...
7  tx = [PropD: PropD + DataL - 1] ./ Fd;
8  % Plot data.
9  stem(tx, x, 'kx');
10 % Set axes and labels.
11 axis([0 30 -1.6 1.6]); xlabel('Time'); ylabel('Amplitude');
12 % Correct for propagation delay
13 PropD = Delay * Fd;tx = [PropD: PropD + DataL - 1] ./ Fd;
14 % Filter at the transmitter.
15 [yc, tc] = rcosflt(x, Fd, Fs, 'sqrt',R,Delay);
16 % Plot data.
17 figure(1);
18 stem(tx, x, 'kx'); hold on;
19 % Plot filtered data.
20 plot(tc, yc, 'r-'); hold off;
21 title('Message data (delayed by 4) and Transmit Filtered data');
22 legend('Message data','Transmit Filtered data');
23 % Set axes and labels.
24 axis([0 30 -1.6 1.6]); xlabel('Time'); ylabel('Amplitude');
25 % Filter at the receiver.
26 [ydown, tdown] = rcosflt(yc, Fd, Fs, 'sqrt/Fs',R,Delay);
27 PropD = 2*Delay * Fd;tx = [PropD: PropD + DataL - 1] ./ Fd;
28 figure(2);
29 stem(tx, x, 'kx'); hold on;
30 % Plot filtered data.
31 %plot(tc, yc, 'r-'); hold off;
32 plot(tdown, ydown, 'r-'); hold off;
33 title('Message data (delayed by 8) and Receive Filtered data');
34 legend('Message data','Receive Filtered data');
35 % Set axes and labels.
36 axis([0 30 -1.6 1.6]); xlabel('Time'); ylabel('Amplitude');
37

```

Appendix 2

219

```

1
2 % Simulation of matched filtering for baseband binary transmission % Random unipolar
  ↳ data is generated, and encoded using a BSPK encoder, and % transmitted through a
  ↳ matched raised cosine filtering system which is % subjected to AWGN noise.
3
4 % Experimental BER vs SNR data is generated using Monte Carlo Simulation
5
6 % Theoretical BER vs SNR data produced from Gaussian Distribution function
7
8
9 clear all;clc;
10
11 SNRmax=12;SNReye1=20;SNReye2=8; % SNR in dB
12
13 $N=50000$; % $N$ is the number of symbols
14
15 Fd=1; % Sample frequency in the output
16
17 Fs=16; % Upsample and Downsample factor for cosine filter
18
19 delay=4; % Delay selected for cosine filter
20
21 R=0.5; % Rolloff factor for cosine filter
22
23 % Generate Random Unipolar Data and Convert to Polar Data (BPSK encoder)
24
25
26 y=randint(N,1,[],1234); % Generate N random unipolar
27
28 symbols
29
30 $y_b=2 * y-1$ ; \quad % Convert symbols to polar symbols
31
32 % Transmit Filter with Upsampling
33
34 $$$
35
36 [yup, xup]=rcosflt(yb,Fd,Fs,'sqrt',R,delay); % Select Raised square root cosine filter
37
38 % using Fs/Fd upsampling with delay
39
40 for SNR=1:SNRmax % Start of SNR loop to generate BER data
41
42 % Generate Gaussian Noise between transmit and receive filtered data
43
44 yup1=awgn(yup,SNR); % Add AWGN noise for selected SNR value
45
46 % Receive filter with downsampling
47
48 $$$
49
50 [ydown,xdown]=rcosflt(yup1,Fd,Fs,'sqrt/Fs',R,delay); % Raised cosine filter % without
  ↳ further
51
52 upsampling
53 yout=downsample(ydown,Fs/Fd); % Downsample by factor Fs/Fd

```

220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276

```

1  % Convert to unipolar data (BPSK decoder), count bit error, adjust for delay %
2
3
4  $\mathrm{yy}=(1+\operatorname{sign}(\$ \text{yout } \$)) / 2 ; \$ \backslash$
5  $y_1=y y(2 * \text{d e l a y}+1: 2 * \text{d e l a y}+N) ; \$ \backslash$
6  $[B E R \text{ n u m}, B E R(S N R)]=\operatorname{symerr}(y, y_1) ; \$ \backslash$
7  end \backslash
8  % Convert polar to unipolar data. \backslash
9  % Delay output by $2 * $ delay \backslash
10  $$$ Count experimental BER. \backslash
11  % End of SNR loop to calculate BER data
12
13  % Theoretical Calculation of BER
14
15  $$$
16
17  $x t=[1: 0.1: S N R \text{ m a x}] ; \quad \%$ Generate sequence SNR=1 to SNRmax in steps of 0.1
18
19  $d B$
20
21  xtc=sqrt(10.\^{xt./10}); % Convert SNR(dB) to SNR(ratio)
22
23  yt=qfunc(xtc); % calculate BER probability using Q function
24
25  % [See Mosa(2007), page 67, equation (2.52)]
26
27  $$$ Output data and plots
28
29  disp('SNR(dB) BER(Exp) BER(Theory)');
30
31  SNRx=[1: SNRmax]; SNRxx=sqrt(10.\^{SNRx./10}); BERT=qfunc(SNRxx);
32
33  for $j=1: $ SNRmax
34
35  s=blanks(3-length(int2str(j)));
36
37  disp(sprintf('%s%2.1f%2.3e%2.3e', s,
38
39  $\operatorname{SNR}\{j\}, \operatorname{BER}\{j\}, \operatorname{BERT}\{j\}$));
40
41  end
42
43  figure(1); % Plot the BER versus SNR (theory and experiment)
44
45  semilogy([1: SNRmax], BER, '*'); % Experimental BER vs SNR plot on log(y)
46
47  scale
48
49  hold on; semilogy(xt, yt, 'r'); % Theoretical BER vs SNR plot on log(y) scale hold off;
50
51  xlabel('SNR(dB)'); ylabel('BER');
52
53  legend('Experimental BER', 'Theoretical BER');
54
55  title('Experimental BER and Theoretical BER vs SNR(dB)');
56
57  hold off;
58
59  % Eye Diagrams
60
61  eyediagram(yup(N/2-100\textit{16:N/2+100}16), 32);
62
63  title('Eye Diagram after Transmit Filter');

```

```

64
65 yup1=awgn (yup, SNReye1) ;
66
67 [ydown, xdown]=rcosflt(yup1, Fd, Fs, 'sqrt/Fs', R, delay);
68
69 eyediagram(ydown(N/2-100\textit{16:N/2+100}16), 32);
70
71 titlestring=strcat('Eye_Diagram_after_Receive_Filter_for
72 SNR=', int2str(SNReye1), 'dB');
73
74 title(titlestring);
75
76 yup1=awgn (yup, SNReye2);
77
78 [ydown, xdown]=rcosflt(yup1, Fd, Fs, 'sqrt/Fs', R, delay);
79
80 eyediagram(ydown(N/2-100\textit{16:N/2+100}16), 32);
81
82 titlestring=strcat('Eye_Diagram_after_Receive_Filter_for
83 SNR=', int2str(SNReye2),
84
85 title(titlestring);
86
87

```

Appendix 3

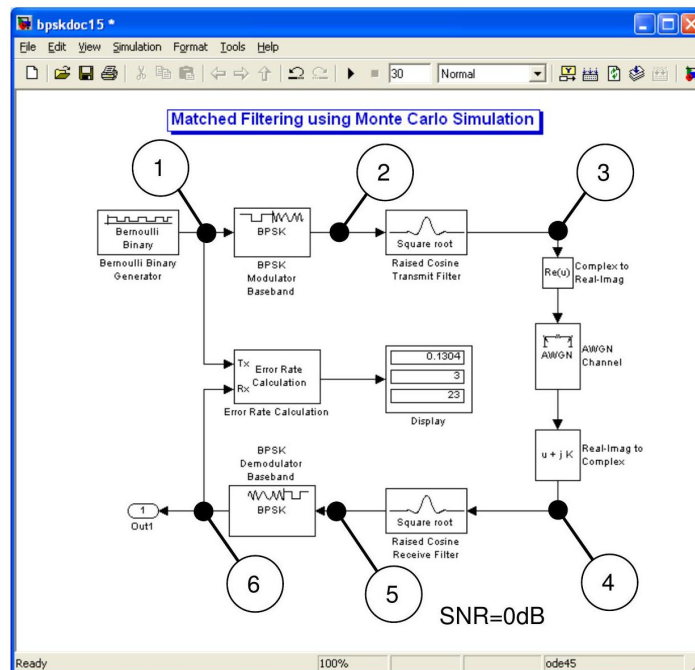


Figure A1. Matched Filtering using Monte Carlo Simulation Demonstration for a 30 random bit sequence, with AWGN set to $SNR = 0$ dB. Labels 1-6 show position of times scopes.

Appendix 4

368

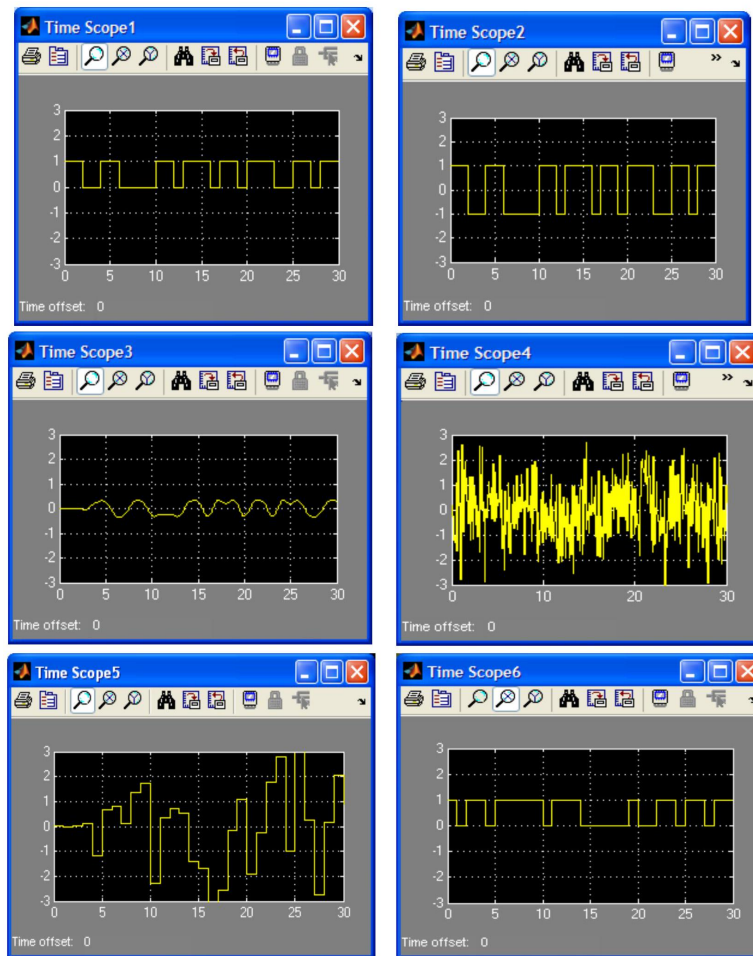


Figure A2. Simulation Demonstration of a 30 symbol input sequence with SNR = 0 dB.

Time scope displays 1-6 are taken at scope points 1-6 as shown in Figure 8 respectively Input sequence = [110011000011011101101110011011] Output sequence (after delay of 8 bits) = [11011100000100110110111] Difference in input and output sequence shows 3 error bits in 23 bits giving BER = 0.1304

369

370

371

372

Appendix 5

373

```
1
2 clear all;clc;\
3 close all;\
4 SNRmax=12; \% SNR in dB\
5 N=12000; \% N is the number of symbols\
6 Fd=1; \% Sample frequency in the output\
7 Fs=16; \% Upsample and Downsample factor for cosine filter\
8 delay=2; \% Delay selected for cosine filter\
9 R=0.5; \% Rolloff factor for cosine filter\
10 y=randint(N,1,[],61); \% Generate N random unipolar\
11 symbols\
12 yb=2*y-1; \% Convert symbols to polar\
13 symbols\
14 [yup,xup]=rcosflt(yb,Fd,Fs,'sqrt',R,delay); \% Select Raised square root\
15 cosine filter\
16 \% using Fs/Fd upsampling with\
17 delay\
18 \%h1=eyediagram(yup(803:end),80);\
19 SNR=30; \% Start of SNR loop to generate BER data\
20 yup1=awgn(yup,SNR); \% Add AWGN noise for selected SNR value\
21 [ydown,xdown]=rcosflt(yup1,Fd,Fs,'sqrt/Fs',R,delay); \% Raised cosine filter\
22 \%eyediagram(ydown,16);\
23 \% without further upsampling\
24 yout=downsample(ydown,Fs/Fd); \% Downsample by factor\
25 FS/Fd\
26 h=commscope('SamplingFrequency',10,'SamplesPerSymbol',32,'Symbol\
27 sPerTrace',1,'NumberOfStoredTraces');\
28 update(h,real(ydown));\
29 h.SamplesProcessed\
30 h.SamplingFrequency\
31 h.PlotType='2D_Line'<\
32 plot(h,'r-');\
33 disp(length(yup));disp(length(ydown));
```

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

Appendix 6

409

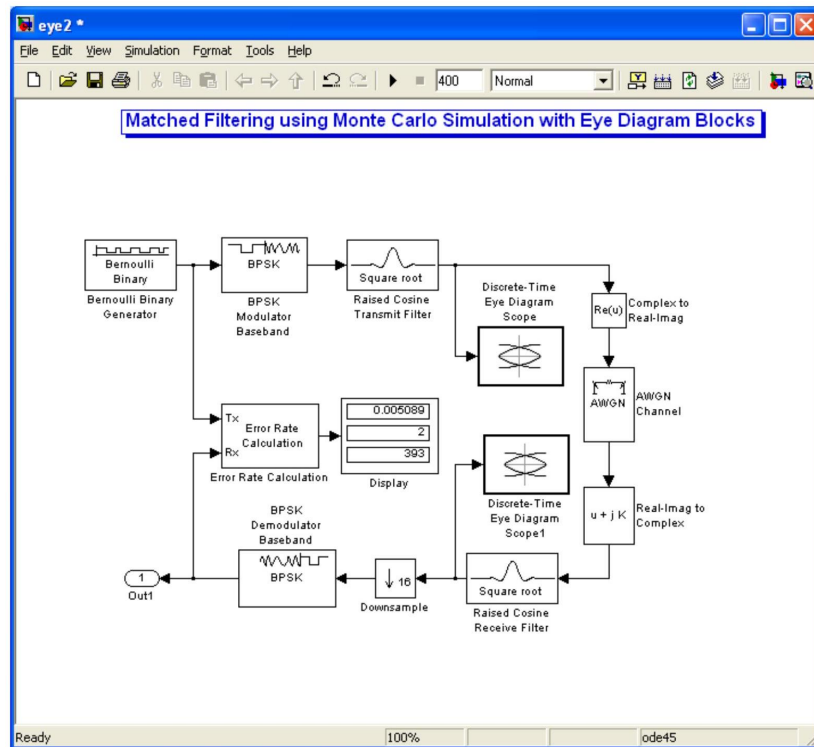


Figure A3. SIMULINK eye diagram generator for Matched Filtering of Baseband Transmission.

Appendix 7

410

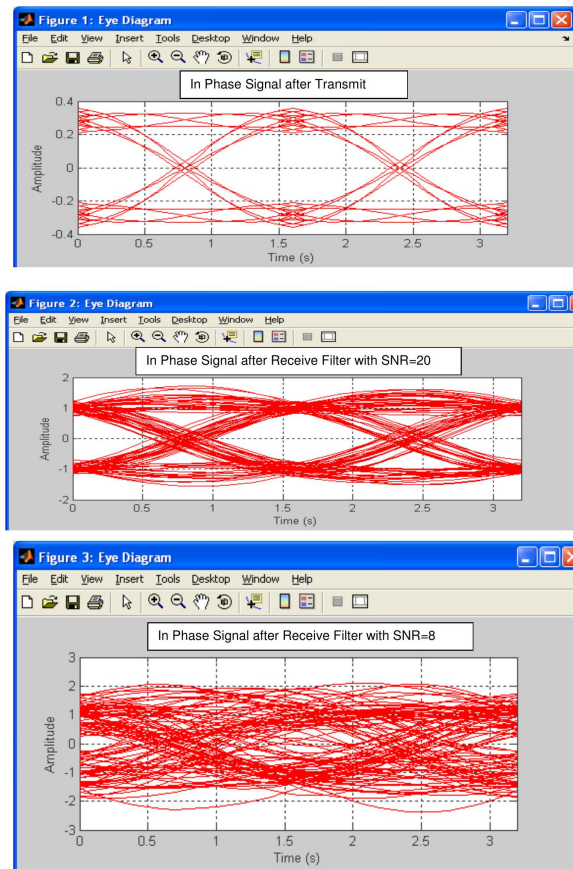


Figure A4. MATLAB program generated eye diagrams after transmit filter and after receive filter with $SNR = 8$ dB and $SNR = 20$ dB.

References

1. Abu-Rgheff, M. A. *Introduction to CDMA Wireless Communications* (Academic Press, 2007). 411
2. Sklar, B. *Digital Communications: Fundamentals and Applications* (Prentice Hall, 2001), 2nd edn. 412
3. The MathWorks. Matlab documentation. <http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.shtml>. Accessed: 2008-02-14. 413
4. Cevrero, A. *et al.* Fast and accurate ber estimation methodology for i/o links based on extreme value theory. *Proceedings -Design, Automation and Test in Europe, DATE 7*, 10.7873/DATE.2013.114 (2013). 414
5. Huo, Y. *et al.* Technology trends for massive mimo towards 6g. *Sensors* **23**, 7, 10.3390/s23136062 (2023). 415
6. Wang, Z. *et al.* A tutorial on extremely large-scale mimo for 6g: Fundamentals, signal processing, and applications. *arXiv preprint arXiv:2307.07340*, 39, 10.48550/arXiv.2307.07340 (2023). 416
7. Lei, H. *et al.* Near-field user localization and channel estimation for xl-mimo systems: Fundamentals, recent advances, and outlooks. *arXiv preprint arXiv:2407.10147*, 9, 10.48550/arXiv.2407.10147 (2024). 417
8. Dong, J. *Estimation of Bit Error Rate of any digital Communication System*. Theses, Télécom Bretagne, Université de Bretagne Occidentale (2013). 418
9. Molisch, A. F. *Wireless Communications* (Wiley-IEEE Press, 2011), 2nd edn. 419
10. Saadat, A., Salman, M. & Saadat, H. Matched filter based timing synchronization for orthogonal frequency division multiplexing systems. In *Proceedings of the 2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)*, 1157–1161, 10.1109/ICIEA.2013.6566541 (IEEE, 2013). 420
11. Shahi, S. N. *Analysis and Mitigation of the Nonlinear Impairments in Fiber-Optic Communication Systems*. Ph.d. thesis, McMaster University, Hamilton, Ontario, Canada (2013). Supervisor: Dr. Shiva Kumar. 421
12. Mestoui, J., Ghzaoui, M. E., Elaage, S., Hmamou, A. & Foshi, J. Ce-ofdm-cdma phase modulation for 5g system. *International Journal of Mathematical, Engineering and Management Sciences* **6**, 1100–1114, 10.33889/IJMEMS.2021.6.4.065 (2021). 422

430

431

432

433

13. Proakis, J. & Salehi, M. *Digital Communications* (McGraw-Hill Education, 2007), 5th edn. 434
14. Liberti, J. C. & Rappaport, T. S. *Smart Antennas for Wireless Communications: IS-95 and Third Generation CDMA Applications* (Prentice Hall PTR, 1999), 1st edn. 435
15. Aadithya, K., Ray, S. & Roychowdhury, J. Bee: Predicting realistic worst case and stochastic eye diagrams by accounting for correlated bitstreams and coding strategies. In *Proceedings of the Asia and South Pacific Design Automation Conference (ASPDAC)*, 366–371, [10.1109/ASPDAC.2015.7059032](https://doi.org/10.1109/ASPDAC.2015.7059032) (2015). 436
16. Hall, T., Delpont, J. & Fourie, C. Determination of the bit error rate due to thermal noise using josim superconducting circuit simulator and the monte carlo method. *IEEE Transactions on Applied Superconductivity* **PP**, 1–5, [10.1109/TASC.2023.3251940](https://doi.org/10.1109/TASC.2023.3251940) (2023). 437
17. Rindhe, B., Digge, J. & Narayankhedkar, S. Modeling of ofdm based system with optical components for bw reconfiguration. *International Journal of Computer Applications* 14–20 (2014). 438
18. Giannakopoulos, G. *Low Power Modem Design: A guidance to use novel algorithms (QPSK) and digital filters (FIR, IIR) with MATLAB, SIMULINK with VHDL and Xilinx FPGA* (LAP LAMBERT Academic Publishing, 2015), 1st edn. 439
19. Güvenkaya, E. & Arslan, H. Frequency domain eye diagram for ofdm. In *Conference Paper*, [10.13140/RG.2.1.3647.6966](https://doi.org/10.13140/RG.2.1.3647.6966) (2013). 440
20. Yadav, S., Kumar, P., Singh, R., Raj, V. & Pradhan, D. Comparative analysis of ber and papr for ofdm, ofdm-cdma, ofdm-mc-cdma & ofdm-idma systems. *International Journal of Electronics and Communication Engineering* **10**, 456–462 (2021). 441
21. Saadat, A., Salman, M., Ajaz, M. & Islam, M. U. Matched filter based timing and frequency synchronization for multiple input multiple output orthogonal frequency division multiplexing systems. *Wireless Personal Communications* **82**, [10.1007/s11277-014-2206-0](https://doi.org/10.1007/s11277-014-2206-0) (2015). 442
22. Rindhe, B., Digge, J. & Narayankhedkar, S. Implementation of optical ofdm based system for optical networks. *International Journal of Electrical and Computer Engineering (IJECE)* **vol. 4**, page 767–781 (2014). 443

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content. 454