Performance of Hypothesis Testing in a Communication System*

Ritabrata Roy

December 11, 2002

Abstract

This exercise aims at examining the basic ideas behind hypothesis testing and its role in a communication system. The binary data transmitted by a source is assumed to be corrupted by white Gaussian noise. At the receiver's end, a decision is made about the original transmitted signal using a simple binary hypothesis. The energy expended and the resulting bit error rate are obtained coresponding to different transmission strategies and variance levels in the noise. Based on this information, an optimum transmission strategy is proposed that provides a minimum bit error rate while maintaining a low power consumption.

1 Introduction

The terms detection, decision making, hypothesis testing, and decoding are almost synonymous. The word *detection* refers to the effort to decide whether some phenomenon is present or not on the basis of some observations. The meaning has been extended in the communication field to detect which one, among a set of mutually exclusive alternatives, is correct. *Decision making* is, again, the process of deciding between a number of mutually exclusive alternatives. *Hypothesis testing* is the same, and here the mutually exclusive

*Submitted as partial fulfillment of course requirement in Stochastic Signals and Systems (ECE 330:541) to Dr. Wade Trappe, Assistant Professor, Rutgers, The State University of New Jersey.

alternatives are called hypotheses. *Decoding* is the process of mapping the received signal into one of the possible set of code words or transmitted symbols [5, 1, 6].

Here a data source is assumed to emit a binary symbol (either a 'zero' or a 'one') every T seconds, for a total of $\frac{1}{T}$ symbols every second. During the transmission of each symbol from the source, a symbol 0 is transmitted as a signal $s_0(t)$, while a symbol $s_1(t)$ corresponds to the symbol 1. In a simple communication model, the transmitted signal may be assumed to be corrupted by white Gaussian noise $w(t)$.

For a receiver that samples the received signal at a rate of N samples per second, the decision hypotheses may be written as:

$$\begin{array}{lcll} H_0 & : & x(n) = s_0(n) + w(n) & \text{for} \quad n = 0, 1, \ldots, N-1 \\ H_1 & : & x(n) = s_1(n) + w(n) & \text{for} \quad n = 0, 1, \ldots, N-1 \end{array} \tag{1}$$

It is assumed that the sampling extracts the true value of the signals $s_j(t)$ and $w(t)$. The noise process may therefore be considered as an independent and identically distributed (iid) Gaussian random variable, $N(0, \sigma^2)$.

The aim of this exercise is to study the possible choices of $s_0(n)$ and $s_1(n)$ that will result in the smallest probability of decision error at different signal-to-noise ratio, SNR=$\frac{\|\mathbf{s}_j\|^2}{\sigma^2}$ where $\|\mathbf{s}\|$ is the Euclidean norm of $\mathbf{s}$.

1.1 Hypothesis Testing

For any binary channel, the transmitted signal over a symbol interval (0, T) is represented by:

$$s_i(t) = \begin{cases} s_0(t), & 0 \leq t \leq T \quad \text{for a binary 0} \\ s_1(t), & 0 \leq t \leq T \quad \text{for a binary 1} \end{cases} \tag{2}$$

The received signal $r(t)$ is typically degraded by noise $n(t)$ and possibly degraded by the impulse response of the channel $h_c(t)$ and may be written as [3]:

$$r(t) = s_i(t) \star h_c(t) + w(t), \qquad i = 1, \ldots, M \tag{3}$$

where $w(t)$ is assumed to be a zero-mean AWGN process and $\star$ represents a convolution operation. For binary transmission over an ideal distortionless channel, convolution with $h_c(t)$ produces no degradation (since $h_c(t)$ is an impulse function for the ideal case), and the representation of $r(t)$ simplifies to:

$$r(t) = s_i(t) + w(t), \qquad i = 1, 2, \qquad 0 \leq t \leq T \tag{4}$$

In the general case of Bayesian Hypothesis Testing, the two hypotheses concerning a real-valued observation Y can be written as [2]:

$$\begin{gathered} H_0 : Y = \epsilon + \mu_0 \\ \text{versus} \\ H_1 : Y = \epsilon + \mu_1 \end{gathered}$$

where ϵ is a Gaussian random variable with zero mean and variance σ^2, and where μ_0 and μ_1 are two fixed numbers with $\mu_1 > \mu_0$ (without loss of generality). In terms of distributions on the observation space, the hypothesis pair may be rewritten as:

$$\begin{gathered} H_0 : Y \sim N(\mu_0, \sigma^2) \\ \text{versus} \\ H_1 : Y \sim N(\mu_1, \sigma^2) \end{gathered}$$

where $N(\mu, \sigma^2)$ denotes the Gaussian(or normal) distribution with mean μ and variance σ^2.

Thus, the likelihood ratio is given by:

$$\begin{aligned} L(y) &= \frac{p_1(y)}{p_0(y)} && (5) \\ &= \frac{\frac{1}{\sqrt{2\pi}\sigma} e^{-(y-\mu_1)^2/2\sigma^2}}{\frac{1}{\sqrt{2\pi}\sigma} e^{-(y-\mu_0)^2/2\sigma^2}} && (6) \\ &= \exp\left\{ \frac{\mu_1 - \mu_0}{\sigma^2} \left(y - \frac{\mu_0 + \mu_1}{2} \right) \right\} && (7) \end{aligned}$$

Thus, the Bayes test is given by:

$$\delta_B(y) = \begin{cases} 1, & \text{if } \exp\left\{ \frac{\mu_1 - \mu_0}{\sigma^2} \left(y - \frac{\mu_0 + \mu_1}{2} \right) \right\} \geq \tau \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where τ is an appropriate threshold.

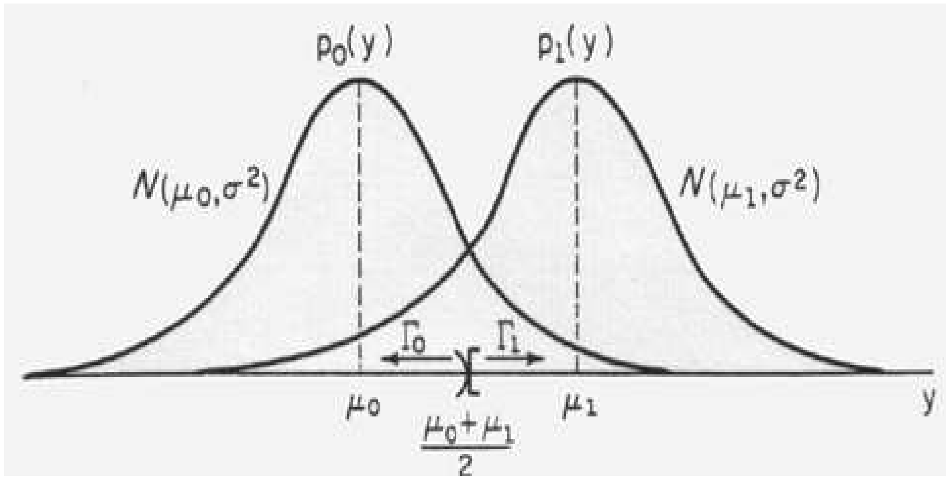

Figure 1: Illustration of Bayesian hypothesis testing with Gaussian error and uniform costs [2].

Rearranging the terms, we get:

$$\delta_B(y) = \begin{cases} 1, & \text{if } y \geq \tau^{'} \\ 0, & \text{if } y < \tau^{'} \end{cases} \quad (9)$$

where $\tau^{'} = \frac{\sigma^2}{\mu_1 - \mu_0} \log(\tau) + \frac{\mu_0 + \mu_1}{2}$.

Fig. 1 shows the case for uniform costs and equal priors, *i.e.*, $\tau = 1$ and $\tau^{'} = (\mu_0 + \mu_1)/2$. Thus, the Bayes rule compares the observation to the average of μ_0 and μ_1. If y is greater than or equal to the average, H_1 is chosen, while if y is less than this average h_1 is chosen.

1.2 Probability of Error and Bit Error Rate

Probability of error $P(e)$ and bit error rate (or BER) are often used interchangeably, although in practice they have slightly different meanings. P_e is a theoretical (mathematical) expectation of the bit error rate for a given system, whereas BER is an empirical (historical) record of a system's actual bit error performance [4]. For instance, if a system has a P_e of 10^{-5}, it means that mathematically, one bit error is expected in every 100,000 bits transmitted. However, if a system has a BER of 10^{-5}, it means that in the past there has been one bit error for every 100,000 bits transmitted. The bit error rate is measured, then compared to the expected probability of error to evaluate a system's performance.

1.3 Signal-to-Noise Ratio

The probability of error is a function of the average signal power to noise power ratio, more commonly referred to as the signal-to-noise ratio (or *SNR*). The SNR is the ratio of the average signal power (the combined power of the carrier and its associated sidebands) to the thermal noise power.

The thermal noise power is expressed mathematically as:

$$N = KTB \tag{10}$$

where N : thermal noise power, W
K : Boltzmann's proportionality constant, 1.38×10^{-23} J/K
T : temperature, K
B : bandwidth, Hz

While the SNR is a standard performance metric in analog communications, in digital communication systems it is more customary to use E_b/N_0, which is a normalized version of SNR. Here, E_b is the bit energy, described as signal power S times the bit time T_b, and N_0 is the noise power spectral density described as the noise power N divided by the bandwidth W. Since bit time T_b and bit rate R are reciprocals of each other, the metric E_b/N_0 may be interpreted as follows:

$$\frac{E_b}{N_0} = \frac{S}{N}\frac{B}{f_b} \tag{11}$$

where E_b/N_0 : energy per bit-to-noise power density ratio
S/N : signal-to-noise power ratio
B/f_b : noise bandwidth-to-bit ratio

A typical curve of Probability of Error versus E_b/N_0 is shown in Fig. 2.

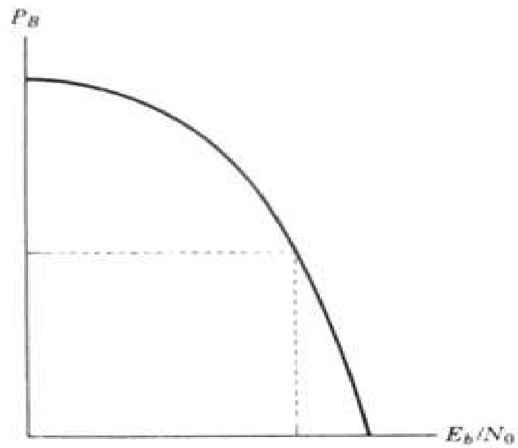

Figure 2: General shape of the P_B vs E_b/N_0 curve [3].

2 Analysis

This exercise calls for the simulation of the binary hypothesis testing model for different choices of s_0 and s_1. In addition, for each choice of these parameters, the performance of the testing procedure is obtained by varying the values of the SNR as well as the number of samples per symbol. Three different hpothesis schemes are proposed in the following subsections.

2.1 On–Off Keying

On–off keying (OOK), or unipolar amplitude keying, is arguably the simplest form of communication. The transmitter uses $s_0(n) = 0$ to represent a "zero" and $s_1(n) = A$ to represent a "one", where A is the amplitude of the transmitted signal. Thus, effectively the transmitter is off when the a zero symbol is being transmitted, which explains the name of the scheme. While the resulting communication leads to conservation of energy, a serious flaw is that when the receiver does not detect any signal, it has no means of knowing whether a stream of zero bits is being transmitted or whether there is a problem with the communication channel.

A simple case of the on-off keying scheme has been illustrated in Fig. 3,

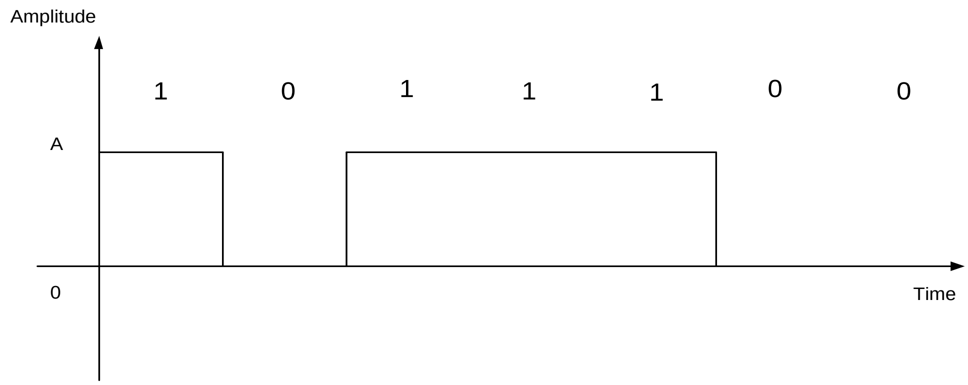

Figure 3: Illustration of a simple on-off keying (OOK) scheme where the transmitted code sequence is 1011100.

where the symbol sequence is 1011100.

The hypothesis relations for this scheme may be written as:

$$\begin{array}{llll} H_0 & : & x(n) = s_0(n) + w(n) & \text{for} \quad n = 0, 1, \ldots, N-1 \\ H_1 & : & x(n) = s_1(n) + w(n) & \text{for} \quad n = 0, 1, \ldots, N-1 \end{array} \tag{12}$$

where $s_0(n) = 0$
$s_1(n) = A$

If zero and one symbols are equiprobable, then the expected energy expended by this keying scheme is given by:

$$\mathrm{E}[\varepsilon] = s_0^2 P(s_0) + s_1^2 P(s_1^2) \tag{13}$$

$$= 0.5(0^2 + 1^2) \tag{14}$$

$$= 0.5 \tag{15}$$

This strategy was implemented in MATLAB (program code included in Appendix A) and the plot of bit error rate was obtained as a function of the signal-to-noise ratio. Fig. 4 and Fig. 5 illustrate two of these case for different values of the number of samples, *viz.*, n=2 and n=5. In both cases, the total number of symbols transmitted is 10000 and the SNR is varied between 1 dB and 8 dB. The amplitude A is given the value 1.

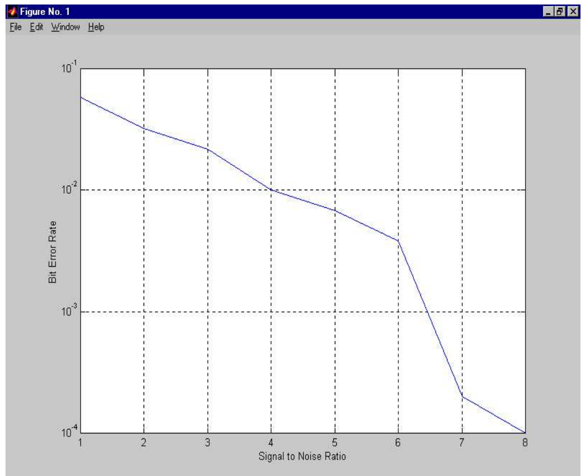

Figure 4: MATLAB plot of bit error rate as a function of signal-to-noise ratio for a unipolar amplitude keying (on–off keying) scheme with 2 samples/symbol.

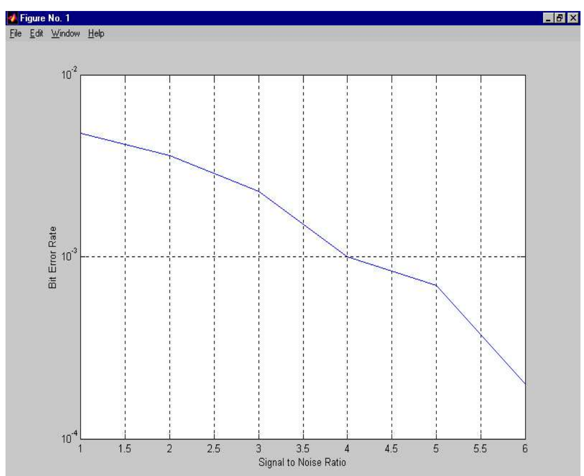

Figure 5: MATLAB plot of bit error rate as a function of signal-to-noise ratio for a unipolar amplitude keying (on–off keying) scheme with 5 samples/symbol.

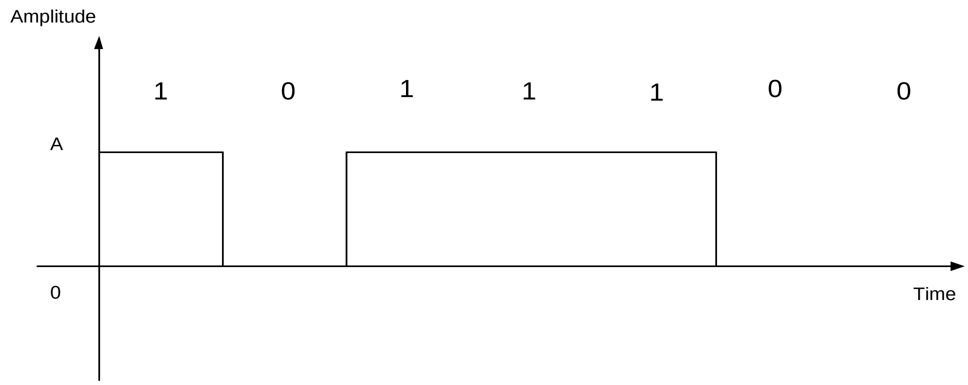

Figure 6: Illustration of a simple bipolar amplitude keying scheme where the transmitted code sequence is 1011100.

2.2 Bipolar Amplitude Keying

Bipolar amplitude keying is a slight modification of the on–off keying scheme discussed earlier. Instead of representing a binary zero with no signal, this scheme uses $-A$ to represent the zero pulse, where A is the amplitude of the signal used to represent binary one.

Thus, the hypothesis relations for this scheme may be written as:

$$\begin{array}{lcll} H_0 & : & x(n) = s_0(n) + w(n) & \text{for} \quad n = 0, 1, \ldots, N-1 \\ H_1 & : & x(n) = s_1(n) + w(n) & \text{for} \quad n = 0, 1, \ldots, N-1 \end{array} \tag{16}$$

where $s_0(n) = -A$
$s_1(n) = A$

A simple case of the bipolar amplitude keying scheme has been illustrated in Fig. 6, where the symbol sequence is 1011100.

This improves over on–off keying by reducing the probability of error in the presence of noise, since symbols zero and one are now separated by $2A$ instead of A, making it less likely for the error threshold to be crossed. It also makes the scheme less ambiguous— since a zero symbol is transmitted as $-A$, when there is no signal, the receiver interprets it as no data being transmitted (although it might also be an effect of poor channel conditions). The tradeoff for the improvement in the bit error rate is a higher consumption of energy, since even a zero pulse is represented by a signal of non-zero amplitude in this scheme.

If zero and one symbols are equiprobable, then the expected energy expended by this keying scheme is given by:

$$\begin{aligned} \mathrm{E}[\varepsilon] &= s_0^2 P(s_0) + s_1^2 P(s_1^2) \qquad (17) \\ &= 0.5(1^2 + 1^2) \qquad (18) \\ &= 1 \qquad (19) \end{aligned}$$

It is observed that the expected energy consumption in this scheme is twice that of the on–off keying scheme.

Since the aim of this transmission mechanism is to reduce the error rate by increasing the "distance" between a transmitted zero symbol and a one symbol, it is instructive to look at a modified form of the on–off keying where the pulses zero and one are represented by 0 and $2A$, *i.e.*, the separation between antipodal symbols is still $2A$. Putting $A = 1$ and assuming equiprobable zeros and ones, the expected energy would be:

$$\begin{aligned} \mathrm{E}[\varepsilon] &= s_0^2 P(s_0) + s_1^2 P(s_1^2) \qquad (20) \\ &= 0.5(0^2 + 2^2) \qquad (21) \\ &= 2 \qquad (22) \end{aligned}$$

Thus, the expected energy consumption in this scheme is twice that of the bipolar amplitude keying scheme, and therefore does not offer any advantages.

The bipolar amplitude keying strategy was implemented in MATLAB (program code included in Appendix B) and the plot of bit error rate was obtained as a function of the signal-to-noise ratio. Fig. 7 and Fig. 8 illustrate two of these case for different values of the number of samples, *viz.*, n=2 and n=5. In both cases, the total number of symbols transmitted is 10000 and the SNR is varied between 1 dB and 8 dB. The data symbols 0 and 1 are represented by $-A = -1$ and $A = 1$ respectively.

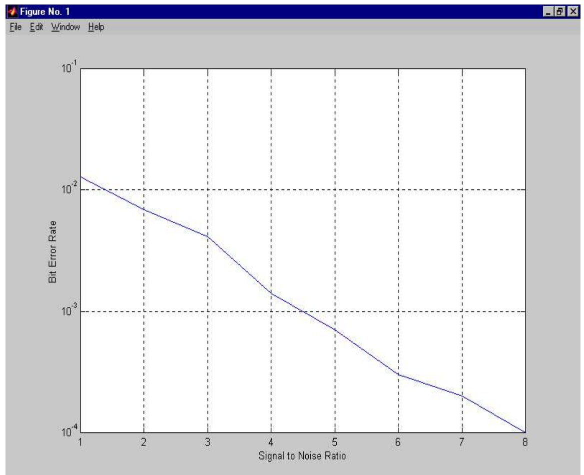

Figure 7: MATLAB plot of bit error rate as a function of signal-to-noise ratio for a bipolar amplitude keying scheme with 2 samples/symbol.

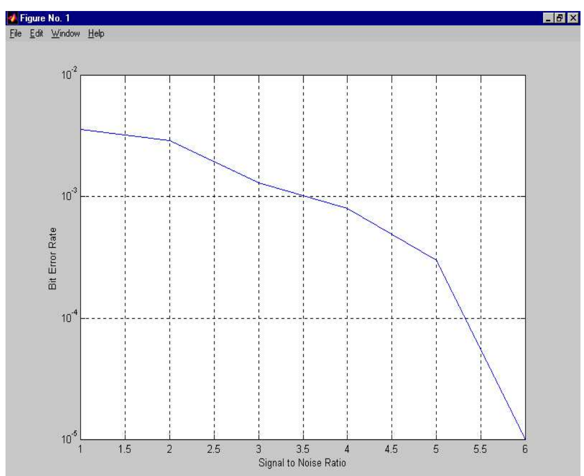

Figure 8: MATLAB plot of bit error rate as a function of signal-to-noise ratio for a bipolar amplitude keying scheme with 5 samples/symbol.

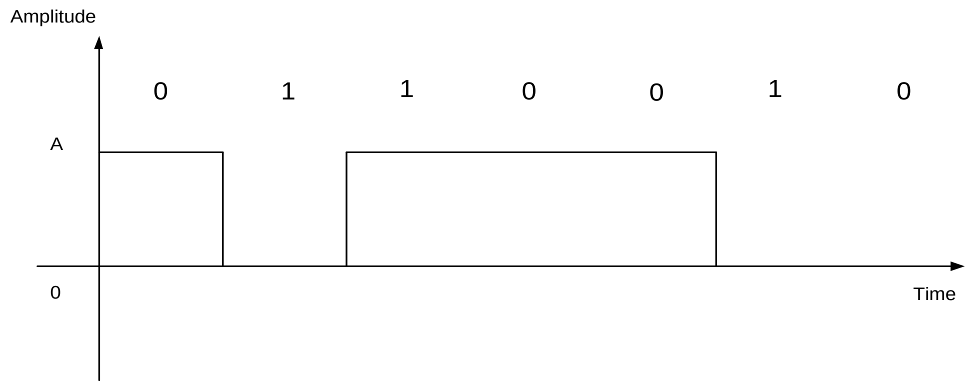

Figure 9: Illustration of a simple phase–change amplitude keying scheme where the generated symbol sequence is 1011100 and the transmitted code sequence is 0110010.

2.3 Phase–Change Amplitude Keying

The third scheme being proposed is a subtle modification of the on–off keying scheme — it sends a symbol A only when there is a change of phase (either from bit 0 to bit 1, or vice versa) and shuts off transmission when there is no change of phase in consecutive data bits. As revealed by the MATLAB code in Appendix C, this requires the original data stream to be coded to carry this information (assuming that the initial signal is high) and then falls into the category of on–off keying. The advantage of this scheme is that it consumes less energy for certain types of data stream where toggling of data between 0 and 1 is relatively infrequent. However, a major disavantage is that if a particular symbol is misinterpreted by the receiver, it has a domino effect on the subsequent symbols *until another error occurs*. This limits the usefulness of the procedure without adequate safeguarding techniques, and is included here only for instructive purposes— it may also be pointed out that for a random set of data that has no particular "pattern", it does not offer energy advantages either.

A simple case of the phase–change amplitude keying scheme has been illustrated in Fig. 9, where the symbol sequence 1011100 is coded as 0110010 (note that it has reduced the number of ones, and hence the transmission energy).

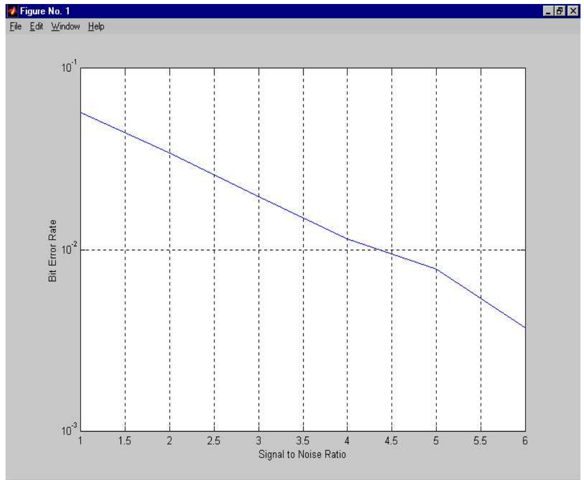

Figure 10: MATLAB plot of bit error rate as a function of signal-to-noise ratio for a phase–change amplitude keying scheme with 2 samples/symbol.

The phase–change amplitude keying strategy was implemented in MATLAB (program code included in Appendix C) and the plot of bit error rate was obtained as a function of the signal-to-noise ratio. Fig. 10 and Fig. 11 illustrate two of these case for different values of the number of samples, *viz.*, n=2 and n=5. In both cases, the total number of symbols transmitted is 10000 and the SNR is varied between 1 dB and 6 dB. A change of phase (either from 0 to 1 or from 1 to 0) is represented by $A = 1$, while $A = 0$ represents no change in phase.

3 Conclusion

The performance of the different transmission schemes outlined in the previous section were compared according to the variation of the bit error rate with the signal-to-noise ratio. The table below summarises the power consumption in each of the methods, for different numbers of samples.

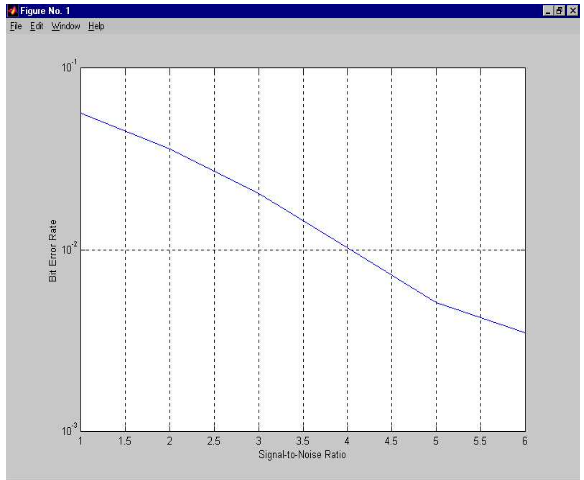

Figure 11: MATLAB plot of bit error rate as a function of signal-to-noise ratio for a bipolar amplitude keying scheme with 5 samples/symbol.

Method	No. of Samples	No. of Symbols	Power Consumed
On–Off Keying	2	10000	0.4960
	5	10000	0.5007
Bipolar Amplitude Keying	2	10000	1.0000
	5	10000	1.0000
Phase–Change Amplitude Keying	2	10000	0.5042
	5	10000	0.4998

As is expected, the bipolar amplitude keying transmission consumes approximately twice the amount of energy as the other two schemes since the transmitter expends the same amount of energy regardless of whether a zero bit or a one bit is transmitted. On the other hand, on–off keying and phase–change amplitude keying consume approximately the same amount of power since they are built upon the same model; the subtle differences in the energy values may be attributed to different random number sequences being generated. It may be pointed out, however, that if the generated data had a specific pattern (long periods of ones followed by long periods of zeros), then the phase–change amplitude keying technique would be much energy-efficient than the bipolar amplitude keying method.

The other issues that deserve careful consideration include the following:

- **The amount of energy used in transmitting a symbol.**
  As has been observed in Section 2.2, the bipolar amplitude keying technique consumes the same amount of energy irrespective of the symbol being transmitted. However, both on–off keying and phase–change keying represent a zero with a lack of a signal, thus consuming no energy in the process, and reducing the net energy consumption.

- **Performance at different signal-to-noise ratios.**
  This has been studies extensively in Section 2 and it was seen that the bit-error-rate falls with increase of the signal-to-noise ration resulting in the classic waterfall curve.

- **Dependence on the number of samples per symbol.**
  The bit error rate versus signal-to-noise ratio curve was plotted for two different values of the number of samples per symbol, *viz.* $n = 2$ and $n = 5$. There is a very slight fall in the bit error rate with increase in the number of samples, which can be attributed to the fact that the detection decision is made over a larger number of measurements, which reduces the probability of error. The difference may be expected to be more marked for a larger increase in the number of samples per symbol, but hardware restrictions limited that approach.

- **Dependence on the likelihood of symbols zero and one.**
  There is no significant difference in the plot of bit error rate versus signal-to-noise ratio when the probability of zero is changed. For instance, Fig. 12 and Fig. 13 plot the BER versus SNR curve for on–off keying when the probability of zero is increased to 0.7 and 0.9 respectively, and there is no noticeable difference in the results compared to those in Section 2.1. However, since the symbol zero (which is now more abundant) requires no energy for transmission, there is a substantial decrease in the energy consumption. A similar power behaviour is seen in the case of phase-change amplitude keying. However, since bipolar amplitude keying utilises the same amount of energy to transmit a zero as to transmit a one, there is no change in the energy characteristics with change in the probability of zero. These results are tabulated below.

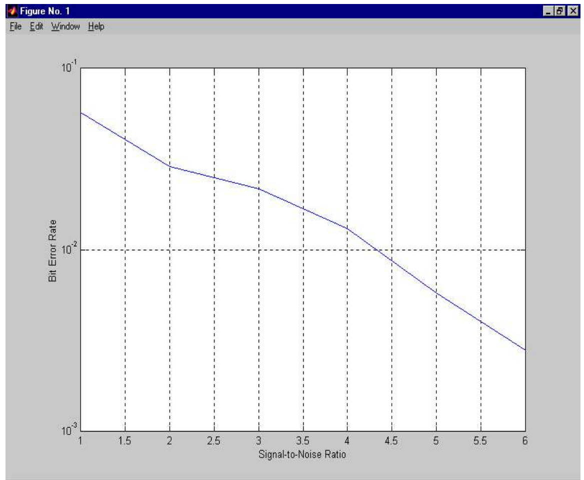

Figure 12: MATLAB plot of bit error rate as a function of signal-to-noise ratio for an on–off keying scheme with 2 samples/symbol and $P(0) = 0.7$.

Method	n	Power $P(0) = 0.5$	Power $P(0) = 0.7$	Power $P(0) = 0.9$
On–off Keying	2	0.4960	0.3011	0.1008
	5	0.5007	0.2969	0.1023
Bipolar Keying	2	1.0000	1.0000	1.0000
	5	1.0000	1.0000	1.0000
Phase-change Keying	2	0.5042	0.4260	0.1702
	5	0.4998	0.4263	0.1758

In conclusion, it is not possible to minimise the bit-error-rate *and* the energy consumption since they have contradictory requirements. Instead, the optimum transmission mechanism should be able to provide sufficiently robust quality and reliability while minimising the energy consumed. Under these conditions, the on–off keying method is probably the best choice for purely random data since it creates a marked improvement in energy efficiency, while not causing a substantial degradation in the bit error rate. If however, the transmission power is not an issue, then bipolar amplitude keying provides more robust performance. It has also been proposed that if the

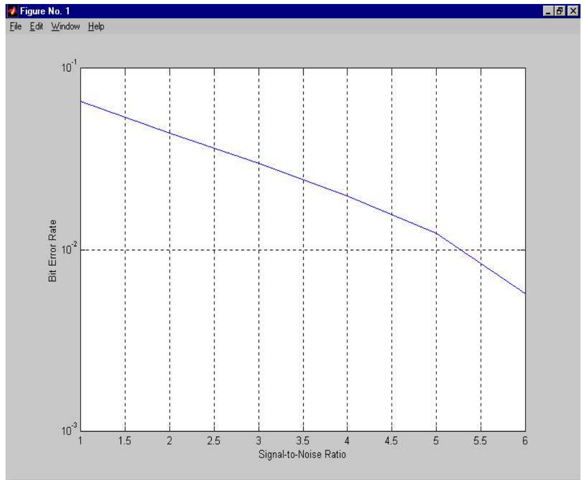

Figure 13: MATLAB plot of bit error rate as a function of signal-to-noise ratio for an on–off keying scheme with 2 samples/symbol and $P(0) = 0.9$.

data exhibits a particular repetitive pattern, then application of the phase–change amplitude keying technique would provide a marked improvement in energy efficiency while maintaining the bit error rate at the level of on–off keying.

A Appendix

The appendix lists the source codes used for simulating the three algorithms discussed in this report. The programs were written in MATLAB version 5.2.0.3084, and were executed in a *Microsoft Windows 98SE* environment.

A.1 On–Off Keying

```
% Hypothesis Testing
n=2; % number of samples per symbol
k=10000; % number of symbols transmitted
A=1; % amplitude of pulse
p=0.5; % probability of zero occurring
s0=0; % corresponding to symbol "zero"
```

```
s1=A; % corresponding to symbol "one"
thresh=0.5*(s0+s1); % threshold for estimation
pwr=0.5*(s0*s0+s1*s1) % average power
error=0; % number of errors
%Outer loop of SNR
for SNRdB=1:6
SNR=10(SNRdB/10);
sigma=sqrt(pwr/SNR);
% Generation of the AWGN noise
N=n*k;
for i=1:N
u1=rand(1);
u2=rand(1);
x1=sqrt(-2*log(u1))*cos(2*pi*u2);
x2=sqrt(-2*log(u2))*cos(2*pi*u1);
rv1(i,1)=x1;
rv2(i,1)=x2;
normvarx(i)=0.5*(rv1(i,1)+rv2(i,1));
end
AWGN=sigma*normvarx;
counter=0;
%Generation of symbols
for i=1:k
varsum=0;
y=unifrnd(0,1);
if (y>p)
symbol(i)=s1;
else
symbol(i)=s0;
end
%Inner loop for samples
for j=1:n
x(j)=symbol(i)+AWGN(n*i+j-n);
varsum=varsum+x(j);
end
if y>p
if varsum/n>=thresh
counter=counter+1;
```

```
end
else
if varsum/n<thresh
counter=counter+1;
end
end
end
numerror=k-counter;
errorprob(SNRdB)=numerror/k;
end
SNRdB=1:6;
semilogy(SNRdB, errorprob)
grid on
xlabel('Signal-to-Noise Ratio')
ylabel('Bit Error Rate')
pwr=(symbol*symbol')/k
```

A.2 Bipolar Amplitude Keying

```
% Hypothesis Testing
n=2; % number of samples per symbol
k=10000; % number of symbols transmitted
A=1; % amplitude of pulse
p=0.5; % probability of zero occurring
s0=-A; % corresponding to symbol "zero"
s1=A; % corresponding to symbol "one"
thresh=0.5*(s0+s1); % threshold for estimation
pwr=0.5*(s0*s0+s1*s1) % average power
error=0; % number of errors
%Outer loop of SNR
for SNRdB=1:6
SNR=10(SNRdB/10);
sigma=sqrt(pwr/SNR);
% Generation of the AWGN noise
N=n*k;
for i=1:N
u1=rand(1);
```

```
u2=rand(1);
x1=sqrt(-2*log(u1))*cos(2*pi*u2);
x2=sqrt(-2*log(u2))*cos(2*pi*u1);
rv1(i,1)=x1;
rv2(i,1)=x2;
normvarx(i)=0.5*(rv1(i,1)+rv2(i,1));
end
AWGN=sigma*normvarx;
counter=0;
%Generation of symbols
for i=1:k
varsum=0;
y=unifrnd(0,1);
if (y>p)
symbol(i)=s1;
else
symbol(i)=s0;
end
%Inner loop for samples
for j=1:n
x(j)=symbol(i)+AWGN(n*i+j-n);
varsum=varsum+x(j);
end
if y>p
if varsum/n>=thresh
counter=counter+1;
end
else
if varsum/n<thresh
counter=counter+1;
end
end
end
numerror=k-counter;
errorprob(SNRdB)=numerror/k;
end
SNRdB=1:6;
semilogy(SNRdB, errorprob)
```

```
grid on
xlabel('Signal-to-Noise Ratio')
ylabel('Bit Error Rate')
pwr=(symbol*symbol')/k
```

A.3 Phase–Change Amplitude Keying

```
% Hypothesis Testing
n=2; % number of samples per symbol
k=10000; % number of symbols transmitted
A=1; % amplitude of pulse
p=0.5; % probability of zero occurring
s0=0; % corresponding to symbol "zero"
s1=A; % corresponding to symbol "one"
thresh=0.5*(s0+s1); % threshold for estimation
pwr=0.5*(s0*s0+s1*s1) % average power
error=0; % number of errors
%Outer loop of SNR
for SNRdB=1:6
SNR=10(SNRdB/10);
sigma=sqrt(pwr/SNR);
% Generation of the AWGN noise
N=n*k;
for i=1:N
u1=rand(1);
u2=rand(1);
x1=sqrt(-2*log(u1))*cos(2*pi*u2);
x2=sqrt(-2*log(u2))*cos(2*pi*u1);
rv1(i,1)=x1;
rv2(i,1)=x2;
normvarx(i)=0.5*(rv1(i,1)+rv2(i,1));
end
AWGN=sigma*normvarx;
counter=0;
%Generation of symbols
for i=1:k+1
varsum=0;
```

```
y=unifrnd(0,1);
if (y>p)
symb(i)=s1;
else
symb(i)=s0;
end
end
for i=1:k
if (symb(i)+symb(i+1)==(s1+s0))
symbol(i)=s1;
else
symbol(i)=s0;
end
%Inner loop for samples
for j=1:n
x(j)=symbol(i)+AWGN(n*i+j-n);
varsum=varsum+x(j);
end
if y>p
if varsum/n>=thresh
counter=counter+1;
end
else
if varsum/n<thresh
counter=counter+1;
end
end
end
numerror=k-counter;
errorprob(SNRdB)=numerror/k;
end
SNRdB=1:6;
semilogy(SNRdB, errorprob)
grid on
xlabel('Signal-to-Noise Ratio')
ylabel('Bit Error Rate')
pwr=(symbol*symbol')/k
```

References

[1] Alberto Leon-Garcia. *Probability and Random Processes for Electrical Engineering*. Addison-Wesley, Reading, MA, 1994.

[2] H. Vincent Poor. *An Introduction to Signal Detection and Estimation*. Springer Verlag, New York, NY, 1998.

[3] Bernard Sklar. *Digital Communications*. Pearson Education Asia, Singapore, 2001.

[4] Wayne Tomasi. *Electronic Communication Systems*. Prentice Hall, Upper Saddle River, NJ, 1998.

[5] David Tse and Robert Gallager. Detection Notes. Technical report, Massachussetts Institute of Technology, 2002.

[6] Roy D. Yates and David J. Goodman. *Probability and Stochastic Processes*. John Wiley, New York, NY, 1999.