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

521328A Simulations and Tools for Telecommunications
Exercise Report

Student	Aissa Azzaz	(2207335)
Student	Abdulmomen Ghalkha	(2207558)

December 28, 2022

CONTENTS

LIST of FIGURES

LIST of SYMBOLS and ABBREVIATIONS

AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
CSI	Channel Status Information
DMRS	Demodulation Reference Signal
eMBB	Enhanced Mobile Broadband
ISI	Inter Symbol Interference
mMTC	Massive Machine Type Communication
MTC	Machine Type Communication
NR	New Radio
OFDM	Orthogonal Frequency Division Multiplexing
PBCH	Physical Broadcast Channel
PDF	Probability Density Function
PSK	Phase Shift Keying
PSS	Primary Synchronization Signal
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
RS	Reference Signal
RSRP	Reference Signal Received Power
RSRQ	Reference Signal Received Quality
RSSI	Received Signal Strength Indicator
SER	Symbol Error Rate
SNR	Signal to Noise Ratio
SSB	Synchronization Signal Blocks
URLLC	Ultra Reliable Low Latency Communication

1 AWGN properties

AWGN channel modeling is a simple modeling technique that is used to simulate and design communications systems. In this part, the model presented in Figure 1.1 is used to study the properties of the AWGN channel block of Simulink.

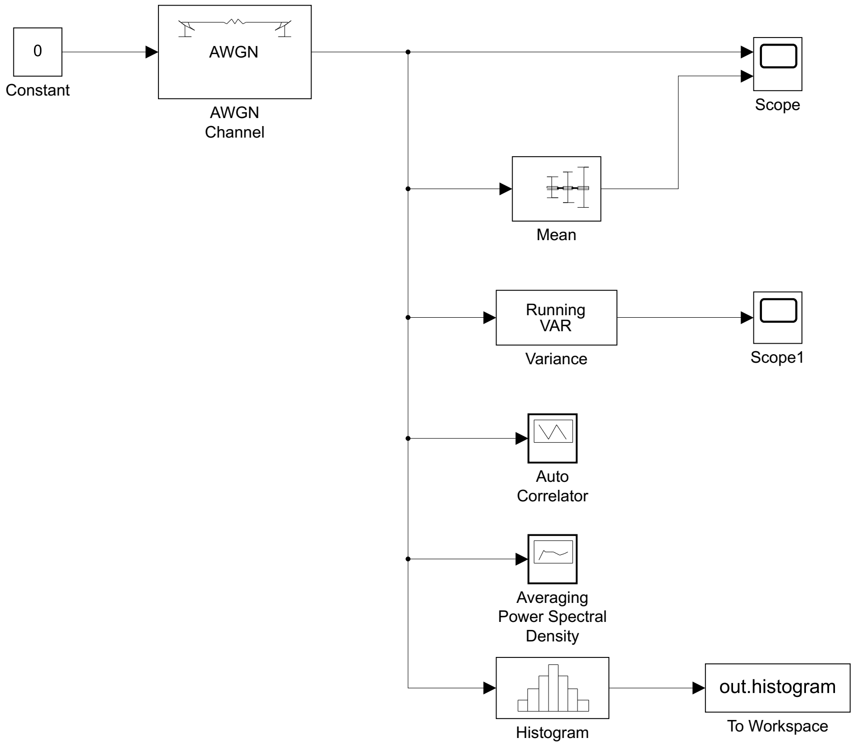


Figure 1.1: Generating AWGN samples in Simulink

1.1 Verifying the AWGN Simulink block

AWGN has several properties. In this part, we shall verify the ones concerning the real AWGN samples.

1.1.1 Zero-Mean

The provided Simulink block has a zero mean as the running mean of its samples converges to zeros as shown in Figure 1.2.

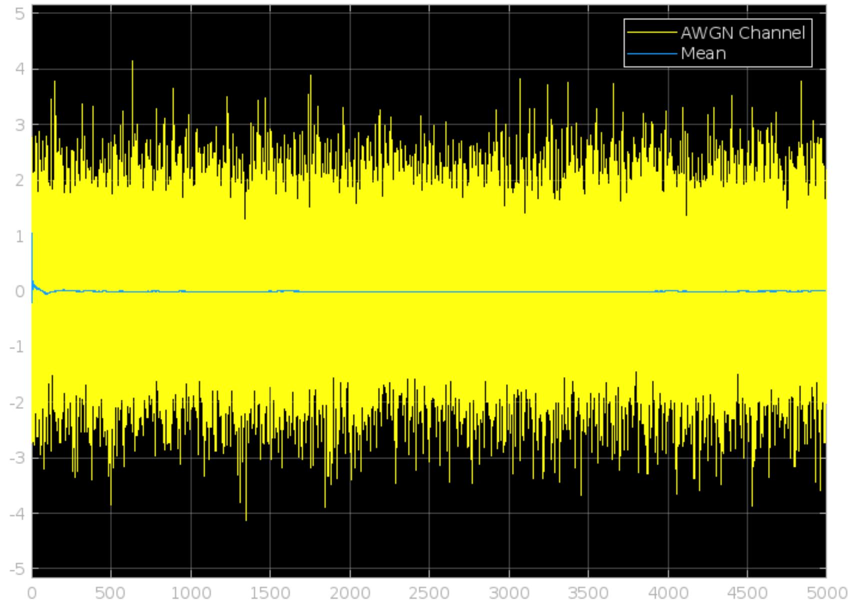


Figure 1.2: Running mean of the AWGN block samples

1.1.2 Constant variance

The provided Simulink block has a constant variance as the running variance of its samples converges to a constant non-zero value as shown in Figure 1.3.

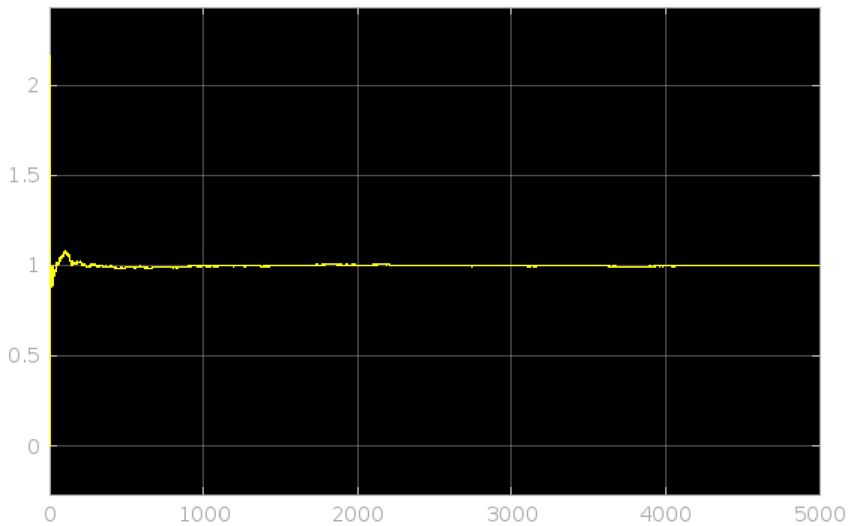


Figure 1.3: Running variance of the AWGN block samples

1.1.3 Whiteness

Whiteness reflects the correlation between the samples. This results in the correlation of each with itself only and not with the other samples, thus an impulse autocorrelation function with by

Fourier transform results in a flat power spectral density. The autocorrelation and power spectral density blocks used results in the autocorrelation function and PSD shown in Figures 1.4 and 1.5.

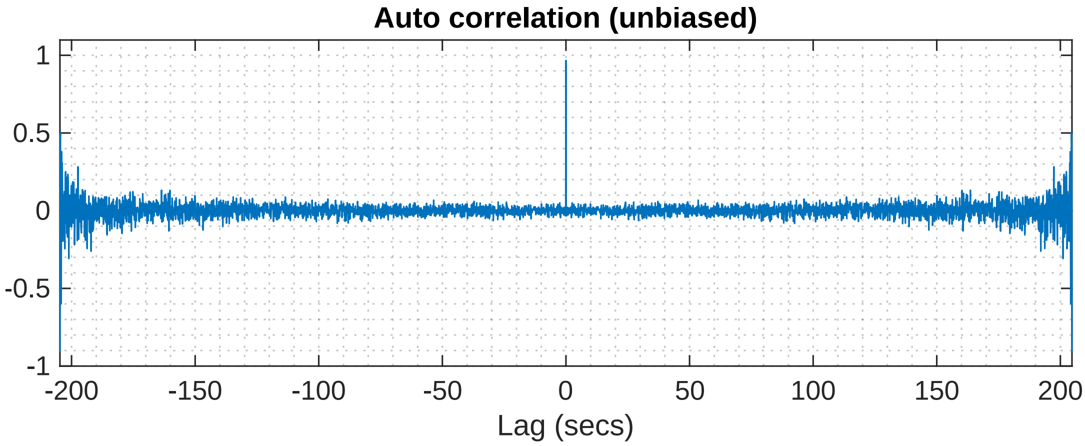


Figure 1.4: Autocorrelation function of the AWGN block samples

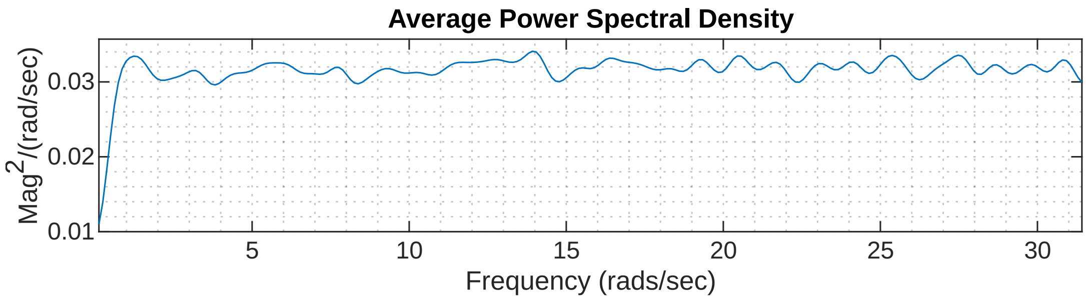


Figure 1.5: PSD of the AWGN block samples

1.1.4 Gaussianity

AWGN samples follow a Gaussian distribution that is characterized by a mean (zero in case of AWGN) and variance related to the SNR of the samples. When normalizing the output histogram results, the Simulink block generates Gaussian samples with zero mean and variance σ^2. This perfectly fits the theoretical Gaussian pdf, as shown in the histogram of Figure 1.6.

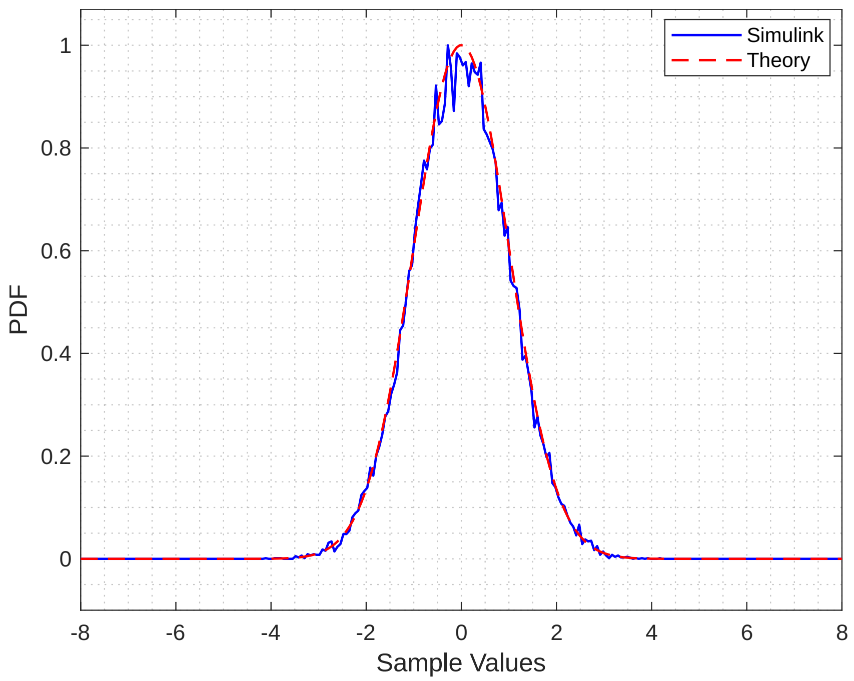


Figure 1.6: Histogram of the AWGN block samples

1.1.5 Conclusion

The AWGN Simulink block generates samples that follow the AWGN properties.

1.2 Histogram limits effects

Changing the upper and lower pairs of the histogram also changes the number of bins. This results in a smoother histogram distribution of the output that approaches more the theoretical pdf, however, the amplitude (the number of samples in each bin) reduces because the samples are spread over more bins, as shown in Figure 1.7, and also changes how wider the histogram is. Good value are the values between 4 and 10 because there enough samples in each bin to reproduce the theoretical PDF and there's not many fluctuations in the histogram amplitude.

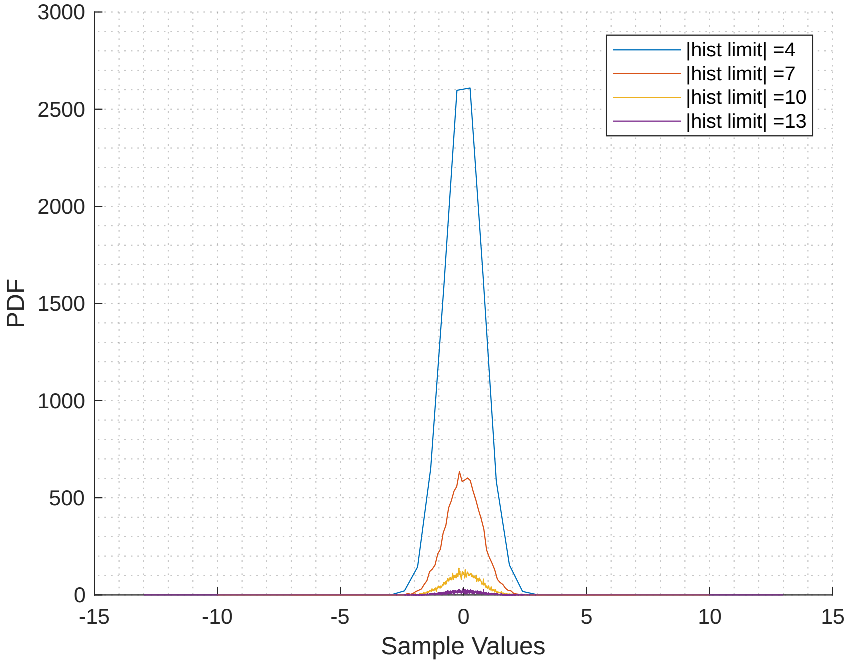


Figure 1.7: Histogram of the AWGN block samples with different limit pairs

2 Random generator seed and result convergence, BER, delay

In this chapter, we investigate the seed effect on the random number generator in the case of BER estimation.

2.1 Model Creation in Simulink

The model is created in simulation as shown in Figure. 2.1.

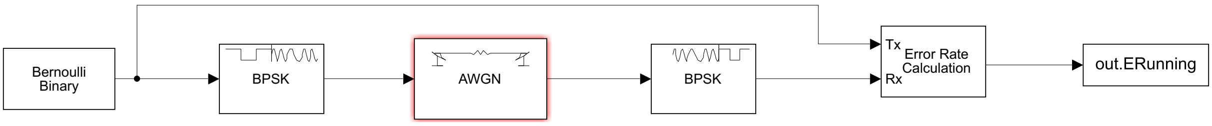


Figure 2.1: Simulation model in Simulink

For the Error Rate Calculation block, we select an 'Output Port' to get a 3-element vector which is then saved in Workspace.

2.2 Simulation Error Expression

In the case of independent errors and insignificant errors $ER << 1$, the relative standard deviation of the simulated error rate can be expressed as $\sigma/ER = {n_e}^{-1/2}$, where e is the number of encountered errors. To find the number of required errors in order to get a relative standard deviation of 10% of the error rate, we use the given expression and follow the calculation below

$$\frac{\sigma}{ER} = {n_e}^{-1/2}$$
$$\frac{\sigma}{ER} = 10\% = 0.1$$

By equating both left and right-hand sides and taking the reciprocal we get

$$\begin{aligned} {n_e}^{1/2} &= 10 \\ {n_e}^{1/2} &= 10^2 \\ n_e &= 100 \end{aligned} \tag{2.1}$$

And therefore, 100 errors are needed to get the given relative standard deviation.

2.3 Seed effect in BER Simulation

In this part, 4 dB was chosen as a value for E_b/N_0, and generated 5 random sequences using different values of the seed $10, 50, 100, 150,$ and 200. The seed value is the starting point of random number generators in order to produce a random sequence and thus starting with the same random seed would result in the same random sequence and conversely using different seed values would produce different random sequences. Therefore, at the beginning with few samples, the different sequences would result in different numbers of errors. However, using a large number of samples all the curves of the running average should converge to the same BER value as shown in Figure. 2.2.

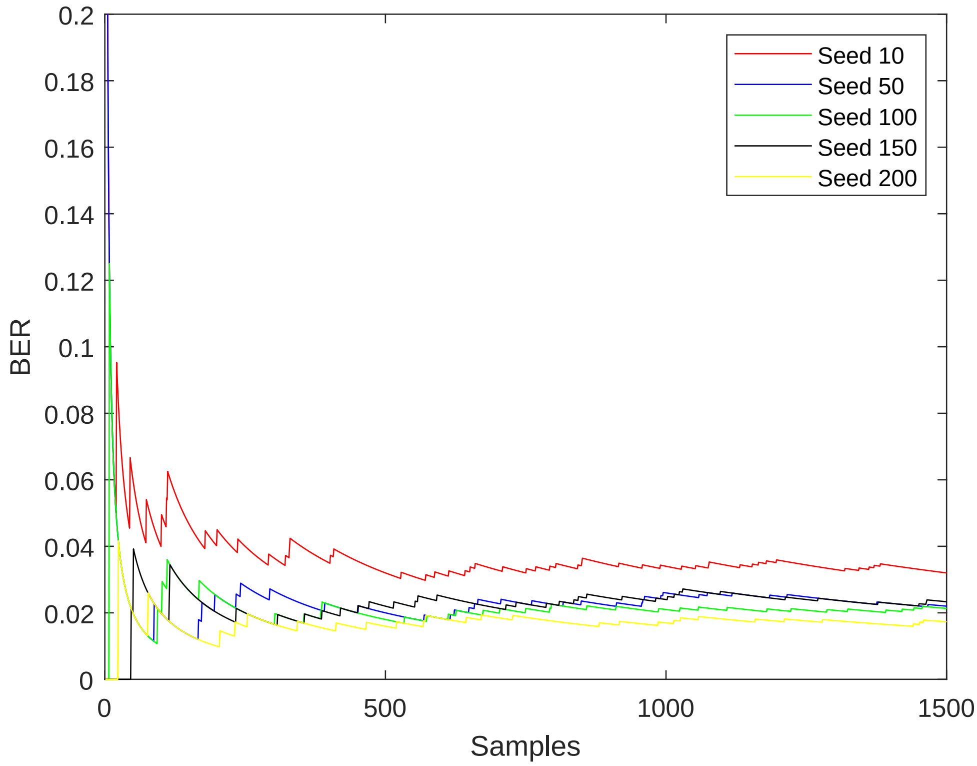


Figure 2.2: BER versus number of samples with different seed values

2.4 Theoretical and Simulated BER

In this part, we're comparing the BER curves generated from the theory as shown in [1] is given by $P_b = \frac{1}{2} erfc(\sqrt{\frac{E_b}{N_0}})$, where E_b is the average energy per bit, N_0 is the power spectral density of the noise. This is how the theoretical BER curve is generated in Simulink. For the simulated BER, we'll use our model for the calculation by setting the stop time to inf, and then a modulated randomly generated sequence is passed through the AWGN channel, and it's then demodulated and compared to the original bit sequence using different values of E_b/N_0. From Figure 2.3, we can see that the simulated BER agrees with the theoretical curve due to the fact that in the analysis of the theoretical BER of the BPSK, the simplest case was assumed which is the AWGN with binary transmission with perfect synchronization. Therefore, the analysis leads to a closed-form expression without further assumptions or simplifications.

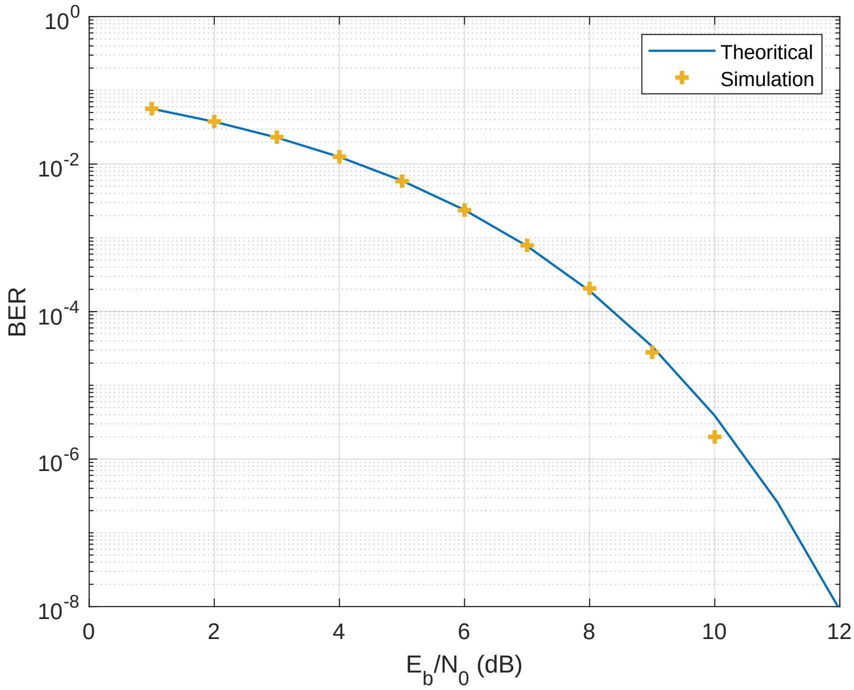


Figure 2.3: Simulated and theoretical BER versus E_b/N_0

2.5 Effect of Receive Delay on BER

Adding a delay of 1, which represents a delay of one sample time makes the main transmitted stream and the delayed stream out of phase and with the assumption that the consecutive bits are independent of each other with a probability of 0.5 for each zero and one we get a BER of 0.5 and doesn't this BER doesn't depend on or improve with values E_b/N_0. As a solution we can sacrifice a total delay of 1 and introduce a delay on the other stream and have both streams with a delay of 1 in order to calculate the BER.

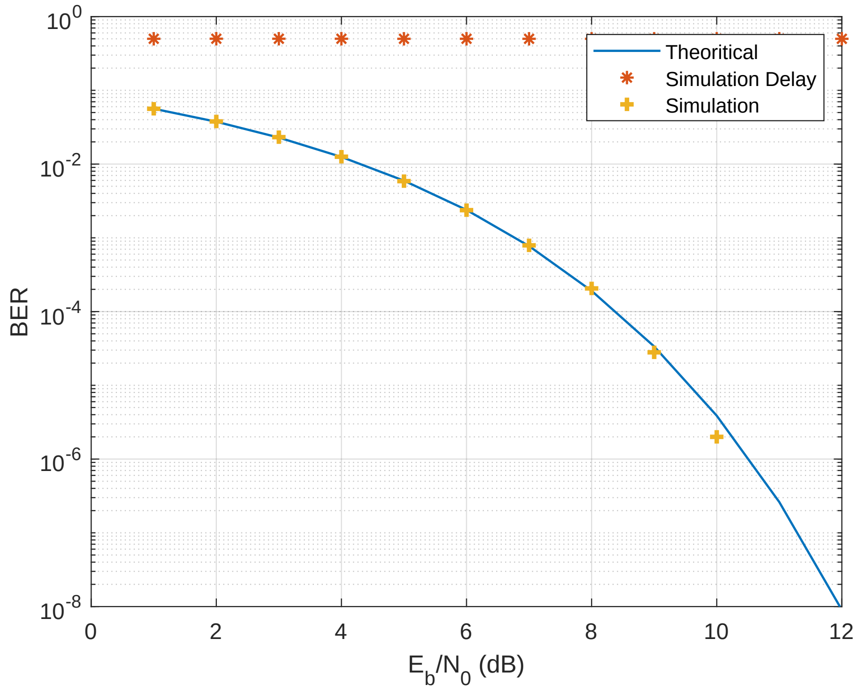


Figure 2.4: Simulated and theoretical BER versus E_b/N_0 with a receive delay

2.6 BER applications

- BER is generally used to evaluate and compare the performance of communication systems both wired and wireless, for example, the main criterion for choosing a communication system is achieving a target minimum BER.
- BER curves are used in the link budget analysis of communication links where the designer tries to meet a target minimum BER and by selecting which modulation scheme will be used he can find the minimum transmitted power.
- When the designer is constrained with a maximum transmitted power he can select which modulation schemes can be used to achieve the target bit rate or requested BER with the help of BER curves.
- Additionally, BER curves are used when in the design and verification of communication devices, for example, transmitters and receivers, and decide whether the design works properly and goes for production or there will be further modifications.

3 Gray and binary coded QAM/PSK, BER, SER

The simulation models used represent a communication link over an AWGN channel using PSK and QAM modulations, Figures 3.1 and 3.2, configured with $M = 8$ and $M = \{16, 64\}$, respectively.

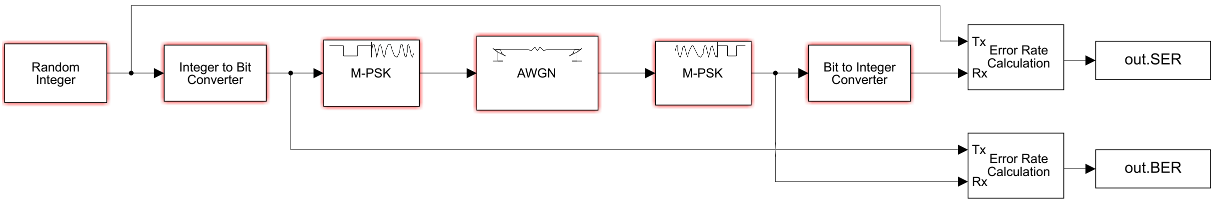


Figure 3.1: Communication over an AWGN channel using PSK modulation

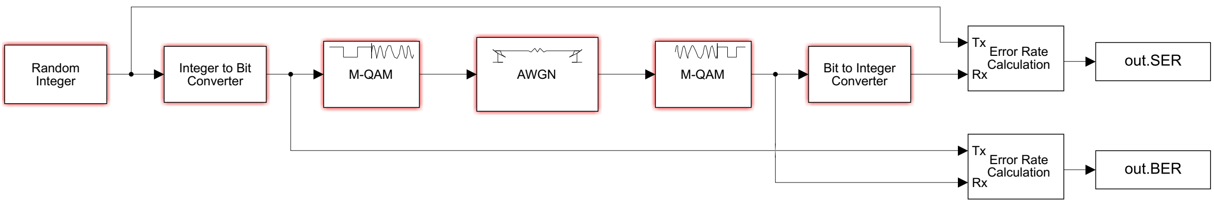
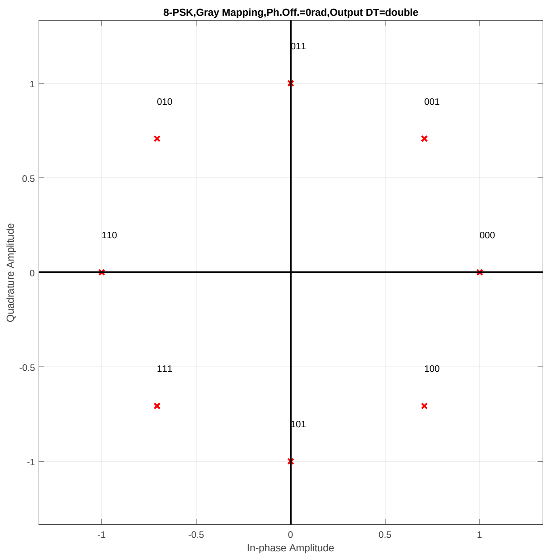


Figure 3.2: Communication over an AWGN channel using QAM modulation

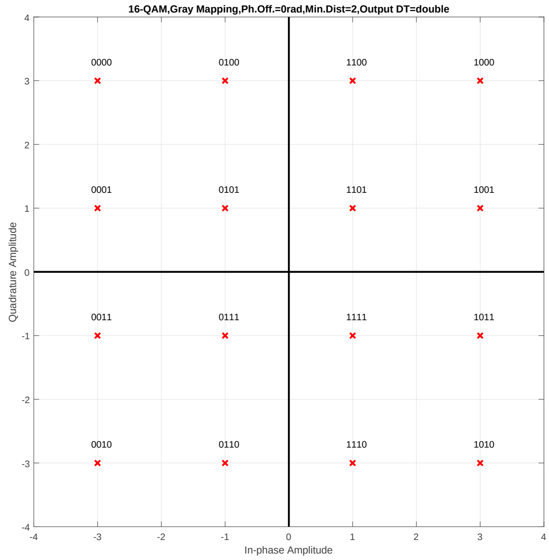
3.1 Constellations of QAM/PSK

The constellation diagram is a plot showing mapping of the bits to symbols in the complex plane denoted by the In-phase (real) and Quadrature (imaginary) plane. Figure 3.3 shows the constellation diagrams of 8-PSK, 16-QAM, and 64-QAM modulations.

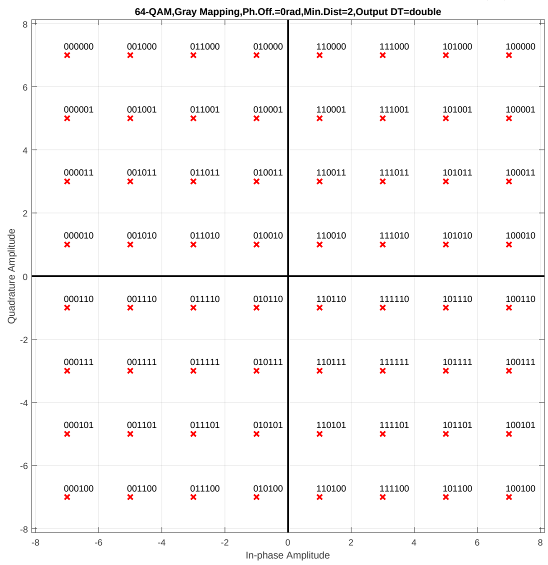
The constellation changes depending on the modulation type and the order. The symbols in 8-PSK are mapped to equally spaced dots along the unit circle due to the fact that PSK encodes the data by changing the carrier's phase. The symbols in the 16/64-QAM are arranged in a ($\sqrt{M}$ by $\sqrt{M}$) rectangular grid because QAM modulation affects the amplitude and phase of the carrier and the block uses rectangular QAM. In addition, the difference between 16-QAM and 64-QAM lies in the number of symbols which affects the spacing between them where the higher the order of modulation is, the higher is the number of symbols and the lower is the spacing between the symbols.



(a) 8-PSK



(b) 16-QAM



(c) 64-QAM

Figure 3.3: Constellation diagrams for the different modulations

3.2 Monte Carlo simulation of BER and SER vs. SNR

The Simulink models shown in Figures 3.1 and 3.2 are used to extract the simulated BERs and SERs. The theoretical BERs and SERs and obtained using the *bertool* analysis app[2] and the *berawgn* function[3], respectively. Figures 3.4, 3.5, and 3.6 show the different theoretical and simulated (binary and gray coded) BER and SER values for the various SNR levels.

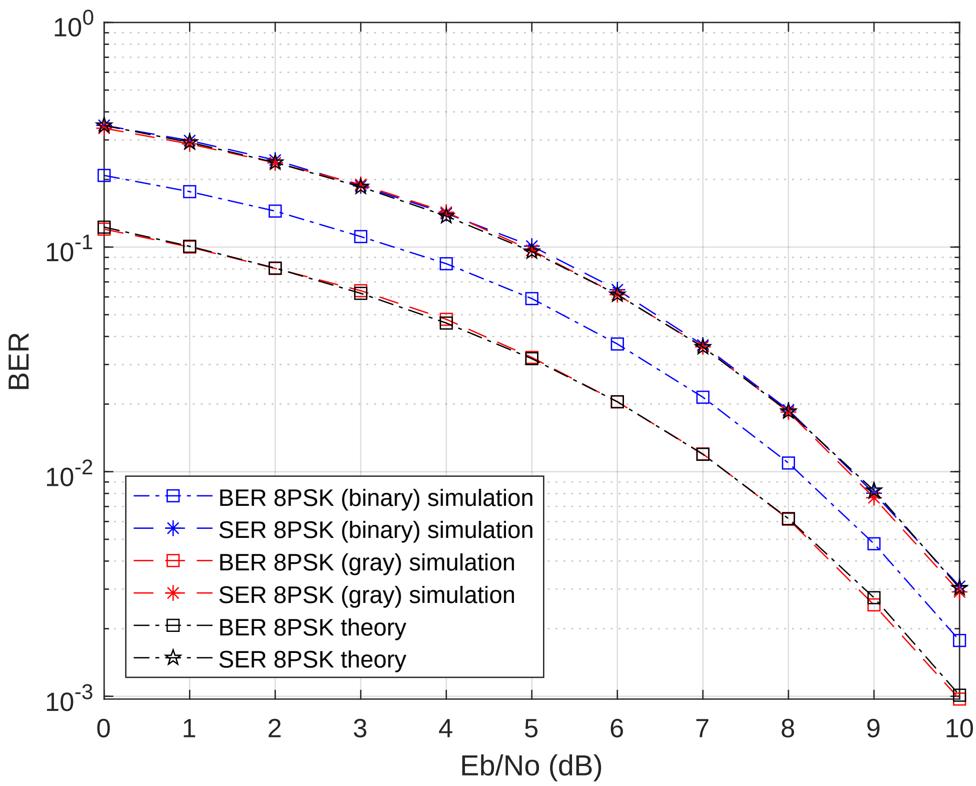


Figure 3.4: BERs and SERs for 8-PSK modulations

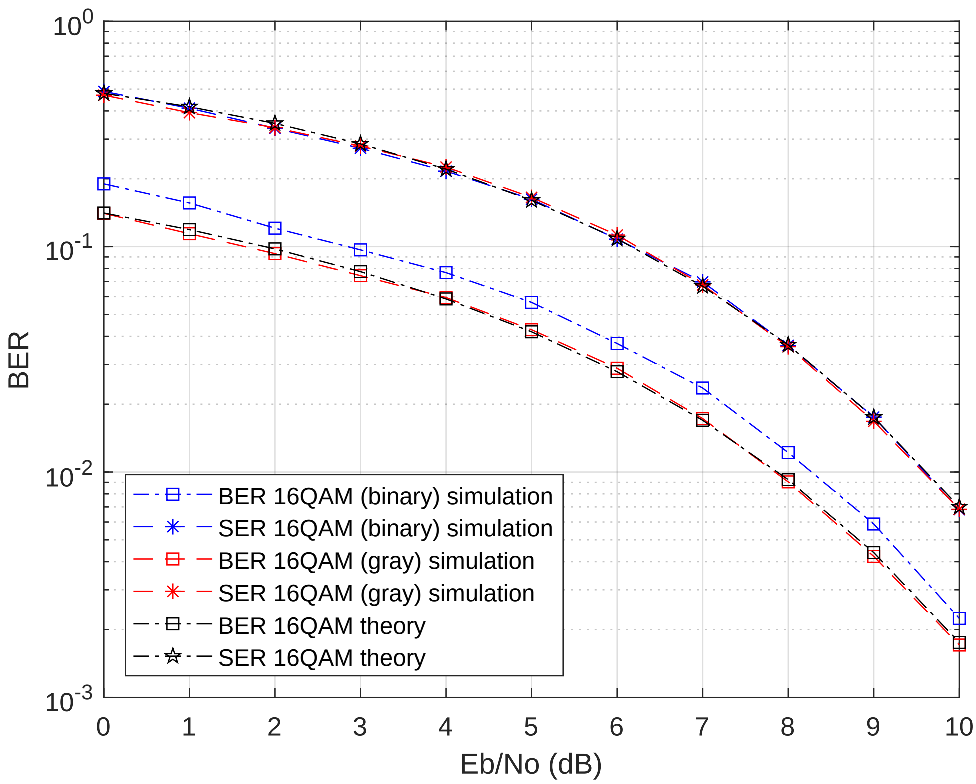


Figure 3.5: BERs and SERs for 16-QAM modulations

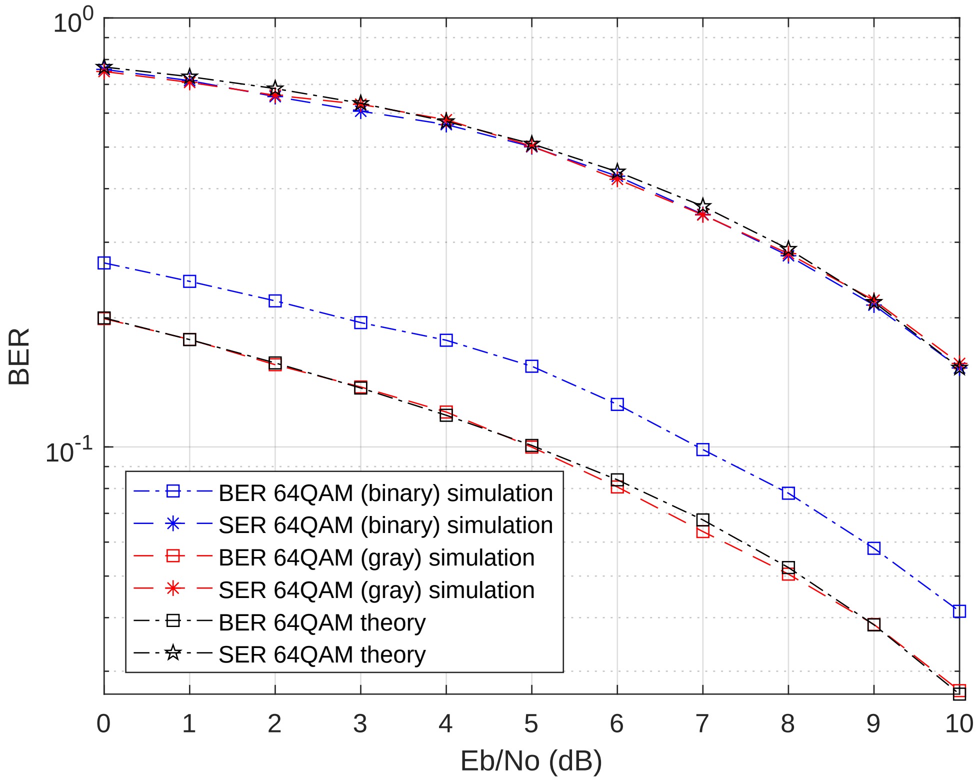


Figure 3.6: BERs and SERs for 64-QAM modulations

3.3 Commenting on the figures

In the three types of modulations, the simulated curves perfectly matched the theoretical ones with negligible differences due the limited simulation runtime. The theoretical BER curves matched the gray coded modulations because the bertool uses gray coding as stated in its documentation.

In addition, the simulated SER curves of the binary and gray coding are identical. However, the simulated BER curves of the binary and gray coding show that, in all modulations, the binary coding scheme experiences more errors. This is because the SER is determined by the geometry of the constellation and the BER is determined based on the SER and the technique (i.e. coding) used to map the bits to symbols.

The main difference between the curves of the different modulation methods is that the order of the modulation severely affects the performance and the achievable BER with a certain SNR value. In this case, 8-PSK achieves lower BER compared to 16-QAM which in itself achieves a lower compared to 64-QAM for a fixed level of SNR.

3.4 Gray coding benefit

In the Gray coded symbol constellation, the neighboring symbols differ only by one bit. When Gray coding is used, the shift of a symbol to one of the closest surrounding symbols in the constellation causes only a single bit error as shown in Figure 3.3. This is beneficial because this minimizes the BER which results in better system performance. For example, with binary coding, receiving an adjacent symbol would lead to 2 erroneous bits compared with Gray coding where it would lead only to 1 erroneous bit.

3.5 Comparing BER and SER curves

For the theoretical and simulated (binary and gray coded) results shown in Figures 3.4, 3.5, and 3.6, the SER curves are a scaled version of the BER ones. As discussed earlier, the SER is mainly determined by the geometry of the constellation and the BER is determined based on the SER and the technique (i.e. coding) used to map the bits to symbols, thus the coding method used affects the scaling ratio between the two curves. For gray coded curves the scaling ratio between the SER and the BER is $\log_2(M)$.

The BER would be equal to the SER only in the case when the number of data bit states is equal to the number of the symbol which only happens with binary modulation (.i.e $M = 2$).

4 5G Communications systems (MATLAB)

In this part, we have a look at the 5G NR Toolbox in MATLAB and go over some scripts implemented in the provided toolbox and finally, we introduce the 5G NR which was given by 3GPP.

4.1 Synchronization Signal Blocks and Bursts

The script simulates the generation of Synchronization Signal Blocks (SSB) and how to combine different SSBs to form synchronization signal bursts in the downlink channel. Each SSB sent from the Cell to the UE occupies 240 subcarriers and spans over 4 OFDM symbols. In the below tasks, we go through the exact content of the SSB block and how it is generated. In this simulation, the SSB is represented with a 240-by-4 matrix.

4.2 SSB and SSB Bursts generation

In Fig. 4.1, the primary synchronization signal was first generated and included within the SSB. It can be noted that the primary signal is in the first OFDM slot, and it takes the carriers from 57 to 183.

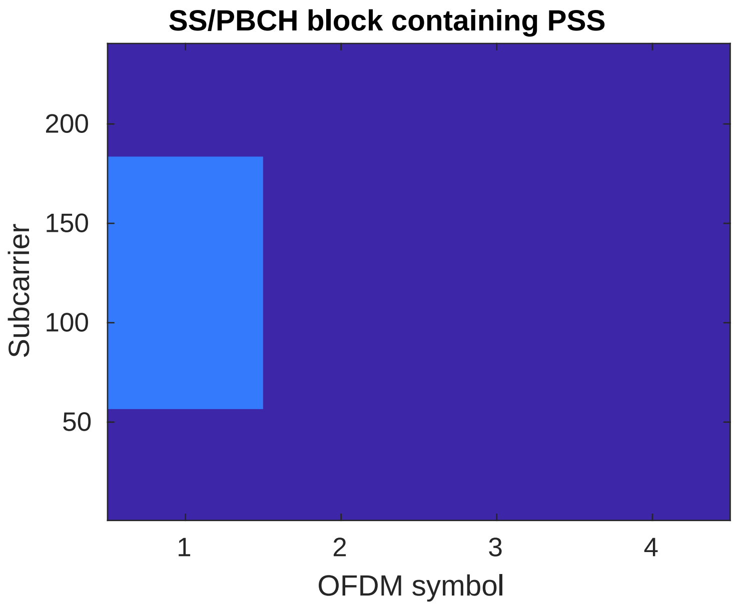


Figure 4.1: Primary synchronization signal

Figure 4.2, includes both the primary and secondary synchronization signals which consist of the same number of BPSK symbols of 127. The reason of the different colors between the primary and secondary SS is that with secondary SS we used a different scaling factor $\beta_{SSS} = 2$ in contrast to 1 in the primary SS.

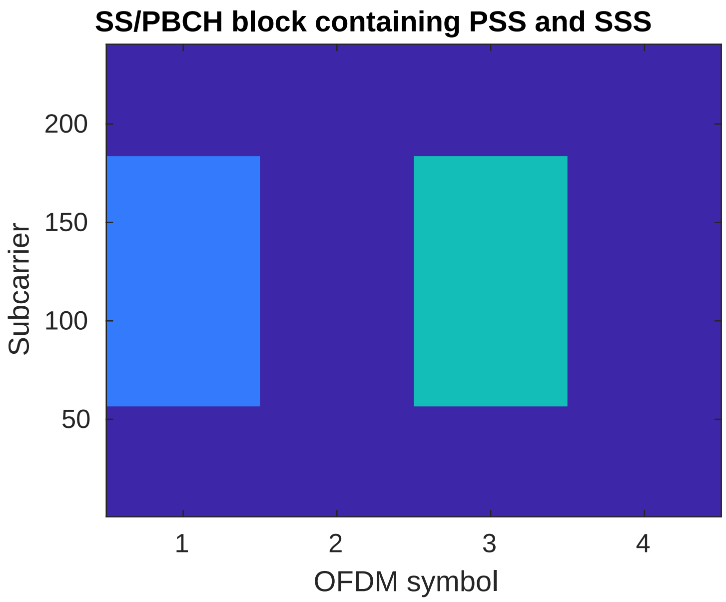


Figure 4.2: Secondary synchronization signal

The third part in the SSB shown in Figure. 4.3, is the physical channel, which carries 864 codeword. This codeword is scrambled depending on the cell identity and the block index, and modulated using QPSK to result in 432 symbols, since each QPSK symbol represents 2 bits. Finally the modulated sequence is mapped to the SS block, and it can be noted that there is a gap between the symbols, which makes the modulated code word to fill the rest of the SS block.

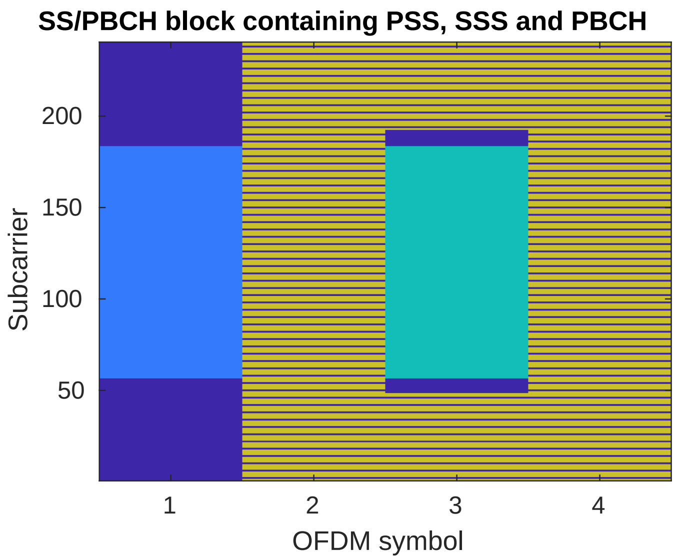


Figure 4.3: SSB includes SSB, SSS, and PBCH

The last step in constructing the SS block is to include the demodulation reference signal or DM-RS to the block which is mainly used to estimate the channel and restore the original signal by equalizing the channel effects. These DM-RS are known for the transmitted and receiver and it's a QPSK signal that is located within the gaps between the symbols in the physical channel slots. Figure 4.4 shows that the gaps between the symbols which have a light color denoting that their values are higher due to the multiplication of the scaling factor $\beta_{PBCH}^{DM-RS} = 4$.

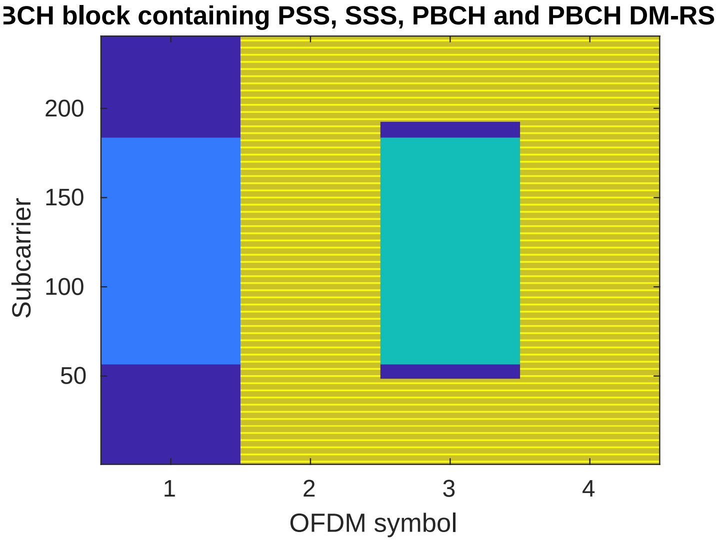


Figure 4.4: DM-RS added to the SSB block

In the last step our SS block is ready and contains all the required signals and channels. However, it's more practical to combine these SS blocks into subframes and frames. In this case we're constructing an SS burst which combines 5 subframes, where each subframe or block is given an index $0, \ldots, L-1$, which will be used to scramble the information sequence. In order to make the burst, we make a loop to keep track of the block index, and other PSS and SSS signals are independent of the block index and added to the burst outside the loop. Figure. 4.5 illustrates the content of the burst, where we can see PSS, SSS, DM-RS, and the content of PBCH.

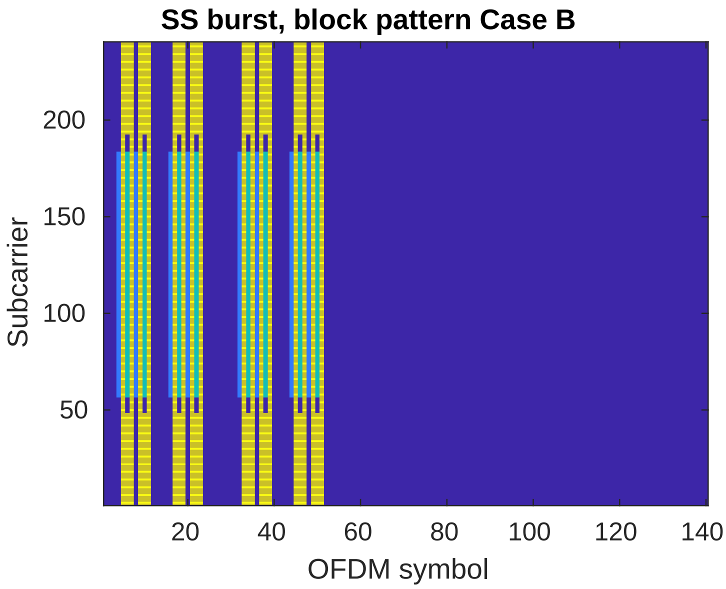


Figure 4.5: SS Burst

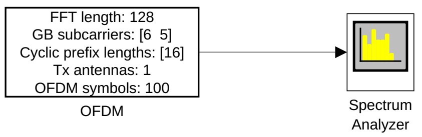
4.3 Generate Wireless Waveform in Simulink Using App-Generated Block

4.3.1 Script Explanation

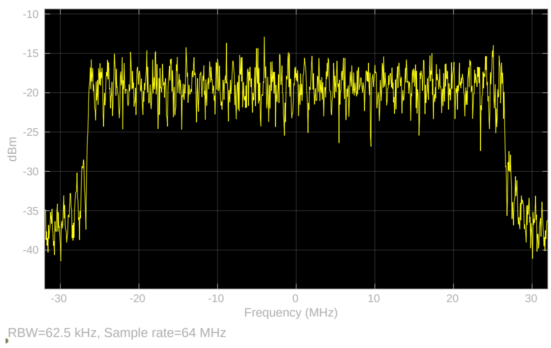
The script constructs Simulink models to generate OFDM waveforms and simulates them in different cases. The generated waveforms are then visualized in the frequency domain using the spectrum analyzer. After that, the script exports an AWGN channel and an OFDM demodulator to Simulink to simulate the effect of the channel on OFDM symbols by visualizing the constellation diagram after the demodulation. Finally, the scripts export a multiband combiner block to Simulink to simulate the aggregation of multiple OFDM sources by shifting them in frequency, and Spectrum Analyzer is used to show the combined result in the frequency domain.

4.3.2 Script Results

In Figure. 4.6a, the OFDM signal generator produces an OFDM signal and the answer is visualized in the spectrum analyzer.



(a) OFDM signal generator



(b) OFDM signal frequency representation

Figure 4.6: Visualizing an OFDM signal in frequency domain

The OFDM generator settings can also be modified before exporting them to Simulink, which can be done using the wireless Waveform Generator. The adjusted settings were the number of FFT lengths, which was changed from 64 to 128 and the subcarrier spacing was changed from 1 MHz to 2 MHz. The resulting spectrum is illustrated in Figure. 4.7, where we can see an increase in the spectrum compared to the first case.

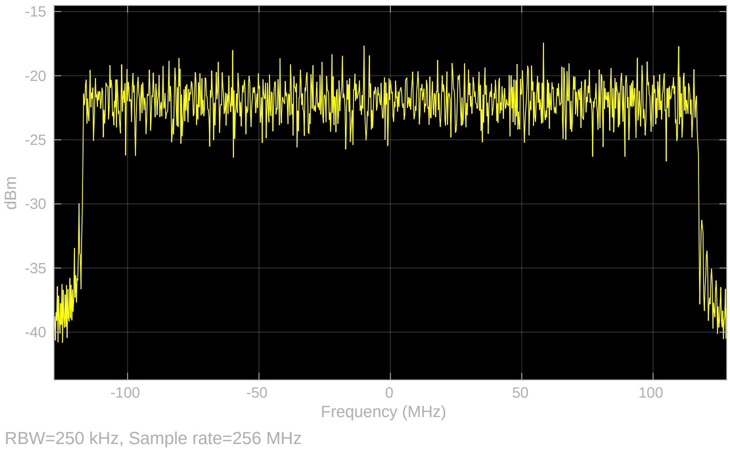
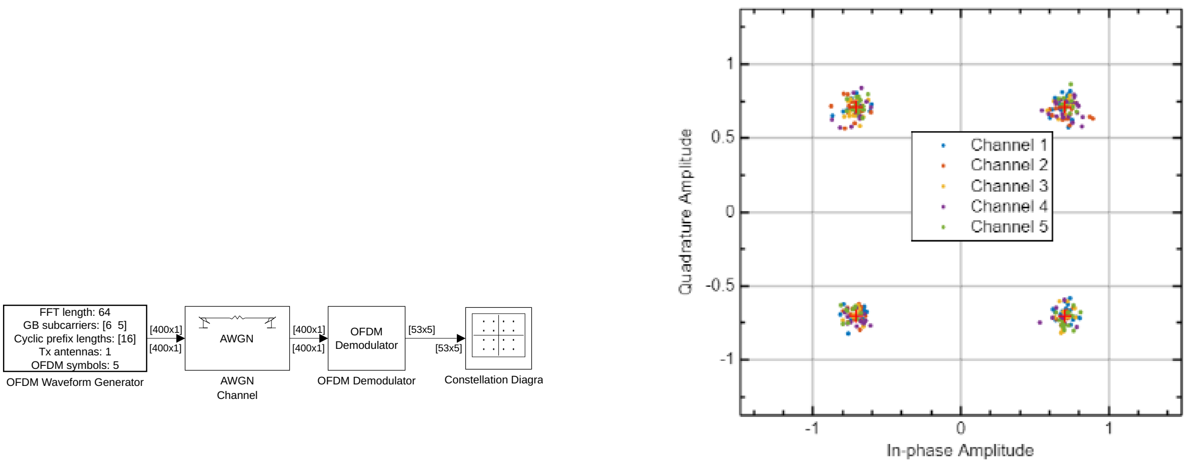


Figure 4.7: Modified OFDM settings signal in frequency domain

Furthermore, the script also exports the OFDM generator with an additive white Gaussian noise block and demodulator block. The received signal is then visualized using a constellation diagram block where we can see that the transmitted signal has been demodulated successfully and the diagram represents the constellation of the QPSK signal as shown in Figure. 4.8



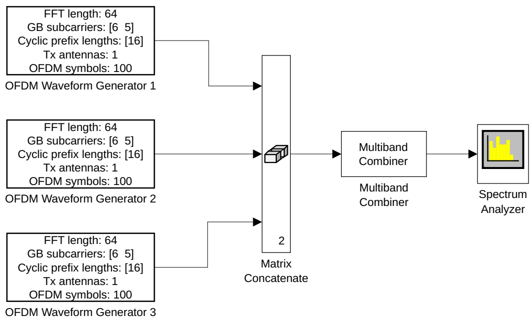
(a) OFDM demodulation system (b) Constellation diagram of the received signal

Figure 4.8: Visualizing an OFDM signal in frequency domain

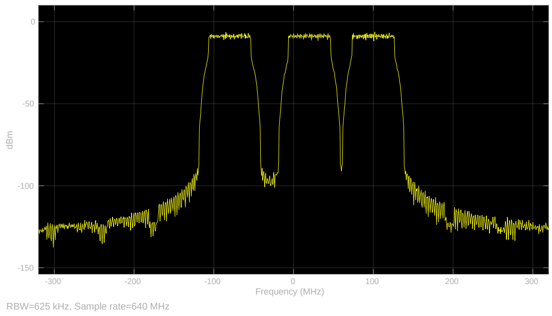
Finally, we combine multiple OFDM signals by combining the output of three OFDM generators with the help of a Multiband Combiner block. The spectrum analyzer shows three OFDM responses separated by guard bands in order to separate them at the receiver. The system model and the result is depicted in Figure. 4.9.

4.4 5G NR CSI-RS Measurements

The 5G NR CSI-RS Measurements example is chosen for this part. The provided script is used to simulate three basic measurements in 5G systems. These basic measurements consist



(a) Multicarrier System Generator



(b) Multicarrier OFDM signal in frequency domain

Figure 4.9: Visualizing an OFDM signal in frequency domain

of CSI-RSRP (CSI reference signal received power), CSI-RSSI (CSI received signal strength indicator), and CSI-RSRQ (CSI reference signal received quality). These basic measurements are used to decide how good the channel is by sending a reference signal that is known for both the transmitter and receiver. Furthermore, the UE (user equipment) changes the current cell that provides better quality or use it in the beam adjustment.

4.5 5G NR CSI-RS Measurements Simulation Results

In Figure 4.10, we plot the reference signal that is used for our measurements. These signals amplitudes and locations are known for both the transmitter and receiver in order to find the channel response and quality from the received signal. We can see that there are two resources (two OFDM symbols) that are occupied by reference signals at location 6 and 10, although in the graph it's shown that the locations are 7 and 11 since it doesn't start with 0 and location is expressed as a zero indexed value.

Furthermore, Figure 4.11, shows the basic measurements on the received resources. In the simulation the power of the reference signal has been set differently for each resource, and for resource 2 the power was three times higher then first resource which explains the difference between the RSRP, RSSI, and RSRQ measurements for the two resources and indicates a better quality for resource 2 which has higher power.

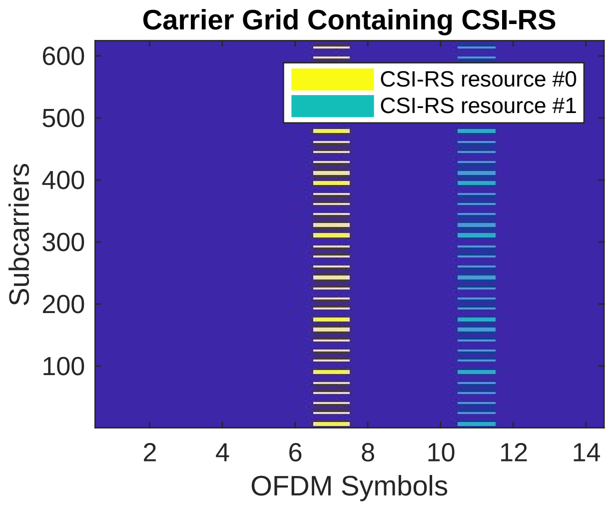
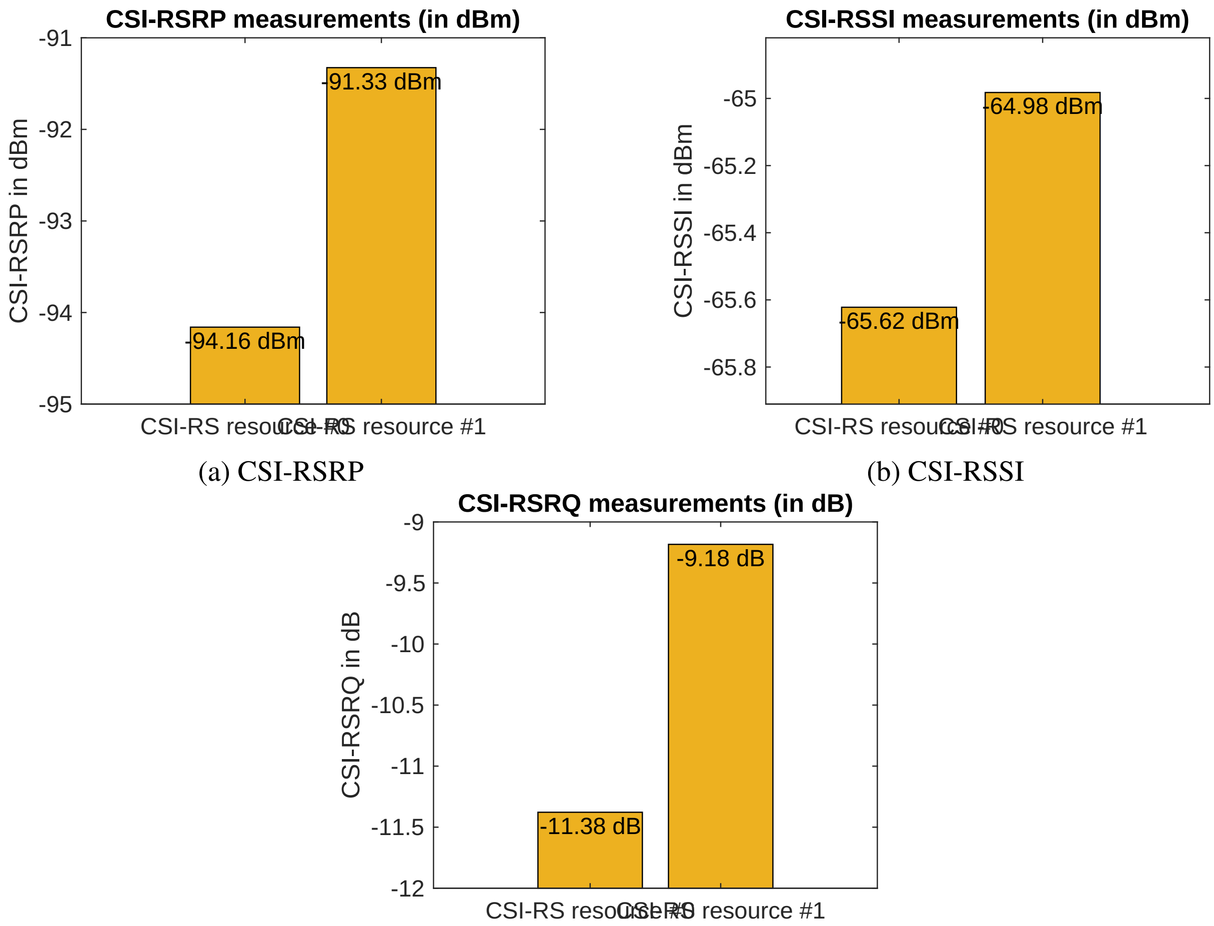


Figure 4.10: CSI-RS resources within a transmitted OFDM frame



(a) CSI-RSRP

(b) CSI-RSSI

(c) CSI-RSRQ

Figure 4.11: RSRP, RSSI, and RSRQ measurements of received OFDM frame using two resource blocks

4.6 5G NR CSI-RS Measurements Simulation Results of Modified Script

In this part we modify some parameters of the provided script. Two parameters are changed in the script. The first parameter is the location of the reference signal resource locations. The new locations are 0 and 7 and the carrier grid is shown in Figure. 4.12.

In addition to that the power of both resources are set equally with the parameter SINRdB which is set equal for both resources. The results shown in Figure. 4.13, which shows almost equal measurements for both resources unlike the previous case where we have a significant difference in the CSI measurements.

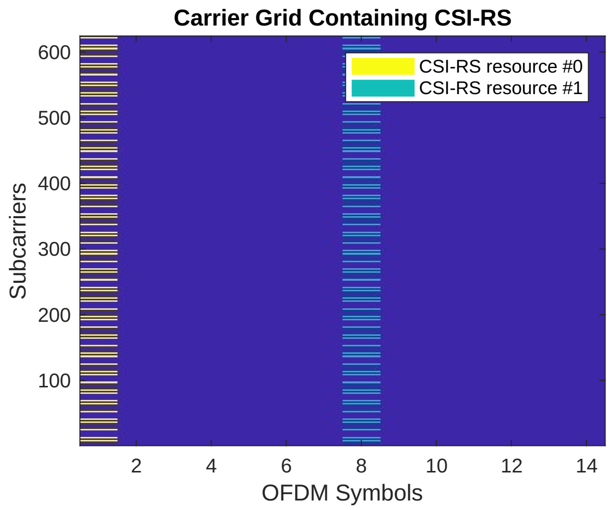
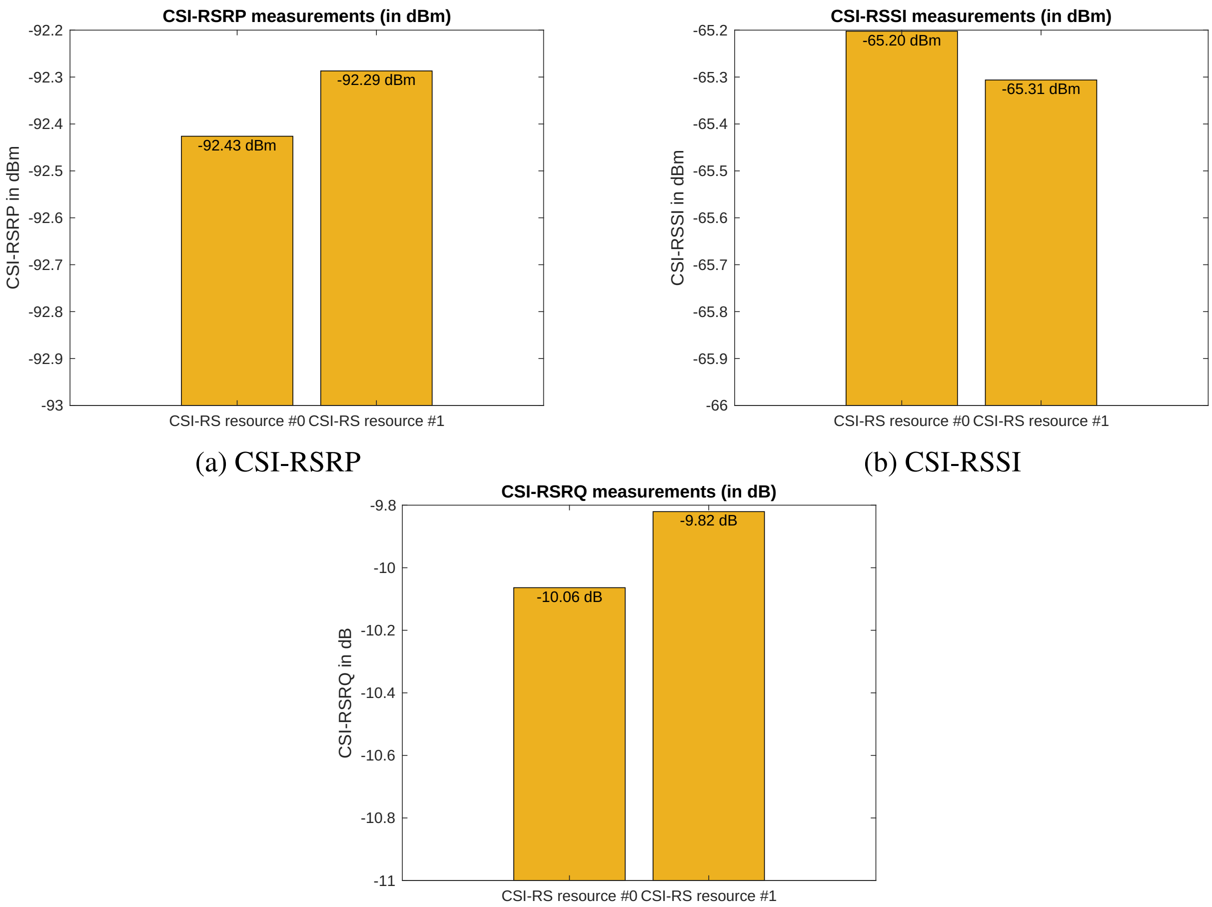


Figure 4.12: CSI-RS resources after modifying the locations



(c) CSI-RSRQ

Figure 4.13: RSRP, RSSI, and RSRQ measurements of received OFDM frame after modifying SINR value

4.7 5G NR Standard

5G New Radio (5G NR) is a new air interface that was developed for 5G. It was built on established technologies to ensure backwards and forwards compatibility for future LTE and 6G technologies, and it was designed to significantly improve the performance, flexibility, scalability, and efficiency of current mobile networks, with the goal of delivering a huge number of diverse services delivered across a diverse set of devices.

There are three general classifications of 5G services, which are outlined here along with some of the advanced wireless technologies required to make them a reality. eMBB (Enhanced Mobile Broadband): Applications that require a large amount of bandwidth. URLLC (Ultra Reliable Low Latency Communication): Services that require extremely high reliability, availability, and security due to latency. Massive Machine Type Communications (mMTC): Low-cost, low-energy devices with large data volumes [4].

5 Self-evaluation

5.1 Self-evaluation of Aissa

- From the different tasks, I learned that simulation runtime affects the simulation results severely. I learned how to quickly iterate on Simulink models with different parameters by using them inside Matlab scripts, which led me to discover new functions and tools in Matlab, such as *berawgn*, *copyobj*, and *squeeze*.
- I struggled with Matlab being non-responsive and crushing sometimes when running simulation models with different parameters in a for loop but overall the work proceeded just fine. I solved the problem of long simulation time by using the online version of Matlab and saving the data of the simulation to generate different plots without the need to rerun the simulation.
- I would evaluate my work and my pair's work as slightly above average and good enough.
- If I started doing the tasks again, I would reread each task's instructions carefully, especially the hints, and highlight every important detail about the simulation parameters. I would also start testing the simulation models with smaller run times and then increase the runtime. In addition, I would, from the beginning, save the simulation results after each run to iterate quickly on the plots.

5.2 Self-evaluation of Abdulmomen

- Doing the tasks really helped me understand how the simulation parameters may affect the simulation results, and how to use Simulink to model simple communication systems and evaluate their performances in terms of BER.
- One of the challenges I faced was that I'm not very familiar with Simulink when I started, although I have some coding experience. However, Mathworks provides very informative documentation that explains every block, and once I am stuck with the functionality or settings of a block I go to the documentation. Another challenge was the long run-time, especially for the Monte Carlo simulation as my computer resources were limited but the problem was solved by running the scripts and Simulink models using the online MATLAB. For task 4, there are many requirements and it took a lot of time to finalize it, and there were many generated figures.
- We tried to do all the tasks and subtasks given in the exercise and I evaluate the work as sufficient.
- We started doing the first tasks early but as the period started and the workload of other courses increased we couldn't finish this work earlier.
- If I had the chance to do this exercise again I'd consider doing it earlier and invest some time every day to do part of the tasks. Furthermore, I'd try to investigate how I can solve the issue of long run-time and start running the simulation with my local computer.

6 Feedback

6.1 Feedback of Aissa

- I am a first year master's student.
- I was familiar with Matlab since I used it in different courses in bachelor degree. I was a bit less familiar with Simulink, which I had only used in one course during the previous period.
- I did Matlab OnRamp course two years ago, and Simulink OnRamp a month before starting this course.
- Approximately 30% of the course content was familiar to me. Specifically, I was familiar with the communications system's general architecture, some channels and their properties, and the BER and SER analysis of the different modulation schemes.
- For me, the course was suitable.

6.2 Feedback of Abdulmomen

- I'm a first-year master's student.
- I'm familiar with Matlab as I have some working experience with it and in my bachelor's three years ago I used it in my thesis that is entitled "Deep learning techniques for OFDM Channel Estimation". However, Simulink was a bit challenging as I wasn't familiar with it that much when I started.
- I did MATLAB OnRamp a long time ago, but I did Simulink OnRamp in this period.
- I'm familiar with 50% of the course content. For example, the BER simulation and channel models.
- The course difficulty was suitable for me.

BIBLIOGRAPHY

[1] Shanmugam K.S. (1979) Digital and analog communication systems. NASA STI/Recon Technical Report A 80, pp. 23225.

[2] Analyze ber performance of communications systems - matlab. https://www.mathworks.com/help/comm/ref/biterrorrateanalysis-app.html, [Online; accessed 5-December-2022].

[3] Ber and ser for uncoded data over awgn channels - matlab berawgn. https://www.mathworks.com/help/comm/ref/berawgn.html, [Online; accessed 5-December-2022].

[4] Kavanagh S. (2022), What is 5g new radio (5g nr). https://5g.co.uk/guides/what-is-5g-new-radio/.