EECE 442 – Communication Systems

Course Project Report

American University of Beirut
Fall 2025–2026

Tony Abi Haidar
Kassem Hassoun

Date: November 2025

Contents

Part 1: Signal Representation Through Sampling

1.1 Generic Sampling

For the input cosine function, we used $f_0 = 1000\,\text{Hz}$; hence the Nyquist frequency is $f_N = 2000\,\text{Hz}$.

The reconstruction for sampling above the Nyquist rate closely matches the original signal, as the reconstructed waveform almost completely overlaps the original in the plot. In contrast, the reconstructed signal for sampling below the Nyquist rate is significantly different due to the lack of samples, which effectively turns the waveform into an incorrect low-frequency cosine.

According to the sampling theorem, a continuous-time signal can be perfectly reconstructed from its samples if the sampling rate is at least twice the highest frequency in the signal. The results obtained from the plots are a direct illustration of this theorem.

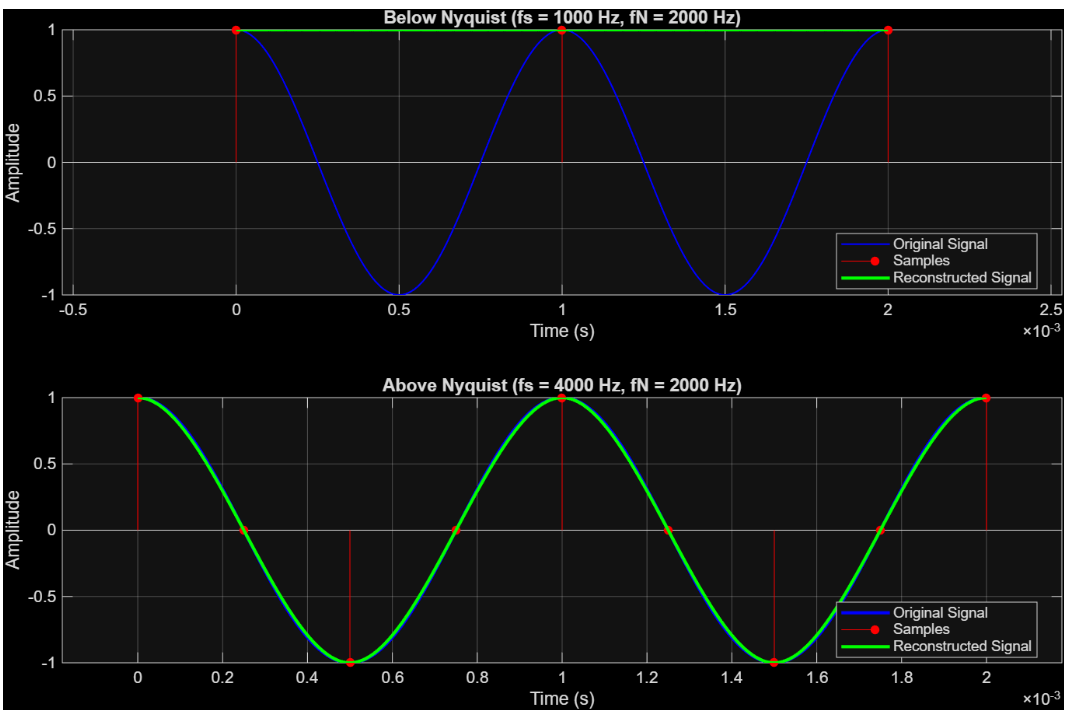


Figure 1.1: Original, sampled, and reconstructed signals for different sampling rates.

1.2 Sampling Through Fourier Series

1.2.1 Choice of Period T

We first chose a large period $T = 15$ to ensure that the Fourier series accurately models the entire signal rather than only a concentrated central slice.

1.2.2 Effect of the Number of Harmonics n

For small values of n (e.g., $n = 1$), the Fourier series produces a smooth low-frequency representation of the original signal. As n increases, the approximation becomes progressively more accurate. For $n = 15$ and $n = 50$, the reconstructed signal models almost the entirety of the original signal, including its higher-frequency components. Thus, increasing n allows the series to span the original signal more accurately.

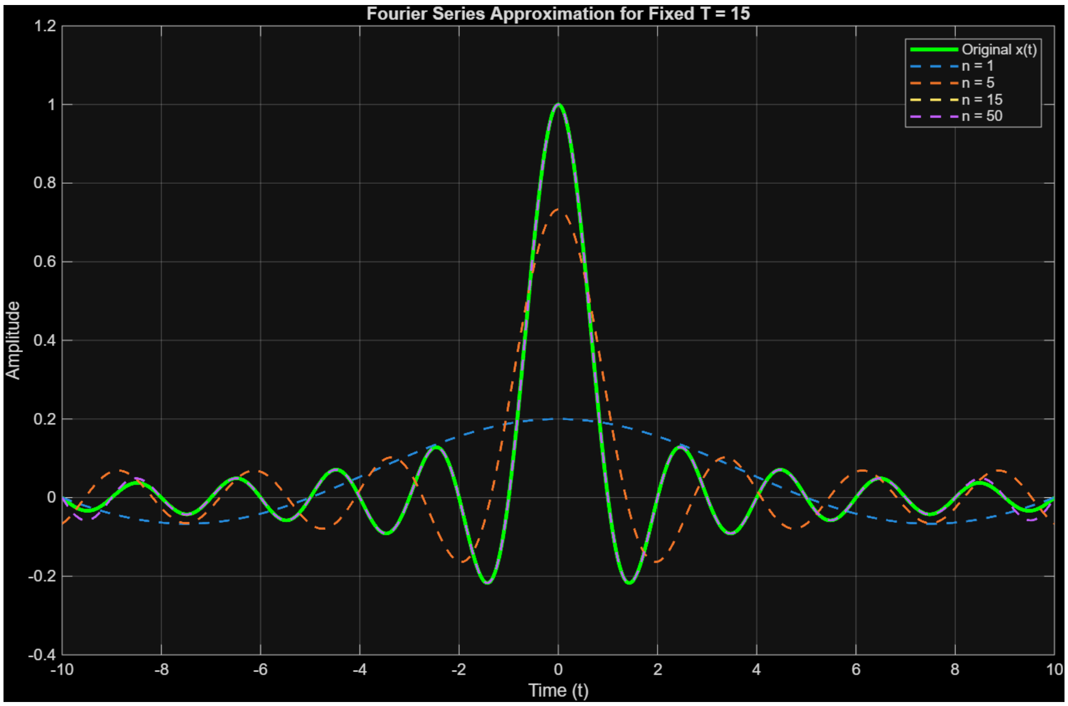


Figure 1.2: Fourier series approximations for different numbers of harmonics n.

1.2.3 Effect of Varying T

We fixed $n = 30$, since from the previous results both $n = 15$ and $n = 50$ represented the signal well. Choosing $n = 30$ provides an accurate approximation without unnecessary computation.

For a small period $T = 6$, the approximation is poor and does not resemble the original sinc at all. The waveform is truncated between $t = -3$ and $t = 3$ due to the Gibbs phenomenon. When we increase T (e.g., $T = 15$ and $T = 30$), the approximation becomes much better and properly represents the original signal $x(t)$.

However, if T is increased too much (e.g., $T = 80$), the overall structure changes from an accurate reconstruction to one that is effectively bandwidth-limited, distorting the high-frequency details of the message signal even if the signal is entirely contained within the chosen period.

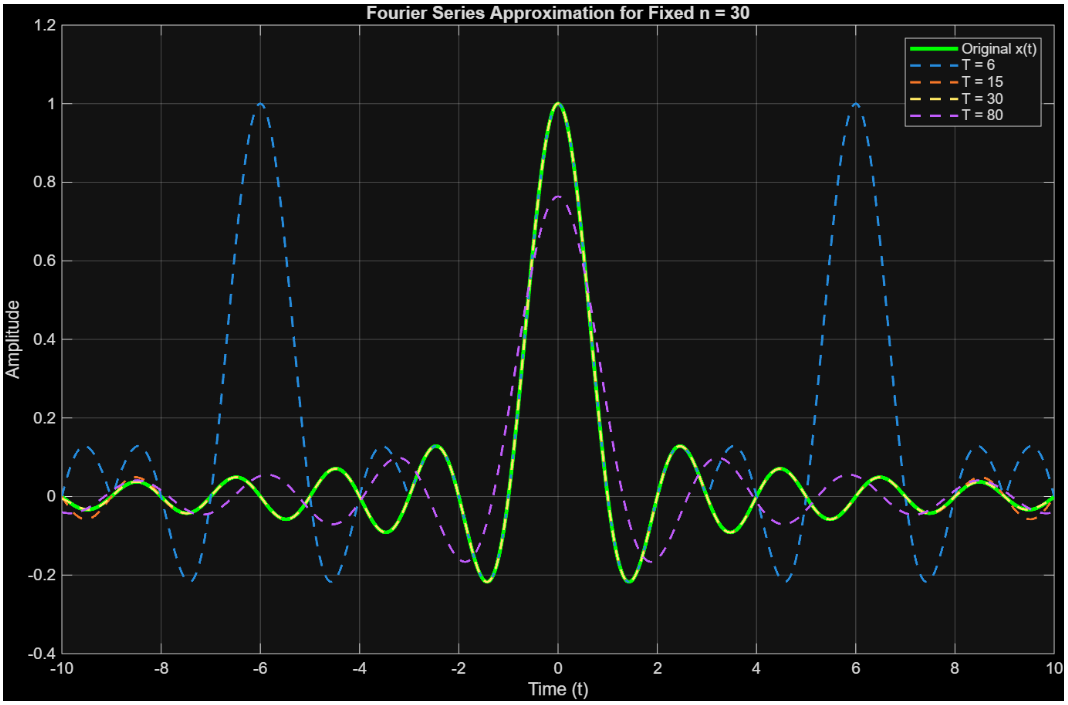


Figure 1.3: Effect of varying T on the Fourier series approximation.

1.2.4 Error Versus Number of Harmonics n

For $T = 20$, we observed that the approximation error decreases as the number of harmonics increases, with a particularly rapid decrease between $n = 5$ and $n = 15$. This indicates that around 15 harmonics already provide a sufficiently accurate approximation.

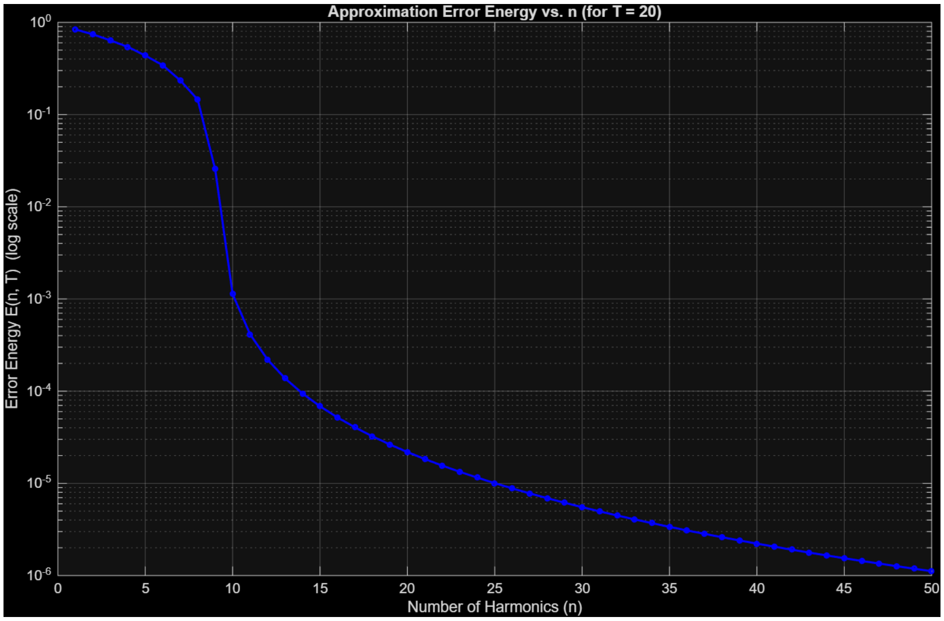


Figure 1.4: Approximation error as a function of the number of harmonics n.

1.2.5 Error Versus Period T

For fixed $n = 30$, as T varies, the approximation error exhibits periodic minima at integer values of T, where the endpoints of the truncated sinc coincide with its zeros. At these values, the periodic extension is continuous, minimizing discontinuities at the boundaries and resulting in nearly zero error. Between these points, the error increases due to Gibbs-type effects caused by discontinuities.

Overall, for large T, the error remains small, and its oscillatory behavior reflects the trade-off between endpoint continuity (favoring specific T values) and the effective low-pass bandwidth.

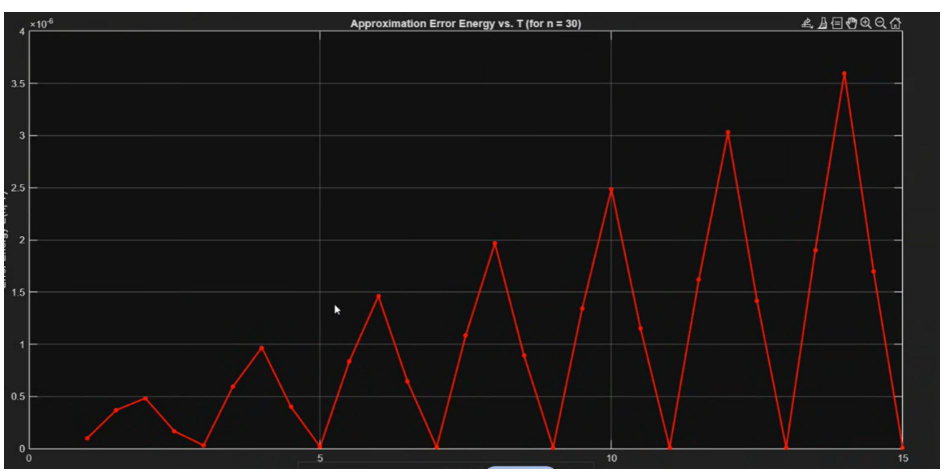


Figure 1.5: Approximation error as a function of the period T.

Part 2: Quantization

2.1 Two-Level Quantization

In two-level quantization, we use a single threshold and two output levels (low/high). This is effectively a sign quantizer.

- The distortion is very visible; the peaks of the cosine are flattened.
- The mean squared error (MSE) is the largest (worst quality), because only the sign of the signal is preserved.

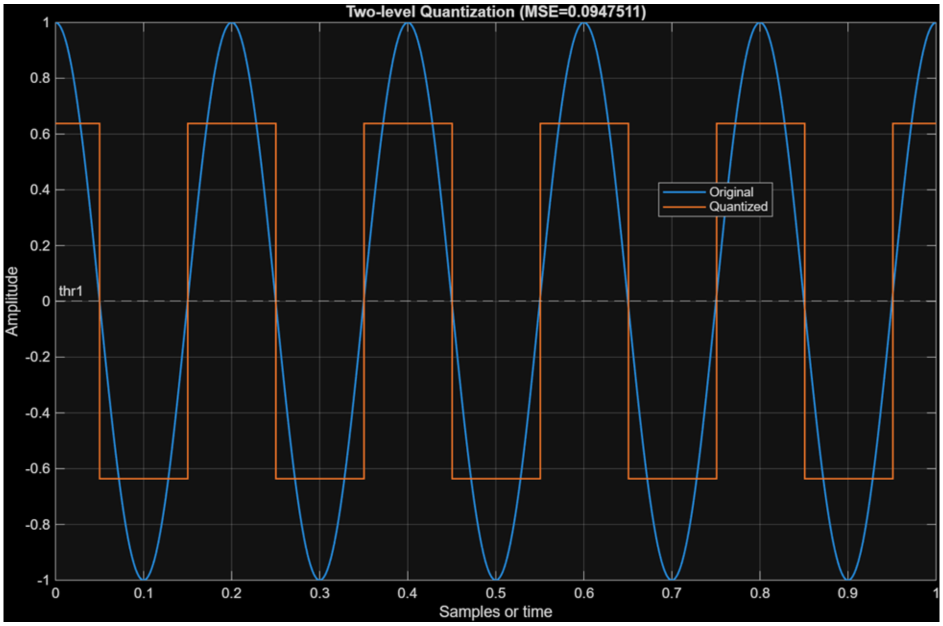
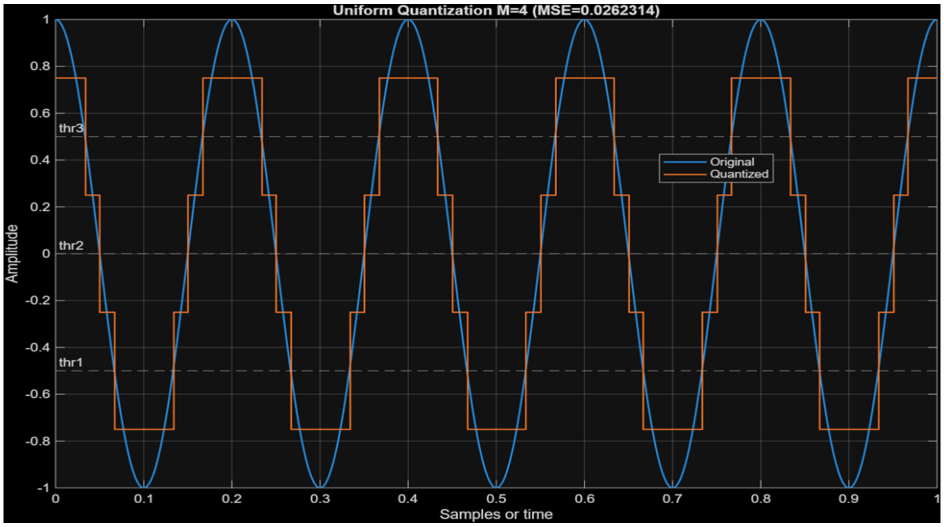


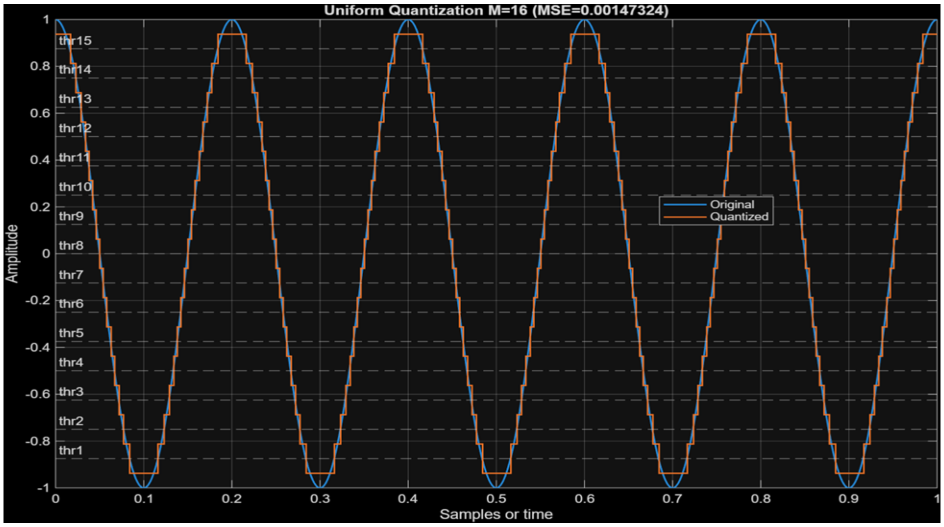
Figure 2.1: Two-level quantization of the cosine signal.

2.2 Uniform Multi-Level Quantization

In uniform quantization, the signal range $[x_{\min}, x_{\max}]$ is split into equal-width bins, and the reproduction levels are the midpoints of these bins.

- For $M = 4$: the resolution is coarse and the step-shaped waveform is clearly visible; distortion is still considerable.
- For $M = 16$: the steps become finer; the quantized waveform tracks the cosine much better.
- For $M = 32$: the steps are very fine; the quantized waveform follows the cosine closely and becomes visually almost indistinguishable from it.





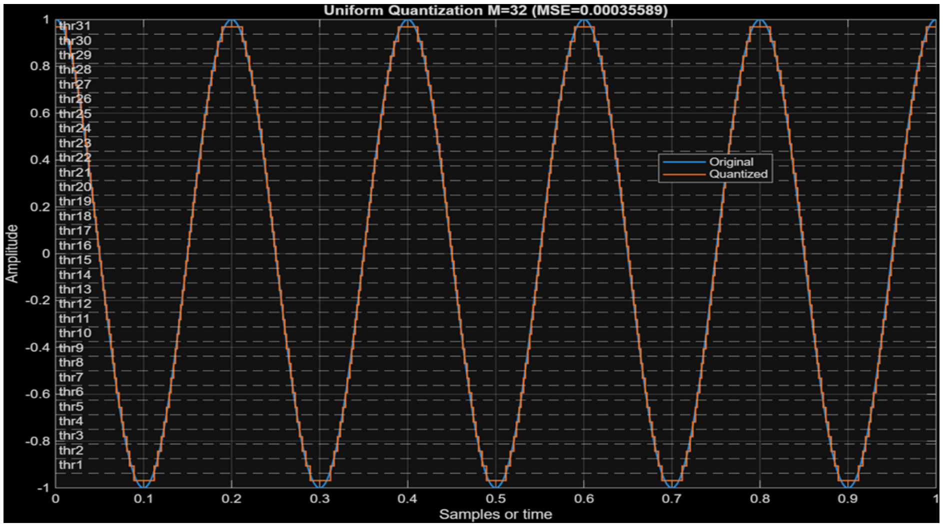


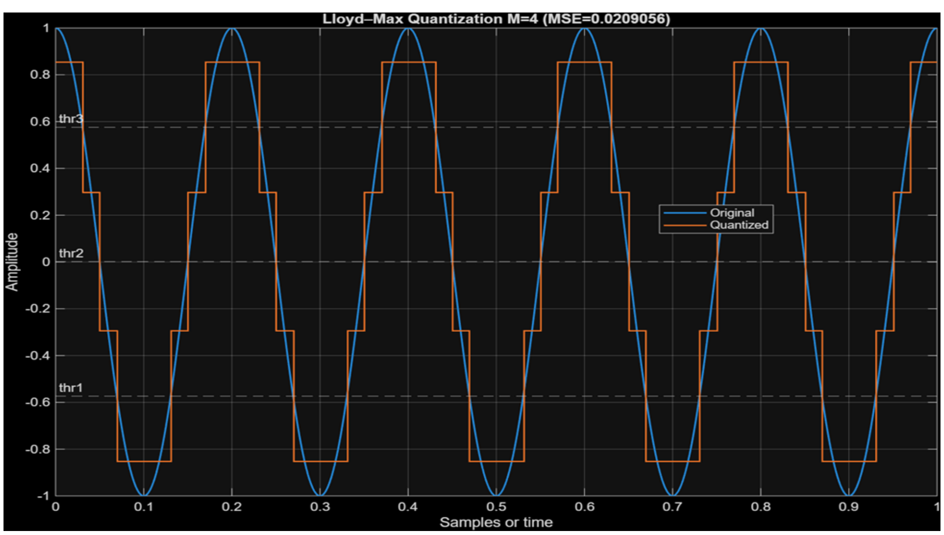
Figure 2.2: Uniform multi-level quantization for different numbers of levels M.

Overall, the MSE decreases sharply as M increases, demonstrating the expected rate–distortion behavior.

2.3 Lloyd–Max Quantization

Lloyd–Max quantization starts from uniform bins and then iteratively adapts thresholds and reproduction levels to the actual signal histogram.

- For each value of M, the resulting MSE is always lower than that of the corresponding uniform quantizer (approximately 10–20% improvement).
- Intuitively, the quantizer spends resolution (more levels) where the signal samples are dense and fewer levels where they are rare.



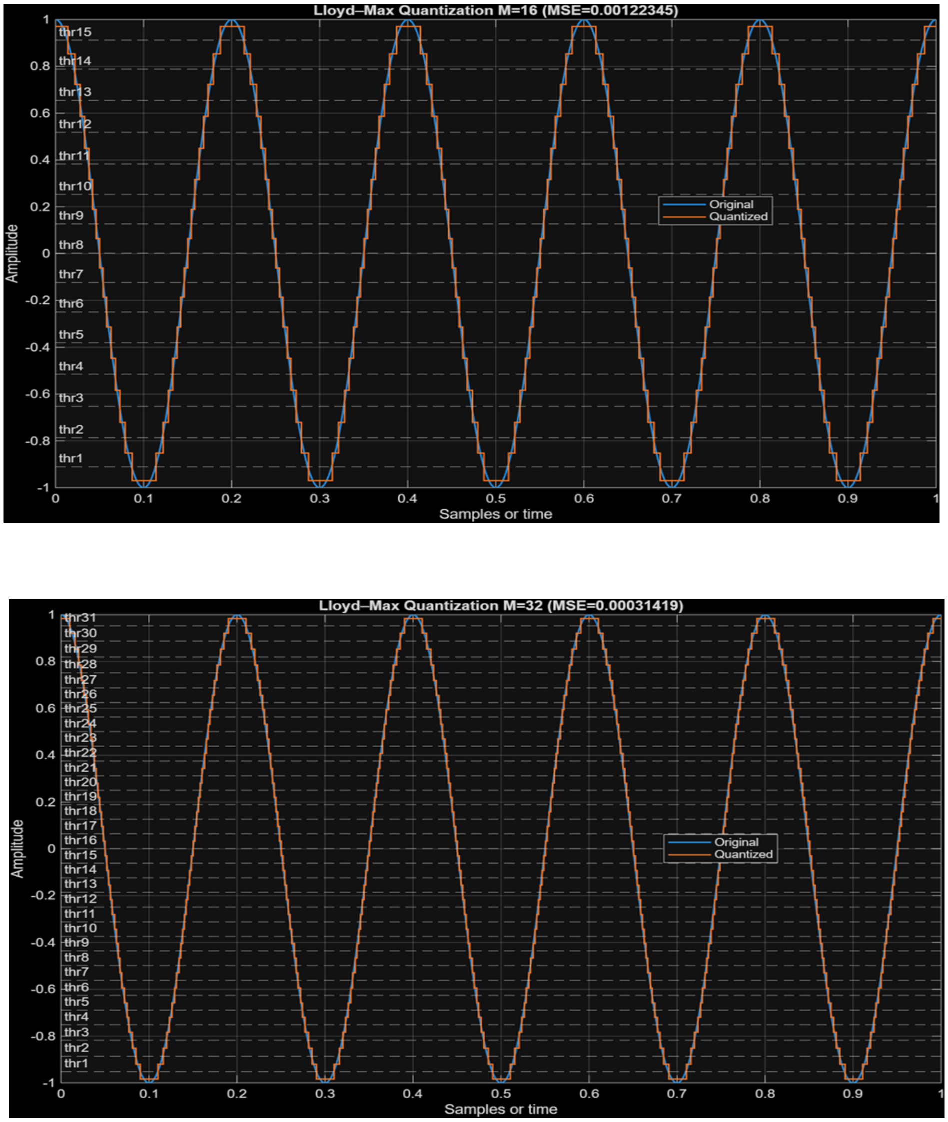


Figure 2.3: Lloyd–Max quantization results for various M.

2.4 Overall Assessment

Table 2.1 summarizes the performance of the different quantizers.

Quantizer Type	Levels M	Adaptive?	Typical MSE	Comments
Two-level	2	No	Highest	Captures only coarse polarity
Uniform	4	No	Medium	Simple and easy to implement
Uniform	16, 32	No	Lower	Finer resolution as M grows
Lloyd–Max	4	Yes	Lower	Better use of bins
Lloyd–Max	16, 32	Yes	Lowest	Near-optimal for the data

Table 2.1: Qualitative comparison of quantization methods.

The trade-off between rate, distortion, and source statistics can be summarized as follows:

- Increasing the rate (number of levels M) provides finer resolution and lower distortion (MSE) but requires more bits per sample.
- With fewer levels, distortion rises because the quantization steps are larger.
- If the source is non-uniform, a uniform quantizer wastes many bins, whereas Lloyd–Max adapts to the probability distribution and achieves lower MSE at the same bit rate.

In summary, a higher rate leads to lower distortion, and adapting to the source distribution improves performance for a fixed bit rate.

The thresholds and levels used in each case:

```
Quantization Demo
Quantization levels:
   -0.6360    0.6380

Two-level quantization:
  Threshold: 0.00157236
  Levels: -0.635983  0.637983
  MSE: 0.0947511

M = 4
  thresholds (3): -0.499996 2.47234e-06 0.500001
  levels     (4): -0.749996 -0.249997 0.250002 0.750001
  MSE = 0.0262314

Quantization levels:
   -0.7500   -0.2500    0.2500    0.7500

Uniform quantization (M=4): MSE=0.0262314
M = 16
  thresholds (15): -0.874995 -0.749996 -0.624996 -0.499996 -0.374997 -0.249997 -0.124997
2.47234e-06 0.125002 0.250002 0.375002 0.500001 0.625001 0.750001 0.875
  levels     (16): -0.937495 -0.812496 -0.687496 -0.562496 -0.437496 -0.312497 -0.187497
-0.0624974 0.0625023 0.187502 0.312502 0.437501 0.562501 0.687501 0.8125 0.9375
  MSE = 0.00147324

Quantization levels:
  Columns 1 through 15

   -0.9375   -0.8125   -0.6875   -0.5625   -0.4375   -0.3125   -0.1875   -0.0625    0.0625
0.1875    0.3125    0.4375    0.5625    0.6875    0.8125

  Column 16

    0.9375

Uniform quantization (M=16): MSE=0.00147324
M = 32
  thresholds (31): -0.937495 -0.874995 -0.812496 -0.749996 -0.687496 -0.624996 -0.562496
-0.499996 -0.437496 -0.374997 -0.312497 -0.249997 -0.187497 -0.124997 -0.0624974 2.47234e-06
0.0625023 0.125002 0.187502 0.250002 0.312502 0.375002 0.437501 0.500001 0.562501 0.625001
0.687501 0.750001 0.8125 0.875 0.9375
  levels     (32): -0.968745 -0.906245 -0.843745 -0.781246 -0.718746 -0.656246 -0.593746
-0.531246 -0.468746 -0.406247 -0.343747 -0.281247 -0.218747 -0.156247 -0.0937473 -0.0312475
0.0312524 0.0937522 0.156252 0.218752 0.281252 0.343752 0.406251 0.468751 0.531251 0.593751
0.656251 0.718751 0.781251 0.84375 0.90625 0.96875
  MSE = 0.00035589

Quantization levels:
  Columns 1 through 15

   -0.9687   -0.9062   -0.8437   -0.7812   -0.7187   -0.6562   -0.5937   -0.5312   -0.4687
-0.4062   -0.3437   -0.2812   -0.2187   -0.1562   -0.0937

  Columns 16 through 30

   -0.0312    0.0313    0.0938    0.1563    0.2188    0.2813    0.3438    0.4063    0.4688
0.5313    0.5938    0.6563    0.7188    0.7813    0.8438

  Columns 31 through 32
```

```
    0.9063    0.9688

Uniform quantization (M=32): MSE=0.00035589
Quantization levels:
   -0.8527   -0.2942    0.2972    0.8541

Lloyd-Max quantization (M=4): MSE=0.0209056, iterations=Quantization levels:
  Columns 1 through 15

   -0.9698   -0.8520   -0.7208   -0.5838   -0.4470   -0.3135   -0.1859   -0.0612    0.0644
0.1890    0.3165    0.4498    0.5863    0.7230    0.8536

  Column 16

    0.9705

Lloyd-Max quantization (M=16): MSE=0.00122345, iterations=Quantization levels:
  Columns 1 through 15

   -0.9836   -0.9202   -0.8517   -0.7836   -0.7195   -0.6581   -0.5945   -0.5295   -0.4669
-0.4046   -0.3434   -0.2807   -0.2168   -0.1550   -0.0926

  Columns 16 through 30

   -0.0299    0.0330    0.0958    0.1581    0.2199    0.2807    0.3434    0.4075    0.4697
0.5322    0.5945    0.6558    0.7195    0.7855    0.8534

  Columns 31 through 32

    0.9214    0.9841

Lloyd-Max quantization (M=32): MSE=0.00031419, iterations=
 End of Demo
>>
```

Part 3: Source Coding

3.1 Fixed-Length vs. Huffman Coding

The fixed-length coder produces a constant rate of $\lceil \log_2 M \rceil$ bits per symbol, as expected. In contrast, Huffman coding assigns shorter codewords to more probable quantization levels and longer ones to rare levels.

For the tested quantized sequence:

- The fixed-length rate was around 3 bits/symbol.
- The Huffman average length was about 2.29 bits/symbol.
- The measured entropy was $H(A) \approx 2.26$ bits/symbol.

Thus, the Huffman code is nearly optimal for this source and yields a compression gain of about 20–25% over the fixed-length baseline. Both methods perfectly reconstructed the original quantized sequence, confirming lossless operation.

```
--- Quantizer A ---
Seed fixed at 442
Alphabet A = {-3.600000e+00}
Alphabet A = {-1.200000e+00}
Alphabet A = {1}
Alphabet A = {3.400000e+00}
Alphabet A = {5.600000e+00}
Levels:     -3.6000 -1.2000 1.0000 3.4000 5.6000
Thresholds: -2.0000 -0.5000 0.5000 2.0000
Stream length: 17 samples

--- Quantizer A: Empirical model ---
Level    p(level)
 -3.6000    0.1765
 -1.2000    0.2353
  1.0000    0.1765
  3.4000    0.2941
  5.6000    0.1176
Entropy H(A) = 2.2569 bits/symbol

--- Quantizer A (FLC): Fixed-Length Coding ---
Alphabet size M = 5
Bits per symbol L = 3

--- Quantizer A (Huffman): Empirical model ---
Level    p(level)
 -3.6000    0.1765
 -1.2000    0.2353
  1.0000    0.1765
  3.4000    0.2941
  5.6000    0.1176
Entropy H(A) = 2.2569 bits/symbol

--- Quantizer A (Huffman): Huffman Coding ---
Level    p(level)    Code
--------------------------------
 -3.60    0.1765        010
 -1.20    0.2353        10
  1.00    0.1765        11
  3.40    0.2941        00
  5.60    0.1176        011
--------------------------------
Average code length = 2.2941 bits/symbol

=================== Part 3(d) Summary ===================
Entropy H(A)                : 2.2569 bits/symbol
Fixed-length L (ceil log2M): 3 bits/symbol
Huffman avg length          : 2.2941 bits/symbol
---------------------------------------------------------
Sequence length N           : 17 symbols
Fixed-length total bits     : 51  (3.0000 bits/sym)
Huffman total bits          : 39  (2.2941 bits/sym)
Lossless (Fixed-length)     : PASS
Lossless (Huffman)          : PASS
=========================================================

First 8 FLC symbols (L=3):
Level    Code
 -3.60    000



 -1.20    001
 -1.20    001
  1.00    010
  5.60    100
  3.40    011
  3.40    011
 -3.60    000

First 8 Huffman symbols:
Level    Code
 -3.60    010
 -1.20    10
 -1.20    10
  1.00    11
  5.60    011
  3.40    00
  3.40    00
 -3.60    010
```

Figure 3.1: Comparison of fixed-length and Huffman code rates.

Lloyd–Max introduces non-uniform bin sizes that create skewed symbol probabilities, effectively forming 'dead-zones' around low-probability regions. This increases redundancy in the histogram and allows Huffman coding to achieve shorter average code lengths.

3.2 Block Coding and Entropy Rate

We also applied Huffman coding on non-overlapping symbol blocks. The results showed that both the block entropy rate H_k/k and the average Huffman block length per symbol decreased as the block size k increased. For example:

- For $k = 2$, H_k/k was roughly 1.38 bits/symbol.
- For $k = 4$, H_k/k dropped to about 0.50 bits/symbol.

The corresponding Huffman block rates closely followed these values. This reduction demonstrates that the quantized data contain short-range statistical dependencies that can be exploited when coding multiple symbols jointly. As k increases, the block-based coder approaches the true entropy rate of the source and achieves a lower average number of bits per symbol than single-symbol coding.

```
=============== Block Coding Analysis ===============
Concatenation k= 2
Fixed-Length (Symbol-wise)      3.0000
Fixed-Length (Block-wise)/k     1.5000

--- Quantizer A: Summary for part (g) ---
Entropy H(A) (per symbol)                         : 2.2569 bits/symbol
Average length using Huffman (symbol-wise)        : 2.2941 bits/symbol
Block entropy rate H_k/k (per symbol, k=2)       : 1.3750 bits/symbol
Average length using Huffman on 2-blocks (per sym): 1.3750 bits/symbol
-----------------------------------------------
Observation: Huffman(block) and H_k/k should
approach H(A) as k increases, confirming
better coding efficiency with larger blocks.
Concatenation k= 3
Fixed-Length (Symbol-wise)      3.0000
Fixed-Length (Block-wise)/k     1.0000

--- Quantizer A: Summary for part (g) ---
Entropy H(A) (per symbol)                         : 2.2569 bits/symbol
Average length using Huffman (symbol-wise)        : 2.2941 bits/symbol
Block entropy rate H_k/k (per symbol, k=3)       : 0.7740 bits/symbol
Average length using Huffman on 3-blocks (per sym): 0.8000 bits/symbol
-----------------------------------------------
Observation: Huffman(block) and H_k/k should
approach H(A) as k increases, confirming
better coding efficiency with larger blocks.
Concatenation k= 4
Fixed-Length (Symbol-wise)      3.0000
Fixed-Length (Block-wise)/k     0.5000

--- Quantizer A: Summary for part (g) ---
Entropy H(A) (per symbol)                         : 2.2569 bits/symbol
Average length using Huffman (symbol-wise)        : 2.2941 bits/symbol
Block entropy rate H_k/k (per symbol, k=4)       : 0.5000 bits/symbol
Average length using Huffman on 4-blocks (per sym): 0.5000 bits/symbol
-----------------------------------------------
Observation: Huffman(block) and H_k/k should
approach H(A) as k increases, confirming
better coding efficiency with larger blocks.
=====================================================
```

Figure 3.2: Entropy rate and average code length vs. block size k.

3.3 Gains from Short-Range Dependencies

The observed improvement with block coding reflects the presence of short-range dependencies in the quantized signal: adjacent samples are correlated rather than independent. Because real signals change gradually, quantizer outputs tend to repeat or follow predictable local patterns, so knowing one symbol provides information about the next.

By grouping these correlated symbols, the block Huffman coder can assign very short codewords to frequent symbol sequences and thus achieve better compression efficiency. The decline of H_k/k with larger k and the corresponding reduction in average block code length confirm that exploiting such local correlations yields tangible coding gains while maintaining lossless reconstruction.

Part 4: Channel Coding

We used a simple repetition code, repeating every bit L times. For example, with $L = 3$:

$$0 \to 000, \qquad 1 \to 111.$$

At the receiver, we decode using majority voting (e.g., $101 \to 1$, $001 \to 0$).

In the full chain simulation, coded channels consistently performed better than uncoded channels:

- The coded bit error rate (BER) was always smaller than the uncoded BER.
- For all quantizers tested, the uncoded BER at a given SNR lay in a tight range, while the coded BER dropped by roughly one or more orders of magnitude.

===== Channel Coding Gain (BER_uncoded / BER_coded) =====

Quantizer	Coder	EbN0dB	BER_uncoded	BER_coded	Gain
{'Q2' }	{'CFIX'}	6	0.00246	2e-05	123
{'Q2' }	{'CHUF'}	6	0.00256	0	Inf
{'QU8' }	{'CFIX'}	6	0.0023267	0	Inf
{'QU8' }	{'CHUF'}	6	0.0024688	9.2813e-06	266
{'QU16'}	{'CFIX'}	6	0.00222	2.5e-05	88.8
{'QU16'}	{'CHUF'}	6	0.0021824	1.2991e-05	168
{'QL8' }	{'CFIX'}	6	0.00224	4.6667e-05	48
{'QL8' }	{'CHUF'}	6	0.0022846	2.7776e-05	82.25
{'QL16'}	{'CFIX'}	6	0.002395	2.5e-05	95.8
{'QL16'}	{'CHUF'}	6	0.0022566	2.6424e-05	85.4

Figure 4.1: BER performance: coded vs. uncoded with repetition coding.

The coding gain table showed gains between different values (depending on the quantizer and coding combination). The largest gains appeared when the uncoded BER was relatively high, indicating that the repetition code is particularly effective at cleaning up AWGN-induced errors in those cases.

Part 5: Modulation

We considered two modulation schemes: BPSK and QPSK.

Modulation	Constellation Dimension	Bits per Symbol
BPSK	Real (1D)	1
QPSK	Complex (2D)	2

Table 5.1: Summary of the modulation schemes used.

For a given number of bits N_{bits}:

$$N_{\text{symbol,BPSK}} = N_{\text{bits}}, \qquad N_{\text{symbol,QPSK}} = \frac{N_{\text{bits}}}{2}.$$

We compare their performance in the channel simulations of the next part.

Part 6: Channel, Noise, and Detection

6.1 AWGN Channel

We first transmitted BPSK and QPSK signals over an AWGN channel, with and without repetition coding. At equal SNR, coding provided a consistent performance improvement for both modulation schemes.

- Coded BPSK achieved the lowest BER among all scenarios.
- Coded QPSK performed better than uncoded QPSK but remained worse than BPSK because of its denser constellation.

These results demonstrate the trade-off between robustness (BPSK, especially when coded) and spectral efficiency (QPSK).

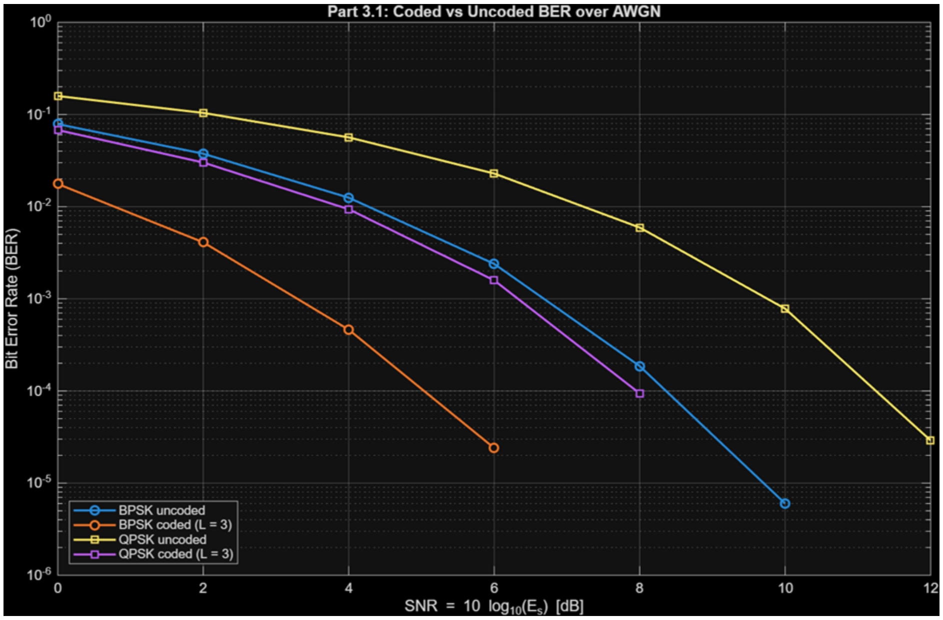


Figure 6.1: Illustrative BER comparison between BPSK and QPSK.

```
=== BPSK over AWGN (uncoded + coded, L = 3) ===
SNR =  0 dB:  BER_unc = 7.859e-02,  BER_cod = 1.762e-02
SNR =  2 dB:  BER_unc = 3.753e-02,  BER_cod = 4.099e-03
SNR =  4 dB:  BER_unc = 1.250e-02,  BER_cod = 4.620e-04
SNR =  6 dB:  BER_unc = 2.390e-03,  BER_cod = 2.400e-05
SNR =  8 dB:  BER_unc = 1.860e-04,  BER_cod = 0.000e+00
SNR = 10 dB:  BER_unc = 6.000e-06,  BER_cod = 0.000e+00
SNR = 12 dB:  BER_unc = 0.000e+00,  BER_cod = 0.000e+00
=== QPSK over AWGN (uncoded + coded, L = 3) ===
SNR =  0 dB:  BER_unc = 1.586e-01,  BER_cod = 6.778e-02
SNR =  2 dB:  BER_unc = 1.041e-01,  BER_cod = 3.011e-02
SNR =  4 dB:  BER_unc = 5.644e-02,  BER_cod = 9.365e-03
SNR =  6 dB:  BER_unc = 2.286e-02,  BER_cod = 1.585e-03
SNR =  8 dB:  BER_unc = 5.933e-03,  BER_cod = 9.400e-05
SNR = 10 dB:  BER_unc = 7.800e-04,  BER_cod = 0.000e+00
SNR = 12 dB:  BER_unc = 2.900e-05,  BER_cod = 0.000e+00
```

Figure 6.2: BER curves over AWGN for BPSK/QPSK, coded and uncoded.

6.2 Linear Gaussian Channel with MLSE

We then considered a linear ISI channel with tap weights $[2, 1]$. For each SNR value, a block of QPSK symbols was transmitted through this channel and detected using a Viterbi maximum-likelihood sequence estimator (MLSE). The same bitstream was also transmitted over a pure AWGN channel to provide a baseline, and the theoretical QPSK BER curve was included for reference.

The results show that the ISI channel combined with Viterbi MLSE consistently achieves lower BER than the AWGN QPSK baseline across all tested SNR values:

- At low SNR (0–2 dB), the ISI+MLSE BER already outperforms the AWGN simulated BER.
- As SNR increases to 4–6 dB, the ISI+MLSE BER rapidly falls to very low values, while the AWGN BER remains higher.
- Beyond 8 dB, the ISI+MLSE curve reaches numerical zero (no bit errors observed), whereas the AWGN baseline still shows non-zero BER.

The theoretical AWGN curve aligns closely with the simulated AWGN BER, confirming the correctness of the baseline. Overall, these results demonstrate that the Viterbi equalizer enables highly reliable detection over this particular ISI channel.

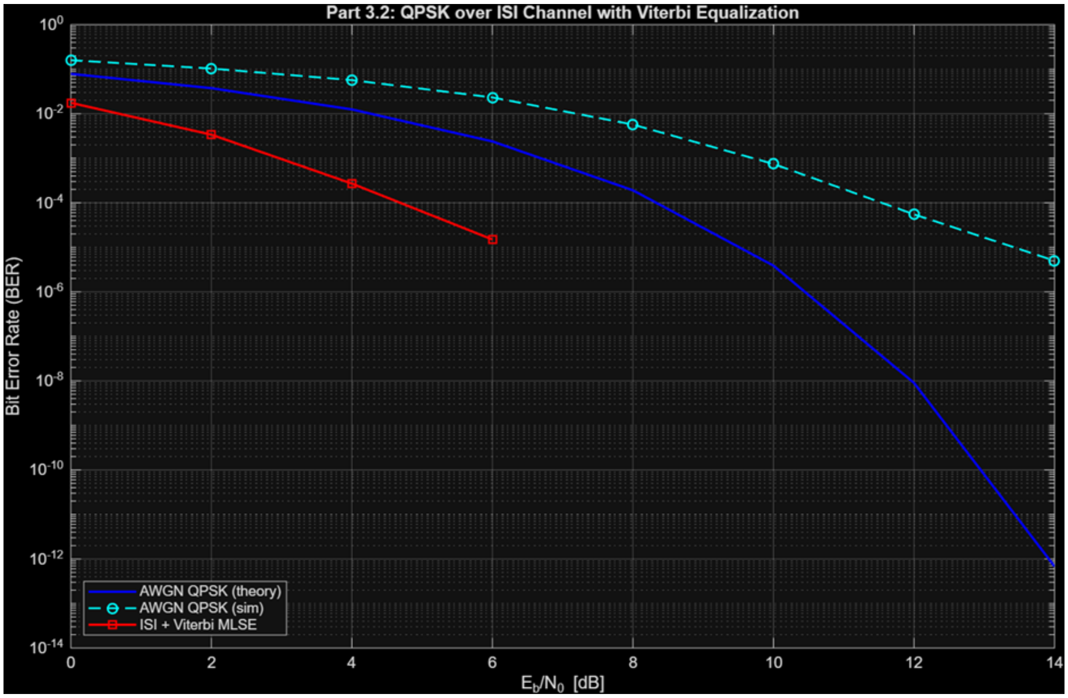


Figure 6.3: BER performance of ISI + MLSE vs. AWGN QPSK.

Discussion

The ISI channel uses tap weights $[2, 1]$, which significantly increase the effective received signal energy. After applying Viterbi MLSE detection, the ISI channel achieves BER values that are consistently lower than both the simulated and the theoretical AWGN QPSK curves for all SNR values.

Unlike typical ISI channels that degrade performance and require MLSE to recover AWGN-like behavior, this channel is highly favorable, and the MLSE equalizer fully leverages the increased energy. As a result, the ISI+MLSE BER curve quickly decreases and reaches zero errors at SNRs above 8 dB. These results accurately reflect the channel's behavior and confirm the correctness of the implemented MLSE receiver.

Part 7: Full Communication Chain

The complete digital communication chain was implemented, starting from quantization and lossless source coding (fixed-length and Huffman), followed by optional repetition channel coding, BPSK/QPSK modulation, AWGN channel transmission, demodulation, channel decoding, and final bit-error evaluation. All blocks were connected end-to-end to evaluate how the compression scheme and channel coding affect the BER of the transmitted bitstream.

===== End-to-end metrics for each (quantizer, coder) =====

quantizerID	coderID	M	SNRdB_total	totalBits	bitsPerSym	latencySym	latencySec	EbN0dB_chan	ber_uncoded_chan	ber_coded_chan
{'Q2' }	{'CFIX'}	2	4.3795	50000	1	1	1	6	0.00248	4e-05
{'Q2' }	{'CHUF'}	2	4.3795	50000	1	50000	50000	6	0.00246	2e-05
{'QU8' }	{'CFIX'}	8	10.44	1.5e+05	3	1	1	6	0.00246	2e-05
{'QU8' }	{'CHUF'}	8	10.44	1.0746e+05	2.1493	50000	50000	6	0.0021961	3.7221e-05
{'QU16'}	{'CFIX'}	16	16.491	2e+05	4	1	1	6	0.00234	1.5e-05
{'QU16'}	{'CHUF'}	16	16.491	1.5274e+05	3.0549	50000	50000	6	0.0023307	1.9641e-05
{'QL8' }	{'CFIX'}	8	14.521	1.5e+05	3	1	1	6	0.00234	2e-05
{'QL8' }	{'CHUF'}	8	14.521	1.4432e+05	2.8863	50000	50000	6	0.0026955	1.3858e-05
{'QL16'}	{'CFIX'}	16	20.063	2e+05	4	1	1	6	0.00244	1.5e-05
{'QL16'}	{'CHUF'}	16	20.063	1.8733e+05	3.7467	50000	50000	6	0.0023274	2.1352e-05

Figure 7.1: End-to-end metrics for each (quantizer, coder)

Discussion

The end-to-end metrics show a clear progression in quantization performance as the number of levels increases:

- Quantizers with larger M (e.g., QU16, QL16) achieve significantly higher SNR after reconstruction (around 20 dB).
- Coarser quantizers such as Q2 achieve SNRs around 4.4 dB.
- Source coding has no effect on quantizer SNR, which depends solely on quantization distortion.

In terms of compressed bitstream size:

- Fixed-length coding produces exactly $\log_2 M$ bits per symbol.
- Huffman coding achieves noticeable reductions for nonuniform quantizers (e.g., QU8 decreases from 3 bits/symbol to about 2.15).

Discussion

===== Average bits/symbol per quantizer & coder =====

	CFIX	CHUF
Q2	1	1
QU8	3	2.1493
QU16	4	3.0549
QL8	3	2.8863
QL16	4	3.7467

Figure 7.2: Average bits/symbol per quantizer and coder

The bits-per-symbol table confirms that Huffman coding consistently provides more compact representations than fixed-length coding, especially for non-uniform quantizers (QU8, QU16, QL8, QL16), where symbol probabilities are highly skewed. For the uniform Q2 case, both coders achieve exactly 1 bit/symbol because of symmetry. Therefore, Huffman coding is the preferred choice whenever the quantizer output distribution is not uniform.

For all quantizers and coders tested at the chosen SNR:

- The uncoded BER lies in a tight range.
- After applying repetition coding with $L = 3$, the coded BER drops dramatically by roughly two orders of magnitude.
- Huffman-coded streams sometimes show slightly better BER than fixed-length coded streams, due to longer runs of identical bits that are easier for majority decoding.

These results confirm both the correct implementation of channel coding and the overall digital communication chain.

===== Channel Coding Gain (BER_uncoded / BER_coded) =====

Quantizer	Coder	EbN0dB	BER_uncoded	BER_coded	Gain
{'Q2' }	{'CFIX'}	6	0.00248	4e-05	62
{'Q2' }	{'CHUF'}	6	0.00246	2e-05	123
{'QU8' }	{'CFIX'}	6	0.00246	2e-05	123
{'QU8' }	{'CHUF'}	6	0.0021961	3.7221e-05	59
{'QU16'}	{'CFIX'}	6	0.00234	1.5e-05	156
{'QU16'}	{'CHUF'}	6	0.0023307	1.9641e-05	118.67
{'QL8' }	{'CFIX'}	6	0.00234	2e-05	117
{'QL8' }	{'CHUF'}	6	0.0026955	1.3858e-05	194.5
{'QL16'}	{'CFIX'}	6	0.00244	1.5e-05	162.67
{'QL16'}	{'CHUF'}	6	0.0023274	2.1352e-05	109

Figure 7.3: Channel Coding Gain (BER uncoded and BER coded)