CE 3006 DIGITAL COMMUNICATION COURSE PROJECT

Group Number: 1

Group Members:

Alex Neo (U1820416G)
Joshua Ng (U1821510F)
Cheng Chee How(U1822033D)
Mark Chua De Wen(U1821979L)
Munmit Singh Dhillon(U1822404L)

Introduction

Background

In a Digital Communication System, message signals are usually generated by the source in analog format which is then converted into digital format before it is transmitted. At the receiver end, the received data is converted back to the analog format. Between the transmitter and the receiver, there consists of encoding, modulating and transmitting the signal over the communication channel.

In this project, we tried the different modulation schemes with Additive White Gaussian Noise (AWGN), such as On-Off Keying (OOK), Binary Phase Shift Keying (BPSK), Binary Frequency Shift Keying (BFSK) and 16 Quadrature Amplitude Modulation (16-QAM), which will varying impacts on the performance of bit error rate. Additionally, we implemented encoding schemes such as Cyclic code, Hamming code, Turbo code and Reed-Solomon Code in an attempt to improve the performance.

Objective

The objective of this project is to develop basic communication using MATLAB. From the use of Matlab, we learned to generate, analyse the different data and apply the knowledge we have learnt throughout this course.

Phase 1: Data Generation

In Phase 1, we will be discussing the topic of data generation. Our focus will be on how we generated random binary data and white gaussian noise. Additionally, we will be discussing how we combined the above mentioned and plotted it against various SNR values to analyse the bit error rate performance.

1. Generation of Binary Data and Noise

In this section we will be covering the steps we took to achieve the requirements that were listed out for this project.

1.1. System layout

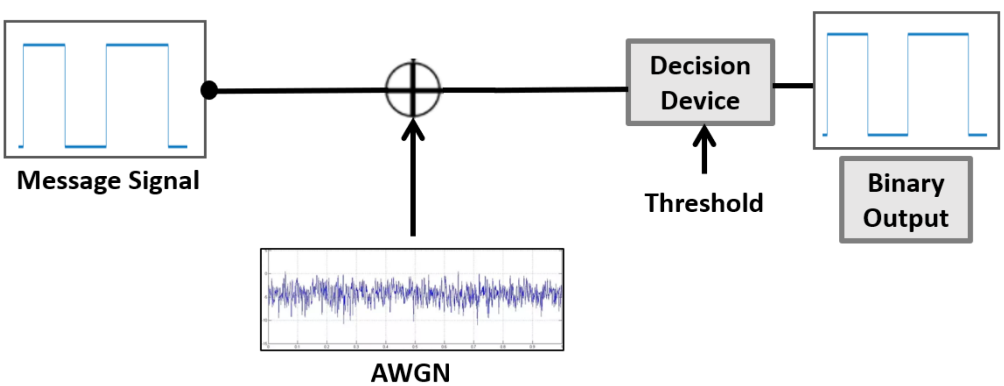


Figure 1: Block Diagram for Data Generation

A. Number of bits for transmission is 1024. (N = 1024)
B. Random binary digits to be generated will be 0 or 1.
    a. This was achieved with the use of the *randi* function in MATLAB for values 1 to N. This is to simulate unipolar line coding.
C. Binary digits to be converted from 0 and 1 to -1 and +1 respectively.
    a. This was achieved by multiplying the random binary digits by two followed by subtracting one. This is to convert the unipolar to a bipolar line coding.
D. Noise was generated using the *randn* function in MATLAB while ensuring that it will have normal distribution with zero mean and unit variance.
    a. A function was generated to vary the SNR values from 0 to 50 with increments of 5. The various SNR values are then used to change the noise variance.

b. SNR (in dB) = 10log10 (S/N) where S is the Signal power (or variance) and N is Noise power (or variance).
c. Once this was generated, the White Gaussian Noise was then added to the transmitted data which becomes our received signal.

E. A threshold function was created to convert the received signal based on the threshold value of 0. Any signal received above the threshold is assigned to 1 while any signal received below the threshold is assigned to 0.
F. A function was created to compute the bit error rate during the transmission with respect to the various SNR values.
    a. The computation of Bit Error Rate $=\frac{Number\ of\ Errors\ During\ Transmission}{Total\ Number\ of\ Bits\ for\ Transmission}$
    b. This was computed for the various values of SNR from 0 to 50 with increments of 5.
G. We then plotted the results in a graph with the X axis being the SNR values while the Y axis being the Bit Error Rate.

1.2. Analysis of Bit Error Rate against SNR values

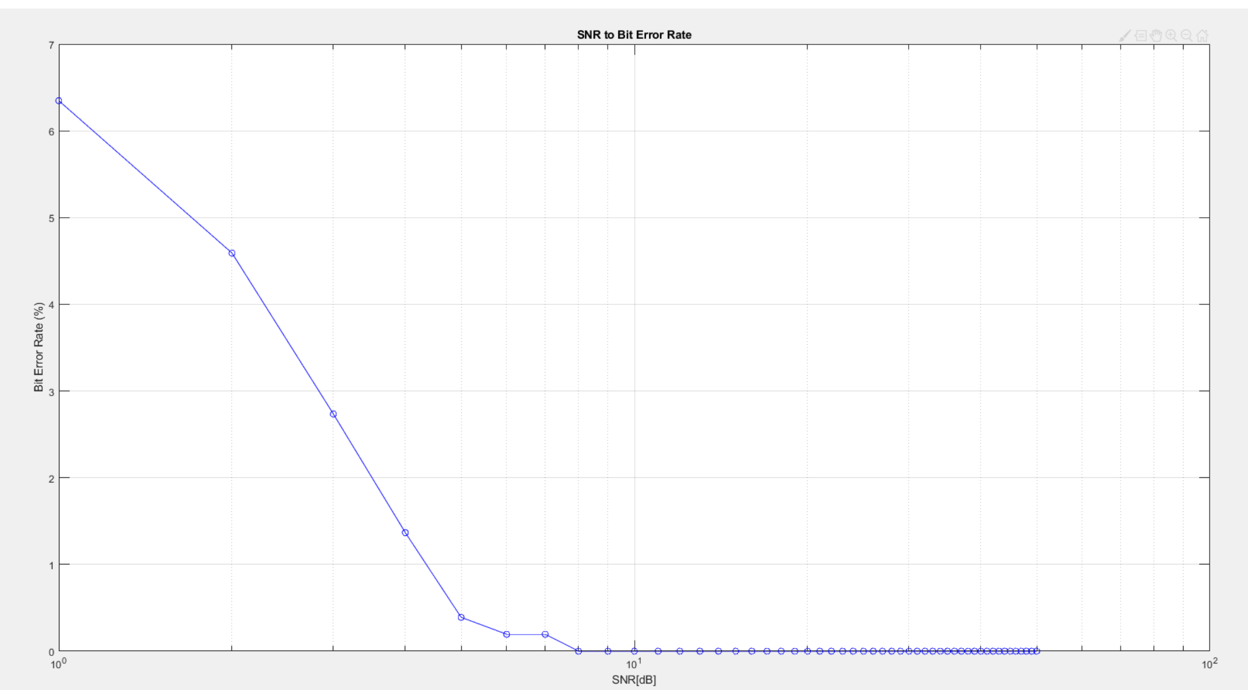


Figure 2: SNR to BER line graph

Phase 2: Modulation for Communication

In this section below, we will be discussing four modulation techniques that we implemented for this project. The modulation techniques are, On-Off Keying (OOK), Binary Phase Shift Keying (BPSK), Binary Frequency Shift Key (BFSK), 16-Quadrature Amplitude Modulation (16-QAM). Additionally, we will be providing in-depth analysis for each of the modulation techniques used and how they compare with each other.

For this project there are some parameters that applied to all the modulation systems used.

The parameters are as follows,

- Only Double-sided Additive White Gaussian Noise Channel will be used
- Message data rate is 1000 bits per second
- Carrier frequency will be 10Khz with the exception of BFSK
- Carrier Sampling frequency is 16 times the carrier frequency (160Khz)
- SNR range from -100 to 0

1. On-Off Keying (OOK)

1.1. Introduction

OOK is the simplest form of Amplitude Shift Keying modulation that represents the digital data. Binary 1 represents the presence of the carrier wave while Binary 0 represents the absence of the carrier wave. Transmission of signal depends on the ‘1’ and ‘0’ of the carrier. Transmission is switched on when it is ‘1’ and off when it is ‘0’. OOK is usually used to transmit Morse code over the radio frequency.

1.2. System layout

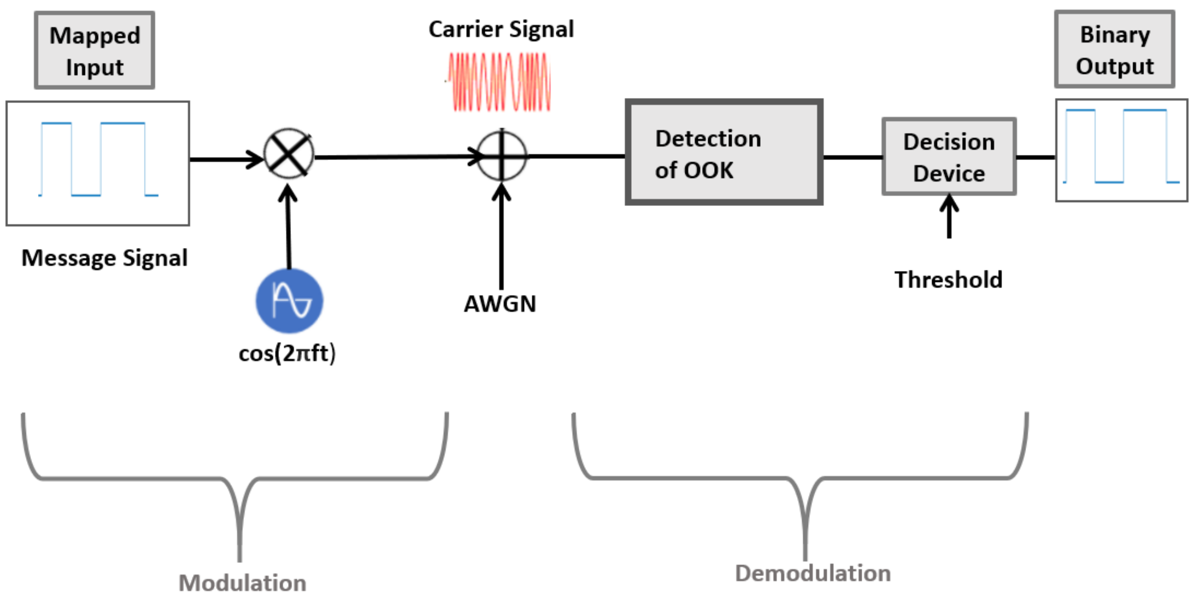


Figure 3: On-Off Keying Block Diagram

For transmission, the binary message signal is processed with the carrier signal of ($\cos(2\pi ft)$). The values in the message bits are mapped with 1 and 0. It is then modulated by multiplying with the carrier signal, $\cos(2\pi ft)$. This results in transmission of signal when the carrier signal is 1 and no transmission when 0.

At the receiving end, the received signal is multiplied by 2 times of the carrier. It is then passed through a low pass filter to filter out the higher frequency of the signal. In the end, it is passed through the decision device with the threshold being set to 0.5 for the delivery of the digital output.

1.3. Analysis of communication system

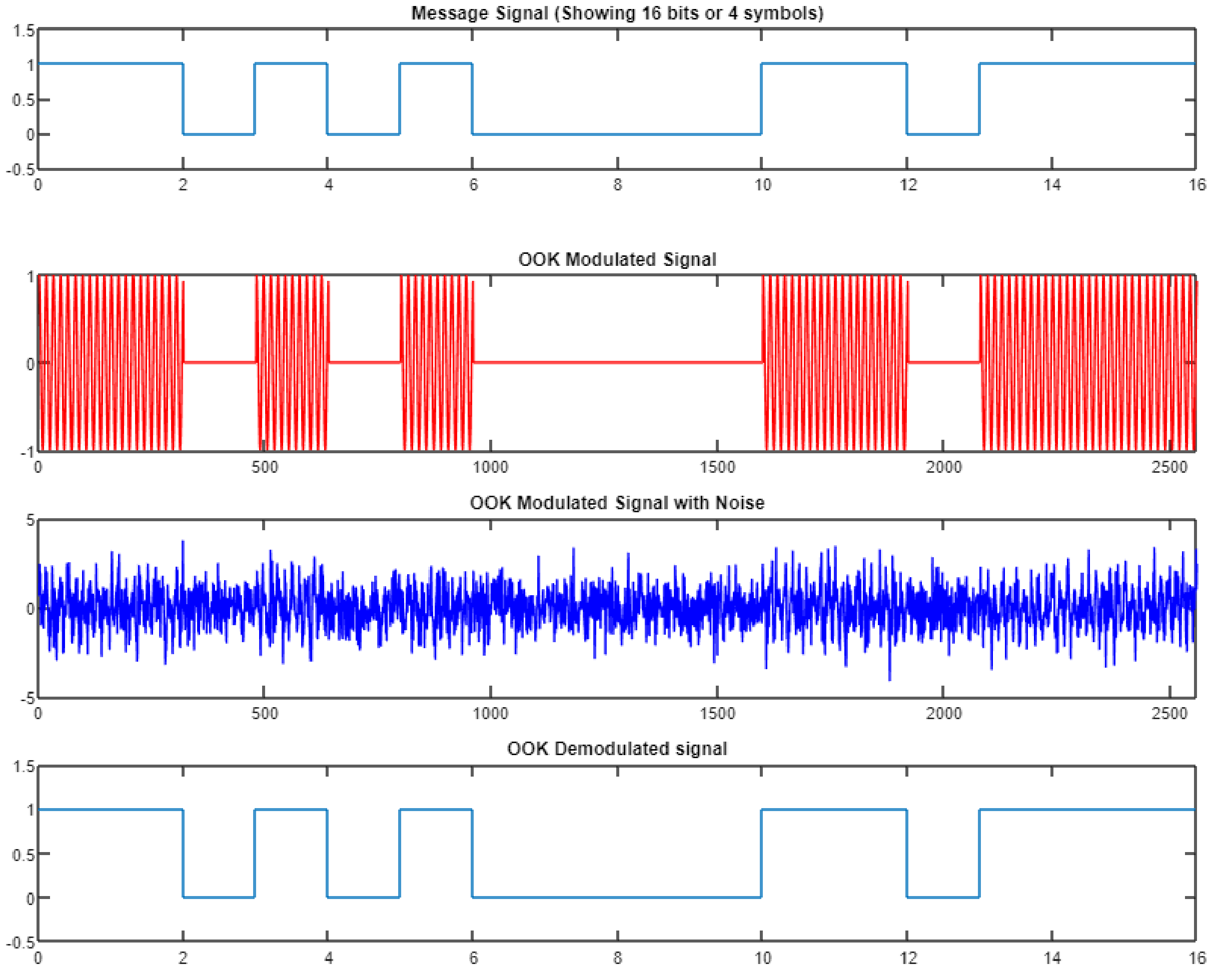


Figure 4: Signal Representations after each stage of communication for OOK

Figure 4 shows the different stages of the digital communication using the On-Off Keying (ASK) technique. The top plot shows the first 16 bits of the random generated binary signal with quantization of binary 1 and 0. After the On-Off Keying Modulation, the transmission of carrier signal is turned on with the logic ‘1’ of the message signal and no transmission signal with logic ‘0’. The third plot shows the carrier signal is transmitted into the channel where the noise is being introduced using the AWGN function. The OOK modulated signal with noise is distorted significantly, making it difficult to see the change. Immediately after exiting the channel, the received signal is demodulated, filtered and transmitted through a decision device to recover the original message signal for the first 16 bit shown in the last plot.

2. Binary Phase Shift Keying (BPSK)

2.1. Introduction

In BPSK, the phase of the signal is used to differentiate bit ‘0’ and bit ‘1’. There are only 2 phases in BPSK, having the ‘0’ bit sinusoidal representation 180 degree out of phase from the ‘1’ but sinusoidal.

2.2. System layout

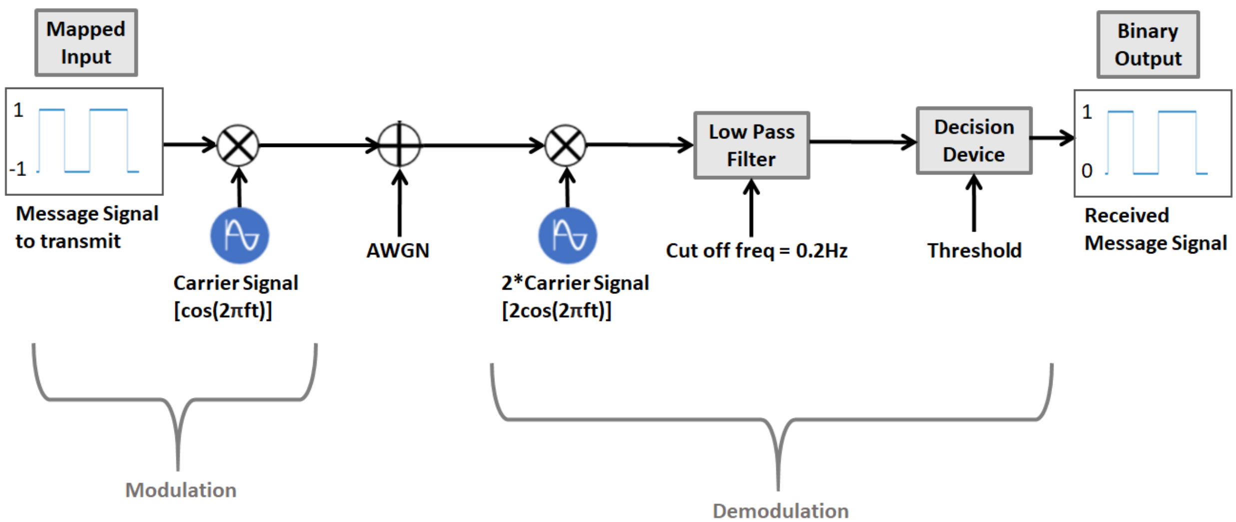


Figure 5: Block Diagram for Binary Phase Shift Key

At the transmitter, instead of multiplying the signal by 2 separate carrier signals that are 180 degree out of phase, the message signal is processed before multiplying with a single carrier signal. The bit values in the message signal are mapped from ‘0’ and ’1’ to ‘-1’ and ‘+1’ respectively. The remapped message signal is then modulated by multiplying it with the carrier signal, $\cos(2\pi ft)$. This results in the ‘0’ bit represented by -$\cos(2\pi ft)$ and the ‘1’ bit represented by $\cos(2\pi ft)$, which is 180 degree off phase from each other.

At the receiver, the received signal is multiplied by 2 times of the carrier, then passed through a high pass filter to remove the imaginary part of the signal, leaving the real part/amplitude of the signal. The signal is then passed through a decision machine with threshold set at 0. Positive amplitude values categorized as bit ‘1’ and negative amplitude values as bit ‘0’.

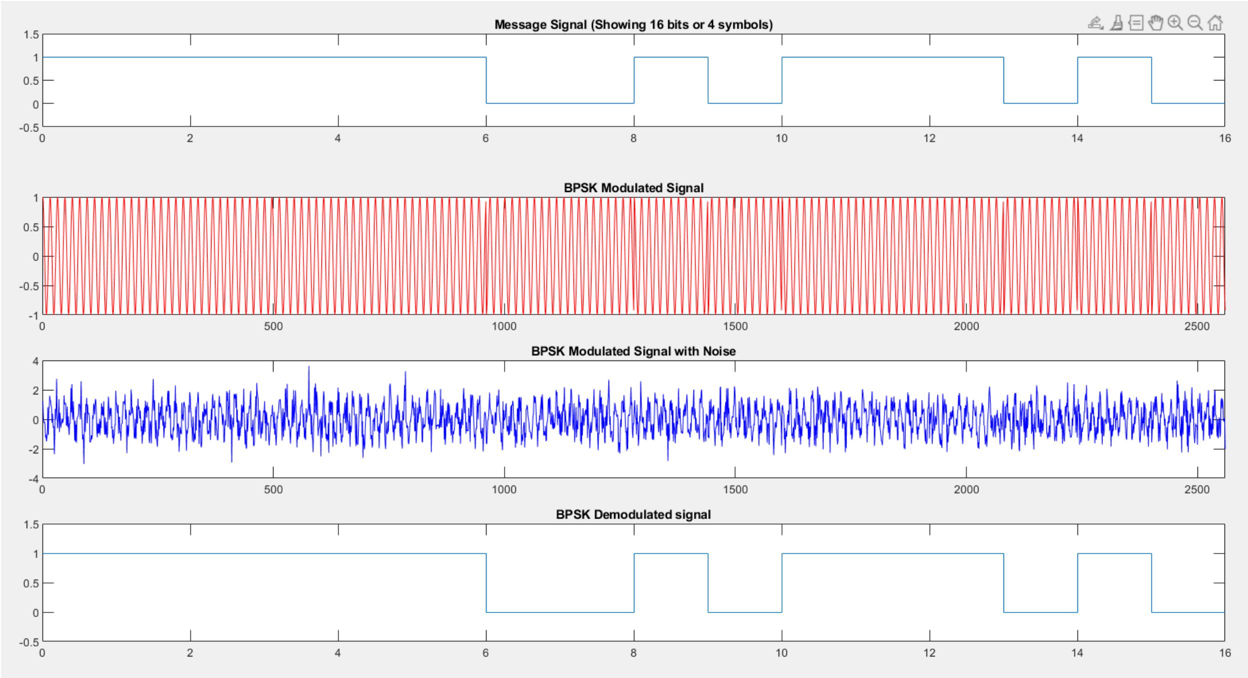


Figure 6: Signal Representations after each stage of communication for BPSK

2.3. Analysis of communication system

Figure 6 displays the different stages of digital communication using the BPSK modulation technique at SNR 0. The first plot shows the first 16 bits of a randomly generated message signal, having a binary quantization of ‘0’ and ‘1’. After BPSK modulation, the signal becomes a cosine wave with the carrier frequency and the amplitude of -1 to 1. Whenever there is a change of bit in the signal, there will be a 180 degree change in phase for the cosine wave. The cosine wave signal is transmitted into the channel, where noise is introduced using the awgn function. The BPSK modulated signal with noise is distorted significantly, making it difficult to visibly see the change in phase. Upon exiting the channel, the received signal is demodulated, filtered and put through a decision machine to recover the original signal. Despite the noise, the original signal for the first 16 bits are recovered accurately.

3. Binary Frequency Shift Keying (BFSK)

3.1. Introduction

In BFSK, the frequency of the carrier signal is the parameter chosen to be a function to represent the signal. For transmission, two local generators used to generate two frequencies are selected to transmit 1 and 0 respectively.

In FSK, there are various ways to generate or modulate the signals such as using distinct frequency sources or using voltage controlled oscillator(VCO). In this implementation, we used the method of switching between distinct frequency sources to generate the carrier signal.

3.2. System layout

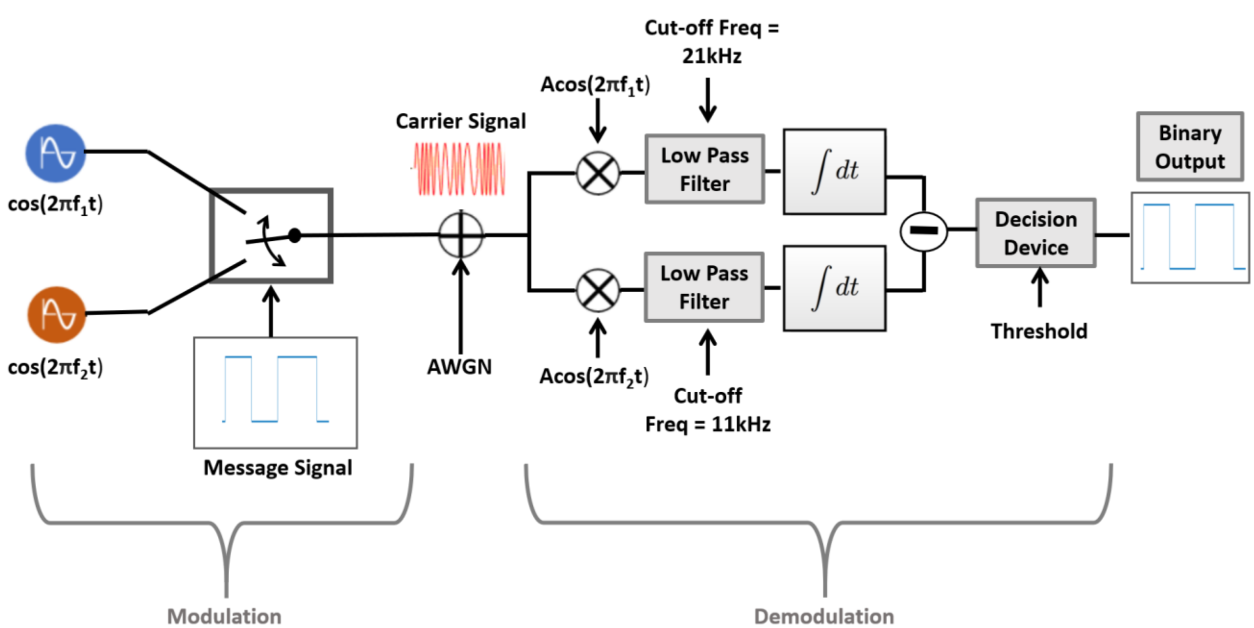


Figure 7: Block Diagram for Binary Frequency Shift Keying

For transmission, the message signal is generated from phase 1's code.
The message signal is then used to generate the carrier signal by switching between both frequencies. The switch will then switch to frequency 1 when the message bit is high and frequency 2 when the message bit is low.

Between the transmitter and receiver, additive white gaussian noise is added to simulate real digital transmission. The received signal is then processed by a coherent BFSK demodulator. The received signal is multiplied by the known frequency 1 and 2 and passed through low pass filters for removal of noise.The output of the low pass filter is then passed into the integrator. The output of integrator 1 is subtracted by integrator 2. The process of multiplying the output and integrating with the known transmitting frequency is also known as matched filtering. The output of the matched filtering is passed through the decision device, if the result is less than the threshold then the output will be low and if the result is more than the threshold, the output will be high.

3.3. Analysis of communication system

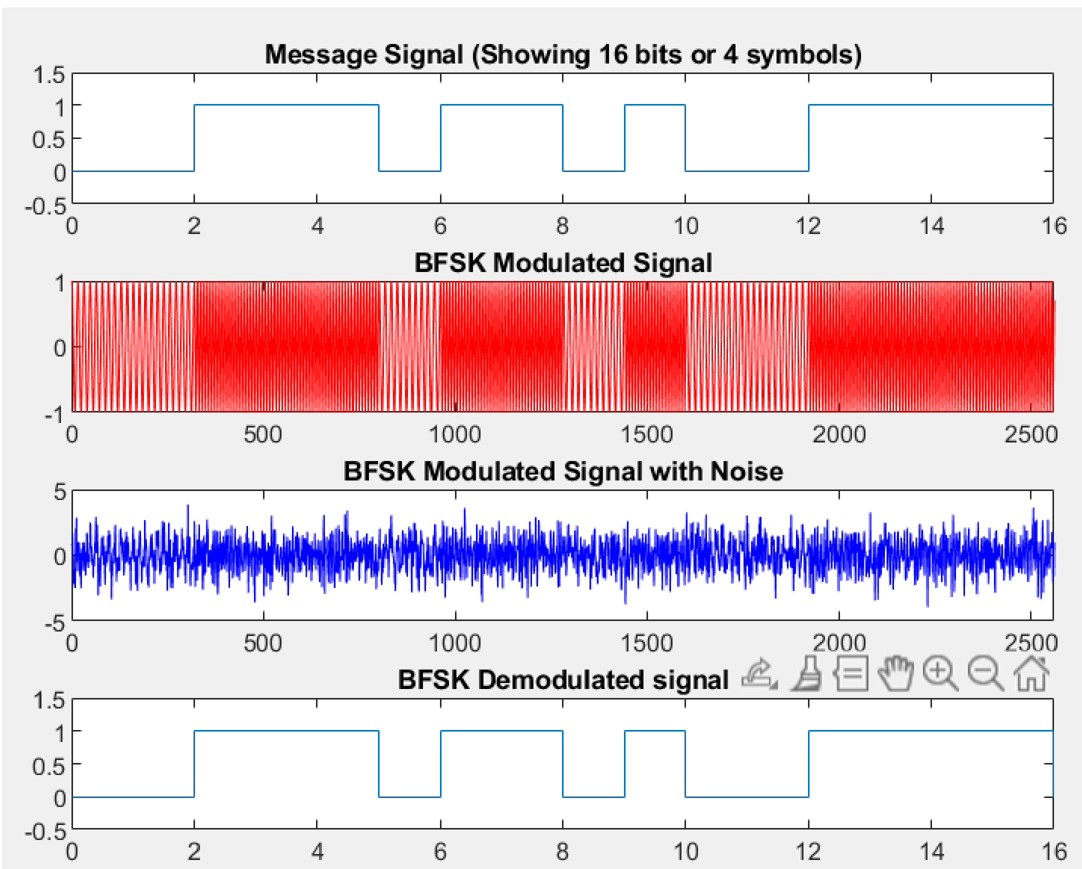


Figure 8: Signal Representations after each stage of communication for BPSK

Figure 8 shows the different signals involved between the different stages of communication. The 1st plot shows the randomly generated message signal from phrase 1 that is sent to the BFSK modulator which will convert the message signal into a carrier frequency that corresponds to 20kHz when message bit is a 1 and 10kHz when the message bit is a ‘0’ which is seen the 2nd plot of figure 8. When the modulated signal is passed through a medium. Additive White Gaussian Noise(AWGN) is added into the signal which can be seen by the difference between the 2nd plot and the 3rd plot. The demodulator will then demodulate the signal and use the multiplier, low pass filter and integrator as described above and if the output is less than the threshold(0), the output is ‘0’ and if it is higher than the threshold(0). The output is ‘1’ which is shown in the 4th plot of figure 8.

4. 16-Quadrature Amplitude Modulation (16-QAM)

4.1. Introduction

Quadrature Amplitude Modulation is a digital modulation technique that uses two carriers which are out of phase with each other by 90°.This means that the two carriers are in quadrature which is the reason this modulation is called Quadrature Amplitude modulation as it alters the amplitude of quadrature carriers for modulating data.

For this project we are implementing 16-QAM with the help of MATLAB’s communication toolbox to simplify the code although manual methods are also included in the source codes as reference.. The number 16 indicates the number of unique symbols that can be modulated allowing 4 bits (log2(16) = 4) of data to be modulated at one time and transmitted over the channel. Because QAM is capable of modulating multiple bits at one time it is capable of achieving better spectral efficiency.

4.2. System layout

16-QAM BLOCK DIAGRAM FOR MATLAB (Transmitter)

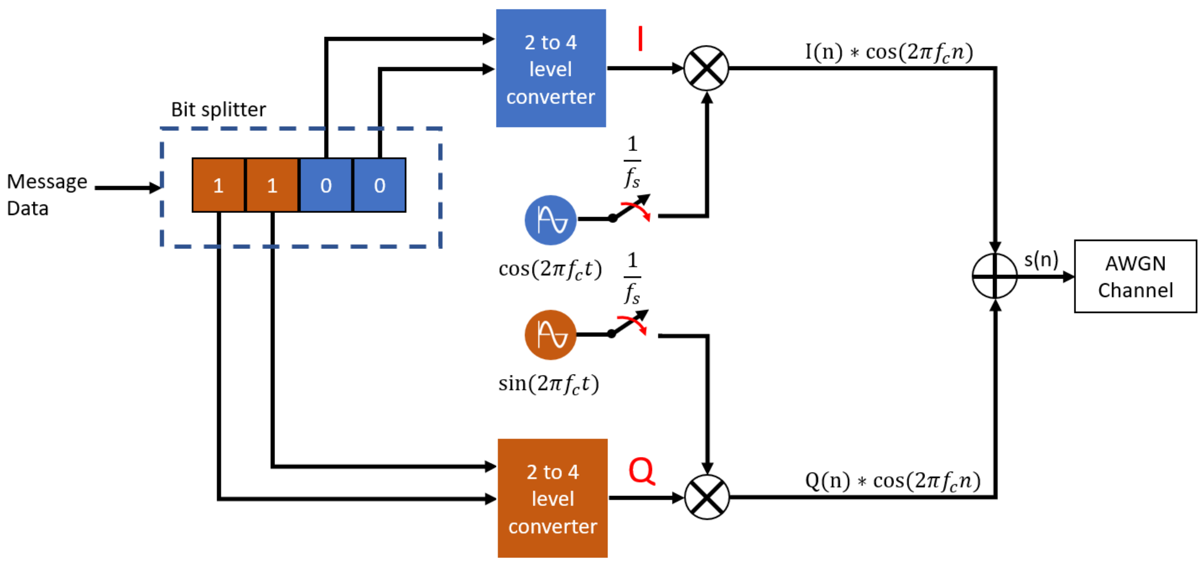


Figure 9: Block Diagram for 16-QAM Modulation

Figure 9 shows a generalized block diagram of the communication system that was designed in MATLAB for this project. The transmitter portion consists of 3 main components. The quadrature carrier waves the 2 to 4 level converter and also the bit splitter. Further details will be provided during the analysis portion of this chapter.

16-QAM BLOCK DIAGRAM FOR MATLAB (Receiver)

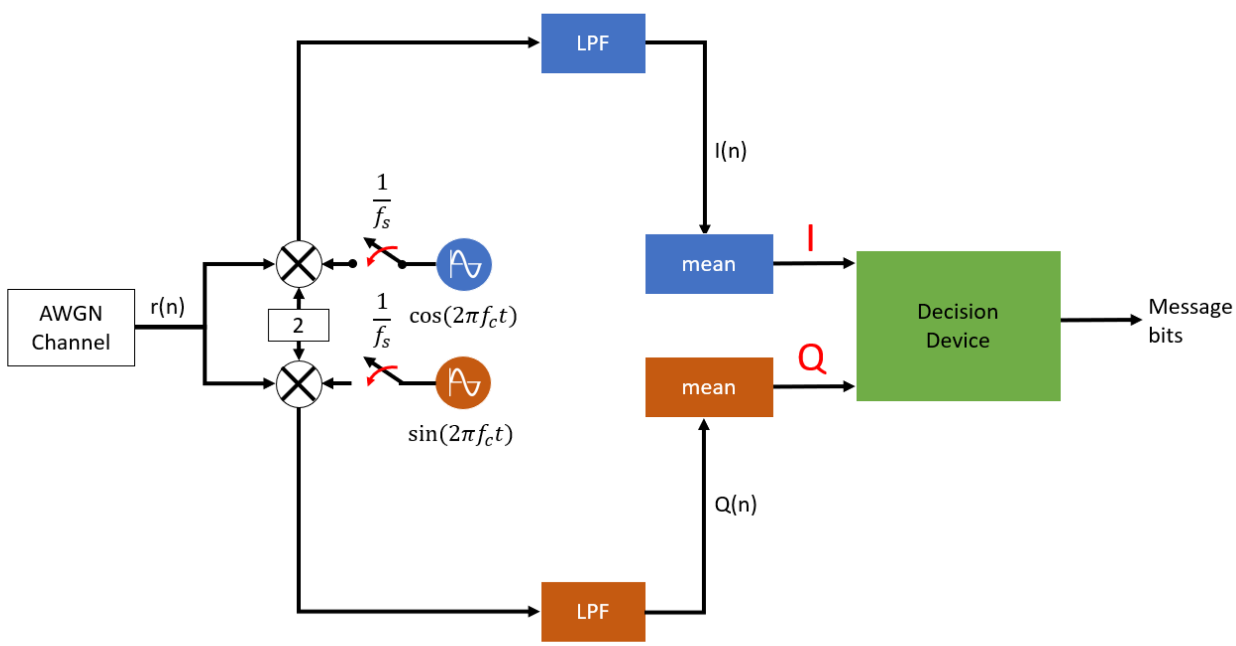


Figure 10: Block Diagram for 16-QAM Demodulation

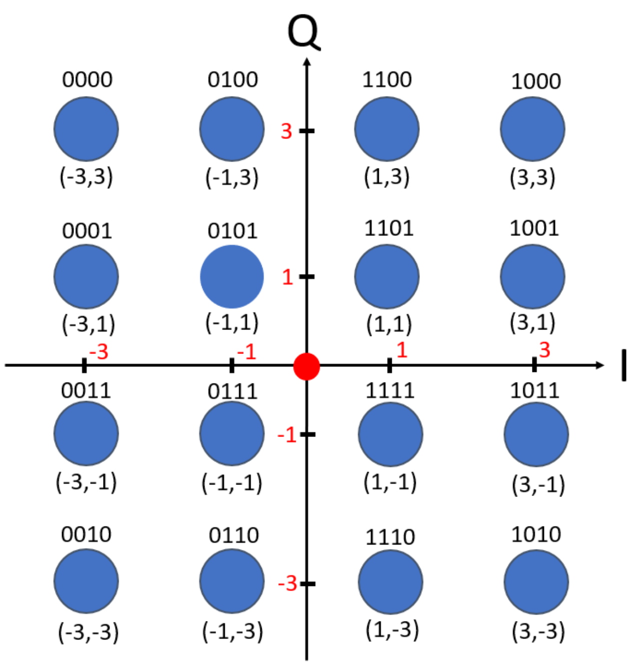
Figure 10 shows the implementation of the receiver side of the 16-QAM system for this project. The receiver consists of 3 main components.The quadrature carriers, low pass filter,a mean block to calculate the mean value of the In-phase and Quadrature data streams and finally the decision device which will make a decision based on the given In-phase and Quadrature values given.

4.3. Analysis of communication system (Transmitter)

4.3.1. Data to symbol mapping

The 16-QAM communication system consists of 16 different symbols which allows 4 bits of data to be modulated and transmitted at one time. The different In-phase and Quadrature components that will be used to map the 16 different symbols is shown in Figure 11 using a constellation diagram.

16-QAM Constellation Diagram (Symbol Mapping)



Symbol	I	Q
0000	-3	3
0001	-3	1
0010	-3	-3
0011	-3	-1
0100	-1	3
0101	-1	-1
0110	-1	-3
0111	-1	-1
1000	3	3
1001	3	1
1010	3	-3
1011	3	-1
1100	1	3
1101	1	1
1110	1	-3
1111	1	-1

Figure 11: 16-QAM Modulation Constellation Diagram

4.3.2. The bit splitter

The bit splitter is responsible for splitting the upper 2 bits and lower 2 bits for generating the appropriate I and Q values that will be used to modulate the carriers. Firstly the bit splitter will read 4 bits of message data. The reason the bit splitter reads 4 bits is because 16-QAM consist of 16 unique symbols that can be mapped. Therefore the amount of bits that can be modulated or mapped to a symbol can be calculated using the formula shown below.

$$Total\ number\ of\ bits\ that\ can\ be\ modulated\ =\ log2(N)$$

$$N\ =\ Number\ of\ mappable\ symbols\ in\ constellation\ diagram$$

The upper 2 bits will be used to generate the Q value and the lower 2 bits will be used to generate the I value

4.3.3. The 2-to-4 level converter

The 2 to 4 converter converts the 2 bits that are fed by the bit splitter into a voltage level or in this case a I or Q value. The design largely depends on the user and even a 2 to 4 decoder can be used where 2 bits will be decoded into one of the 4 outputs which indicate the different values that I and Q can take which is -3,-1,1 and 3.

4.3.4. The two carrier waves

The two carrier waves are the key components that allow QAM to modulate the signals in such a way that allows the manipulating of the phase and amplitude of the carrier signal. This is because of the relationship between two signals that are out of phase from each other by 90°.

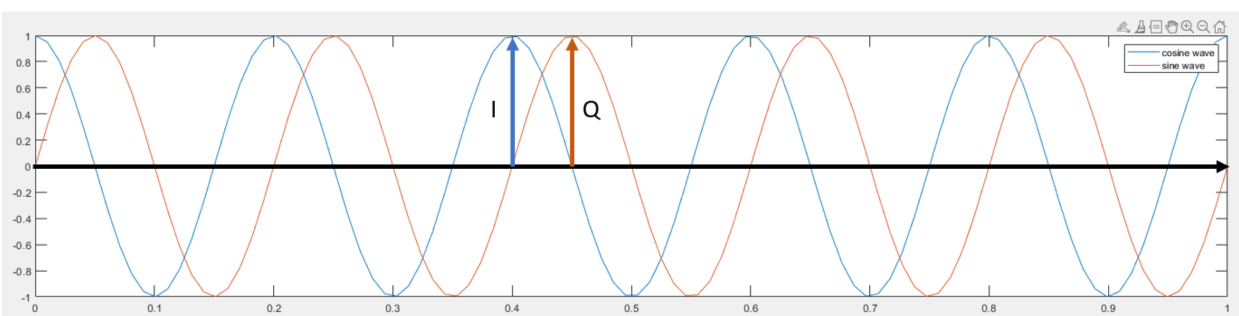


Figure 12: Carrier Signals

Figure 12 shows two carrier waves that are in quadrature to each other. The first signal which is the blue signal is the cosine wave while the orange signal is sine wave. Let's now take note of the amplitude and label the amplitude of the cosine wave as ‘I’ and the amplitude of sine wave as ‘Q’.

In QAM a carrier wave is modulated by a message signal by altering the amplitude and phase. Altering the carrier’s amplitude is a known technique also called amplitude shift keying (ASK) and altering the phase is called phase shift keying (PSK). But this just complicates the circuit design as the amplitude and phase shift are all variables which increases complexity.

However there is a way to create an amplitude and phase shift on a carrier by modifying the amplitudes of the carrier wave and its 90° phase shifted version and adding them together which will create an amplitude altered and phase shifted version of the carrier.

We now modify the amplitude of the carrier by setting the I value to -1 and the Q value to 0 which means it should create the carrier wave with its amplitude multiplied by -1 which is also an inverted cosine wave..

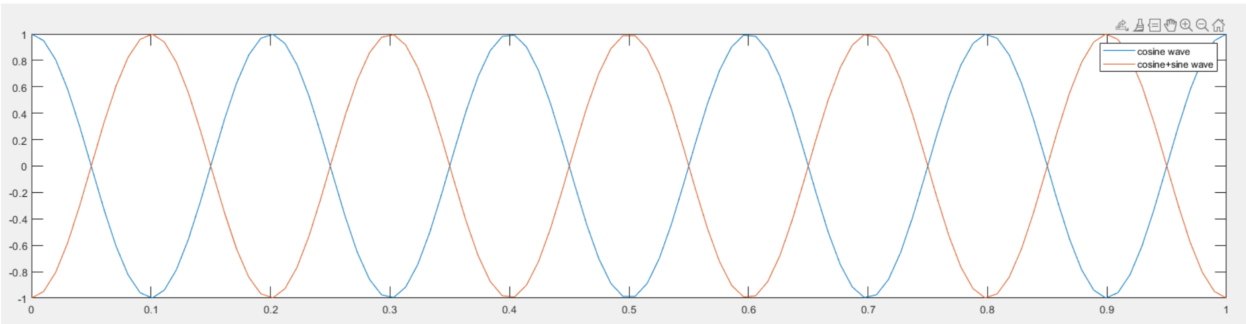


Figure 13: Carrier signals with amplitude altered

As shown in figure 13 the amplitude of the resulting carrier is now a cosine wave with amplitude -1 or an inverted cosine wave.

Next we alter the amplitude and phase of the carrier shown in blue by altering the I and Q values to -1 and -3 respectively.

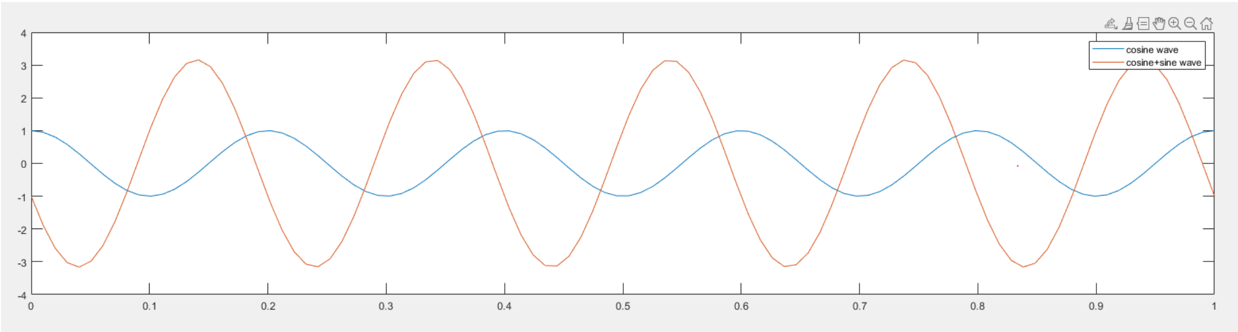


Figure 14: Carrier signals with amplitude and phase altered

As shown in figure 14 the amplitude and phase of the carrier is now altered which means that I and Q values can be used to alter a carrier wave amplitude and phase and to achieved a certain amplitude change or phase shift the encoder will need to use the right I and Q values that give the correct signal.

This allows the QAM encoder circuit to be simpler to design as the encoder just consists of 2 ASK encoder circuits with two different carriers with a linear adder which reduces the circuit complexity.

4.4. Analysis of communication system (Receiver)

4.4.1. Demodulation of received signal using carrier waves

For this project we are using coherent detection for the received signal which means that the two carriers are phase locked with the transmitter. The goal of the demodulator is to extract the I and Q data streams of the received signal.

To retrieve the I and Q data streams we will first multiple the received signal with the two different carriers multiplied by 2.

$$r(n) = I(n) * cos(2\pi fcn) + Q(n) * sin(2\pi fcn)$$

$$y1(n) = (I(n) * cos(2\pi fcn) + Q(n) * sin(2\pi fcn)) * 2 * cos(2\pi fcn)$$

$$y1(n) = 2 * (I(n) * cos(2\pi fcn) * cos(2\pi fcn) + Q(n) * sin(2\pi fcn) * cos(2\pi fcn))$$

$$y1(n) = 2 * \frac{1}{2} * (I(n) * cos(4\pi fcn) + I(n) * cos(0) + Q(n) * sin(4\pi fcn) + Q(n) * sin(0))$$

$$y1(n) = I(n) * cos(4\pi fcn) + I(n) + Q(n) * sin(4\pi fcn) + 0$$

$$y2(n) = (I(n) * cos(2\pi fcn) + Q(n) * sin(2\pi fcn)) * 2 * sin(2\pi fcn)$$

$$y2(n) = 2 * (I(n) * cos(2\pi fcn) * sin(2\pi fcn) + Q(n) * sin(2\pi fcn) * sin(2\pi fcn))$$

$$y2(n) = 2 * \frac{1}{2} * (I(n) * sin(4\pi fcn) - I(n) * sin(0) + Q(n) * cos(0) - Q(n) * cos(4\pi fcn))$$

$$y2(n) = I(n) * sin(4\pi fcn) - 0 + Q(n) - Q(n) * cos(4\pi fcn))$$

y1(n) and y2(n) shows the resulting signal after multiplying the received signal with the cosine carrier and sine carrier respectively.

From the resulting signal equations it is shown that the I(n) and Q(n) components are inside but however there is a sin and cos component in the signal which prevents the demodulator from extracting the I and Q functions.

Fortunately, to extract the I and Q components the demodulator will need to filter out the cosine and sine components. This is a job for the low pass filter which will be explained in the next chapter.

4.4.2. Low-pass filter and the mean

The low pass filter is designed to filter out the high frequency components or in this case also used to filter the cosine and sine components of y1(n) and y2(n). For this project the cut off frequency of the low pass filter is 10Khz this is to filter out the sine and cosine components and return the I and Q data streams that were used to modulate the carrier at the transmitter.

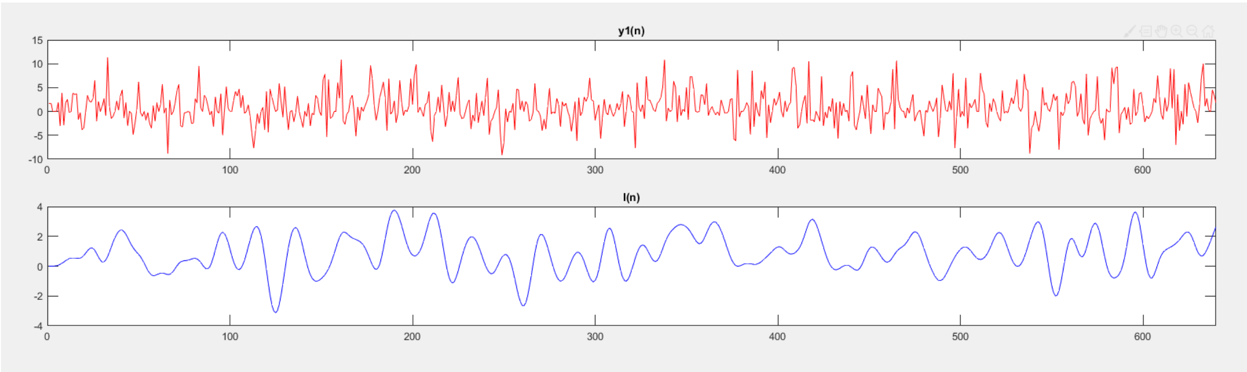


Figure 15: Signal before and after passing through Low-Pass Filter

Figure 15 shows the I(n) data stream after sending the y1(n) through the low pass filter. This data stream will then be sent through the mean block to calculate the mean of the I or Q values over a symbol period. This will give an approximate value which will be sent to the decision device to convert to the relevant symbols.

4.4.3 Decision device

The final block is the decision device which makes a decision by mapping the received I and Q values to a symbol on the constellation diagram. This can be done by using hard decisions methods or soft decision methods. The resulting signal will then be used to compare with the message data to calculate the bit rate error.

4.5. The communication system in action

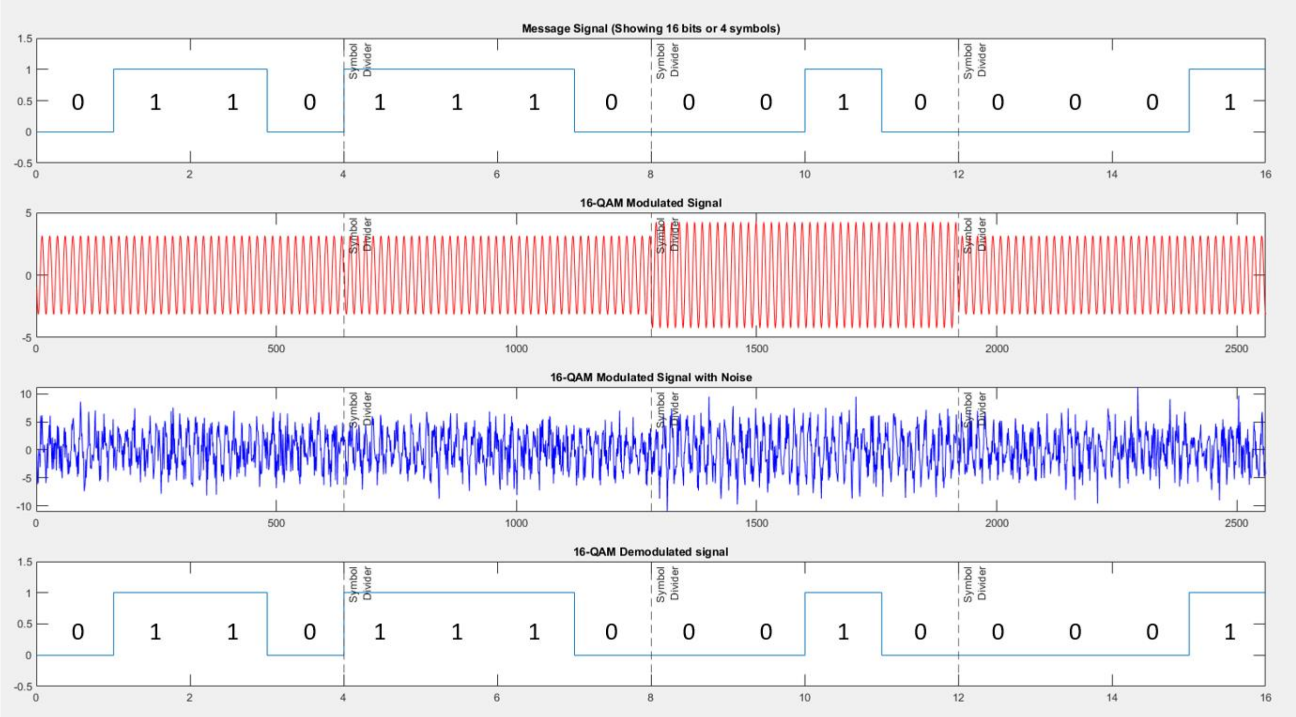


Figure 16: Signal Representation after each stage of communication for 16-QAM

Figure 16 shows the different stages of the waveform when a 16 bit message data is sent through the 16-QAM communication system the first row shows the message data that is going to be modulated while the second row shows the modulated carrier based on the symbols. As each symbol is 4 bits the carrier is modulated by 4 bits. The modulated signal might look the same between 0110 and 1110 but as the phase shift is small and the frequency of the carrier is high it may not be visible to the naked eye.

5. Analysis of different modulations (BER explanation)

The transmission accuracy of each modulation technique is evaluated using bit rate error, the percentage of bits in the message that is erogenous. This is repeated for a range of signal to noise ratios (SNR), -100 to 0.

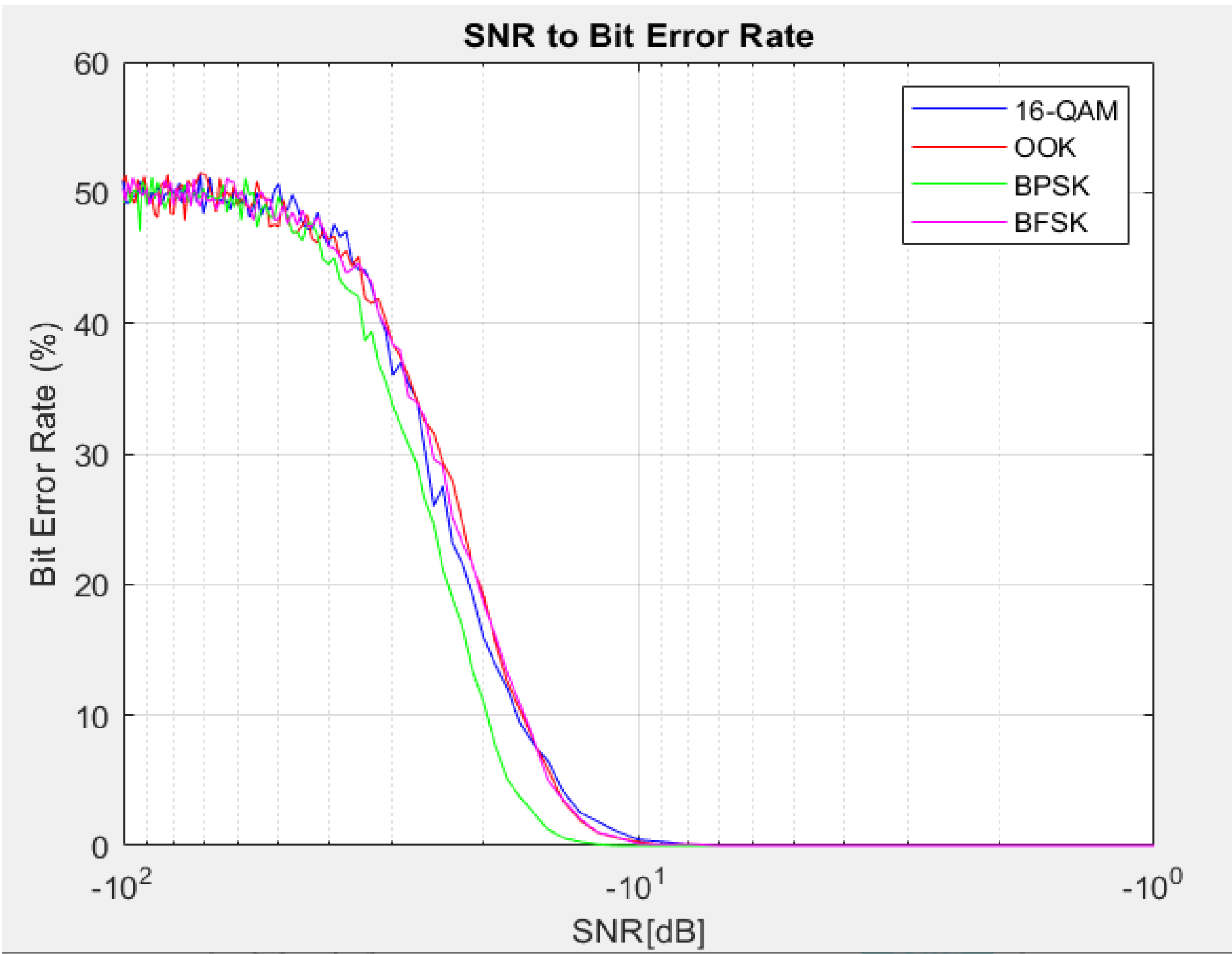


Figure 17: Comparison of BER from SNR of -100 to 0

BPSK performs the best in terms of bit error rate (BER), having a lower BER at than the rest of the modulation techniques for SNR below -10 as reflected in Figure 17. OOK, BFSK and 16-QAM performs similarly, equalizing with BPSK at around -10 SNR. This is due to the robustness of BPSK, having the ‘1’ and ‘0’ separated by 180 degree phase shift of the carrier. Other than BER, there are some other factors that contribute to the usefulness of the modulation technique, as mentioned in Table 1.

	Advantages	Disadvantages
OOK	• Low complexity • Low cost	• Low bit rate • Susceptible to noise
BFSK	• Less susceptible to noise than OOK • Less susceptible to errors than OOK	• High bandwidth requirement
BPSK	• Robust • Low bit rate error	• Relatively high bandwidth requirement
16-QAM	• Best spectral efficiency	• Complex • Susceptible to noise

Table 1: Comparison of different modulation techniques

ASK and OOK are commonly used in short-range wireless applications due to their lower susceptibility to noise. The low complexity and low cost also makes the products more commercially affordable.

BFSK is commonly used for telemetry, weather balloon radiosondes, caller ID, garage door openers, and low frequency radio transmission. Those applications have further distance, hence noise and error rate will be too high for OOK. Hence, BFSK can be used at the cost of higher bandwidth requirements.

BPSK is the most robust and has the lowest BER out of the 4 modulation techniques. Hence, it is used for long range communication systems such as satellite communication. This comes at the trade of lower bandwidth efficiency, although better than BFSK.

16-QAM is commonly used for digital cable television and cable modem applications. 16-QAM can modulate 16 bits at the same time, giving it high spectral efficiency. Spectral efficiency is important in those applications as many people are sharing the channel.

Phase 3: Basic Error Control Coding to Improve Performance

1. Introduction

1.1. Cyclic Code(12,4)

A cyclic code is a linear block code, with the circular shifts of each codeword that gives another word that belongs to the code. They have algebraic properties that are convenient for efficient error detection and correction.

Advantages:

- The well defined mathematical structure allows for efficient decoding schemes.
- The methods used for error detection are simple and easy to implement.
- These methods do not need look-up table decoding.
- It is possible to detect error burst using cyclic codes.

Disadvantages:

- Even though error detection is simpler, error correction is slightly more complicated due to the complexity of the combination logic circuit used.

1.2. Hamming Code(7,4)

Hamming code is a linear code that is useful for error detection up to two immediate bit errors. It is capable of single-bit errors.

In Hamming code, the source encodes the 4 bit message by adding 3 redundant bits in the message, resulting in a 7 bit encoded message. These redundant bits are mostly inserted and generated at certain positions in the message to accomplish error detection and correction process.

Advantages:

- Hamming code is effective on communication networks where the whole messages are known for the single-bit errors.

- The low complexity of hamming codes makes it best suitable for use in computer memory and single-error correction.

Disadvantages:

- Hamming code can only correct single bit errors. Multibit errors might result in flipping of a correctly received bit, increasing BER.

Hamming codes are used in many applications where single bit transmission errors are common, such as DRAM memory chips and satellite communication hardware.

1.3. Turbo Code(4 iterations)

Turbo codes is a family of error correction codes that is capable of approaching the maximum channel capacity or the Shannon limit. Turbo encoders use two different encoders running in parallel with an interleaver to generate the parity bits to be transmitted over the channel. The classic turbo code encoder uses an iterative approach which consists of two Recursive Systematic Convolutional encoders. The resulting parity codes will then be concatenated with the message for transmission.

The message bits will be sent in to the first encoder and a randomized version will be sent to the second encoder also called interleaving where the bit positions of the message is shuffled. This will add some randomness to the parity bits which is key in approaching capacity. Unlike all the other codes used in this project which uses a hard-decision approach to convert to bit values the deuses a soft approach which outputs a range of probabilities that the bit could be.

Advantages:

- Performance extremely close to the Shannon limit
- Performs well with noisy environments

Disadvantages:

- More complex to implement correctly

1.4. Reed-Solomon Code(8,4)

Reed-Solomon code are block-based error correcting codes.The encoder divides the message stream into blocks of digital data and adds extra redundant bits to each block depending on the current input.This means that the encoder takes *k* data symbols of *s* bits each and adds parity symbols to make an *n* symbol codeword.There are n-k parity symbols of s bits .The decoder processes each block which to correct errors and recover the original data.

Advantages:

- Decrease power usage which improves the performance of error correction in Reed-Solomon.Coding gain is high.
- Most widely used to correct burst error.

Disadvantages:

- Reed-Solomon Codes do not perform considerably well in BPSK modulation.

2. Analysis of the different codes performance

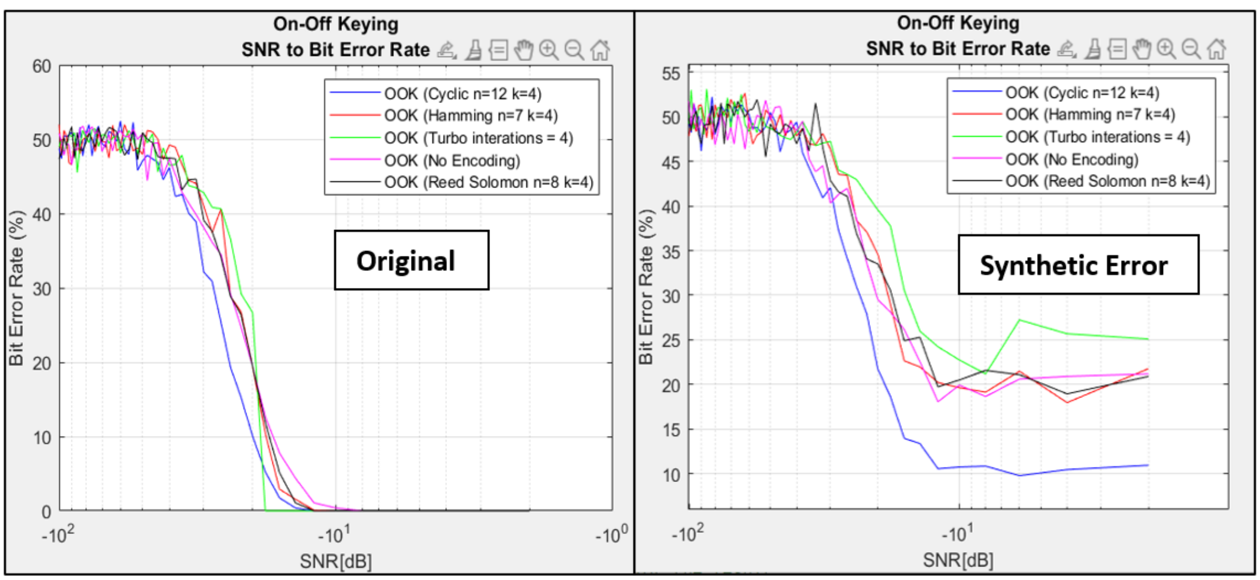


Figure 18: Performance of different encoding schemes

OOK will be used for channel coding comparisons between the 4 schemes for consistency. For each coding scheme, the code length and message length is selected to optimize the BER. The resulting bit length after channel encoding are 3072, 1792, 3084 and 2048 for cyclic, hamming, turbo and reed-solomon encoding schemes respectively. The bit length is an important factor as it will affect the speed of message transfer.

As the performance for each channel encoding scheme is similar, synthetic error has been introduced to the demodulated signal before decoding. 20% of the total bits are error induced. Cyclic encoding outperformance the other encoding schemes in both the original and error induced signals, as verified by figure 18 above. With proper selection of the number of iterations, turbo will result in lower BER for lower SNR values.

Conclusion

In this project we started with generating message data bits for transmission over an additive white gaussian noise channel without modulation to see the effects of noise on the message signal. Next we modulated the signals using four different modulation methods.

Each modulation has its merits while Binary phase shift keying demonstrates the best bit error performance but it modulates 1 bit at a time as compared to 16-QAM which have similar bit error performance but modulates 4 bits at once which achieve better spectral efficiency compared to all the other modulation methods used.

Finally we use four different error correction codes to improve the bit error performance of the various modulation schemes which showed improvement however the improvements were marginally better.

That being said there are many parameters to consider and these results are not conclusive as our testing parameters might not reflect well with the typical environment that this signals.

In conclusion, choice of modulation methods and error correction codes depends heavily on knowledge of the environment to achieve the best performance and each modulation and error correction codes have their own merits and using just the bit error performance is not enough to conclusive rule a modulation or error code better than the other.