EEN1043 Wireless/Mobile Communications

Assessment 1

Kyle Sheehy

21404424

BASK

Bask is the simplest of the three modulation schemes. It entails turning the carrier signal either on or off for 1 and 0 respectively. In actuality, the amplitude should be ± A, where A is the amplitude of the carrier signal, but implementing this scheme results in more of a phase shift scheme rather than amplitude. Because of this, utilising unipolar BASK is a more effective way of demonstrating the modulation. This could result in a weaker decision boundary however as the spread of bits is tighter due to the range being halved: [-A, A] = 2*[0, A].

For any given bit steam, using BASK has a simple methodology. Once the time per bit has been declared (T_{BASK}), the reference signal must be declared ($Reference_{BASK}$):

$$Reference_{BASK} = \cos(2\pi F_C T_{BASK})$$

Where F_C is the carrier frequency. For each bit within the stream, if the bit is 1 then A_1, or 5, is multiplied by the carrier signal. If the bit is 0, then 0 is multiplied by the carrier signal. These 2 rules results in the for loop declared in lines 50-59. This modulated signal now gets added to the bask signal array, bask_signal, and the subsequent array is changed to a numpy array for processing.

The linear Eb/N0 is calculated using the following formula:

$$EbN0_{LINEAR} = 10^{\frac{EbN0}{10}}$$

This has to be converted to a linear scale because in order to calculate bit error rate (BER), the scale must be linear not logarithmic. Decibels are in logarithmic form and subsequent calculations wouldn't be correct if this conversion did not take place.

Energy per bit is calculated using the following (line 42):

$$Eb_{BASK} = A1^2 \times \frac{t\_bit_{BASK}}{4}$$

The threshold for unipolar BASK is $A_1/4$. This is obtained through integration of both R0 and R1 which is the correlation of the output when bits 0 and 1 are transmitted respectively. As R0 is 0 because of A0 being 0 for bit 0, the threshold simplifies to:

$$threshold_{optimal} = \frac{R0 + R1}{2}$$

$$threshold_{simplified} = \frac{R1}{2}$$

We know what R1 is:

$$R1 = \frac{A1 \times t\_bit_{BASK}}{2}$$

Therefore:

$$threshold_{simplified} = \frac{A1}{4}$$

Noise power (N0) must be calculated in order to obtain the variance and standard deviation of the noise. It relies on the Eb/N0 value being in a linear scale. The formula for N0 used in this code is given below:

$$N0 = \frac{Eb_{BASK}}{EbN0_{LINEAR}}$$

Variance is an important factor to have calculated correctly as this allows for better noise application. This is important because it allows noise be added that is representative of the channel rather than regular Gaussian White Noise. Variance is given by:

$$\sigma_{NOISE}^2 = N0 \times \frac{fs}{2}$$

Where fs is the sampling frequency of 100Hz. From Variance, we can calculate the standard deviation by taking the square root, given by the equation below:

$$\sigma_{NOSIE} = \sqrt{\sigma_{NOISE}^2}$$

To simulate channel characteristics, adding Gaussian White Noise is imperative. This can be done by generating a series of random Gaussian values the length of the signal and multiplying it by the standard deviation derived above. This allows the variance of the noise to dictate the channel conditions by subsequently adding the transmitted signal and this noise together. This is in essence, the received signal. Ideally, there would also be a H factor convoluted with the transmitted signal for actual channel simulation.

The BASK demodulation employs a correlation receiver approach. For each bit period, the received signal segment is correlated with the reference carrier signal by computing their dot product and normalizing by the sampling frequency. This correlation operation effectively matches the received signal against the expected bit '1' waveform, maximizing the signal-to-noise ratio at the decision point. The resulting correlation value is then compared against the optimal threshold to determine whether a '0' or '1' was transmitted. This approach is mathematically equivalent to a matched filter and provides optimal performance in additive white Gaussian noise channels.

$$rx_{SYMBOL} = \sum \frac{rx_{SEGMENT} \times Reference_{BASK}}{fs}$$

If the result from this calculation is greater than the threshold then the array stores this as a '1' and the opposite is true for '0'.

The constellation points for the first 20 symbols are saved to an array. I have also done this for the first 5000 symbols as this gave a more accurate representation of the overall transmission simulation. The BASK tx and rx symbols have had jitter added to them on the y-axis to increase interpretation of the graph. There is no quadrature component in the BASK and I have not implemented one in the code.

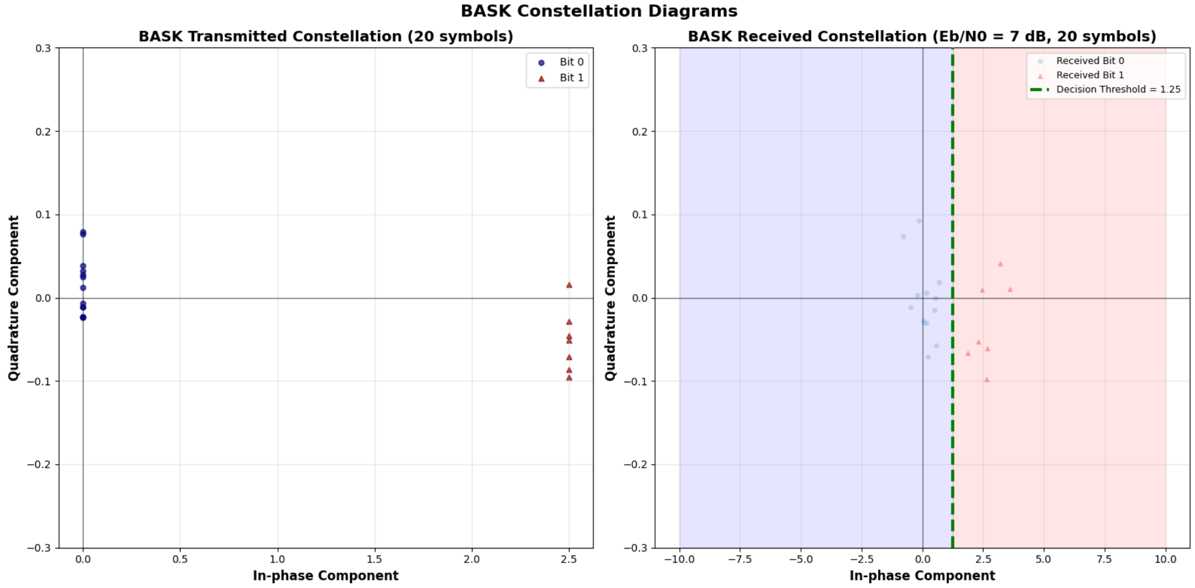

Figure 1 – BASK Transmission and Received Constellation Diagrams for 20 bits.

The jitter is clearly highlighted here, showcasing the 20 bits more easily

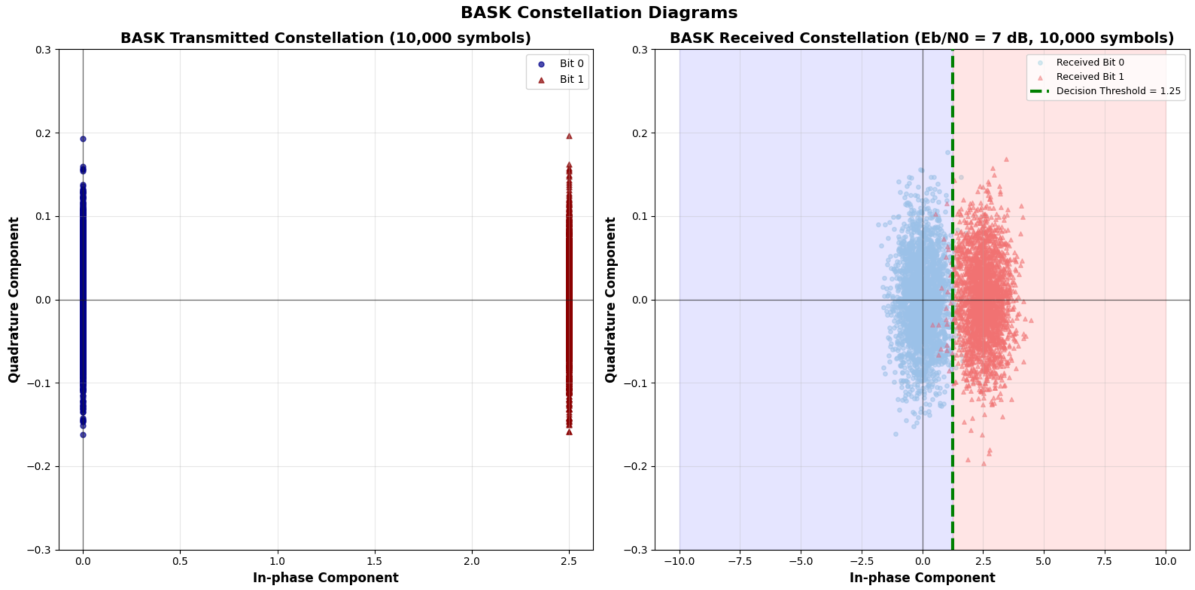

Figure 2 – BASK Transmission and Received Constellation Diagrams for 5000 bits.

The jitter makes the bits much easier to interpret intuitively here.

BASK constellation diagrams illustrate unipolar amplitude shift keying where bit '1' maps to carrier amplitude A_1=5V and bit '0' to zero amplitude. The correlation receiver produces in-phase values around 0 (bit '0') and 2.5 (bit '1'), with the optimal decision threshold at 1.25. Figure 1 shows 20 symbols for clear individual symbol visualisation, while Figure 2 demonstrates 5000 symbols revealing the statistical Gaussian distribution caused by AWGN at Eb/N0 = 7 dB. Artificial y-axis jitter aids visual interpretation without affecting signal processing, as BASK operates purely on the in-phase axis. The decision regions and threshold line illustrate how the correlation receiver achieves optimal bit detection in the presence of noise.

Bit Error Rate (BER) quantifies communication system performance as the ratio of incorrectly received bits to total transmitted bits. The inverse relationship between BER and Eb/N0 reflects the fundamental principle that higher signal energy relative to noise reduces error probability.

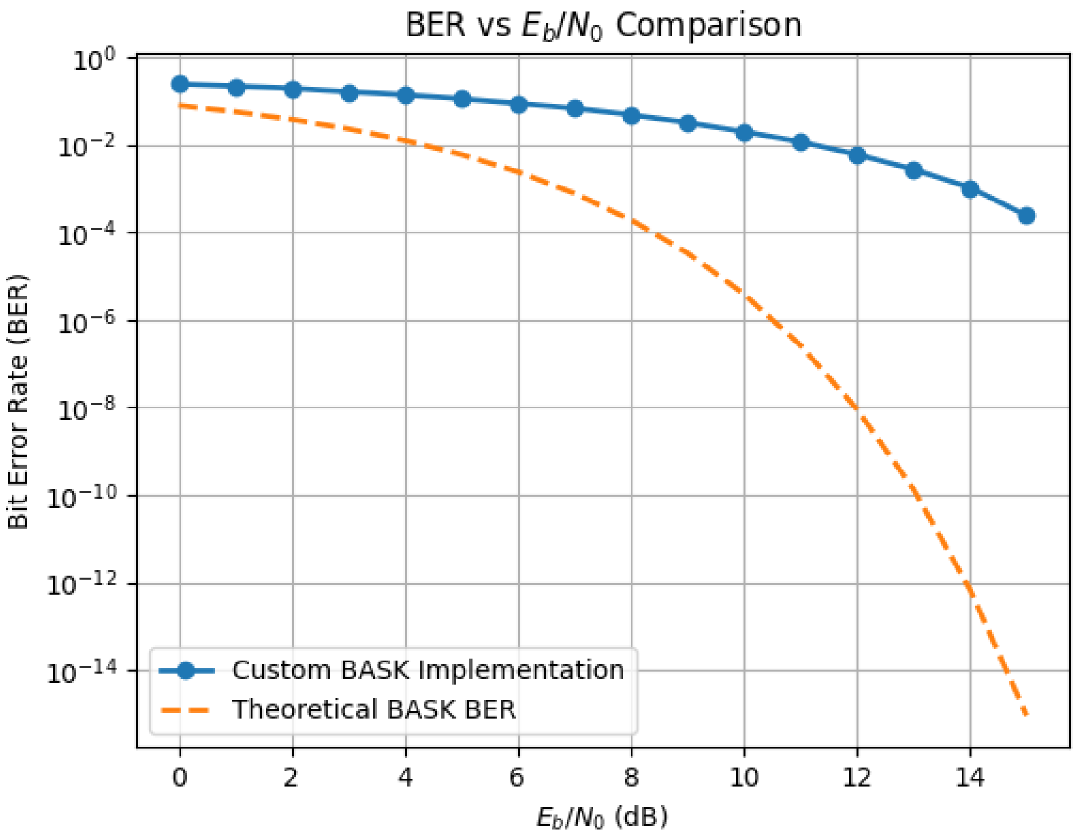

Figure 3 – BER for BASK modulation scheme showcasing an increase in Eb/N0 leads to a reduction in BER. There is a large difference between my implementation of BASK and the theoretical version which is calculated using: BER = 0.5 × erfc(√(Eb/2N0)) for unipolar signalling. Both curves exhibit the characteristics of a waterfall curve.

The simulated curve shows higher BER than theoretical due to practical implementation factors including finite precision arithmetic, discrete-time sampling effects, and the specific correlation receiver implementation. The theoretical curve assumes ideal continuous-time processing. The 3 dB performance penalty of unipolar BASK compared to bipolar signalling is evident in the theoretical curve, which uses the erfc(√(Eb/2N0)) formula rather than the standard erfc(√(Eb/N0)) for bipolar systems.

At higher Eb/N0 values (>10 dB), the curves begin to converge, suggesting that noise becomes the dominant factor rather than implementation imperfections. The simulation achieves BER ≈ 10^{-4} at 15 dB Eb/N0.

QPSK

QPSK fundamentally differs from BASK by modulating signal phase rather than amplitude. While BASK transmits 1 bit per symbol using amplitude variation (A_1 or 0), QPSK achieves 2 bits per symbol through four distinct phase states, doubling the spectral efficiency.

The QPSK signal is expressed as $s(t) = A \cdot \cos(2\pi fc \cdot t + \phi_K)$ where ϕ_K represents one of four possible phases. This creates a two-dimensional signal space using in-phase (I) and quadrature (Q) components, with constellation points at $(I,Q) = (A \cdot \cos(\phi), A \cdot \sin(\phi))$.

Gray coding maps bit pairs to phases to minimize bit errors: 00→45°, 01→135°, 11→225°, 10→315°. This ensures adjacent constellation points differ by only one bit, reducing error propagation when noise causes symbol misdetection. Each symbol maintains constant energy Es while delivering twice the information capacity of BASK.

As QPSK spans both the x and y axes, the corresponding signals are split to be dealt with separately – I-Carrier and Q-Carrier, where I-Carrier is along the x-axis (In-Phase) and Q-Carrier along the y-axis (Quadrature). The implementation converts the bit stream into QPSK symbols using binary-to-decimal conversion (line 122). This maps 00→0, 01→1, 10→2, 11→3, enabling efficient symbol indexing for constellation lookup.

Two orthogonal reference carriers are generated:

$$I_{CARRIER} = \cos(2\pi fc\, t)$$

$$Q_{CARRIER} = -\sin(2\pi fc\, t)$$

In using sine for the $Q_{CARRIER}$, quadrature is ensured by implementing a 90° phase shift. The negative sign ensures proper quadrature relationship for correlation detection.

This is shown in figure 4, where 5000 bits have been transmitted to allow intuitive understanding of the transmission and received constellations.

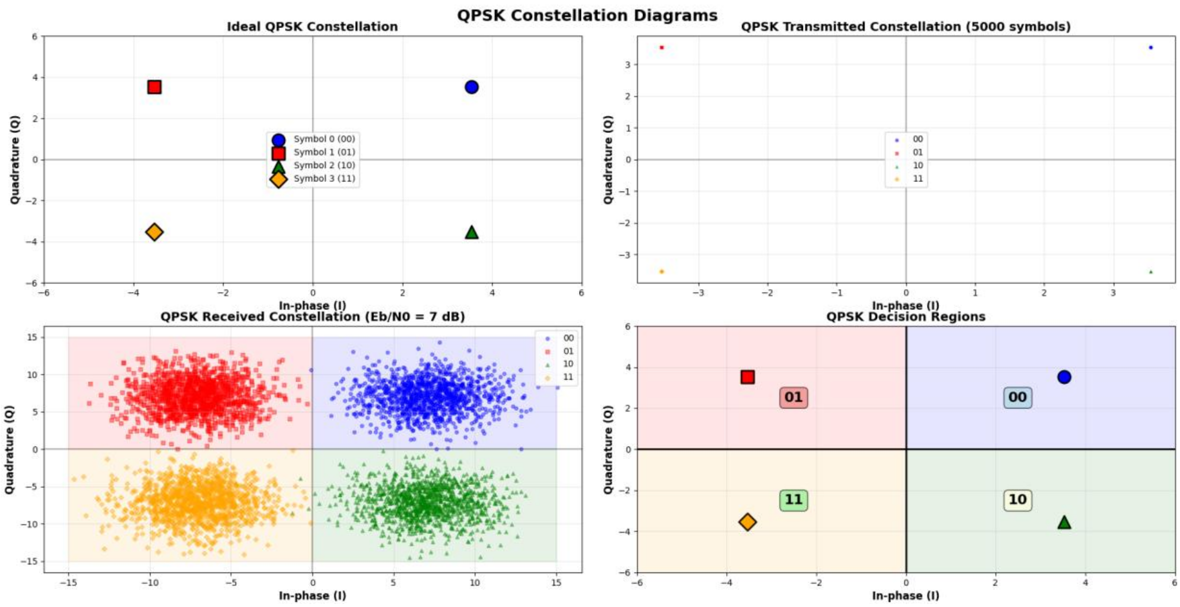

Figure 4 – Transmitted and Received Constellation Diagrams for QPSK for 5000 symbols. Transmission diagrams are on the top row and Received on the bottom. The bottom right depicts

the decision regions for each transmitted symbol. The bottom left shows the actually received symbols and where they lie within the decision regions, for Eb/N0 = 7dB.

The received constellation (bottom left, figure 4) shows Gaussian noise clouds around each ideal point, with the decision boundaries (I=0, Q=0 axes) effectively separating the quadrants for symbol detection.

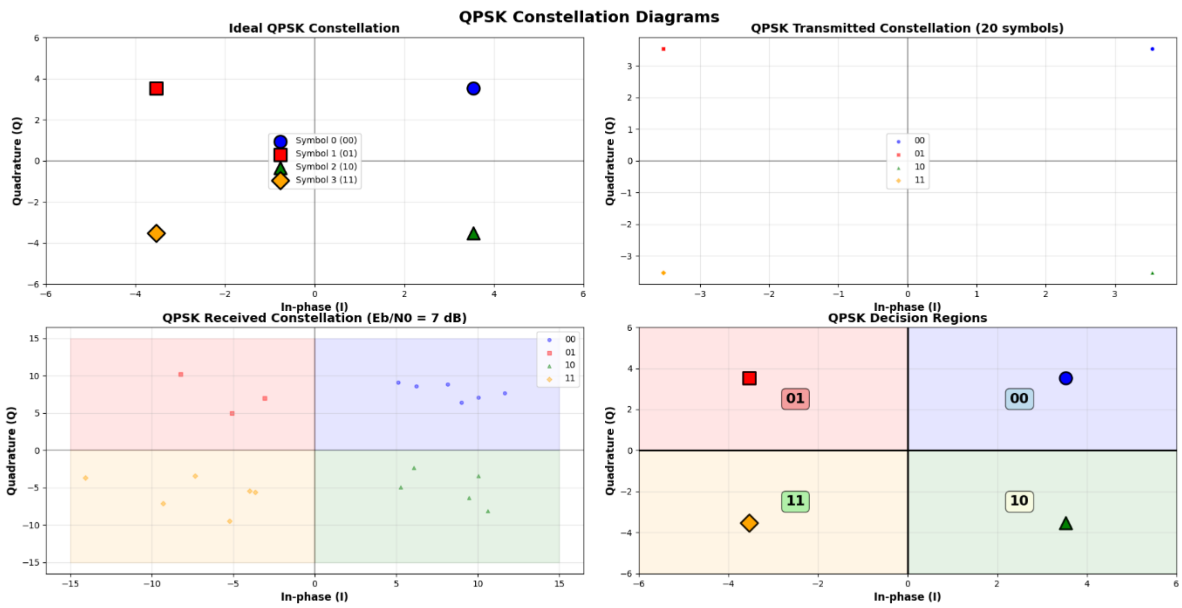

Figure 5 – Transmitted and Received Constellation Diagrams for QPSK for 20 symbols

QPSK symbol duration (T_{SYMBOL} = 2 seconds) is twice the bit duration (compared to BASK), maintaining the same data rate as BASK while transmitting 2 bits per symbol. This results in 200 samples per symbol at fs = 100Hz, providing accurate resolution for correlation processing.

Each QPSK symbol generates sperate I and Q waveforms

$$I_{WAVEFORM} = I_{AMPLITUDE} \times \cos(2\pi fc\ t)$$

$$Q_{WAVEFORM} = Q_{AMPLITUDE} \times (-\sin(2\pi fc\ t))$$

The transmitted signal combines both of these waveforms

$$s(t) = I_{WAVEFORM} + Q_{WAVEFORM}$$

S(t) is called $\text{symbol}_{WAVEFORM}$ in the code (line 171).

This signal is then added to the array $QPSK_{SIGNAL}$ which in turn, is converted to a numpy array.

Demodulation is performed using dual correlation

$$I_{CORRELATION} = \sum (I_{CARRIER} \times received_{SEGMENT}) \times \frac{2}{fs}$$

$$Q_{CORRELATION} = \sum (Q_{CARRIER} \times received_{SEGMENT}) \times \frac{2}{fs}$$

The factor 2/fs compensates for the discrete-time approximation of continuous correlation, ensuring proper amplitude scaling.

Euclidian distance is used in the symbol decision. The receiver selects the constellation point with minimum distance, implementing optimal maximum likelihood for AWGN channels.

After symbol detection, this process reverses the mapping

$$Bit\ 1 = symbol\ //\ 2\ (most\ significant\ bit)$$

$$Bit\ 2 = symbol\ \%\ 2\ (least\ significant\ bit)$$

This converts decimal symbol values back to binary bit pairs for BER calculation.

The conversion of Eb/N0 to linear scale is the same as BASK. The energy per bit is calculated by dividing the energy per symbol by 2. Both formulae are given below, as well as calculation of N0, noise variance, and noise standard deviation:

$$Es_{QPSK} = \frac{A^2 \times t_{SYMBOL}}{2}$$

$$Eb_{QPSK} = \frac{Es_{QPSK}}{2}$$

$$N0 = \frac{Eb_{QPSK}}{EbN0_{LINEAR}}$$

$$\sigma_{NOISE}^2 = N0 \times \frac{fs}{2}$$

$$\sigma_{NOISE} = \sqrt{\sigma_{NOISE}^2}$$

BER follows the same principles here as it did in BASK – both schemes are implementing a correlation-based detection system.

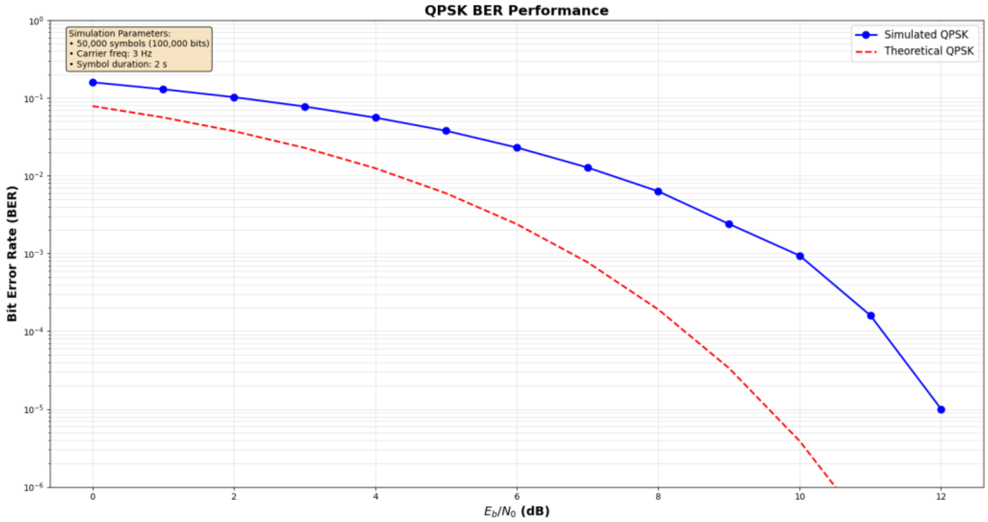

Figure 6 – BER vs Eb/N0 for QPSK. Again, an increase in Eb/N0 demonstrates a decrease in BER. Both the implemented QPSK and the theoretical QPSK are waterfall graphs. Above 10 dB Eb/N0, the curves converge as AWGN becomes the dominant limitation rather than implementation effects, with both achieving BER ≈ 10^{-5} at 12 dB.

8PSK

8PSK follows the same ideology as QPSK but with 8 different phases utilised instead of 4. This allows for 3 bits to be transmitted per symbol instead of 2. The 3-second symbol duration maintains the same bit rate as BASK while achieving 3 times the spectral efficiency through phase diversity rather than increased symbol rate. The setup of this scheme is the same as for QPSK.

The 8PSK signal is expressed as $s(t) = A \cdot \cos(2\pi f_c \cdot t + \phi_K)$ where ϕ_K represents one of eight possible phases. This creates a two-dimensional signal space using in-phase (I) and quadrature (Q) components, with constellation points at $(I,Q) = (A \cdot \cos(\phi), A \cdot \sin(\phi))$.

I did not implement Gray coding for 8PSK, the bit mapping is given here: 000→22.5°, 001→67.5°, 010→112.5°, 011→157.5°, 100→202.5°, 101→247.5°, 110→292.5°, 111→337.5°. Each symbol maintains constant energy. The absence of Gray coding means adjacent constellation points may differ by up to 3 bits, potentially tripling bit errors when noise causes symbol misdetection to neighbouring points.

The x and y axes are utilised in 8PSK the same as for QPSK, resulting in the I and Q carrier being split again. The 8PSK symbols also utilise binary-to-decimal conversion (line 262) for mapping the symbol indexes for constellation hookup.

The same orthogonal carriers are created, and the constellation diagram is depicted below

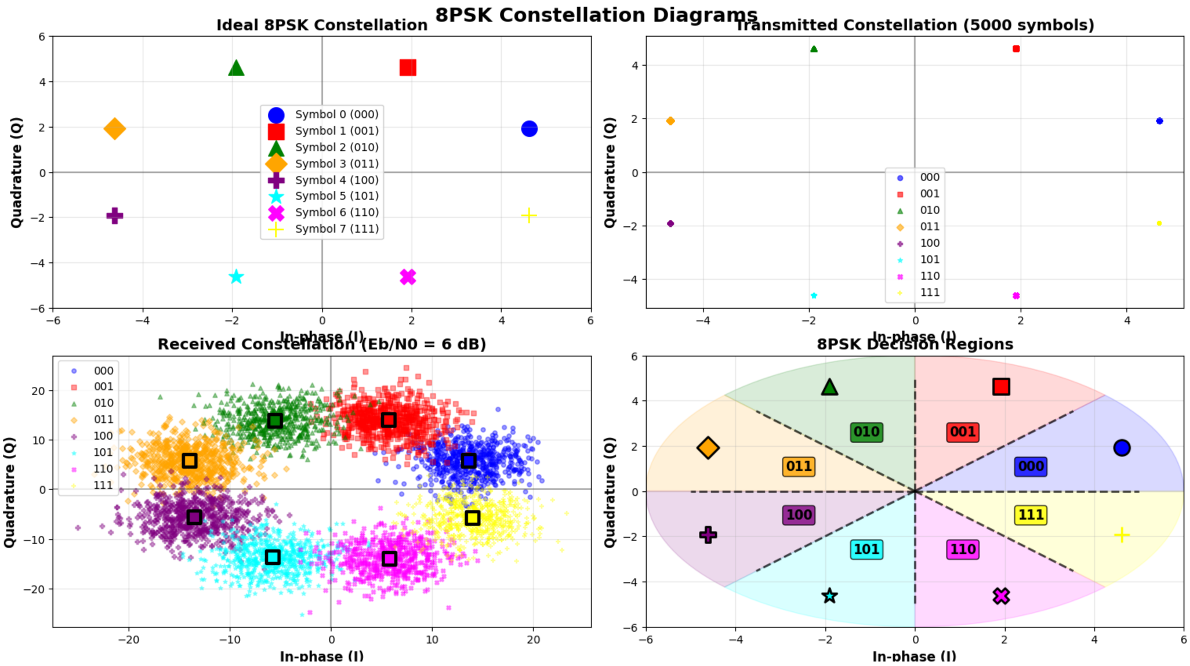

Figure 7 – 8PSK Transmitted and Received Constellation Diagrams for 5000 symbols. Transmitted diagrams are on the top row and Received on the bottom. The bottom right depicts the decision regions for each transmitted symbol. The bottom left shows the actually received symbols and where they lie within the decision regions, for Eb/N0 = 7dB.

The received constellation on the bottom left also depicts the ideal symbol received with a black box. Euclidian distance to this black box is what the correlation scheme uses to determine what symbol has been received. These ideal symbols are more clearly represented in the bottom right constellation diagram.

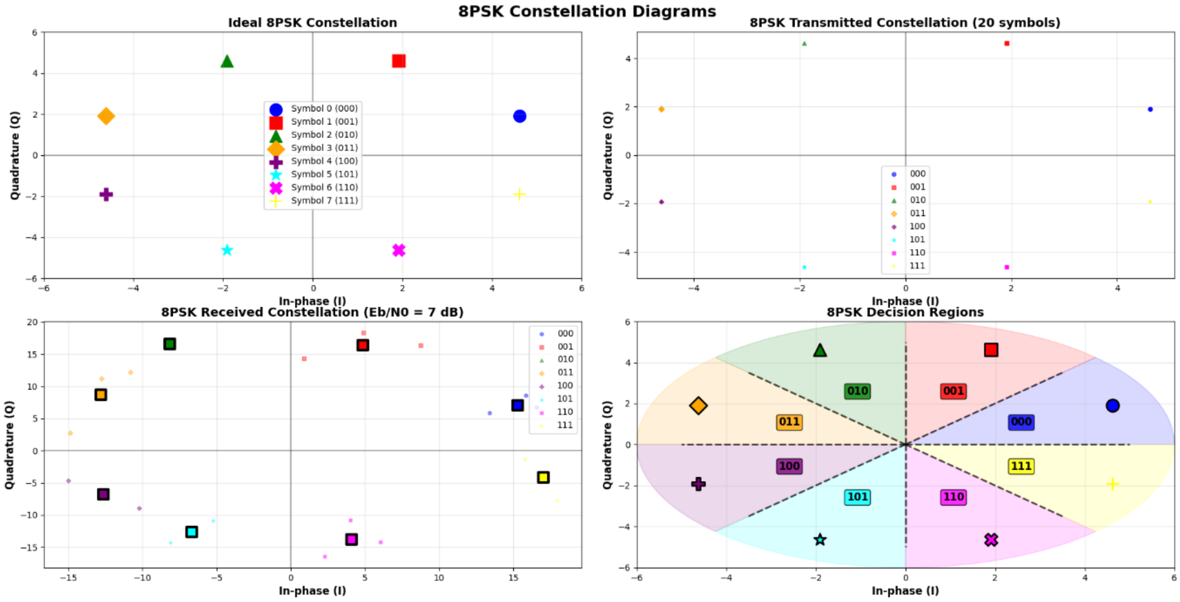

Figure 8 – 8PSK Transmitted and Received Constellation Diagram for 20 symbols.

Comparing figures 7 and 8, it is much easier to see the decision regions in figure 7 and the cluster of bits colours each sections accordingly. The 8 decision regions form 45° pie-slice sectors, with boundaries at angles 0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°. Each region encompasses a 22.5° angular span around its constellation point.

8PSK symbol duration (T_{SYMBOL} = 3 seconds) resulting in 300 samples per symbol at fs = 100Hz, providing accurate resolution for correlation processing.

Each 8PSK symbol generates sperate I and Q waveforms, the same as QPSK above. There is no need to redo each of these equations so only ones that are changed will be detailed. Dual correlation processing computes I_corr and Q_corr values, then applies Euclidian distance across all 8 constellation points

$$distance = (I_{CORR} - I_{CONSTELLATION})^2 + (Q_{CORR} - Q_{CONSTELLATION})^2$$

The receiver selects the constellation point with minimum distance, implementing optimal maximum likelihood for AWGN channels. AWGN noise variance (N0×fs/2) accounts for power distribution across both I and Q channels, ensuring proper SNR representation in the two-dimensional signal space. The maximum likelihood detector evaluates all 8 constellation points per symbol, requiring 8 distance calculations compared to QPSK's 4, increasing computational complexity by 100%.

After symbol detection, this process reverses the mapping

$$Bit\ 1 = (symbol \gg 2)\ \&\ 1\ (most\ significant\ bit)$$

$$Bit\ 2 = (symbol \gg 1)\ \&\ 1\ (middle\ bit)$$

$$Bit\ 3 = (symbol)\ \&\ 1\ (least\ significant\ bit)$$

This converts decimal symbol values back to binary bit pairs for BER calculation. The core of this lies in the bit extraction process using bitwise operations. Since 8PSK encodes 3 bits per symbol, each symbol value (0-7) must be converted back into its constituent 3 bits. Right shift operations (>>) combined with bitwise AND operations (& 1) extract individual bits. (Lines 375 – 383)

The extracted bits are placed into an array at their correct positions. Per symbol, the 3 bits are stored at indices 3*i, 3*i+1, and 3*i+2, which effectively reconstructs the original bit stream from the symbol stream. This reconstructed bit sequence can then be compared to the original transmission stream to calculate the BER. The bit-to-symbol conversion using weighted binary:

$$symbol = bit[0] \times 2^2 + bit[1] \times 2 + bit[2] \times 1$$

The conversion of Eb/N0 to linear scale is the same as BASK and QPSK. The energy per bit is calculated by dividing the energy per symbol by 3. The same formulae are given below, with the only changes being to nomenclature:

$$Es_{8PSK} = \frac{A^2 \times t_{SYMBOL}}{2}$$

$$Eb_{8PSK} = \frac{Es_{8PSK}}{3}$$

$$N0 = \frac{Eb_{8PSK}}{EbN0_{LINEAR}}$$

$$\sigma^2_{NOISE} = N0 \times \frac{fs}{2}$$

$$\sigma_{NOISE} = \sqrt{\sigma^2_{NOISE}}$$

The 8PSK BER is given below:

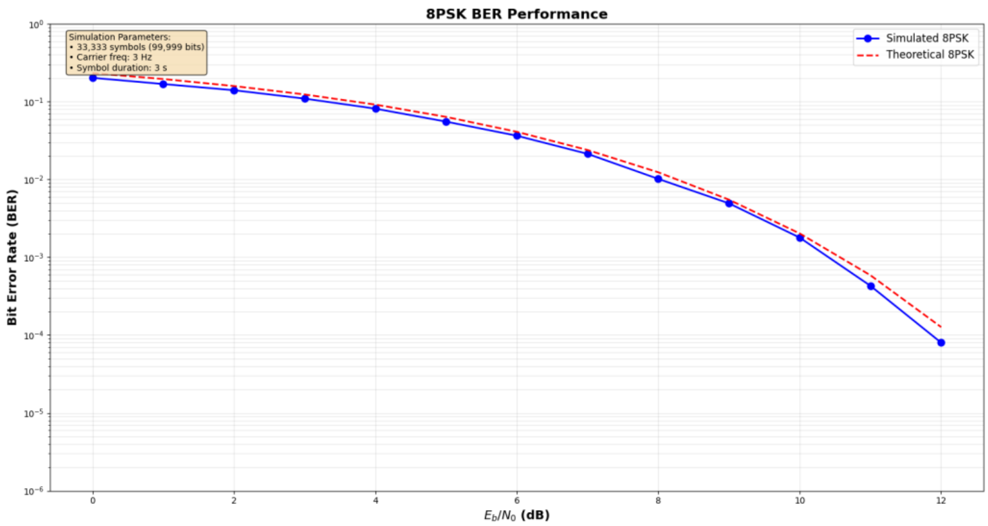

Figure 9 – 8PSK BER vs Eb/N0. Both the implemented 8PSK and theoretical 8PSK have a decreasing BER for increasing Eb/N0.

It is important to note that the simulated 8PSK outperforms the theoretical one. This highlights how the actual channel conditions are not represented in the code and simply adding noise to the signal does not accurately depict real life scenarios, which the theoretical model is based on. Should a H factor have been incorporated, it is almost guaranteed that the simulated version would have performed worse (higher BER) than the theoretical.

Comparing BASK vs QPSK vs 8PSK

We know from above that 8PSK has the best transmission rate, but it has the worst BER vs Eb/N0. These have been graphed together below. This comparison shows both the simulated BER vs Eb/N0 values and the theoretical ones. Arguments can be made that some schemes are better because they are more closely representative of the theoretical values, but this simulation does not take into account the actual channel conditions, which is represented by H.

$$y(t) = x(t) \otimes h(t) + n(t)$$

In time domain, the transmission signal and the channel conditions (h) must be convolved but in the frequency domain, convolution is regular multiplication:

$$Y(f) = X(f) \times H(f) + N(f)$$

If the channel conditions had been implemented in these simulations, then there would be a bigger difference between the simulated BER vs Eb/N0 values and the theoretical ones.

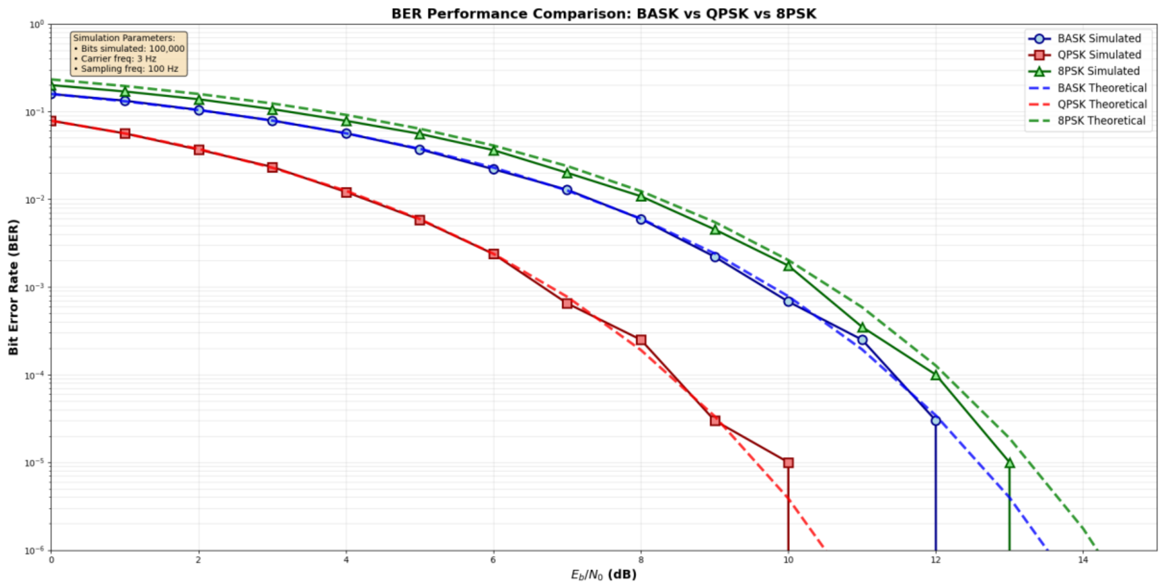

Figure 10 – All BER vs Eb/N0 values

From figure 10, we can conclude that BASK shows superior performance to 8PSK at low Eb/N0 due to unipolar signalling's robustness.

Figure 10 demonstrates the fundamental trade off between spectral efficiency and error performance. QPSK achieves optimal balance, maintaining BASK-equivalent BER while doubling the spectral efficiency. 8PSK suffers approximately 4-5 dB penalty at BER = 10^{-4} compared to QPSK due to reduced minimum distance between constellation points, despite achieving 3x spectral efficiency (compared to BASK).

For QPSK, the minimum distance is given by

$$distance = 2 \times \sqrt{2} \times A$$

For 8PSK, the distance is given by

$$distance = 2 \times A \times \sin\left(\frac{\pi}{8}\right) \approx 0$$

This explains the approximate 4.2dB theoretical penalty for 8PSK.

BASK provides robust performance with minimal complexity but poor spectral efficiency. QPSK represents the optimal compromise for most applications. 8PSK maximises spectral efficiency at the cost of increased power requirements and implementation complexity.

There are multiple implementation complexities that should be considered when choosing a scheme. BASK implements single correlation with threshold detection. It is the simplest and requires minimal processing. QPSK has dual I/Q correlation with 4-point ML detection. It is more advanced and subsequently requires more processing. 8PSK also employs dual I/Q correlation but with 8-point ML detection. This requires 100% more processing power than that of QPSK.

The constellations of each scheme influence their performance. BASK's one-dimensional signalling suffers from amplitude noise vulnerability. QPSK's square constellation provides optimal minimum distance utilisation. 8PSK's circular constellation reduces minimum distance compared to QPSK, which explains its performance degradation.

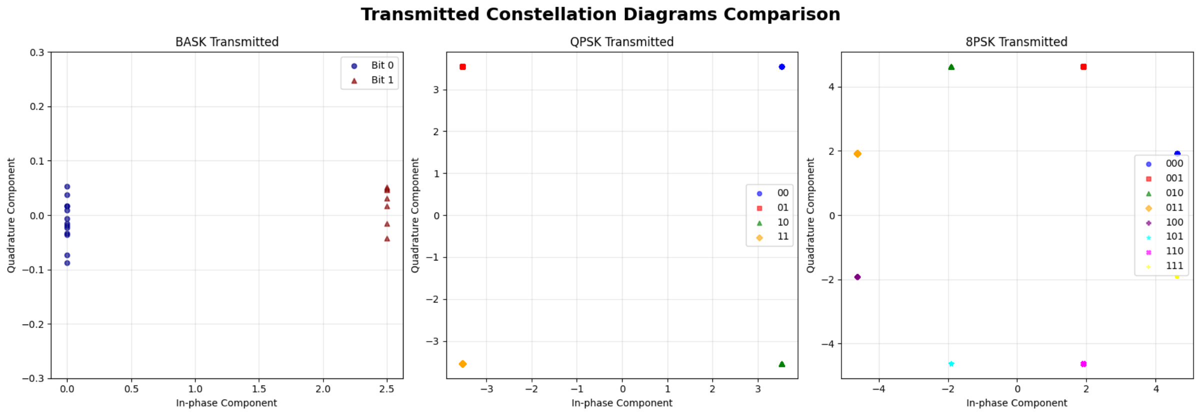

Figure 11 – Transmission constellation diagrams for each of the 3 schemes, again jitter has been added to the BASK scheme for visualisation purposes.

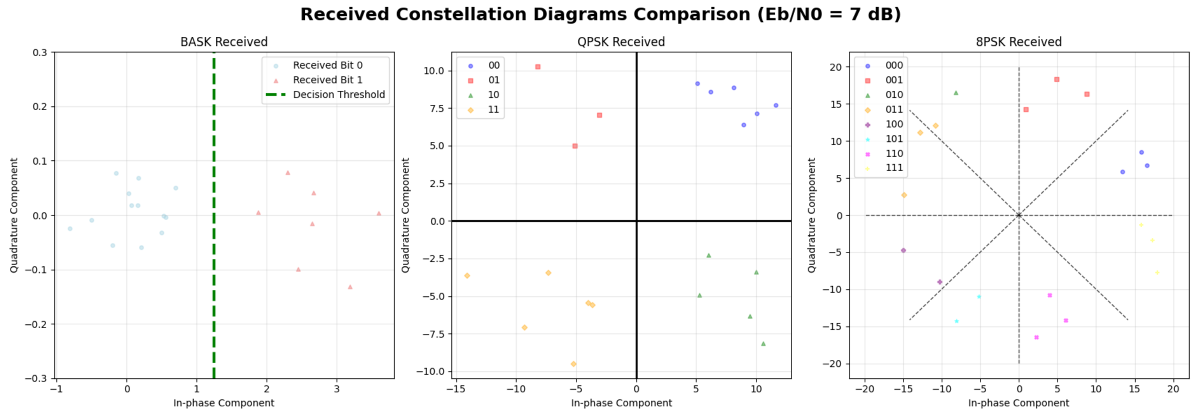

Figure 12 – Received constellation diagrams for each of the 3 schemes for 20 symbols. Each constellation diagram has the decision boundaries drawn in for clarity.

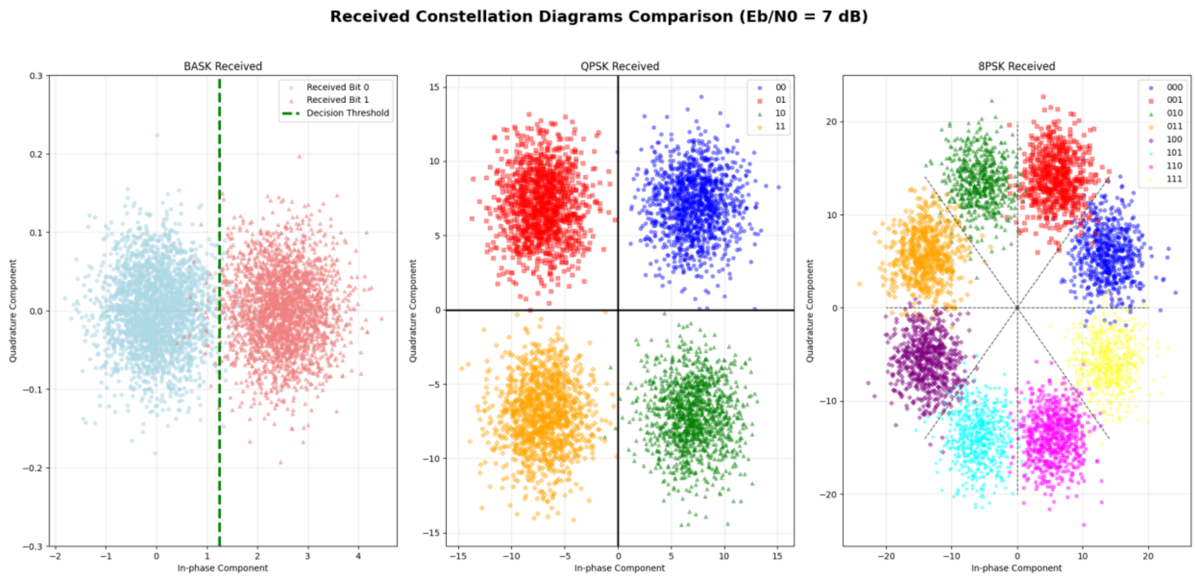

Figure 13 – Received constellation diagrams for each of the 3 schemes for 5000 symbols. Each constellation diagram has the decision boundaries drawn in for clarity.