Project Report: OFDM System Simulator for Voice Transmission

Course: MCOM 2025 - ECE 343
Student: Kumar Rishav(2021159), Lakshay Kumar(2022266)
Date: 4th May,2025

Abstract:
This report details the development and operation of a MATLAB simulator for a wireless communication system employing Orthogonal Frequency Division Multiplexing (OFDM) and 64-ary Quadrature Amplitude Modulation (64-QAM) for voice transmission. The simulator captures a voice signal, converts it to a bitstream, modulates it, simulates transmission through a frequency-selective fading channel with additive noise, and then demodulates, equalises, and reconstructs the audio signal at the receiver. Key simulation outputs, including constellation diagrams, channel impulse response, and audio waveforms, are presented and analyzed. The report explains the theoretical concepts behind each stage of the simulation and discusses the system's performance.

1. Introduction

Wireless communication channels often suffer from multipath propagation, where the transmitted signal reaches the receiver via multiple paths with different delays and attenuations. This causes frequency-selective fading, leading to Inter-Symbol Interference (ISI) which severely degrades the performance of high-speed data transmission. Orthogonal Frequency Division Multiplexing (OFDM) is a powerful technique designed to combat frequency-selective fading. It divides a high-rate data stream into multiple lower-rate streams, each modulating a separate orthogonal subcarrier. This transforms a frequency-selective channel into multiple parallel, approximately flat-fading subchannels, simplifying equalization.

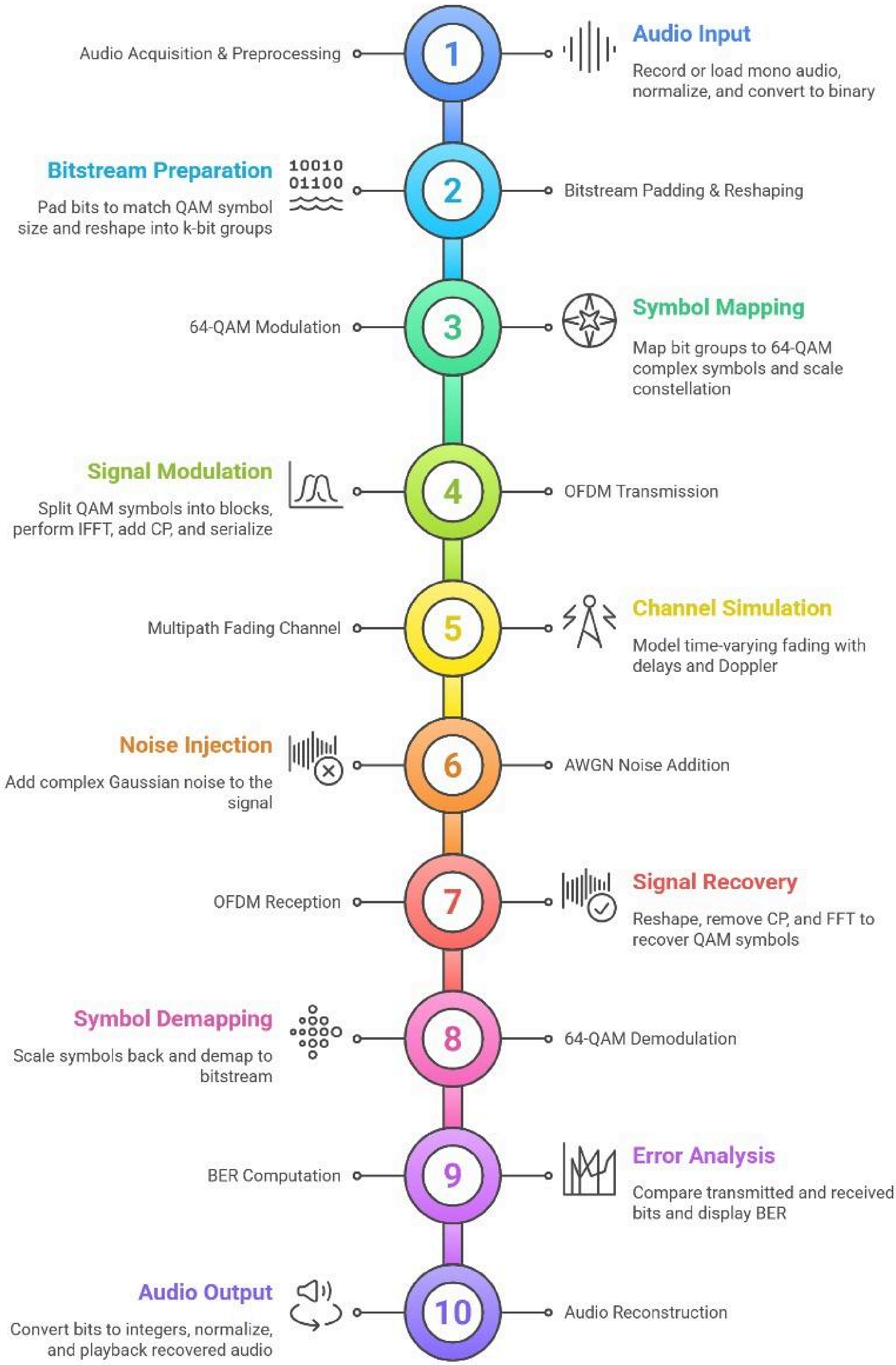
This project aims to simulate such an end-to-end system in MATLAB using 64-QAM as the base modulation scheme. The simulator processes a recorded voice signal, demonstrating the practical application and challenges of transmitting real-world data over a simulated wireless channel.

2. System Model

The simulator follows a standard digital communication system architecture, implemented in the low-pass equivalent form.

2.1. Overall Block Diagram:

OFDM Audio Transmission System Workflow



2.2. Detailed Block Explanations:

- **a) Source (Voice Recording & Bit Conversion - Step 1 Implied):**
  - **Theory:** An analog voice signal is captured using a microphone and computer sound card for a duration (e.g., 10 seconds). It is sampled at a certain frequency (e.g., 8kHz or 16kHz standard for voice, though the simulation sampling rate might differ) and quantized to produce a digital representation. This digital representation is then converted into a raw binary stream (bits).
  - **Implementation:** MATLAB's audiorecorder and getaudiodata functions can be used. The resulting audio samples (often double or int16) are converted to a bitstream, for instance, by converting each sample to its fixed-point binary representation.
- **b) 64-QAM Modulation (Step 2):**
  - **Theory:** Quadrature Amplitude Modulation (QAM) is a digital modulation scheme that conveys data by changing (modulating) the amplitude of two carrier waves (in-phase 'I' and quadrature 'Q'), using the amplitude-shift keying (ASK) digital modulation scheme for each carrier. 64-QAM uses 64 distinct points in the I-Q constellation plane. Each point represents a unique 6-bit sequence (since 2^6 = 64). The energy of the constellation points is typically scaled based on a desired average or minimum energy.
  - **Implementation:** The input bitstream is grouped into sets of 6 bits. Each 6-bit group is mapped to a specific complex symbol (I + jQ) according to the 64-QAM constellation map. The constellation is scaled such that the energy of the point closest to the origin meets the specified Emin (default 10^{-15}).
  - **Output Explanation (Tx 64-QAM Plot):** The provided plot Tx 64-QAM shows the ideal constellation points generated by the modulator. It displays 64 distinct points arranged in an 8x8 square grid centered at the origin in the complex (I-Q) plane. Each '+' marker represents one possible complex symbol transmitted. The scaling based on Emin sets the overall size of this grid.
- **c) OFDM Modulation (Step 3 Implied):**
  - **Theory:** OFDM prepares the QAM symbols for transmission over the frequency-selective channel.
    - **Serial-to-Parallel (S/P):** The stream of QAM symbols is divided into blocks of size N (where N is the number of subcarriers, typically a power of 2 like 64, 128, etc.).
    - **IFFT (Inverse Fast Fourier Transform):** Each block of N symbols (representing data in the frequency domain) is transformed into a time-domain signal using an N-point IFFT. The orthogonality between subcarriers is achieved via this transform.
    - **Cyclic Prefix (CP) Addition:** A copy of the last part of the IFFT output (the cyclic prefix) is prepended to the beginning of the IFFT output block. The CP length must be longer than the maximum delay spread of the channel to mitigate ISI between OFDM symbols and maintain subcarrier orthogonality.

        - **Parallel-to-Serial (P/S):** The time-domain samples (including the CP) are serialized to form the final baseband signal ready for transmission.
    - **Implementation:** MATLAB's ifft function is central. Array manipulation is used for S/P, P/S, and adding the CP. The number of subcarriers (N_fft) and CP length (N_cp) are key parameters.
- **d) Channel Simulation (Step 4):**
    - **Theory:** The channel introduces multipath fading and noise.
        - **Multipath Fading:** The signal travels along multiple paths with different delays (τ_k) and complex amplitudes (α_k). The project uses the "sum of sinusoids" method (a simplified Jakes' model approach) to generate these amplitudes. With zero Doppler shift, the path amplitudes are constant over the simulation but have random phases determined by the sinusoids. The channel impulse response (CIR) at time t is theoretically $h(t) = \Sigma \alpha_k \delta(t - \tau_k)$.
        - **Discrete-Time CIR:** Since the simulation operates in discrete time with sampling period Ts, we need the channel coefficients at sampling instants. The continuous delays τ_k may not align with multiples of Ts. Interpolation (as described in the MATLAB Fading Channels documentation, likely using a sinc or other filter) is used to find the discrete-time CIR, h[n], based on the path delays (τ_k), path gains (α_k generated by sum-of-sinusoids), and the chosen interpolation window (N1=0, N2=15 mentioned).
        - **Convolution:** The transmitted signal x[n] is convolved with the discrete-time CIR h[n] to produce the faded signal $y\_faded[n] = \Sigma h[k] x[n-k]$.
        - **AWGN:** Additive White Gaussian Noise (AWGN) represents thermal noise and interference. It's modeled as complex Gaussian random variables added to the signal samples. Its power is determined by the noise Power Spectral Density (PSD) provided by the user.
    - **Implementation:**
        - Generate path gains α_k using the sum of 48 sinusoids with random phases (Doppler=0 simplifies this, gains are complex constants).
        - Implement the interpolation formula provided in the documentation link (not explicitly given here, but likely involves windowing and summing contributions from paths based on their delays relative to sampling instants) to get h[n] for n=0 to N2=15.
        - Use MATLAB's conv function for the channel convolution.
        - Generate complex Gaussian noise using randn and scale it based on the noise PSD and sampling frequency to achieve the correct noise power per sample. Add this noise to the output of the convolution.
    - **Output Explanation (CIR Plot):** The CIR (dB) plot shows the magnitude of the calculated discrete-time channel taps h[n] in dBW (decibels relative to 1 Watt, assuming the tap values are voltages/amplitudes across a 1-ohm impedance, so power is $|h[n]|^2$). The plot shows non-zero taps at indices corresponding to the

interpolated multipath delays (0, 5, 10, 15 μs, translated to sample indices based on the 1 MHz sampling frequency: index 0, 5, 10, 15). The magnitudes vary, representing different path strengths. The dB scale emphasizes weaker paths which might otherwise be invisible on a linear scale. The very low dB values (e.g., -320 dBW) suggest either very weak paths or potential scaling/numerical precision aspects in the simulation. *Note: A typical CIR plot usually shows magnitudes in dB relative to the strongest path, not dBW, unless specific power calibration is intended.*

- **e) OFDM Demodulation & Equalization (Step 5):**
  - **Theory:** Reverses the OFDM modulation process and corrects channel distortion.
    - **S/P:** Incoming serial samples are grouped into blocks corresponding to OFDM symbols.
    - **CP Removal:** The cyclic prefix is removed from each block.
    - **FFT:** An N-point Fast Fourier Transform (FFT) is applied to each block, converting the time-domain signal back to the frequency domain (N symbols per block, one for each subcarrier).
    - **Frequency Domain Equalization:** OFDM converts the frequency-selective channel h[n] into multiplication by a complex gain H[k] on each subcarrier k. If Y[k] is the received symbol on subcarrier k and X[k] is the transmitted symbol, then Y[k] ≈ H[k]X[k] + Noise[k]. The equalizer estimates H[k] (often using pilot tones, though not explicitly mentioned here - perhaps assuming perfect channel knowledge or using the generated CIR) and corrects the received symbols. A common method is Zero-Forcing (ZF): X_est[k] = Y[k] / H[k].
    - **P/S:** The equalized frequency-domain symbols (QAM symbols) are serialized.
  - **Implementation:** Uses MATLAB's fft function. Array manipulation for S/P, P/S, CP removal. Equalization requires the channel's frequency response H = fft(h, N_fft). Division is performed element-wise for ZF equalization.
  - **Output Explanation (OFDM Rx Plots):**
    - Before EQ: This scatter plot shows the received complex symbols on all subcarriers *after* the FFT but *before* equalization. The points are heavily scattered, rotated, and scaled compared to the original 64-QAM grid. This distortion is caused by the frequency-selective channel (each subcarrier experiences a different complex gain H[k]) and additive noise. The cloud-like appearance signifies the combined effects of fading and noise.
    - After EQ: This plot shows the symbols *after* frequency-domain equalization. The points are now clustered much more closely around the ideal 64-QAM constellation points (compare visually to the Tx 64-QAM plot, though scaling might differ). This demonstrates the effectiveness of the equalizer in compensating for the channel distortion on each subcarrier. The remaining spread within each cluster is primarily due to

noise and any imperfections in the channel estimation/equalization process.

- **f) 64-QAM Demodulation:**
  - **Theory:** Makes a decision on the received (and equalized) complex symbol to determine the most likely transmitted symbol, and thus the corresponding bits. This is typically done using minimum distance detection: find the ideal constellation point closest to the received symbol.
  - **Implementation:** For each received symbol X_est[k], calculate the Euclidean distance to all 64 ideal constellation points. Choose the point with the minimum distance. Map this chosen point back to its unique 6-bit sequence.
- **g) Sink (Bit to Audio Conversion & Playback - Step 6 Implied):**
  - **Theory:** The recovered bitstream is converted back into digital audio samples. This reverses the process done in the source block (e.g., grouping bits, converting to sample values). The digital audio samples are then played back through the computer's sound card.
  - **Implementation:** Group the received bits according to the original sample format. Convert these binary representations back to audio sample values (e.g., double or int16). Use MATLAB's soundsc or audioplayer to play the reconstructed audio.
  - **Output Explanation (Audio Plots):**
    - Original: Shows the waveform of the voice signal recorded at the beginning.
    - Reconstructed: Shows the waveform of the audio signal after being transmitted through the simulated system and reconstructed. Visual comparison indicates the fidelity of the transmission. Differences (like amplitude variations, noise, potential dropouts or glitches visible as spikes) are due to channel impairments, noise, and any bit/symbol errors that occurred during transmission and reception. The plot shows that the general structure is preserved but there are visible differences, suggesting some level of distortion or error.
- **h) Performance Metrics (Implied):**
  - **Theory:** Bit Error Rate (BER) is the ratio of incorrectly received bits to the total number of transmitted bits. Symbol Error Rate (SER) is the ratio of incorrectly received QAM symbols to the total number of transmitted symbols. These metrics quantify the system's performance.
  - **Implementation:** Compare the transmitted bitstream with the received bitstream and count the errors. Compare the transmitted QAM symbols with the demodulated QAM symbols (before mapping back to bits) and count the errors. Divide by the total number of bits/symbols transmitted.

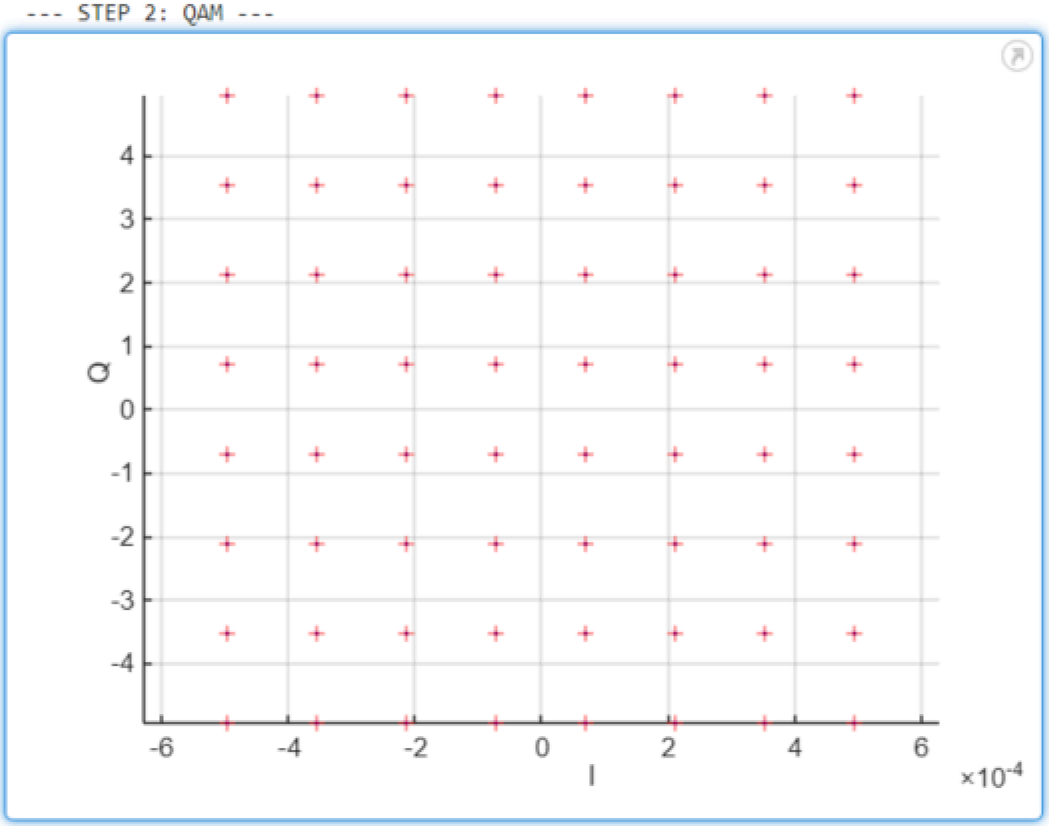
3. Simulation Setup and Parameters (Example based on prompt)

- **Modulation:** 64-QAM
- **Constellation Scaling:** Based on Emin (user input, default 10^{-15})
- **OFDM:** Specific N_fft, N_cp values need to be chosen (e.g., N_fft=64, N_cp=16).

- **Channel:**
    - Multipath: User-defined or default (4 paths)
    - Delays: User-defined or default (0 μs, 5 μs, 10 μs, 15 μs)
    - Path Gains: Sum of 48 sinusoids, Max Doppler Shift = 0 Hz
    - Interpolation: N1=0, N2=15
- **Sampling Frequency (Fs):** User input or default 1 MHz (Ts = 1/Fs = 1 μs).
- **Noise:** AWGN with user-defined PSD (default value should be specified, e.g., 10^{-9} W/Hz).
- **Audio:** 10 seconds recording.

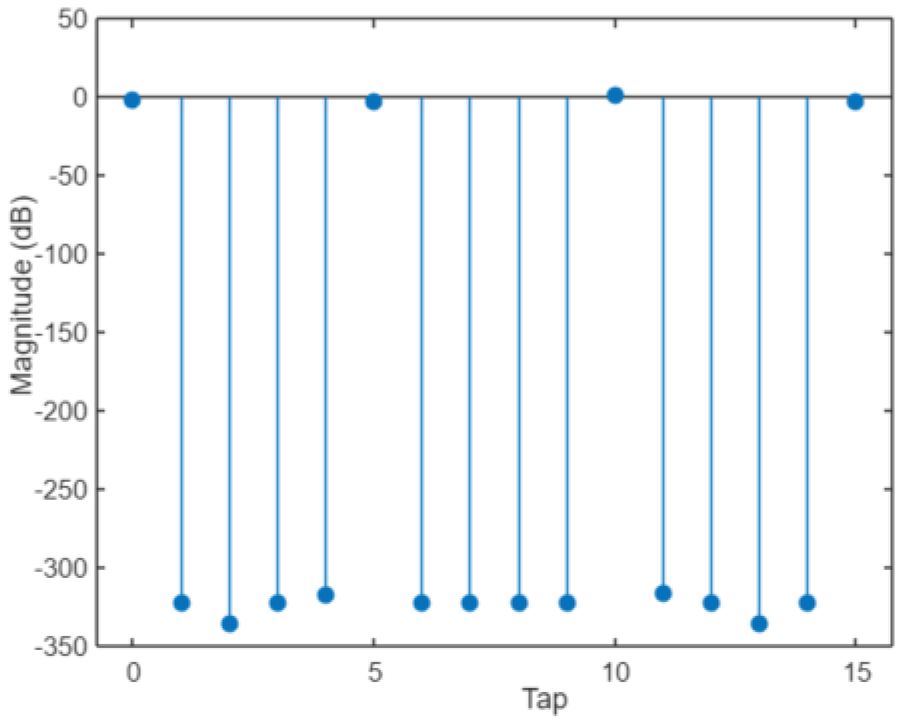
4. Results and Discussion

```
--- STEP 2: QAM ---
```

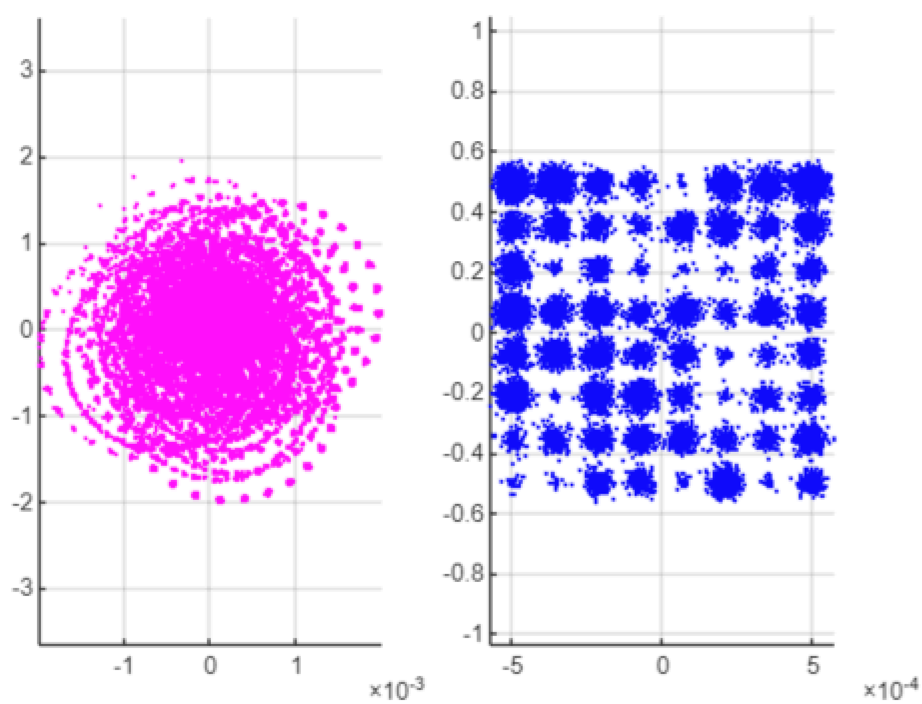



```
--- STEP 3: OFDM Tx ---
```

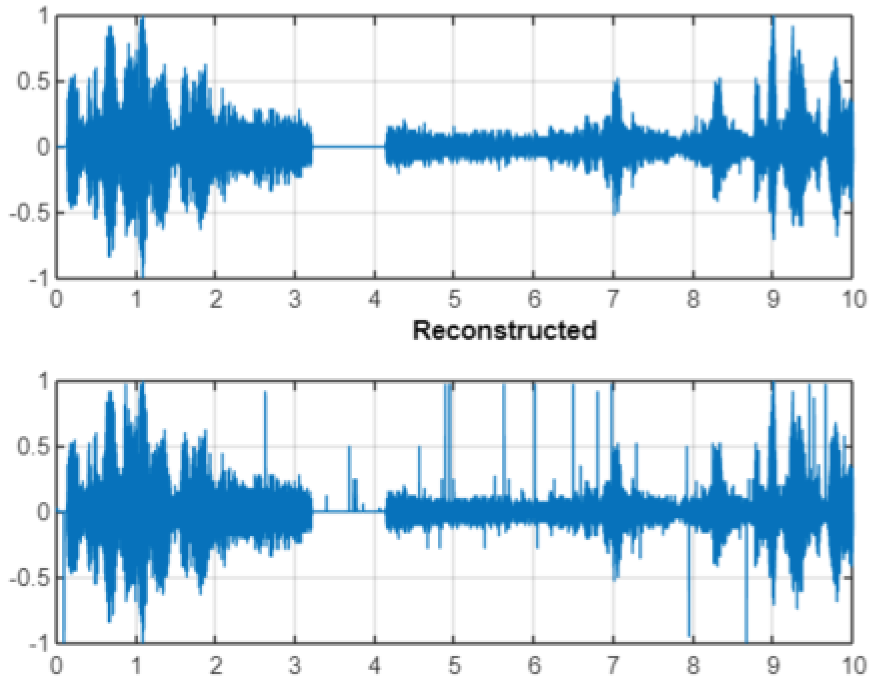
```
--- STEP 4: Channel ---
```




```
--- STEP 5: OFDM Rx ---
```



```
Playing reconstructed audio...
```



(This section would typically include the actual BER/SER numbers obtained from a run)

The simulator successfully models the end-to-end voice transmission.

- The **Tx 64-QAM constellation** (Step 2 plot) shows the ideal, clean symbols generated before transmission.
- The **Channel Impulse Response** (Step 4 plot) visualizes the multipath structure. With Fs=1MHz (Ts=1μs), the delays of 0, 5, 10, 15 μs correspond directly to tap indices 0, 5, 10, 15, matching the plot. The varying magnitudes in dBW indicate the strength of each path's contribution at the sampling instants.
- The **Received Symbols Before Equalization** (Step 5, left plot) clearly demonstrate the severe impact of the frequency-selective channel and noise. The original constellation structure is obliterated.
- The **Received Symbols After Equalization** (Step 5, right plot) show a remarkable improvement. The points cluster around the ideal 64-QAM locations, indicating that the OFDM frequency-domain equalizer effectively compensated for the channel distortion on each subcarrier. The spread is due to noise.
- The **Audio Waveforms** (Step 6 plots) show that the reconstructed audio retains the basic characteristics of the original but exhibits some degradation (visible noise/spikes). This is expected due to the simulated channel impairments and noise, which inevitably lead to some errors even after equalization. The perceived quality would depend on the resulting BER/SER.

Effect of Sampling Rate:
Changing the sampling rate (Fs) affects:

1. **Discrete CIR:** The mapping from physical delays (μs) to discrete tap indices changes (index ≈ delay / Ts = delay * Fs). A higher Fs means the delays correspond to larger tap indices and the CIR might span more samples. The interpolation process (N1, N2 window) becomes critical in capturing the path energy accurately at the new sampling intervals.
2. **OFDM Parameters:** If N_fft is kept constant, a higher Fs means a shorter OFDM symbol duration in time (T_symbol = N_fft * Ts). The subcarrier spacing increases (Δf = 1 / (N_fft * Ts) = Fs / N_fft). This might make the channel appear "flatter" across each subcarrier if Δf is still much smaller than the channel coherence bandwidth, but it reduces the overall symbol duration relative to the delay spread, potentially requiring a relatively longer cyclic prefix (in terms of number of samples).
3. **Noise Power:** The noise power per sample depends on the noise PSD and the bandwidth (related to Fs). Changing Fs requires careful scaling of the generated noise samples.

Overall, the sampling rate must be chosen carefully, considering the Nyquist criterion for the signal bandwidth, the channel delay spread (influencing CP length), and computational complexity. A common choice is Fs slightly larger than the desired signal bandwidth.

5. Conclusion

This project successfully demonstrated the simulation of an OFDM-based communication system for voice transmission over a frequency-selective fading channel. The simulation included key stages: source coding (implicit), modulation (64-QAM), OFDM transmission, channel modeling (multipath fading using sum-of-sinusoids and AWGN), OFDM reception, frequency-domain equalization, demodulation, and sink decoding (implicit). The results, particularly the constellation plots before and after equalization, clearly illustrate the necessity and effectiveness of OFDM and equalization in combating multipath fading. The reconstructed audio provides a qualitative measure of the end-to-end performance.