FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING
MASTER'S PROGRAMME IN WIRELESS COMMUNICATIONS ENGINEERING

LABORATORY WORK REPORT

Wireless Communications I 521395S

Student	Aissa Azzaz
OID	1.2.246.562.24.15372681760

October 26, 2022

CONTENTS

LIST of FIGURES

LIST of TABLES

LIST of SYMBOLS and ABBREVIATIONS

$\frac{E_b}{N_0}$	Bit energy to noise density
λ	Wavelength
f_D	Doppler Spread
P_e	Probability of Error (P_e or BER)
AWGN	Additive White Gaussian Noise
BCH	Bose–Chaudhuri–Hocquenghem (type of coding)
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
dB	Decibels (relative to a 1 Watt)
dBm	Decibels (relative to a 1 mWatt)
DPSK	Differential Phase Shift Keying
ISI	Inter Symbol Interference
LOS	Line of Sight
MRC	Maximum Ratio Combining
NLOS	Non-Line of Sight
PDF	Probability Density Function
PSK	Phase Shift Keying
QAM	Quadrature Amplitude Modulation
SDR	Software Defined Radio
SER	Symbol Error Rate
SNR	Signal to Noise Ratio

1 SIGNAL PROPAGATION

1.1 TWO-RAY MODEL

The provided *TwoRayModel.m* Matlab file simulates a two-ray model problem. The parameters of the problem are Transmitted power P_t, Transmitter height h_t, Receiver height h_r,Separation distance d, Antenna gains product G_l, Field gains product G_R, Frequency f =, Speed of light c, Ground reflection R.

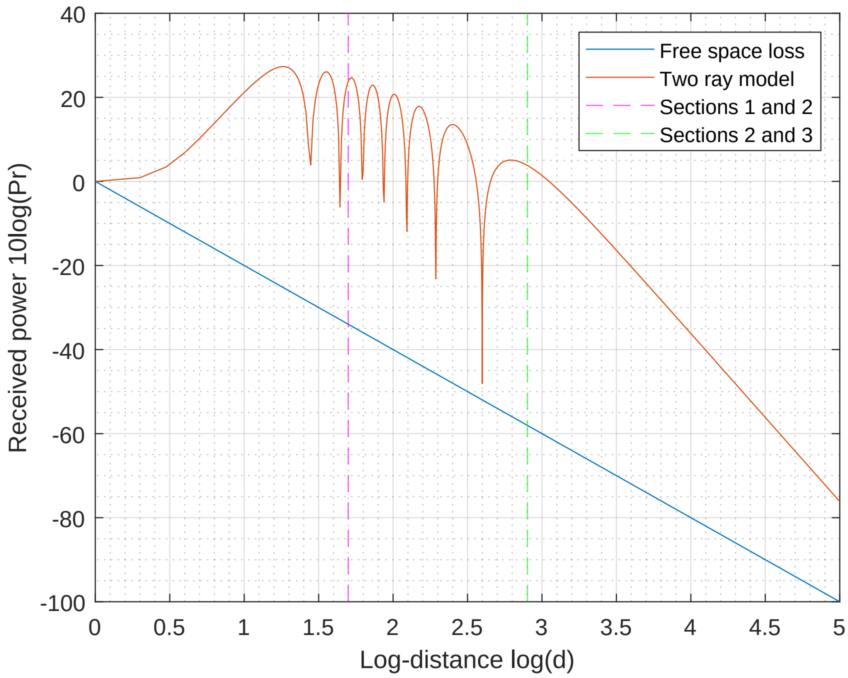


Figure 1.1: Received power versus distance with free space path loss model and two-ray model.

The simulation generates Figure 1.1, which shows the received power P_t vs transmitter-receiver separation distance d with a free space path loss channel and a two ray modeled channel.

The free space path loss model shows that the power decays linearly with $\log(d)$ and with a constant slope. The two-ray model could be divided into three sections. The first section (before the magenta dashed line) ends at $\log_{10}(d) = 1.7 \approx \log_{10}(h_t)$, in which the received power increases because the two rays constructively add up, and therefore the channel enforces a gain on the transmitted signal.

The second section (between the magenta and green dashed lines) starts from a distance greater than h_t and ends at some critical distance d_c. In this section, the received power varies drastically with the distance due to the two rays adding up constructively and destructively with different phases which results in small scale fading as illustrated by the various minima and maxima which are decaying by -20 dB/decade.

In the third section (after the green dashed line), the received power falls off faster than the free space path loss model and this due to the two ray adding up destructively after the critical distance d_c.

The critical distance d_c

The critical distance is the distance at which the two rays only add up destructively. In the Figure 1.1, the critical distance is marked by the green dashed line at $\log_{10}(d_c) = 2.9031$ and therefore, $d_c = 800m$.

The critical distance is verified by calculation [1, p, 36] as

$$d_c = \frac{4h_t h_r}{\lambda} = 800m$$

1.2 RAY TRACING IN URBAN ENVIRONMENT

The provided *OuluRaytrace.m* Matlab file contains multiple code sections. The first section opens *siteviewer* and loads a 3D map of Oulu city center, Figure 1.2, with data exported from openstreetmap.org.

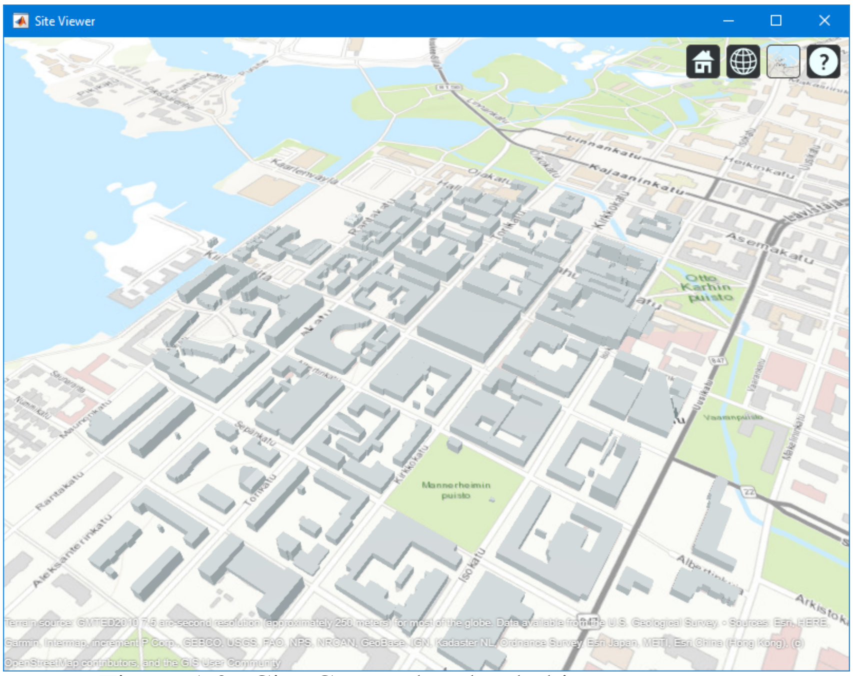


Figure 1.2: City Center data loaded into *siteviewer*.

The second section, Figure 1.3, places a transmitter at a certain location.

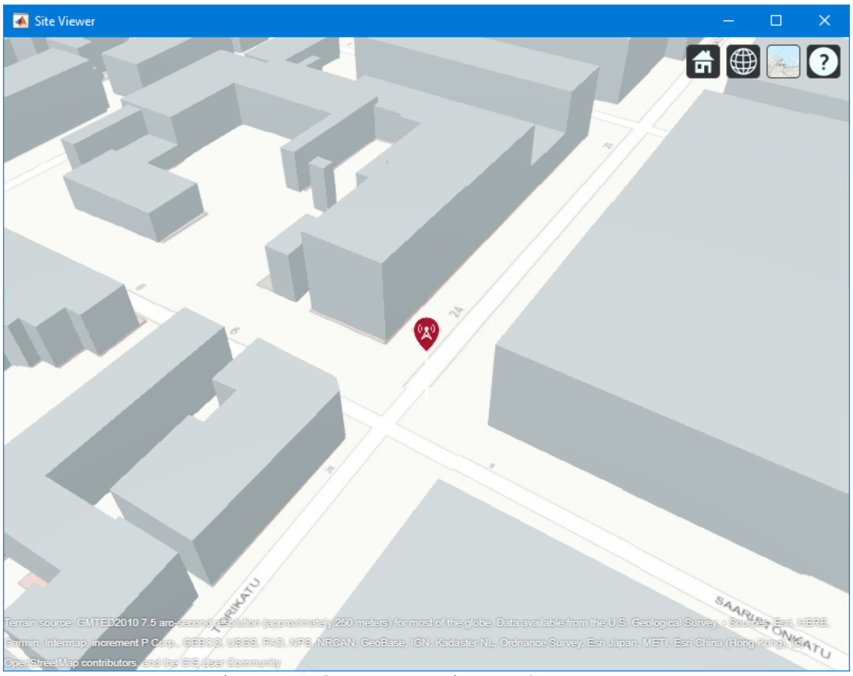


Figure 1.3: Transmitter placement.

The third section sets the parameters for ray tracer (in this case 0 reflections for line of sight), and calculates line of sight coverage for the transmitter antenna restricted to 200 meters, Figure 1.4.

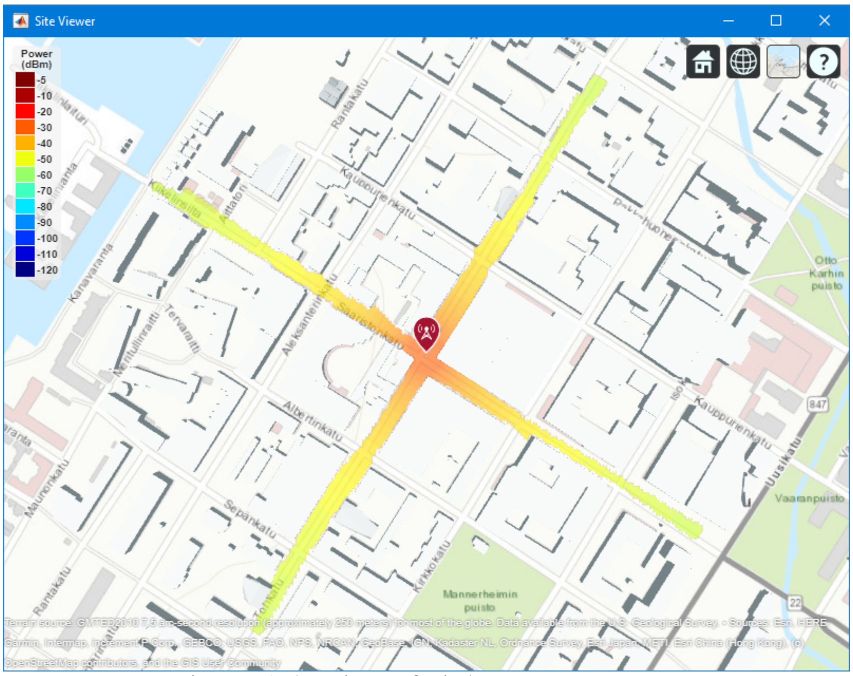


Figure 1.4: Line-of-sight coverage map.

The fourth section sets the location and parameters of receiver antenna outside line of sight,

Figure 1.5.

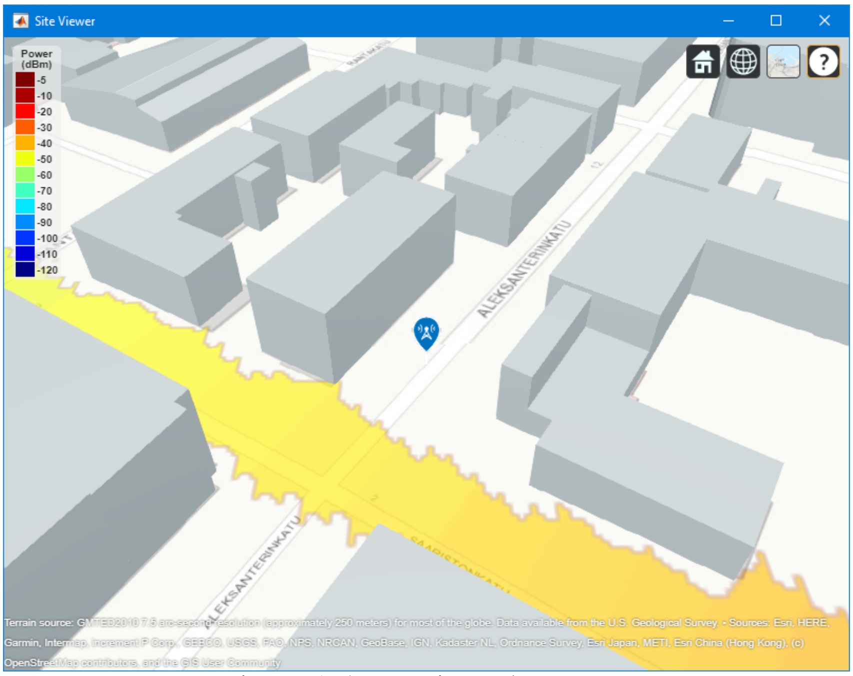


Figure 1.5: Receiver placement.

The fifth section uses the ray tracer to calculate the reflection path (.i.e non-LOS ray trace) and the power of the received signal, Figure 1.6. The non-LOS received power is $P_{RX1} = -63.5455 dBm$.

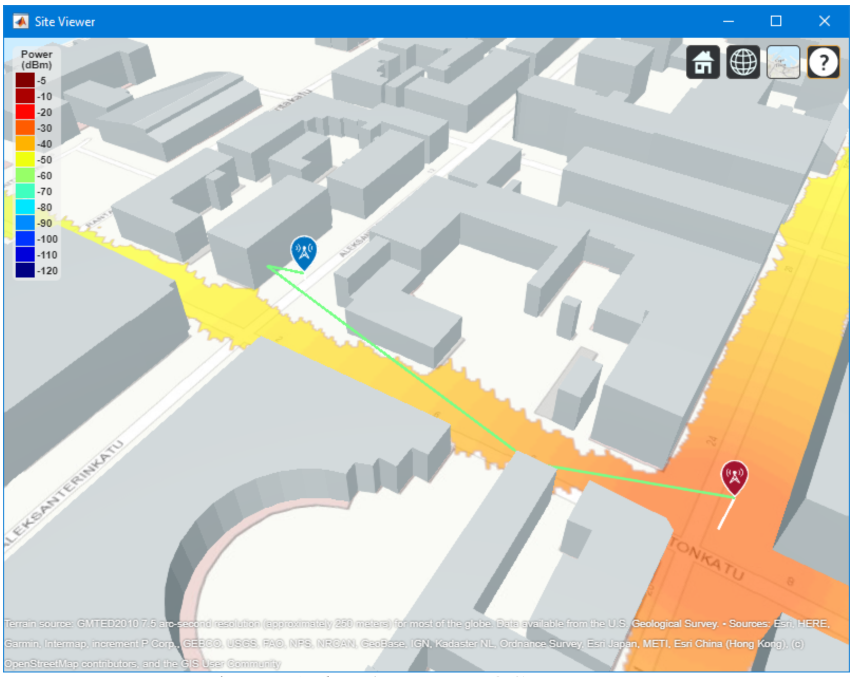


Figure 1.6: The non-LOS ray trace.

The sixth section clears the *siteviewer* and sets a new receiver in line of sight, Figure 1.7.

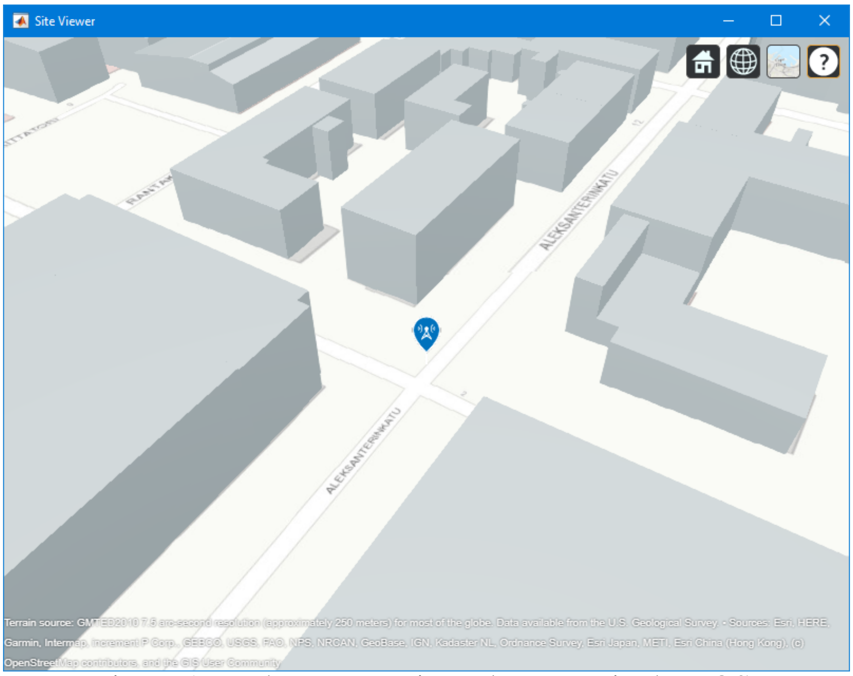


Figure 1.7: The new receiver placement in the LOS.

The seventh section calculates received power at the new receiver due to a single ray (.i.e the first line of sight trace), Figure 1.8. The LOS received power by the new receiver is $P_{RX2,0} = -44.4773 dBm$.

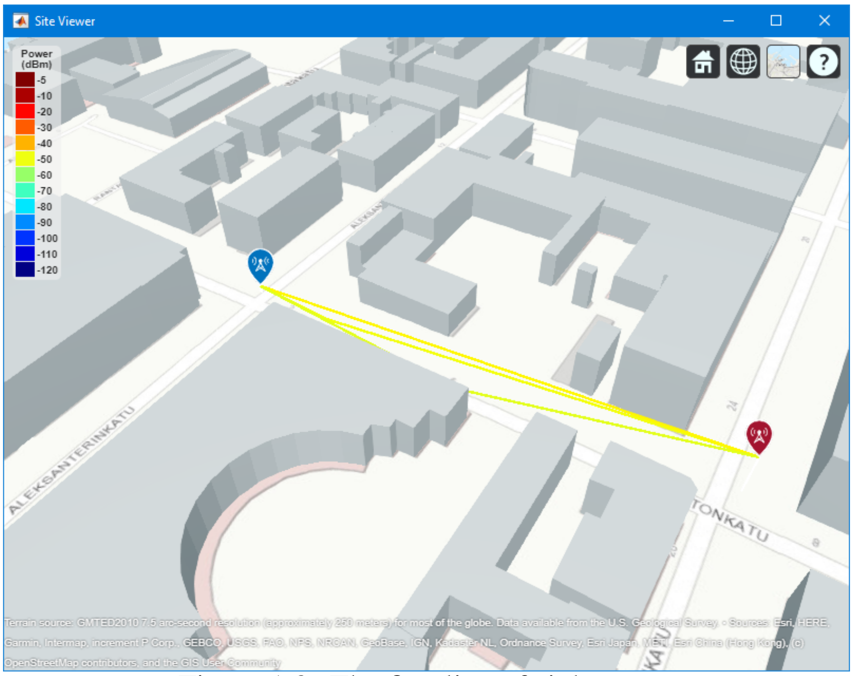


Figure 1.8: The first line of sight trace.

The eighth section sets the parameters of the ray tracer to get 9 reflections, Figure 1.9, and calculates the received power. With the nine reflections enabled, the received power is $P_{RX2,9} = -43.3401 dBm$.

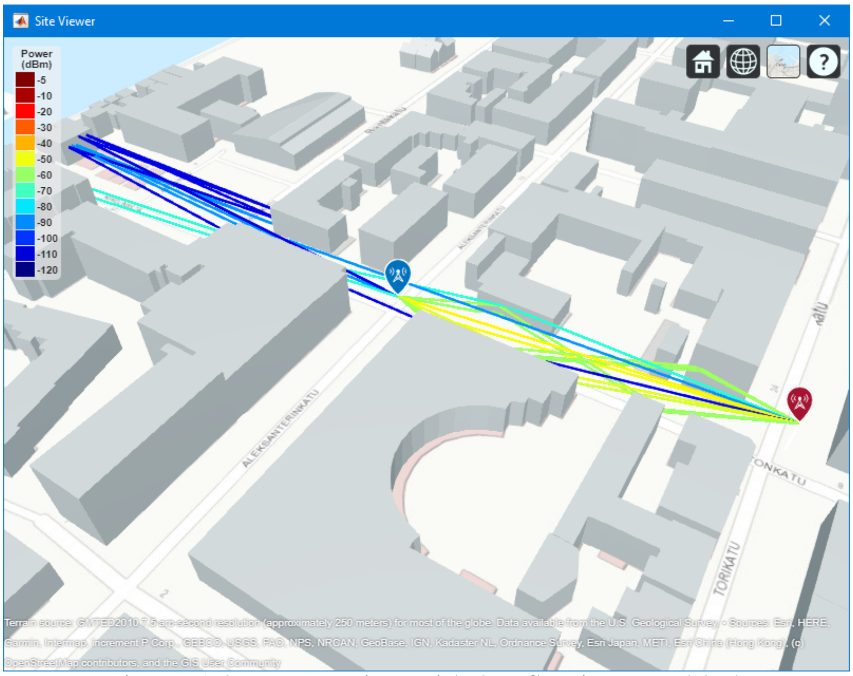


Figure 1.9: Ray tracing with 9 reflections enabled.

When the receiver was positioned in the LOS, the received power with no reflections enabled was $P_{RX2,0} = -44.4773 dBm$, however, when it was placed in a non-LOS position, the received power was $P_{RX1} = -63.5455 dBm$. The received power increased when we changed the receiver from the non-LOS to LOS position. This is because the line-of-sight ray experiences a shorter and direct path distance which reduces its path loss compared with the non-LOS rays and because the non-LOS rays don't fully bounce off the walls with all their power to reach the non-LOS receiver.

Without reflections, the power received by the LOS receiver was $P_{RX2,0} = -44.4773 dBm$, and with 9 reflections enabled this power was $P_{RX2,9} = -43.3401 dBm$. There is a small increase of $1.1372 dBm$ when adding reflections, but it is not a significant change. This because the added reflected rays don't contribute significantly to the received power since they lost a considerable portion of it in the longer path they took and when bouncing off walls which have a certain reflection coefficient that determines the ratio of the absorbed power by the wall to the power directed back to the receiver path.

2 MPSK IN AWGN CHANNEL

2.1 BPSK IN AWGN CHANNEL

The *MPSK_in_AWGN.slx* Simulink model, shown in Figure 2.1, represents a model for a communication system in an AWGN channel. In this model, a random number generator is used to generate random integers which are then converted to bits and transmitted through the AWGN channel model after being modulated using an MPSK modulator. After the channel, an MPSK demodulator extracts the bits and a convertor converts them to integers. Also, two error rate calculation blocks are used, the one on the top for the symbols (.i.e the integers) and it outputs the SER, and the one on the bottom for the bits, and it outputs the BER.

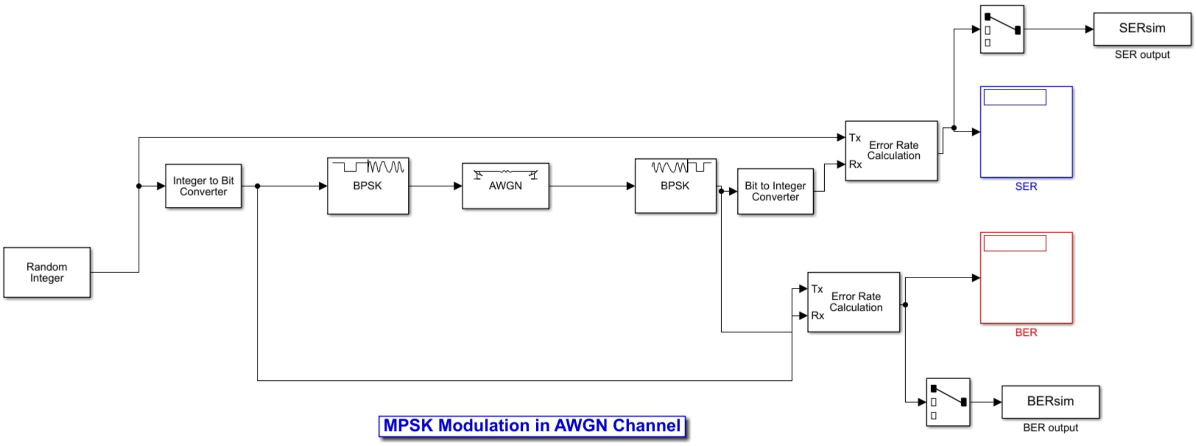


Figure 2.1: MPSK_in_AWGN.slx Simulink file

The *MPSK_control.m* Matlab file is used to run the *MPSK_in_AWGN.slx* Simulink model and controls its parameters. In this part, M is set to 2, and this sets the modulation type to be BPSK. The file also runs the Simulink model with various SNR ($\frac{E_b}{N_0}$) levels to demonstrate the effect of the AWGN channel on the BER performance of the chosen modulation.

The BER performance of the BPSK modulation for the different SNR levels is presented on Table 2.1 and plotted in Figure 2.3. The BER decreases logarithmically as the SNR value increases where the BER is reduced by one decade approximately with each 2 dB increase in SNR.

Table 2.1: Effect of SNR on BER in BPSK

Eb/No (dB)	BER
0	0.0816
1	0.0569
2	0.0381
3	0.0223
4	0.0122
5	0.0059
6	0.0023
7	0.0007

To compare the simulation data, Matlab's *bertool* is used. The tool is configured to reassemble the simulation conditions, as shown in Figure 2.2, and its results are compared against the simulation in Figure 2.3. The simulation results totally match the theoretical results generated using the BER tool.

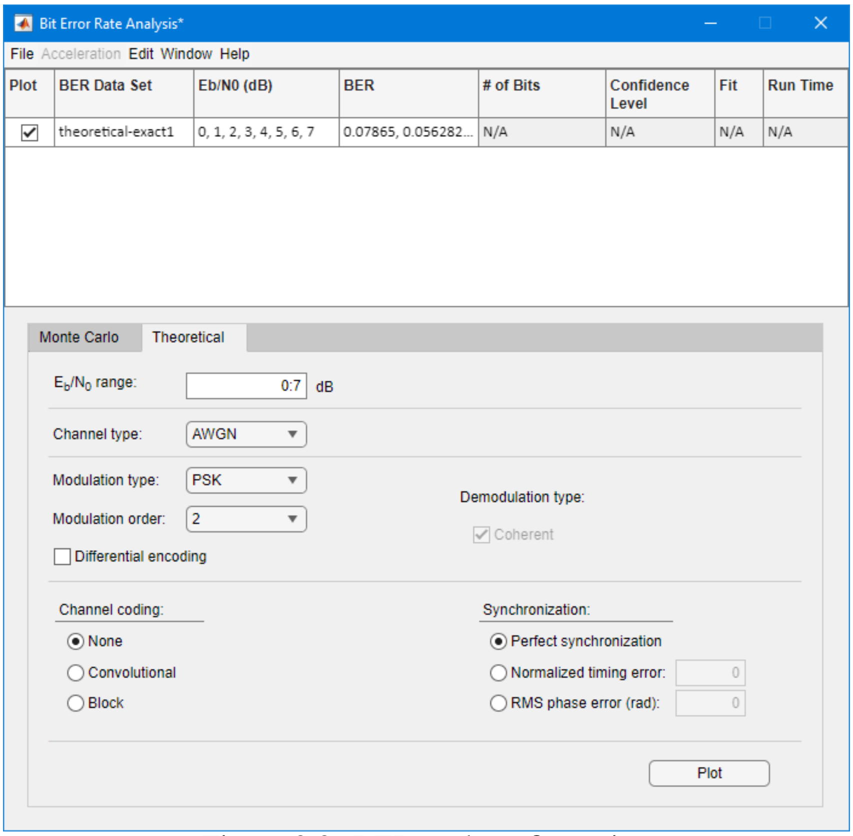


Figure 2.2: BER tool configuration

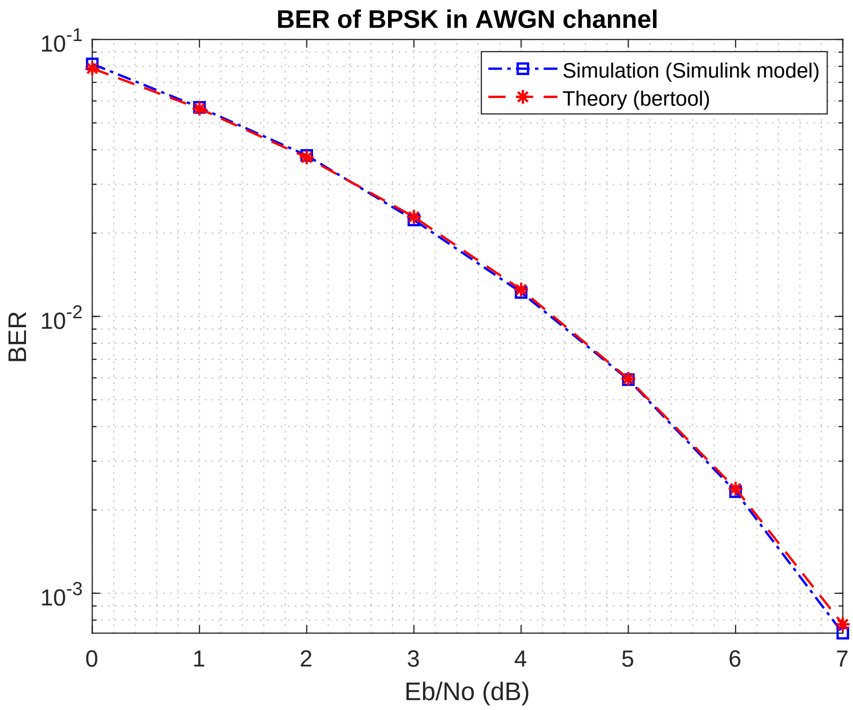


Figure 2.3: Simulated and theoretical BER in BPSK for different SNRs

2.2 8PSK IN AWGN CHANNEL

In this part, the simulation parameter M is set to 8 to obtain an 8PSK modulation. Since the modulation order is greater than 2, the symbols of the modulator are a combination of its input bits. The SER and BER performance of the system are presented in Table 2.2, and they are plotted in Figure 2.4.

Table 2.2: Effect of SNR on BER and SER in 8PSK

Eb/No (dB)	BER	SER
0	0.1278	0.3585
1	0.1036	0.2995
2	0.0844	0.2491
3	0.0639	0.1906
4	0.0487	0.1461
5	0.0339	0.1017
6	0.0214	0.0642
7	0.0126	0.0378

In the 8PSK performance plots presented in Figure 2.4, the BER is always lower than the SER and the ratio between them is approximately fixed to 3 for are levels of SNR. This is intuitive because a symbol error might result from more than one bit error, and the SER is determined by the geometry of the constellation and the BER is determined based on the SER and the technique used to map the bits to symbols. The MPSK modulator and demodulator used in the Simulink model use Gray coding to map the bits. The bit error probability P_b (.i.e BER) of 8PSK with

Gray coding is given in terms of the symbol error probability P_s (.i.e SER) by [1, eq 6.3] as

$$P_b = \left[\frac{P_s}{\log_2(M)}\right]_{M=8} = \frac{P_s}{\log_2(8)} = \frac{P_s}{3}$$

The factor of 3 in the theoretical equation matches the results shown in Figure 2.4.

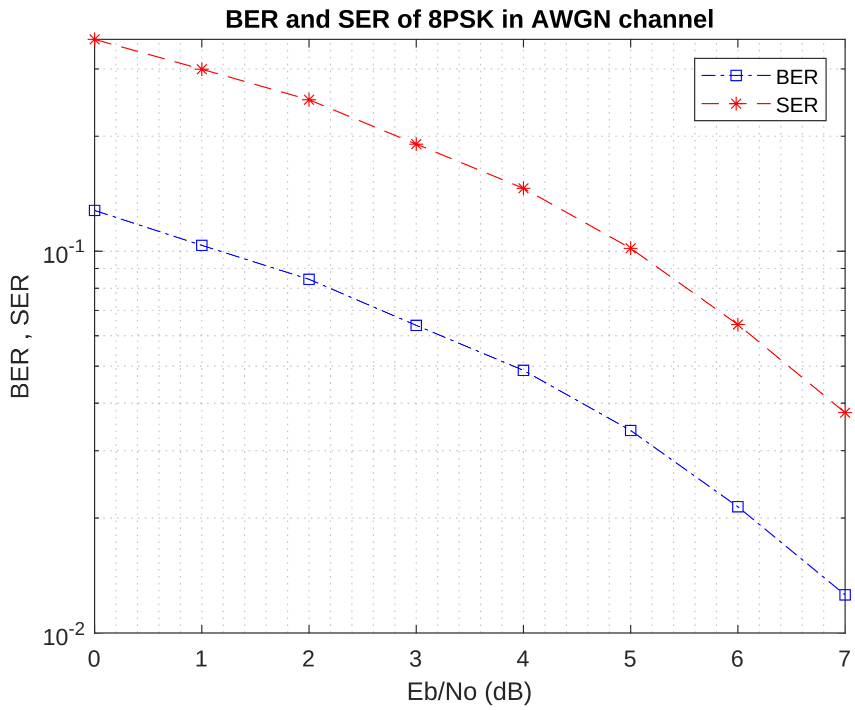


Figure 2.4: Simulated BER and SER in 8PSK for different SNR levels

The BER of 8PSK and BPSK is shown in Figure 2.5. For any level of SNR, the BER increases as the order of modulation increases. This is due to the fact that an increase in the modulation order results in a decrease in the spacing between the symbols which results in a higher SER and BER. To compensate for the increase in BER and support higher data rates (provided by higher modulation orders), we need to increase the spacing between the symbols and this could be achieved by using QAM modulation.

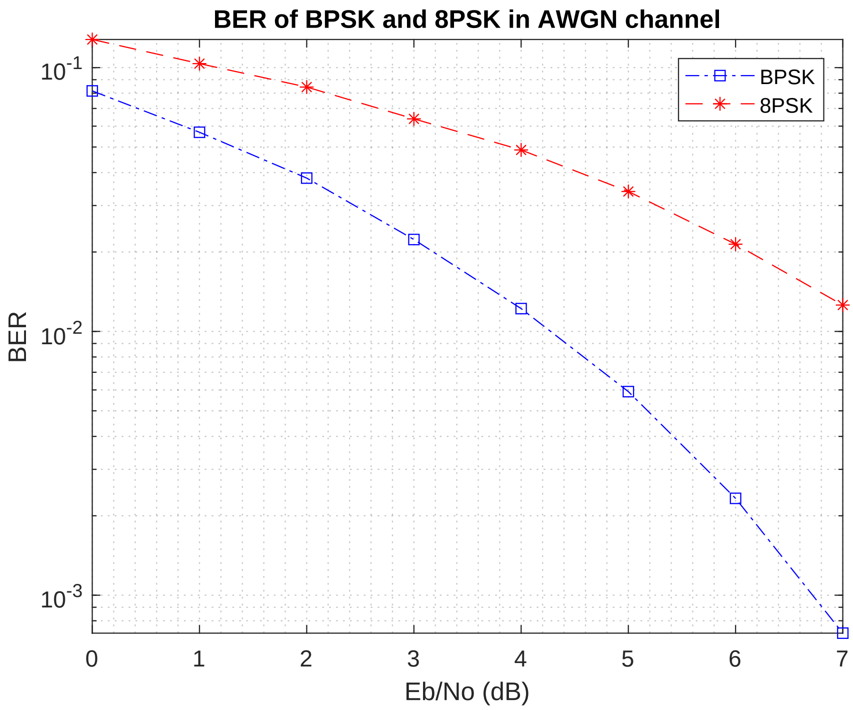


Figure 2.5: Simulated BER in BPSK and 8PSK for different SNR levels

Another way to enhance the performance of the communication and lower the BER without the need for an increased SNR is to use coding. The simplest form of coding is to apply Gray coding on the bits when they are mapped to symbols. Gray coding is a method of representing binary numbers in which each two consecutive numbers differ only one bit[2]. Figure 2.6 shows an 8PSK constellation diagram with symbols being Gray coded. Using gray code minimizes the bit error probability by ensuring that the closest symbols do not induce significant bit changes. This is due to the fact that when a symbol error occurs, it is more likely that only one bit is erroneous and with Gray coding the symbol can only make an error by being detected as an adjacent symbol. This is not true for regular representation of binary numbers, in which not every single bit error mapped to an adjacent symbol and more than one bit error might lead to an adjacent symbol, for example two bits change between 3 = 011 and 4 = 100, as shown in Table 2.3.

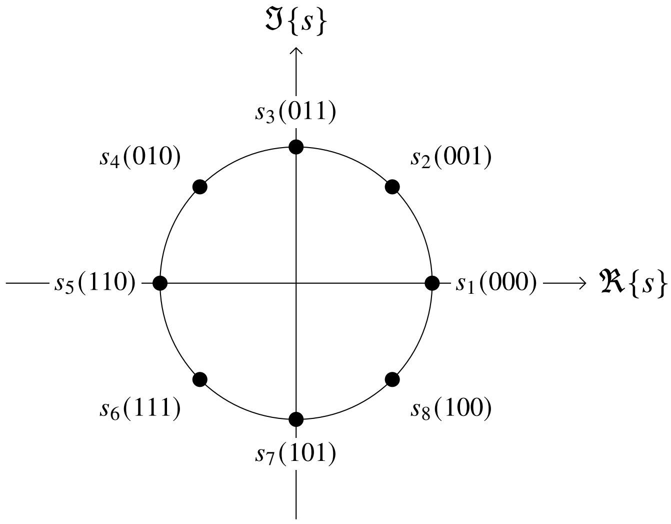


Figure 2.6: Constellation diagram for 8PSK

Table 2.3: Regular and Gray coded binary numbers representations

Number	Regular Binary	Gray Coded Binary
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

3 DPSK IN RAYLEIGH FADING CHANNEL

3.1 Doppler shifts

The Doppler shift is given by the following equation[1, eq 2.3] as

$$\Delta f = f_D = \frac{v}{\lambda}\cos(\theta) \tag{3.1}$$

where λ is the wavelength at the carrier frequency $\left(\lambda = \frac{c}{f}\right)$, v is the speed of the receiver, and θ is the angle of the arrival of the wave. To get the maximum Doppler shift $\cos(\theta)$ is set to 1 by setting $\theta = 0°$. Thus, equation 3.1 reduces to

$$\Delta f_{max} = f_D = \frac{v}{\lambda} \tag{3.2}$$

The Doppler shifts are:

- For velocity a ($v = 5.5 km/h$):

$$f_{D,a} = 4.0741\ Hz$$

- For velocity b ($v = 175.5 km/h$):

$$f_{D,b} = 130\ Hz$$

By setting the *MaximumDopplerShift* property in the *comm.RayleighChannel* object, the amount of Doppler shift in the Rayleigh fading channel can be controlled. Figure 3.1 shows the step response of the Rayleigh fading for the velocities (a) and (b).

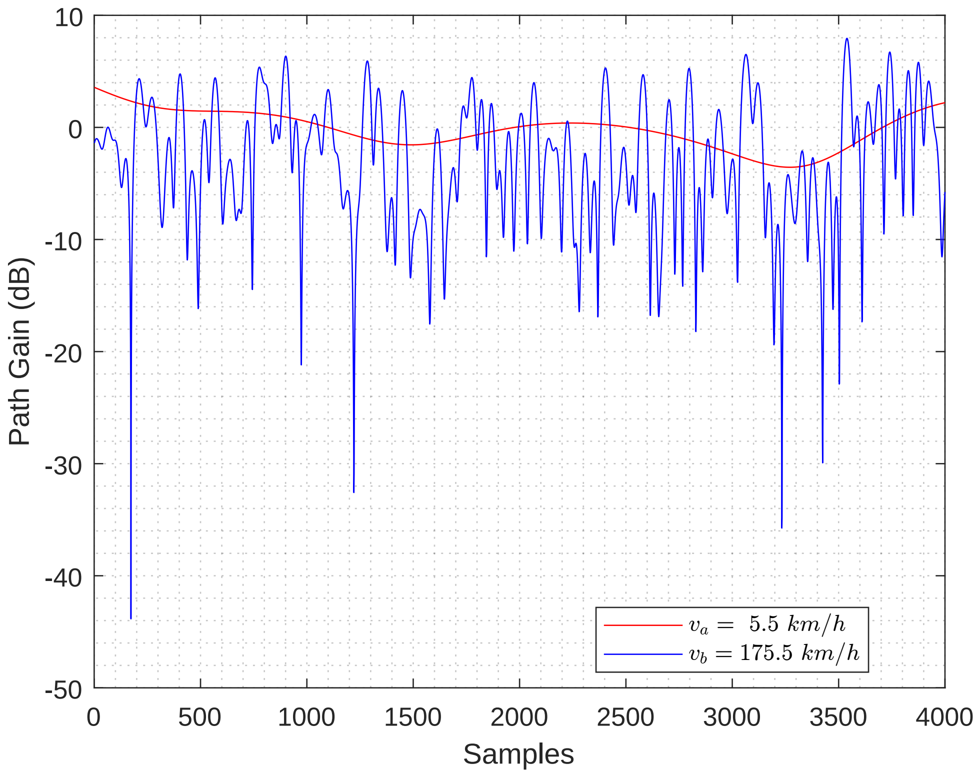


Figure 3.1: Step response of Rayleigh fading channel for the different velocities

From Figure 3.1, with velocity a (5.5 km/h), the different samples experience relatively constant gain response with a small variation that ranges between +4 dB to -4 dB, thus the channel with velocity is a slow fading channel. However, for velocity b (175.5 km/h), the different samples experience a severe variation in gain ranging between +8 dB and -40 dB, thus with velocity b, the channel is fast fading.

3.2 BER performance for $f_{D,a}$

The *DPSK_fading.m* Matlab file plots the empirical and theoretical BER for DPSK signal through a Rayleigh channel in the existence of AWGN. The empirical BER is obtained by replicating the communication system and the channel effect (AWGN, Rayleigh) and comparing the transmitted and received messages. The theoretical BER is calculated using the *berfading* function. For a Doppler shift $f_{D,a}$ resulting from velocity v_a, the theoretical and empirical BER match each other, Figure 3.2.

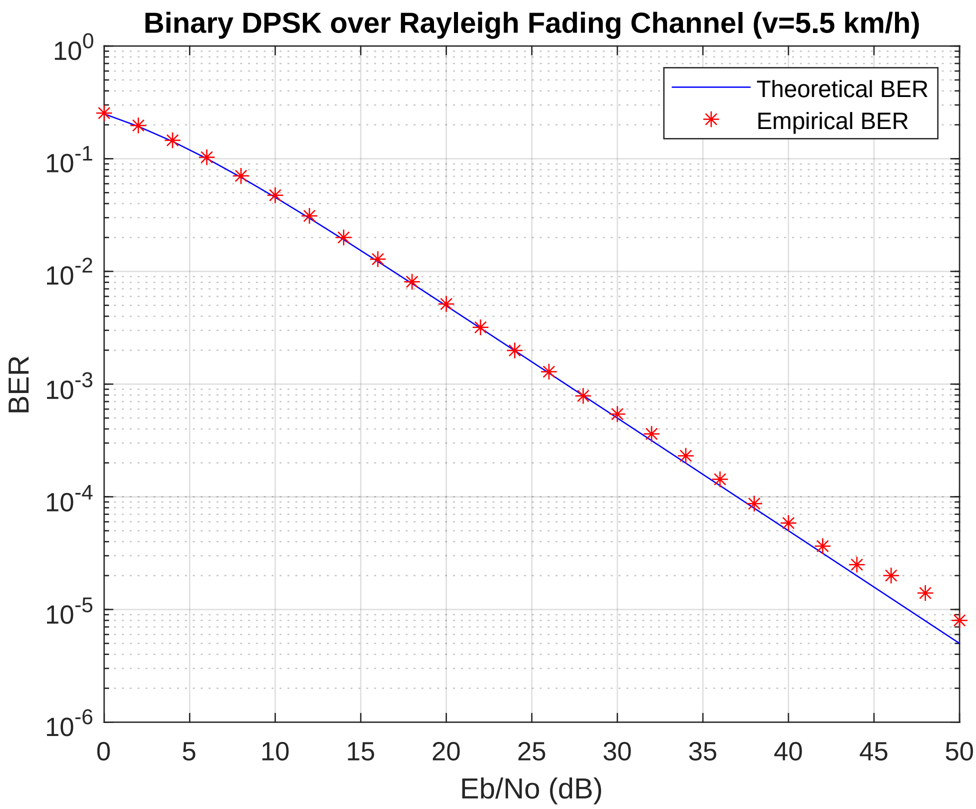


Figure 3.2: Empirical and theoretical BER in a Rayleigh fading channel with $f_{D,a}$

3.3 BER performance for $f_{D,b}$

For a Doppler shift $f_{D,b}$ resulting from velocity b, the theoretical and empirical BER don't match each other, Figure 3.3. The theoretical results are the same as the ones obtained in task 2, but the empirical ones start to flatten out at a BER of 10^{-3} and SNR of 30 dB. This difference is due to the irreducible error floor in a Rayleigh channel[1, p, 192] resulting from the Doppler spread (Doppler shift effect). The empirical simulation is replicating the complete communications link and the used *comm.RayleighChannel* modeling function is configured with Doppler shift, but the theoretical one implemented using the *berfading* function doesn't account for Doppler shift by assuming slow fading as stated in its documentation.

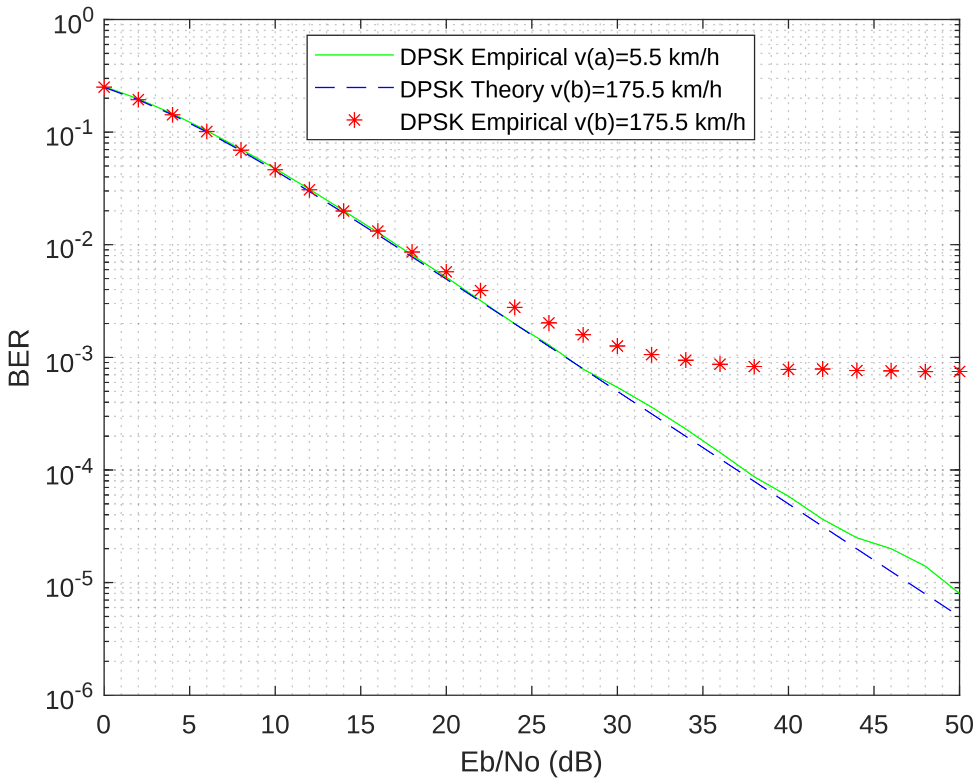


Figure 3.3: Empirical and theoretical BER in a Rayleigh fading channel with $f_{D,a}$ and $f_{D,b}$

3.4 The irreducible error floor of DPSK modulation

Deriving the equation for the irreducible error floor

Assuming the two transmitted bits in DPSK are equiprobable, we conclude that the probability of error is the same for transmitting a 0-bit or a 1-bit.

Assuming we sent two successive identical symbols $s(k-1)$ and $s(k)$ at times $(k-1)T_b$ and KT_b, the two received symbols are:

$$\begin{cases} r(k-1) = g_{k-1}s(k-1) + n_{k-1} \\ r(k) = g_k s(k) + n_k \end{cases}$$

where g_{k-1} and g_k are the fading gains and n_{k-1} and n_k are the AWGN components at the corresponding time intervals. Assuming that the fading is independent of k, the mean and variance of the fading gain are $\eta = E\{g_k\}$ and $\sigma^2 = \frac{1}{2}E\{|g_k - \eta|^2\}$, and for Rayleigh distribution, the PDF of g_k is

$$p(g_k) = \frac{g_k}{\sigma^2} e^{-g_k^2/(2\sigma^2)}, \quad x \geq 0$$

The adjacent complex fading amplitudes have correlation $E\{(g_{k-1}-\eta)^*(g_k-\eta)\} = 2\rho\sigma^2$ where $-1 \geq \rho \leq 1$ is the correlation coefficient which depends on the fast-fading channel model.

Using equation [1, Eq 5.63], the decision rule for DPSK is based on

$$z_k = r(k)r^*(k-1) = [g_k s(k) + n_k]\,[g_{k-1}s(k-1) + n_{k-1}]^*$$

The components $r(k-1)$ and $r(k)$ are complex Gaussian RVs since both the fading amplitude and the additive noise are complex. Therefore, z_k represents a Hermitian quadratic form of complex variables as

$$z_k = A\,|X_k|^2 + B\,|Y_k|^2 + C\left|X_k Y_K^*\right| + C^*\left|X_k^* Y_k\right|$$

Setting $D = z_k|_{r_k=1}$ to denote the decision variable corresponding to sending a 1-bit. Then, based on the previous decision rule, the average bit error is $P_b = p\{D \leq 0\}$. Following [3, Appendix B] and [4, Appendix 9A] approaches, we get

$$A = 0\,;\; B = 0\,;\; C = 1\,;\; X_k = r(k-1)\,;\; Y_k = r(k)$$

$$P_b = \frac{\frac{v_2}{v_1}}{1 + \frac{v_2}{v_1}}\;;\; \eta = \frac{v_2}{v_1} = \frac{1+\bar{\gamma}(1+\rho)}{1+\bar{\gamma}(1-\rho)}$$

where $\bar{\gamma}$ is the average fading SNR. By replacing, the error probability is

$$P_b = \frac{1}{2}\left[\frac{1+\bar{\gamma}(1-\rho)}{1+\bar{\gamma}}\right]$$

The error floor is obtained by letting $\bar{\gamma} \longrightarrow \infty$ as

$$P_{floor} = \lim_{\bar{\gamma}\to\infty} P_b = \lim_{\bar{\gamma}\to\infty} \frac{1}{2}\left[\frac{1+\bar{\gamma}(1-\rho)}{1+\bar{\gamma}}\,\frac{\frac{1}{\bar{\gamma}}}{\frac{1}{\bar{\gamma}}}\right] = \lim_{\bar{\gamma}\to\infty}\frac{1}{2}\left[\frac{\frac{1}{\bar{\gamma}}+(1-\rho)}{1+\frac{1}{\bar{\gamma}}}\right] = \frac{1-\rho}{2}$$

For a Gaussian Doppler power spectrum model, the Doppler power spectrum function is given by [1, Table 6.12]

$$S_c(f) = \frac{P_0}{\sqrt{(\pi)}} B_D e^{-\frac{f^2}{B_D^2}}$$

and the correlation coefficient is

$$\rho = \frac{A_C(T)}{A_C(0)} = e^{-(\pi B_D T)^2}$$

where B_D is the Doppler bandwidth and T is the symbol/bit rate,

Calculating the numerical value of the irreducible error floor

$$B_D = f_D = 130\ Hz$$

$$T = \frac{1}{\text{sample rate}} = 100\ \mu s$$

$$\rho_c = e^{-(\pi B_D T)^2} = 0.99335$$

$$P_{floor} = \frac{1-\rho_c}{2} = 833.286433172098 \times 10^{-6}$$

Simulating the irreducible error floor of the Gaussian Doppler power spectrum

To simulate the irreducible error floor of the Gaussian Doppler power spectrum, the property *DopplerSpectrum* in the *comm.RayleighChannel* object is changed to *doppler('Gaussian')*, as shown in Listing 3.1.

```
smplrte = 10000 ; % Set sample rate for Rayleigh channel
dpplrshft = dpplrshft_b ; % Set Doppler shift value for Rayleigh channel ( same as Task 1.b )
chan = comm.RayleighChannel ...
        ('SampleRate', smplrte, ...
            'MaximumDopplerShift', dpplrshft, ...
            'DopplerSpectrum', doppler('Gaussian')) ; % to get a Gaussian Doppler power spectrum
```

Listing 3.1: Code to obtain Gaussian Doppler power spectrum in a Rayleigh fading channel

Comparison

The simulated value of the irreducible error floor is

$$P_{floor} = 884 \times 10^{-6}$$

The calculated and simulated values of the irreducible error floor are approximately identical as shown in Figure 3.4 below.

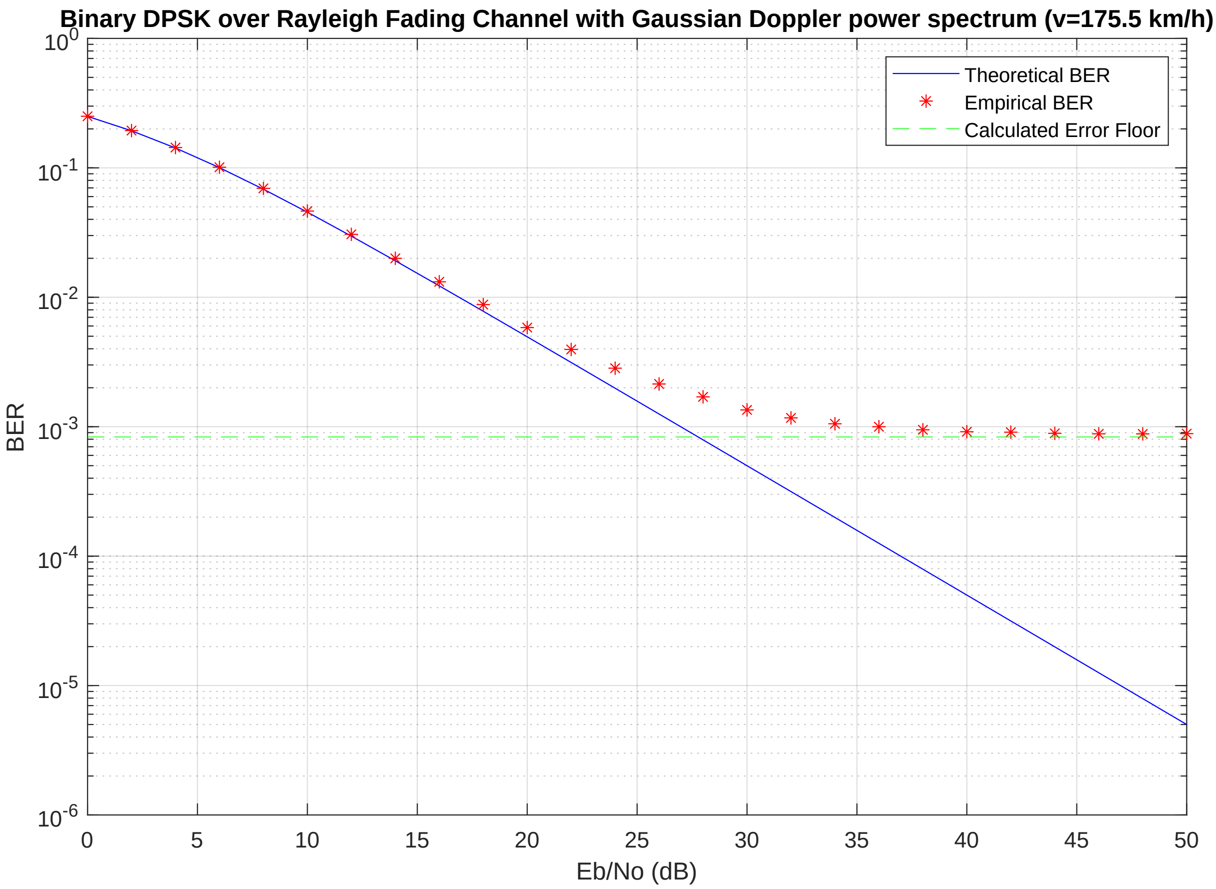


Figure 3.4: BER in a Rayleigh fading channel with $f_{D,b}$ and Gaussian Doppler power spectrum

4 Diversity

4.1 Interpreting the results

The provided *mimo.m* Matlab file implements different diversity schemes over Rayleigh fading channel, and also calculates the BER for the different diversity schemes for various SNR values.

It starts by initializing the simulation parameters (number of transmit and receive antennas, modulation order, SNR values . . .), then it generates the transmitted and received packets with different realizations of the channel for each packet, after that it calculates the BEr for each packets and averages it for each SNR value. The code generates the plots on Figure 4.1 showing the BER performance of the used diversity techniques for each SNR value.

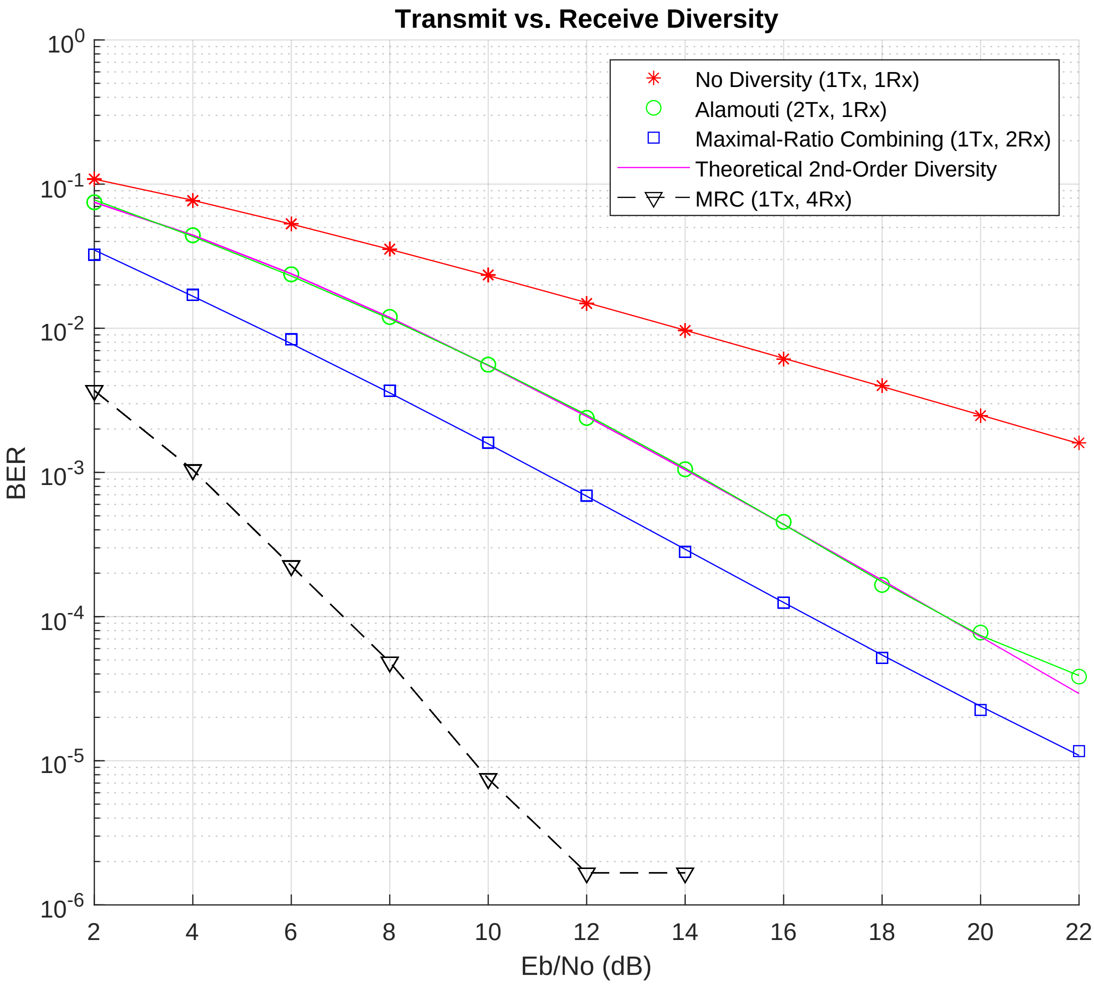


Figure 4.1: Transmit vs receive diversity for the given diversity schemes

For a given SNR value (γ), the BER (i.e. probability of error) for BPSK modulation in Rayleigh fading channel is given by [1, eq 6.58]

$$P_b = \frac{1}{4\gamma_b}$$

and the BER for BPSK modulation in Rayleigh fading channel with 2 receive diversity branches

($M = 2$) and maximum ratio combining (MRC) for is given by [1, eq 7.20, 7.23]

$$P_b = \left(\frac{1-\Gamma}{2}\right)^2 \sum_{m=0}^{2} \binom{1+m}{m} \left(\frac{1+\Gamma}{2}\right)^m \approx \alpha_M \left(\frac{2}{\beta_M \gamma}\right)^M$$

where $\Gamma = \sqrt{\frac{\gamma}{1+\gamma}}$. The BER for BPSK modulation in Rayleigh fading channel with 2 transmit diversity branches ($M = 2$) and Alamouti scheme (space-time transmit diversity "STTD") is given by [1, eq 7.20, 7.23]

$$P_b \approx \alpha_M \left(\frac{2M}{\beta_M \gamma}\right)^M$$

Thus for any SNR value, the no diversity system presents the worst BER performance compared to other systems and as the SNR increases, the BER of the other systems becomes far lower than that of no diversity system as reflected in the simulation results shown on Figure 4.1.

The simulation results also show that using two transmit antennas and one receive antenna (Alamouti scheme) gives worse performance than the maximal-ratio combined (MRC) system of one transmit antenna and two receive antennas. The ratio between the simulated BER performances is 4 which matches the approximated theoretical equation stated above in which the ratio is the factor $M^M = 2^2 = 4$ in the numerator of the Alamouti scheme approximated BER equation. In terms of SNR, in the 2×1 system the SNR should be multiplied by a factor of M (M=2 in this case .i.e increase by 3dB) to preserve the same BER performance as in the 1×2 system.

The theoretical performance of second-order diversity system matches the performance of the 2×1 system (Alamouti scheme) because it is implemented using the *berfading* command which normalizes the total power across all the diversity branches, as stated in its documentation.

Increasing the diversity branches further enhances the performance of the system and this is reflected on the BER results of the 1×4 MRC system as it results in the lowest BER performance for any SNR value compared to the other systems.

4.2 Adding a new curve

Implementing a new curve could be easily done using the provided *mrc1m.m* Matlab file as shown in Listing 4.1. The added curve presents the performance of MRC with 6 receive diversity branches as shown in Figure 4.2. The added 1×6 MRC system outperforms all the previous systems, and it achieves a lower BER record for lower SNR values.

```
%   Maximal-Ratio Combining for 1xM antenna configurations
%   function mrc1m(M, frLen, numPackets, EbNo) is used
BER16(idx)=mrc1m(6,120,10000,EbNo(idx));
```

Listing 4.1: Added code to generate the MRC1×6 BER performance

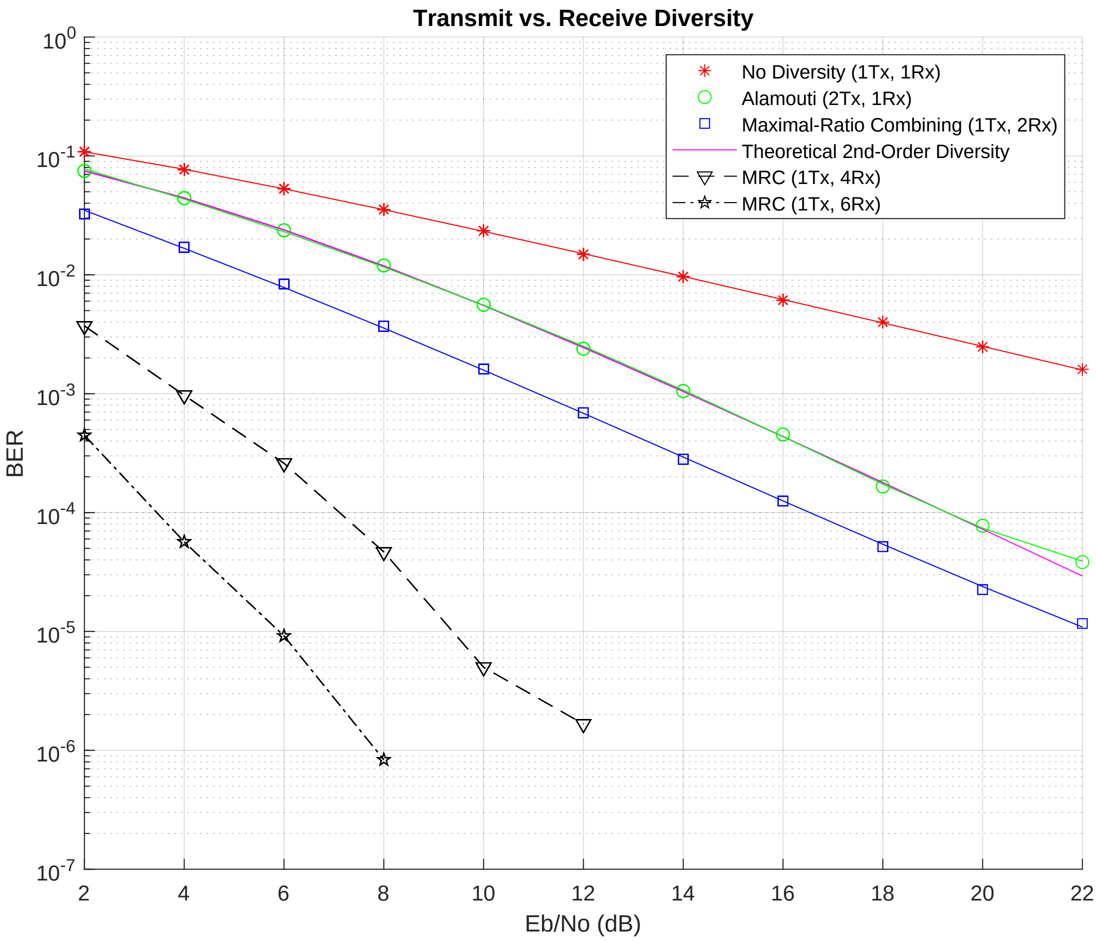


Figure 4.2: Transmit vs receive diversity for the given and additional diversity schemes

5 Performance of Channel Coding methods

5.1 Error correction mechanism

The provided *codingexample_control.m* Matlab file sets the blocks parameters for the *codingexample.slx* Simulink model which represents two communication links over AWGN channel using BPSK modulation. The first link embodies a Hamming error correction code h(7,4) and the second does not have any error correction as shown in Figure 5.1 below.

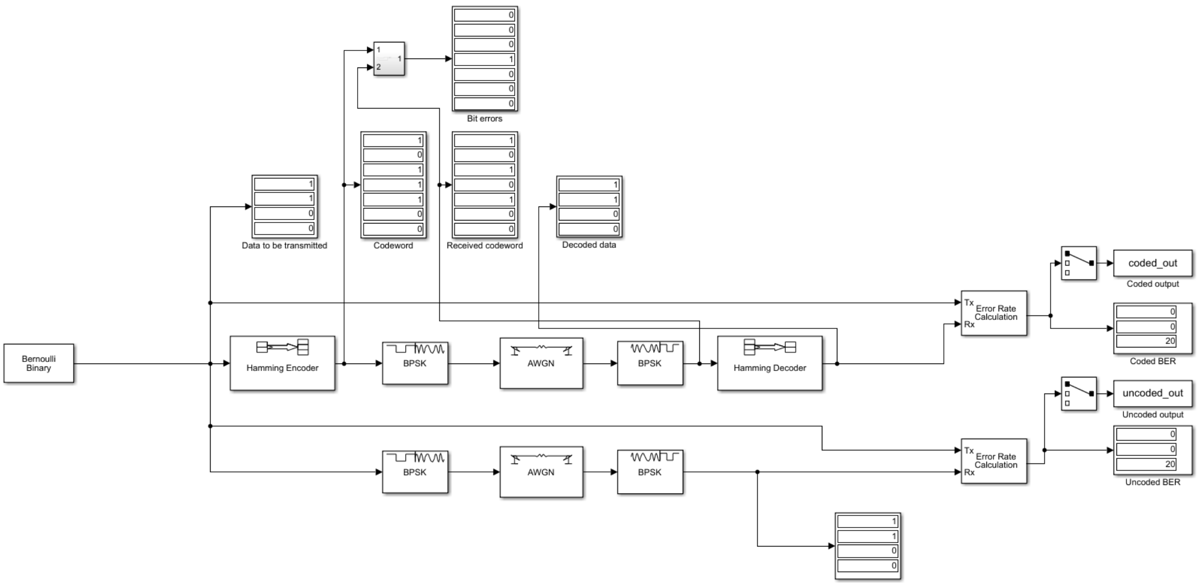


Figure 5.1: *codingexample.slx* Simulink model

The "bit errors" block compares the output of the Hamming encoder at the transmitter side with the input of the Hamming decoder at the receivers and presents a logic "1" if there's a mismatch between the bits of the two words (.i.e a bit error exists). When getting a single bit error as indicated in Figure 5.1, there exists a difference between the generated code word (u = 0011101) and received code words (y = 0010101). However, in this the error is corrected by the hamming encoder and there's no difference between the transmitted and received messages ($x_{Tx} = x_{Rx} = 0011$).

Error correcting works by adding extra bits (N-K) to the message to be transmitted (K bits) and thus getting a code word (N bits). The receiver does error detection in which the encoder does a mathematical operation on the received code word to obtain a syndrome word. Based on the latter and the decoding lookup table, the decoder determines the error vector. If the error vector is not zero then, the decoder corrects the error by adding the error vector to the received code word to obtain the corrected message.

In the previous case of Figure 5.1, the transmitted message was x_{Tx} = (0011), and the received code word was y = (0010101) and since the used code is a hamming (7,4) code[5, p71], the syndrome is

$$s = yH^T = [0010101] \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = [011]$$

Using the Hamming (7,4) decoding Table[5, p70], the error vector is

$$e = [0001000]$$

Thus, the received message is

$$x_{Rx} = y + e = [0010101] + [0001000] = [0011101]$$

$$u_{Rx} = x_{Rx}[1 : k] = [0011] = u_{Tx}$$

If there's two bit error, the Hamming (7,4) code is able to detect them by having a non-zero syndrome, but it's not able to correct them as it is only capable of correcting single bit errors ($[\frac{d_{min}-1}{2}] = [\frac{3-1}{2}] = 1$). When there's more bit errors (three or more), the code is not able to detect them as it is only cable of detecting at maximum two bit errors ($d_{min} - 1 = 3 - 1 = 2$).

5.2 Performance of error correcting code versus SNR

The section of the *codingexample_control.m* Matlab file runs the *codingexample.slx* Simulink model with various values of SNR and extracts the BER for the coded and uncoded systems. Table 5.1 shows the performance results for the 0 dB and 7 dB SNR values. The coded system starts to achieve lower BER than the non coded system at SNR values greater than 6.5 dB. At lower SNR values, the coding gain is less than unity resulting in a worse performance than the uncoded system. This is due to the fact that in the coded system, the transmitter is sending more bits and with such a severe degradation in the channel, the BPSK modulator reduces more errors which combined with the redundant bits, do not help in the error correction process resulting in more errors, and thus we get a higher BER.

Table 5.1: Coded and uncoded BER for BPSK modulation with Hamming (7,4) code

Eb/No (dB)	Uncoded BER	Coded BER
0	0.0744	0.1193
7	0.0007	0.0006

5.3 Error correction performance with higher code rate

A higher code rate R_c can be achieved by increasing the code length N and the word length K. The Hamming (15,11) is used for to achieve the increased code rate of $R_c = \frac{11}{15} = 0,733$

compared to the code rate of $R_c = \frac{4}{7} = 0,571$ of the last question. Table 5.2 shows the BER performance of the uncoded systems in comparison with the higher code rate system at SNR values of 0 dB and 7 dB. Similar to the previous section, the Hamming (15,11) coded system achieves poor performance compared to the uncoded system at lower SNR values. However, due to the increased code rate and message length, the threshold in which the Hamming (15,11) coded system starts to outperform the uncoded system is reduced to 4 dB. The BER performances of the uncoded system and both coded ones are shown in Figure 5.2, in which the higher code rate system achieves lower BER at all SNR levels due to the increased number of redundant bits and the higher error detection capability (4 bits).

Table 5.2: Coded and uncoded BER for BPSK modulation with Hamming (15,11) code

Eb/No (dB)	Uncoded BER	Coded BER
0	0.0743	0.1167
7	0.0007	0.0002

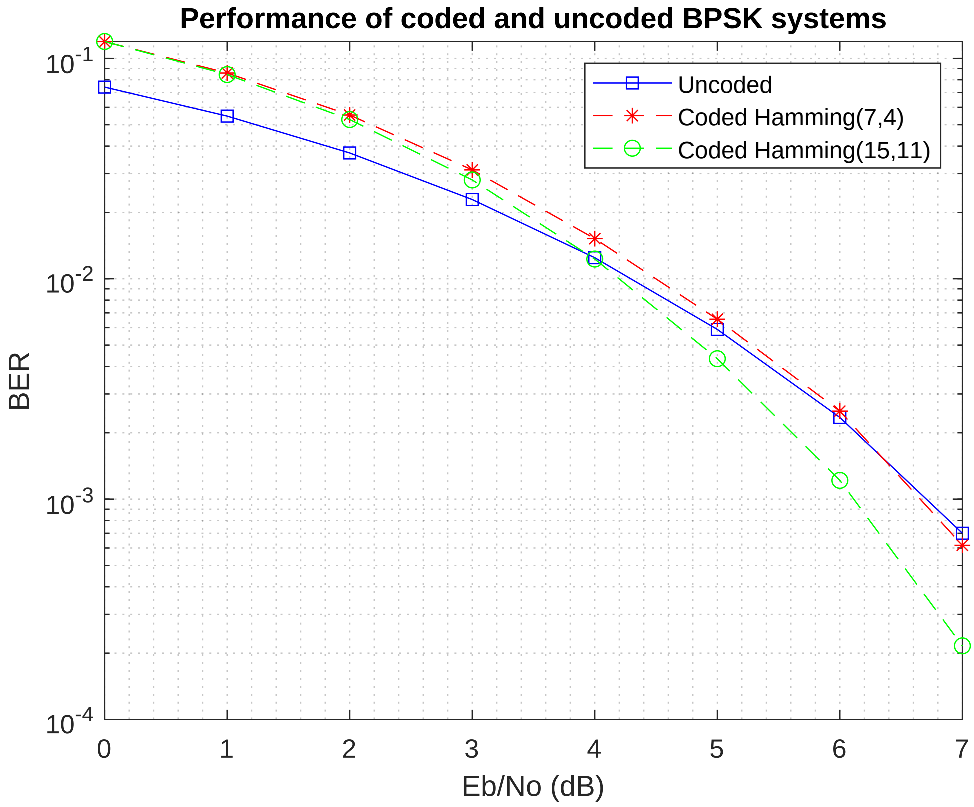


Figure 5.2: Coded and uncoded BPSK BER performance

6 Extra tasks

6.1 Extra Task 1

The implement communication link, shown in Figure 6.1, uses a random integer generator as a symbol generator. Then the integers are converted to bits and applied to the input of a 16QAM modulator which generates the transmitted signal that faces an AWGN channel. At the receiver side, the signal is demodulated using the 16QAM demodulator and the bits are extracted and compared against the transmitted ones to determine the BER. Also, a constellation diagram block is used to visualize the effect of different SNR levels on the samples of the received signal. The 16QAM system requires a considerable amount of SNR to properly function correctly as shown Figure 6.3 shows the constellation diagram for 7 dB and 12 dB of SNR. Compared to the BPSK and 8PSK studied in the section, the 16QAM excels at bandwidth efficiency by cramming more bits into one symbol. However, Figure 6.4 shows that its BER performance is worse than the earlier systems.

Adding a coding techniques would enhance the BER of the system. Figure 6.2 shows the same system with a convolutional encoder and Viterbi Decoder. At reasonable SNR levels, the coded system performed better than most system as shown in Figure 6.4, and for low SNR levels the coded system performed even worse than all the other systems.

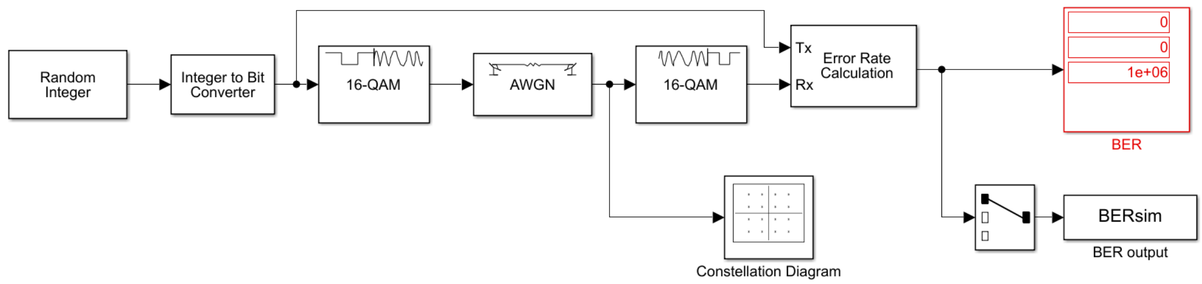


Figure 6.1: Implemented channel

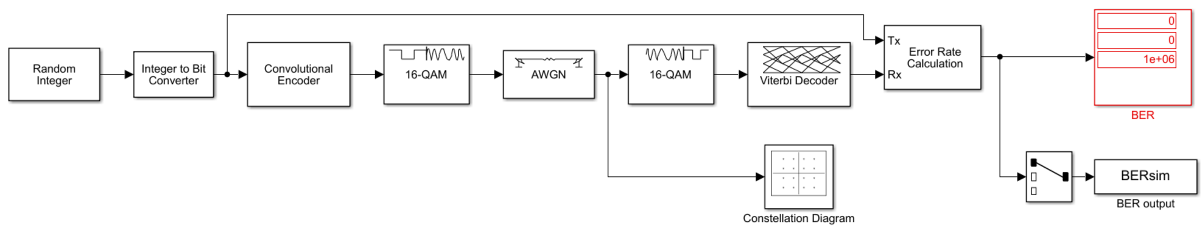
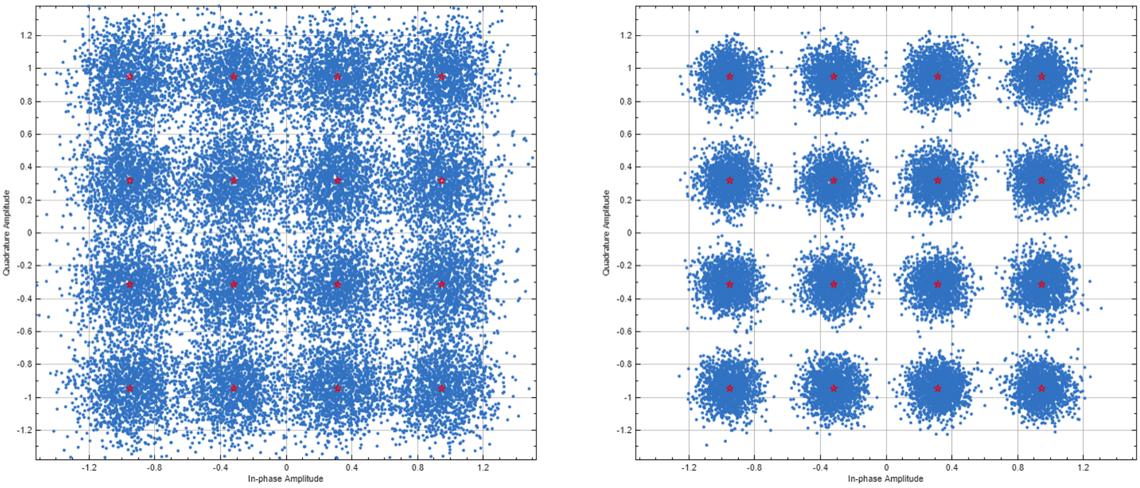


Figure 6.2: Implemented channel



(a) 7 dB (b) 12 dB

Figure 6.3: Constellation diagrams for different SNR levels

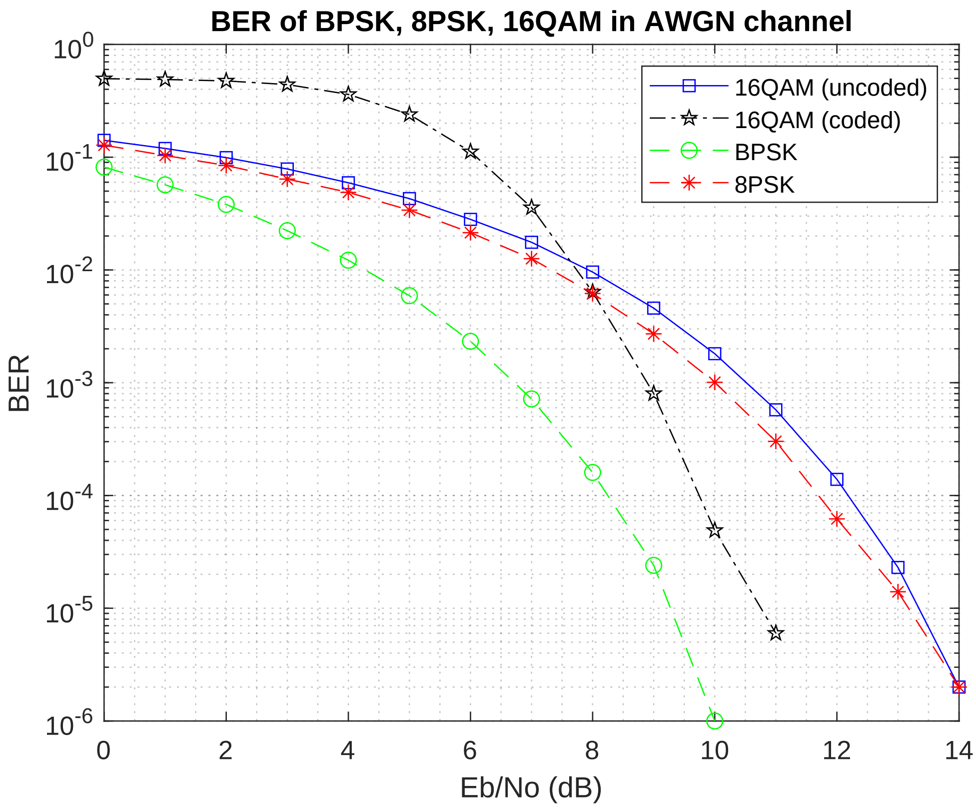


Figure 6.4: BER comparison

6.2 Extra Task 2

To correct two or more errors, a different coding method than those of section 5 should be used. Figure 6.5 shows a modified version of the Simulink model used in section 5. Here, BCH

Decoder and BCH Encoder blocks are used and configured to implement a BCH (15,5) coding which is able to correct $t = 3$ bit errors. The modified system achieves lower error probability compared to the uncoded and the hamming (7,4) coded systems as shown in Figure 6.6.

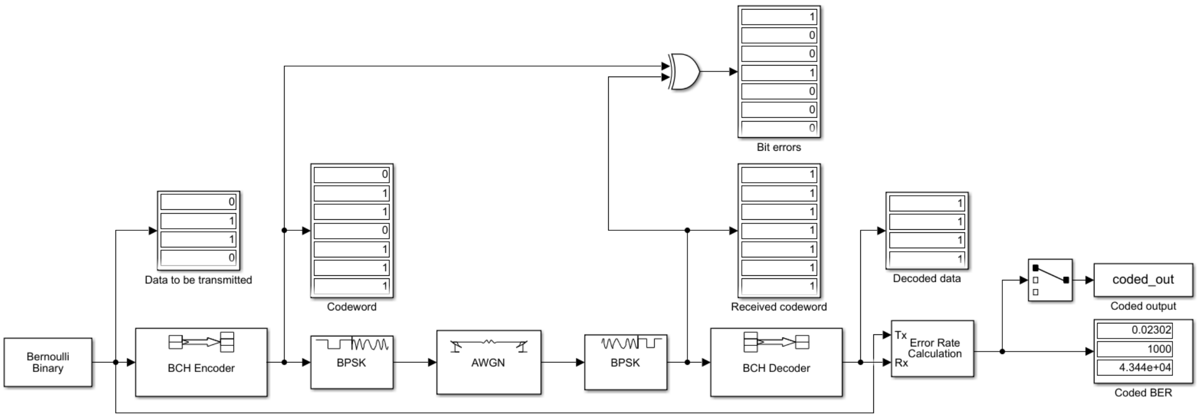


Figure 6.5: Modified *codingexample.slx* Simulink model

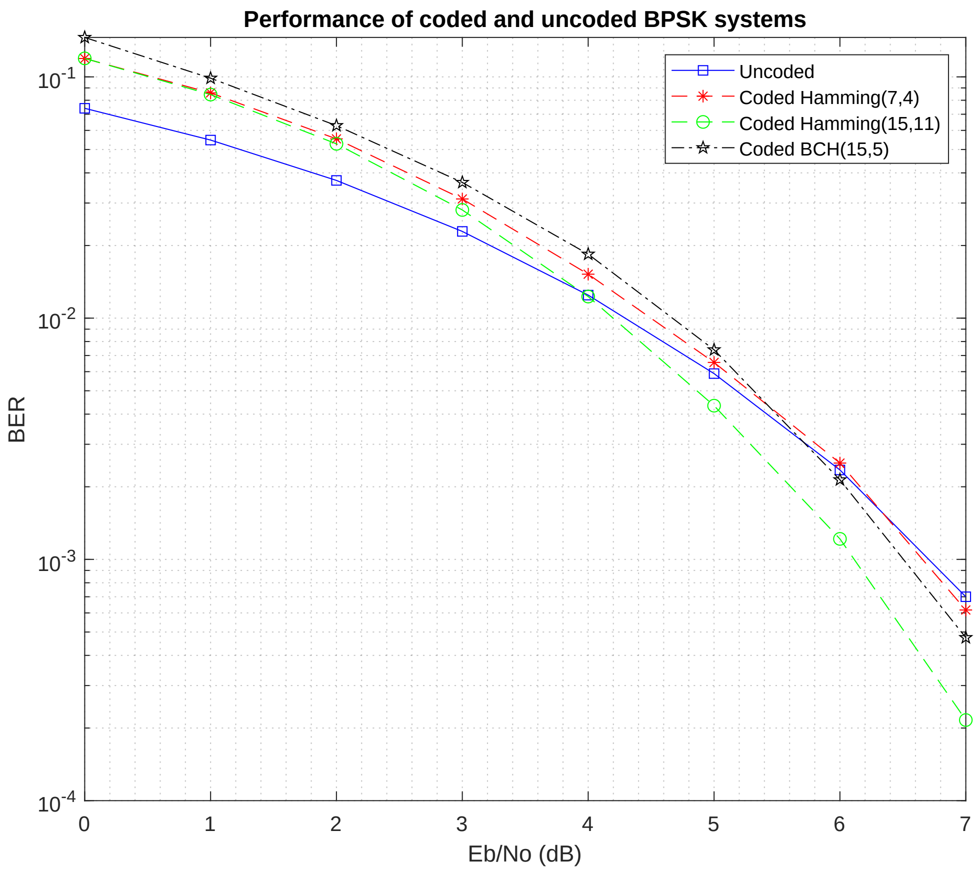


Figure 6.6: Coded (Hamming, BCH) and uncoded BPSK BER performance

7 SELF-EVALUATION AND FEEDBACK

Self-evaluation

- I did find Wireless communication I laboratory work useful and educational. It helped me to grasp the theory presented in the lectures and have an idea how it functions at a deeper level.
- I learned how to write more MATLAB code, and how to use Simulink. I also learned more about ray tracing and how it is used in practice.
- The work proceeded fine, I didn't have problems with the code, but I wasn't sure from my understanding of the theory, so I had to first understand the concepts more and then try to do the lab.
- I have used MATLAB in the previous years, but I just tinkered a few times with Simulink. This laboratory was the first time I use Simulink to model a certain system, but it wasn't hard at all.
- I would evaluate my work as average (maybe slightly above average) but not excellent.
- If I started doing the tasks again, I would make sure to do them directly after finishing the homework assignments and before the exams. Although I did the first tasks early in the semester, If I did it again, I would start doing the final tasks earlier.

Feedback

- I think the tasks are already perfect to examine the practicality of the lecture's content. I wish the questions were more direct. For example, specifying the diversity of techniques to be implemented in task 5.2 would be very helpful.
- The coding section needs more tasks, I think. For example, adding a subtask to simulate Turbo codes and interleaving would be very informative.
- I wish the lab used other software than MATLAB as it gets you hooked to the proprietary world software, but I understand that this is a wireless communications course not a simulation one, and it's easier to use one software platform for all simulations.
- I wish the laboratory work gave some notions of SDR (software defined Radio), and GNU Radio Companion. All that can be achieved with MATLAB and Simulink could be done with Python and GNU Radio, especially in the coding section. I think another extra task to examine these tools would be a rich learning experience for wireless communications students. For example, simple SDR peripherals (RTL SDR dongles) would be used to examine some modulation type and the student might have to examine the captured packets/spectrum and extract certain information from it (modulation type, hopping sequence, preambles, data, . . .). Also, the data could be provided to the student directly without the need of him to capture anything or use any SDR peripherals and MATLAB or Simulink could be used for that not just GNU Radio.
- I would like to be involved in putting my proposal into practice.

BIBLIOGRAPHY

[1] Goldsmith A. & (Firm) K. (2005) Wireless Communications. Cambridge Core, Cambridge University Press.

[2] Wikipedia contributors (2022), Gray code — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Gray_code&oldid=1112953976, [Online; accessed 1-October-2022].

[3] Massoud Salehi P. & Proakis J. (2007) Digital Communications. McGraw-Hill Education, fifth edition.

[4] Simon M.K. (2000) Digital communication over fading channels : a unified approach to performance analysis. Wiley series in telecommunications and signal processing, John Wiley & Sons.

[5] Kokkonen T. (2022), Oulu University, Wireless Communication 1, CODING: Block Codes lecture slides. https://opas.peppi.oulu.fi/en/course/521395S/10679.