EE6143 – Advanced Topics in Communications

Carrier and Clock Synchronization for QPSK

Project Report

Student Name : P. Kavinesh Kumar

Roll No. : EE25M021

---

1 Objective

The goal was to implement the receiver tasks specified in the project:

1. Generate QPSK frames with a 200-symbol antipodal preamble
2. Apply clock offset, carrier offset and noise
3. Estimate and correct coarse carrier offset using squared-preamble DFT comb
4. Estimate timing using the preamble
5. Recover the clock using an interpolating Early–Late loop
6. Estimate residual frequency/phase and correct it
7. Evaluate SER vs E_b/N_0 and compare with theory

The report focuses on how each step is implemented in MATLAB and why.

2 Frame Generation and Preamble

Random data symbols are first generated and mapped to QPSK constellation points using Gray mapping. The mapping used is

$$00 \to (x > 0, y > 0), \qquad 01 \to (x < 0, y > 0),$$
$$11 \to (x < 0, y < 0), \qquad 10 \to (x > 0, y < 0).$$

This ordering is intentional: only one bit changes between any two adjacent constellation points. If the alternative mapping

$$10 \to (x < 0, y < 0), \qquad 11 \to (x > 0, y < 0)$$

were used, many symbol errors would flip two bits. With Gray mapping, most symbol errors translate into single–bit errors, improving performance without changing the modulation or signal power.

The transmitted frame has the structure

$$[\text{ 200-symbol preamble }] \ + \ [\text{ 2000-symbol data block }].$$

The same structure is repeated across frames during the simulation. The preamble is an alternating antipodal sequence,

$$s[n] = \frac{1}{\sqrt{2}}(1 + j)\,(-1)^n,$$

placed at the start of every frame.
This particular choice is motivated by three useful properties:

- Squaring the sequence removes modulation entirely,
- The squared signal produces a single strong spectral component,
- The resulting tone allows simple and robust frequency estimation with a DFT comb.

The length of data block can be increased further but for the sake of computational time the length of data frame per monte carlo is taken as 2000.

3 Clock Offset, Carrier Offset and Noise

- The nominal sampling rate is $2R_s$. A random clock offset within ± 50 ppm is applied by resampling the transmit signal using linear interpolator. This simulates realistic timing mismatch.
- Carrier offset is applied through:

$$r(t) = x(t)e^{j2\pi f_{\text{cfo}} t},$$

  with f_{cfo} chosen in the allowed ppm range.
- AWGN is finally added to interpolated samples such that the SER range covers 10^{-4} to 10^{-1}.
- Before staring to estimate the clock and carrier offsets, 0–15 samples are dropped randomly to simulate uncertain start alignment.

4 Coarse Carrier Frequency Estimation and Choice of Comb Resolution

The coarse carrier frequency is estimated using the preamble. Since the preamble symbols alternate in sign, squaring the received preamble

$$x[n] \to x[n]^2$$

removes the QPSK modulation and produces a single complex tone whose frequency is twice the carrier frequency offset. This allows frequency estimation to be performed simply by testing a set of candidate frequencies (a DFT–comb) instead of computing a full FFT.

The comb scans only the permitted frequency range, and the frequency giving the largest spectral peak is selected. The corresponding carrier offset estimate is

$$\hat{f}_{\text{cfo}} = \frac{\hat{f}_2}{2}.$$

How the spacing of the comb was determined. The comb spacing is not arbitrary. After coarse correction, a fine phase tracker operates by averaging phase over short blocks (about $L \approx 10$ symbols). For this tracker to work, the residual phase rotation over one block must stay small compared to the QPSK decision boundary of $\pi/2$ radians. This imposes a constraint on the allowable

residual frequency offset.
Because the squared signal contains a tone at twice the carrier offset, the DFT–comb must be spaced such that, after dividing by two, the remaining error is within this residual bound. This leads to a practical rule:

$$\Delta f_{\text{comb}} \approx 2\,\Delta f_{\text{res}},$$

so that the coarse estimate always lands close enough for the fine tracker to complete the correction reliably.

Noise robustness. To further improve stability, the preamble is divided into segments and the periodograms from each segment are averaged. This strengthens the dominant peak and reduces random fluctuations due to noise.

5 Matched Filtering

The received signal is filtered again using the same RRC filter. This simply completes the raised-cosine pair.

6 Clock Recovery (Early–Late Interpolator)

Clock recovery is performed using an Early–Late timing error detector with fractional interpolation. Instead of sampling strictly every *sps* samples, the receiver maintains a fractional timing variable μ that represents the current sampling phase.

For each symbol instant, three interpolated values are obtained:

$$x_{\text{early}} = x(\mu - \Delta), \qquad x_{\text{center}} = x(\mu), \qquad x_{\text{late}} = x(\mu + \Delta),$$

where Δ is a small fractional offset (here chosen as $sps/8$). The timing error is computed using both I and Q components:

$$e = \Re\{x_{\text{center}}\}\left(\Re\{x_{\text{late}}\} - \Re\{x_{\text{early}}\}\right) + \Im\{x_{\text{center}}\}\left(\Im\{x_{\text{late}}\} - \Im\{x_{\text{early}}\}\right).$$

If the sampling point is late, the late sample becomes stronger than the early sample, giving a positive error; if it is early, the sign reverses. Thus the error indicates the direction in which the sampling instant should move.

Window Averaging and Thresholding

Instead of correcting the timing every symbol, the error values are averaged over a sliding window of approximately ten symbols. The average error is then compared to a small threshold. The logic is:

- If the average error is significantly positive, the sampler is late and the timing phase is advanced slightly;
- If the average error is significantly negative, the sampler is early and the phase is delayed slightly;
- If the error remains within the threshold, the sampler proceeds without correction.

Only a very small step size is applied to the timing phase at each correction, so that the loop follows slow ppm clock drift while avoiding oscillations.

Relationship Between Clock Offset and Interpolator Phase

The interpolator maintains a fractional sampling phase that determines where each symbol is taken relative to the incoming samples. Denote this phase by ϕ_k, obtained by reducing the timing state modulo the samples-per-symbol value.

When the transmitter and receiver clocks are perfectly matched, this phase remains almost constant. However, in the presence of a small clock mismatch, the sampling point slowly drifts forward or backward. With every symbol, the fractional phase shifts by a tiny amount that is proportional to the clock offset. As the shift accumulates, the phase eventually wraps around by one full sample, producing the characteristic sawtooth pattern observed in the plots.

The number of symbols required for one sawtooth cycle is inversely related to the magnitude of the clock offset. A larger offset causes the drift to build up faster, so the sawtooth repeats more frequently; a smaller offset results in a much slower repetition rate.

Interpretation of Results

The recorded interpolator traces clearly reflect this behaviour. Clock offsets of ± 50 ppm produce rapid sawtooth cycling, while a 20 ppm offset leads to a noticeably longer cycle.
Changing the sign reverses the direction of drift, but the magnitude of the periodicity remains determined by the offset size.

These observations confirm that the clock recovery loop is tracking timing correctly and that the interpolator behaviour is consistent with the expected effect of clock mismatch.

Why This Design Was Chosen

This version of the Early–Late loop has several advantages:

- Interpolation allows true fractional timing adjustment rather than nearest sample decisions
- Averaging over a short window suppresses noise spikes and stabilizes the loop
- The threshold prevents unnecessary jitter when the timing error is already small
- The loop naturally tracks slow clock offsets without requiring an explicit PLL design

Overall, this approach provides reliable clock recovery while remaining simple and numerically stable, and it performs well under the clock offsets and noise onditions specified in the project.

7 Residual Phase and Frequency Tracking

Even after coarse CFO removal, a small frequency error remains.
Method I used is as follows:

1. Compare received preamble samples with known preamble.
2. Compute unwrapped phase difference.
3. Fit a straight line: slope $\rightarrow$ residual frequency.

Then, for data:

- Perform decision-directed correction blockwise,
- Accumulate phase and de-rotate progressively.

This keeps constellation rotation small without impacting SER much.

8 Symbol Detection and SER Evaluation

Once carrier, timing, and residual phase corrections are applied, the received symbols are projected onto the QPSK decision regions and assigned to the nearest constellation points. These detected symbols are then compared with the original transmitted symbols, and symbol errors are counted.

8.1 Monte–Carlo Averaging Across Frames

The symbol error rate is not obtained from a single transmission. Instead, a large number of independent frames are transmitted and decoded, and the error counts are accumulated across all of them.

Let N_{total} denote the total number of transmitted symbols across all frames, and N_e the total number of incorrectly detected symbols. The symbol error rate is then estimated as

$$\text{SER} = \frac{N_e}{N_{\text{total}}}.$$

This constitutes a Monte–Carlo simulation. Each frame represents a new and independent realization of:

- Random data
- Additive noise
- Carrier frequency offset within the allowed range,
- Clock frequency offset,
- Random number of dropped samples at the start.

If the error rate were computed from only one frame, the value would fluctuate significantly due to randomness. By averaging across many frames, the estimate converges to the true error probability and produces a smooth and stable SER curve.

8.2 Comparison With Expected Performance

The simulated SER is compared with the theoretical QPSK SER in AWGN. The key observations are:

- When carrier and clock offsets are zero, the simulated curve closely matches the expected theoretical curve

- When offsets are introduced, the performance remains close to the theoretical bound once synchronization and phase tracking are applied
- No significant error floor appears, confirming that all the offsets are corrected effectively.

These results indicate that the synchronization chain functions correctly and that the modem behaves equivalently to an ideal QPSK system once properly aligned.

9 Plots

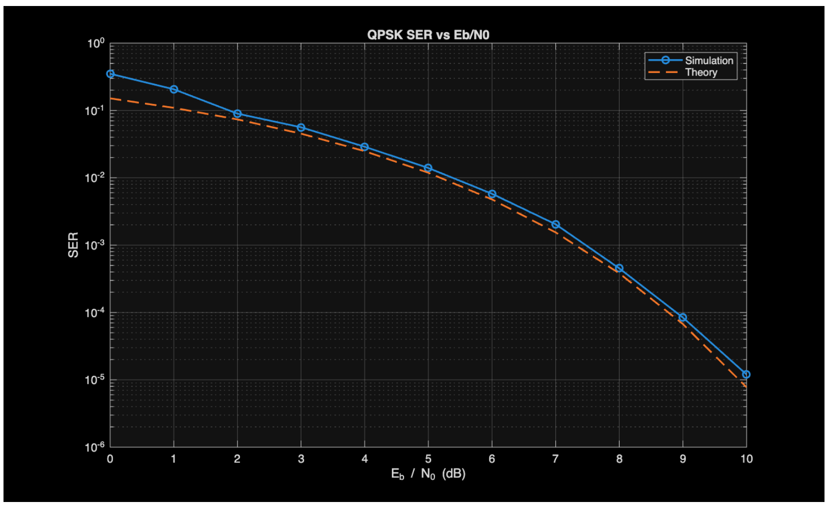

Figure 1: Symbol Error Rate (SER) versus Eb/N0 for QPSK where there is no CFO and clock error.

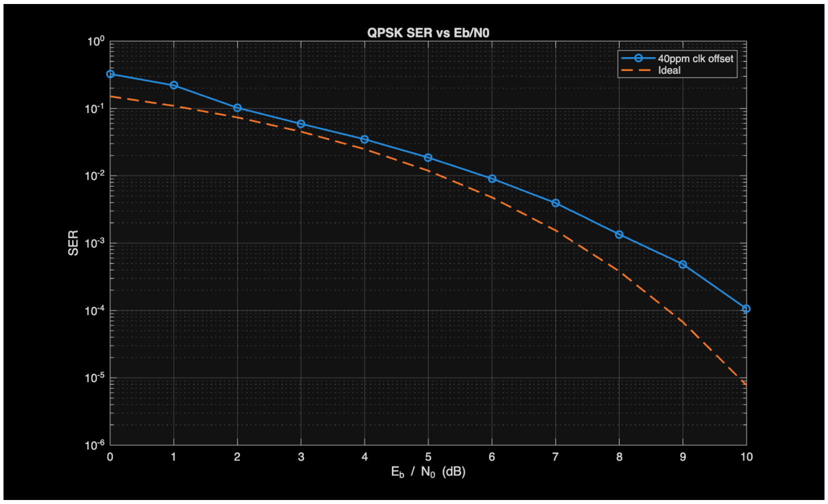

Figure 2: Symbol Error Rate (SER) versus Eb/N0 for QPSK where there is clock error but no CFO.

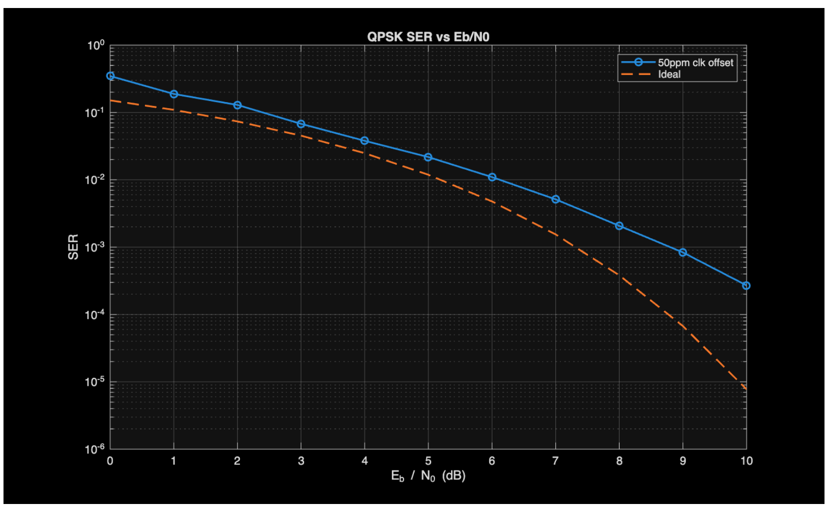

Figure 3: Symbol Error Rate (SER) versus Eb/N0 for QPSK where there is clock error but no CFO.

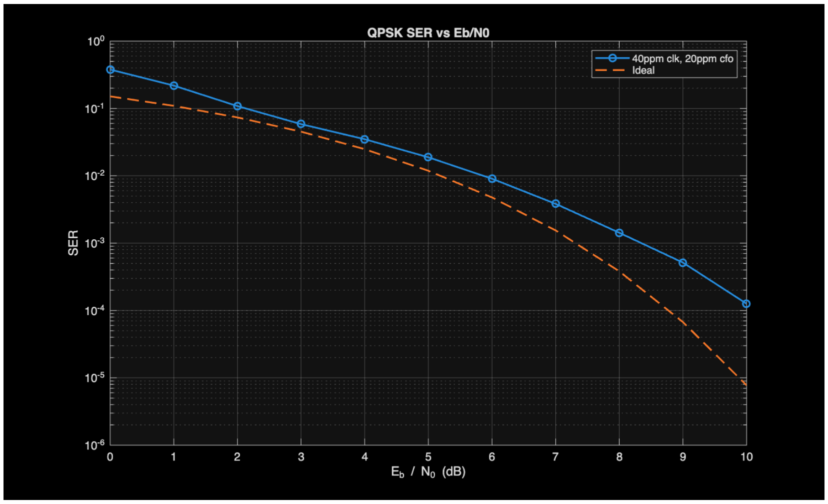

Figure 4: Symbol Error Rate (SER) versus Eb/N0 for QPSK where there is CFO and clock error.

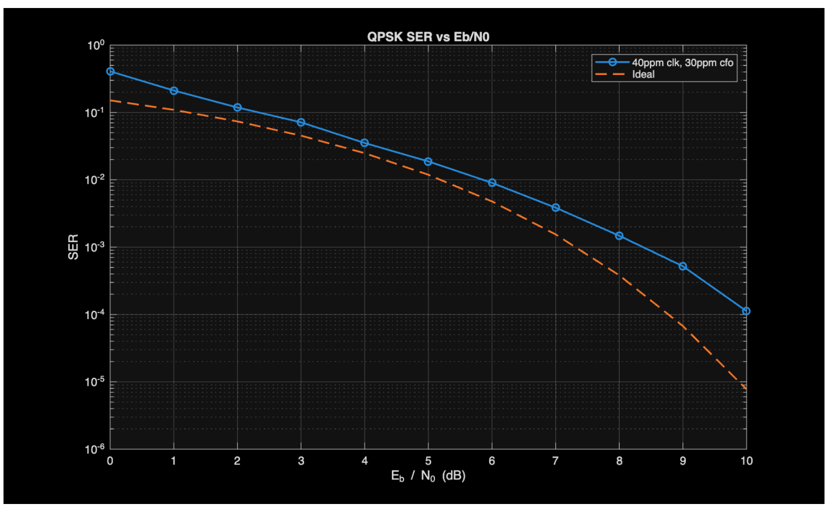

Figure 5: Symbol Error Rate (SER) versus Eb/N0 for QPSK where there is CFO and clock error.

10 Conclusion

This work presented the design and evaluation of a complete synchronization chain for a QPSK receiver. The system combines coarse carrier estimation from the squared preamble, matched filtering, timing recovery through an Early–Late interpolator, and residual phase correction. Together, these stages allow the receiver to operate reliably in the presence of carrier offset, clock mismatch and noise.

Across all operating conditions tested, the detected symbols closely followed the expected performance, and the resulting error rates remained near the theoretical limits. This demonstrates that the synchronization approach is both robust and well suited for the conditions specified in the project.