Performance Evaluation of Digital Modulation Schemes Over Diverse Wireless Channel Models

RM IPR PROJECT DETAILS INFO FILE

MAIN PAGES TO REFER

Page 2,3 ,10&11

CONTENTS

Research Paper
Techniques and channels
Outputs
Code

Performance Evaluation of Digital Modulation Schemes Over Diverse Wireless Channel Models

Research papers with links

1. Comparisons of PSK, APSK, and QAM over AWGN and fading channels(https://www.ewadirect.com/proceedings/ace/article/view/10070?)

2. A MATLAB Simulink Study on the Performance of QAM Modulation Scheme in AWGN, Rayleigh, and Rician Fading Channels: BER Analysis" (2023)

(https://archive.org/details/5575_20230609?)

3. Evaluating the BER Performance for M-ary QAM in AWGN, Rayleigh, Rician and Nakagami-m Fading Channels" (2023, IEEE ICCCNT)

4. Bit Error Rate Analysis of M-ARY PSK and M-ARY QAM Over Rician Fading Channel(https://arxiv.org/abs/2002.07392)

5. Evaluation of SNR for AWGN, Rayleigh & Rician Fading Channels Under DPSK(https://www.sciencepublishinggroup.com/article/10.11648/j.wcmc.20150301.12?)

Excel performance summary csv file is generated

and

heat map is used for easy visualization as line plot was messy

GUI interface for diff snr value

Recommended (IEEE-style) Logical Order by Complexity & Modulation Type:

🟦 **Phase Shift Keying (PSK):**

1. **BPSK** – baseline, most robust, lowest spectral efficiency
2. **QPSK** – standard in many systems like LTE
3. **OQPSK** – refinement of QPSK for spectral shaping
4. **π/4-QPSK** – differential phase QPSK, used in fading channels
5. **8PSK** – higher spectral efficiency than QPSK
6. **16PSK** – even higher PSK, less common due to BER

🟧 **Quadrature Amplitude Modulation (QAM):**

7. **64QAM** – widely used in LTE/Wi-Fi
8. **256QAM** – higher data rate, lower noise robustness

🟥 **Differential:**

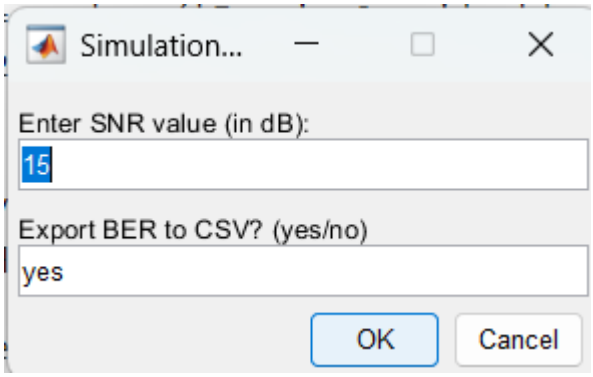
9. **DPSK** – differential encoding without coherent detection

🟪 **Multi-Carrier:**

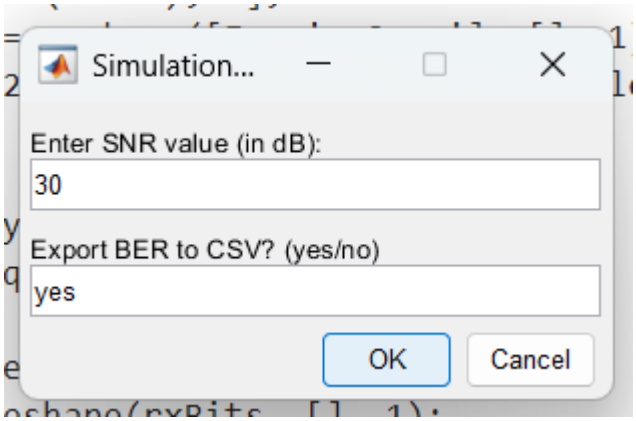
10. **OFDM** – robust multi-carrier scheme (can include QAM inside)

Channel Model	Key Feature	Application Area
AWGN	Noise only	Benchmark / theoretical analysis
Rayleigh	Multipath fading (no LOS)	Urban mobile, deep fading zones
Rician	Multipath + dominant LOS	Suburban, indoor LOS scenarios
Nakagami-m	Flexible multipath fading	Satellite, vehicular, general
Shadowed	Rayleigh + log-normal fading	Outdoor, rural, obstructed areas

Outputs



enter any snr value of choice



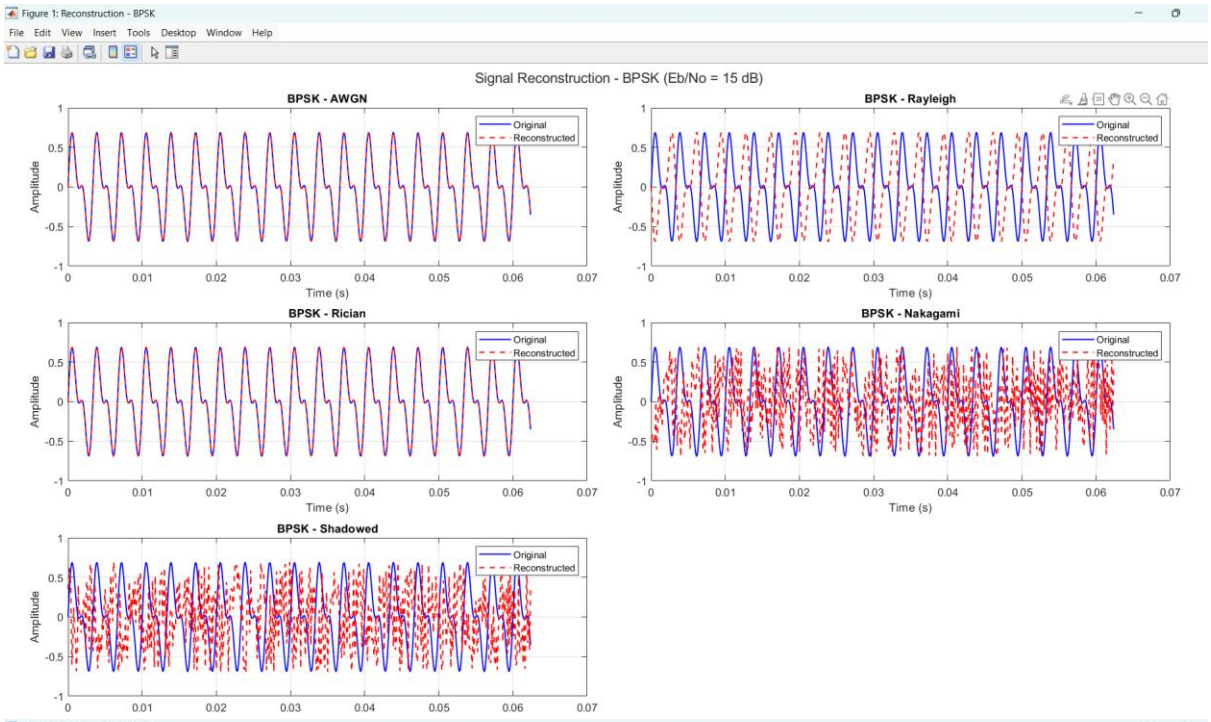


Figure 1: Reconstruction - BPSK
File Edit View Insert Tools Desktop Window Help
Signal Reconstruction - BPSK (Eb/No = 15 dB)
BPSK - AWGN
BPSK - Rayleigh
BPSK - Rician
BPSK - Nakagami
BPSK - Shadowed
Original
Reconstructed
Amplitude
Time (s)

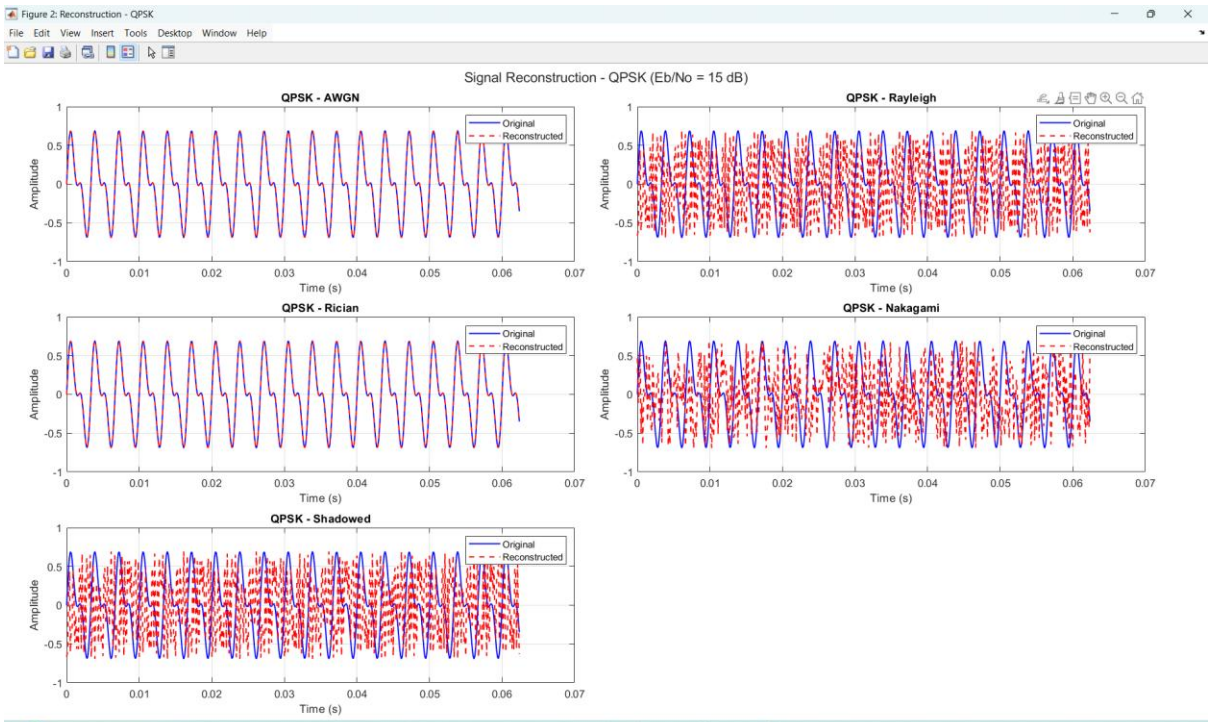


Figure 2: Reconstruction - QPSK
File Edit View Insert Tools Desktop Window Help
Signal Reconstruction - QPSK (Eb/No = 15 dB)
QPSK - AWGN
QPSK - Rayleigh
QPSK - Rician
QPSK - Nakagami
QPSK - Shadowed
Original
Reconstructed
Amplitude
Time (s)


22°C
Mostly clear
Search
ENG
IN
22:50
11-07-2025

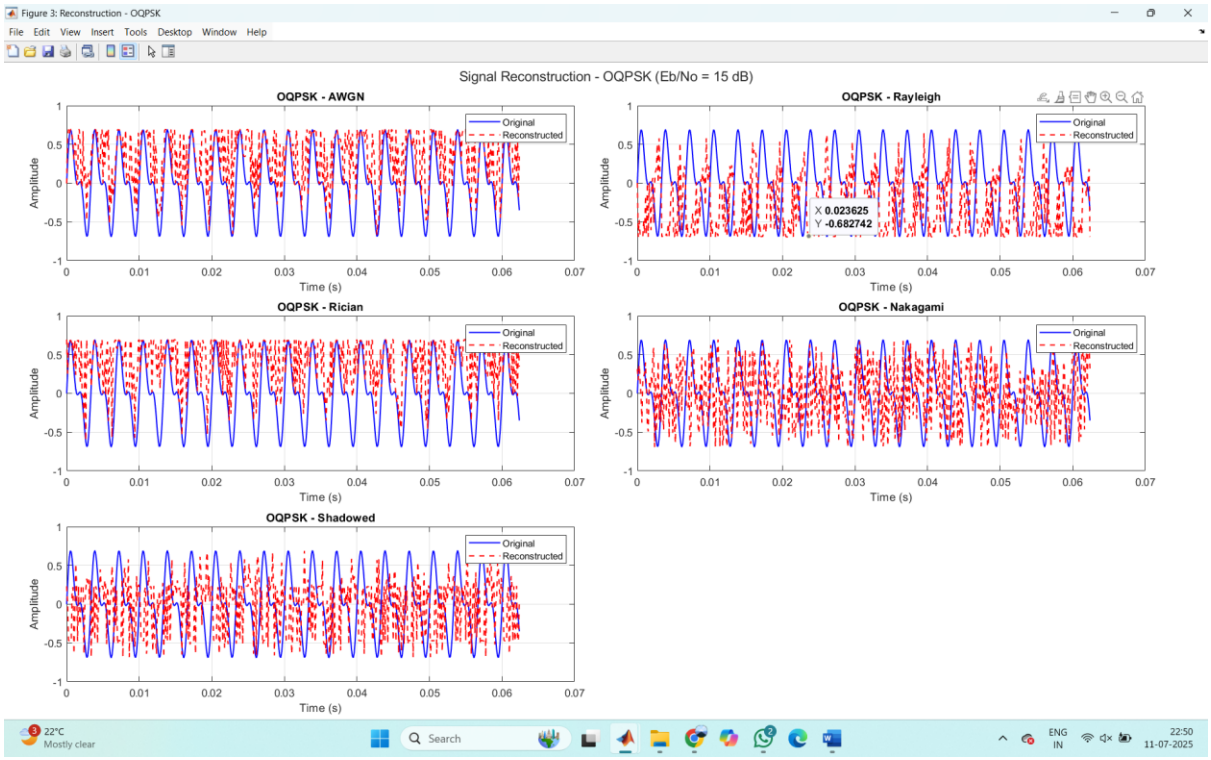


Figure 3: Reconstruction - OQPSK
File Edit View Insert Tools Desktop Window Help
Signal Reconstruction - OQPSK (Eb/No = 15 dB)
OQPSK - AWGN
OQPSK - Rayleigh
OQPSK - Rician
OQPSK - Nakagami
OQPSK - Shadowed
Original
Reconstructed
Amplitude
Time (s)
X 0.023625
Y -0.682742
22°C
Mostly clear
Search
ENG
IN
22:50
11-07-2025

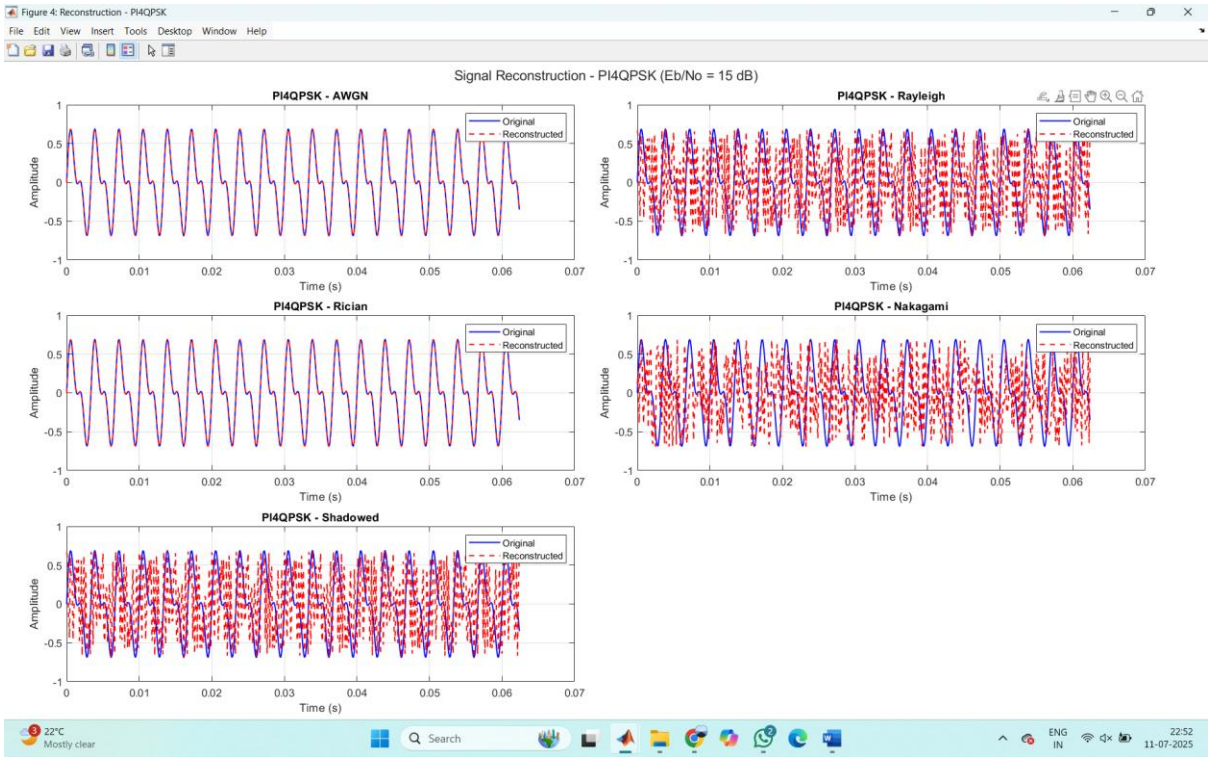


Figure 4: Reconstruction - PI4QPSK
File Edit View Insert Tools Desktop Window Help
Signal Reconstruction - PI4QPSK (Eb/No = 15 dB)
PI4QPSK - AWGN
PI4QPSK - Rayleigh
PI4QPSK - Rician
PI4QPSK - Nakagami
PI4QPSK - Shadowed
Original
Reconstructed
Amplitude
Time (s)
22°C
Mostly clear
Search
ENG
IN
22:52
11-07-2025

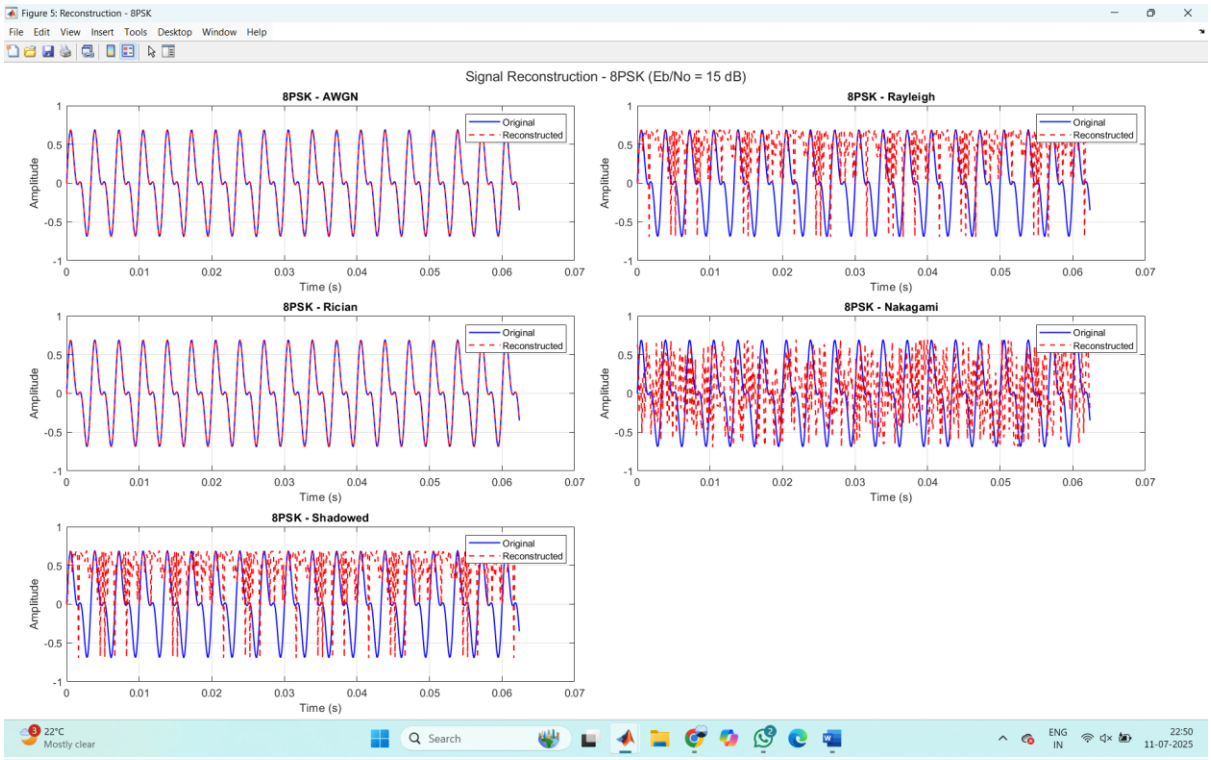


Figure 5: Reconstruction - 8PSK
File Edit View Insert Tools Desktop Window Help
Signal Reconstruction - 8PSK (Eb/No = 15 dB)
8PSK - AWGN
8PSK - Rayleigh
8PSK - Rician
8PSK - Nakagami
8PSK - Shadowed
Original
Reconstructed
Amplitude
Time (s)
22°C
Mostly clear
Search
ENG
IN
22:50
11-07-2025

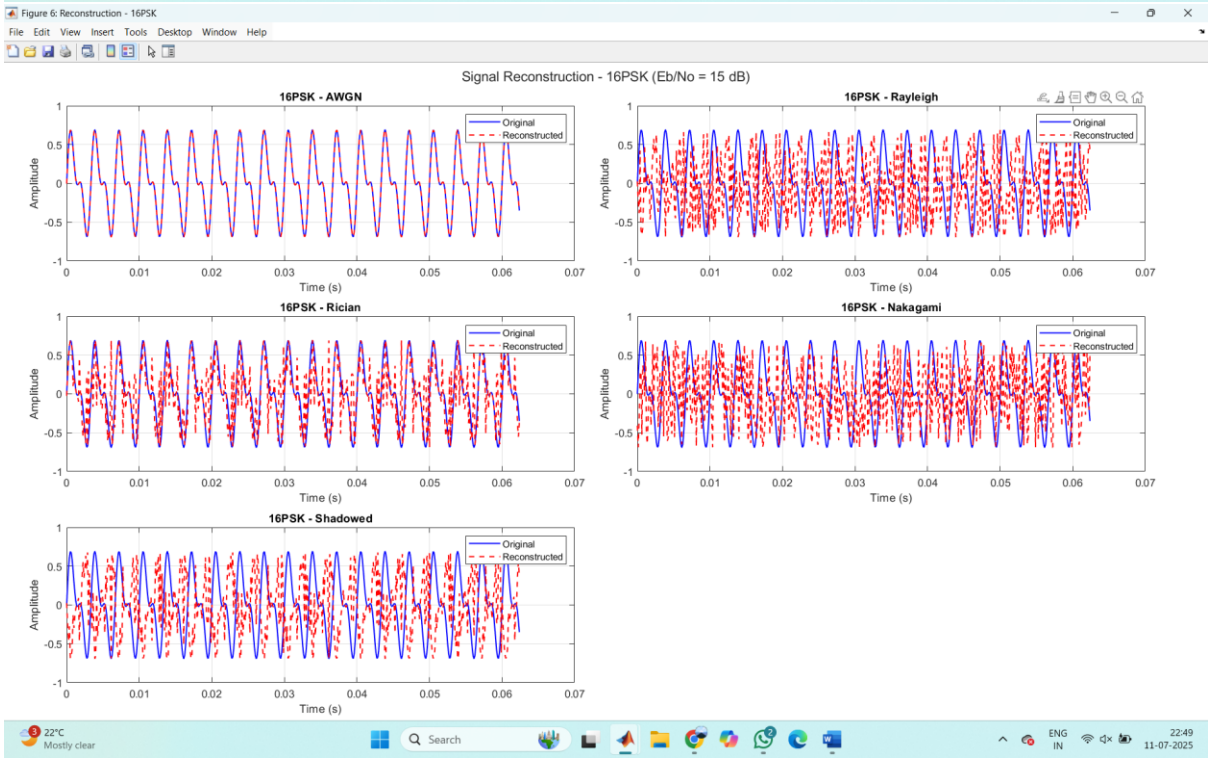


Figure 6: Reconstruction - 16PSK
File Edit View Insert Tools Desktop Window Help
Signal Reconstruction - 16PSK (Eb/No = 15 dB)
16PSK - AWGN
16PSK - Rayleigh
16PSK - Rician
16PSK - Nakagami
16PSK - Shadowed
Original
Reconstructed
Amplitude
Time (s)
22°C
Mostly clear
Search
ENG
IN
22:49
11-07-2025

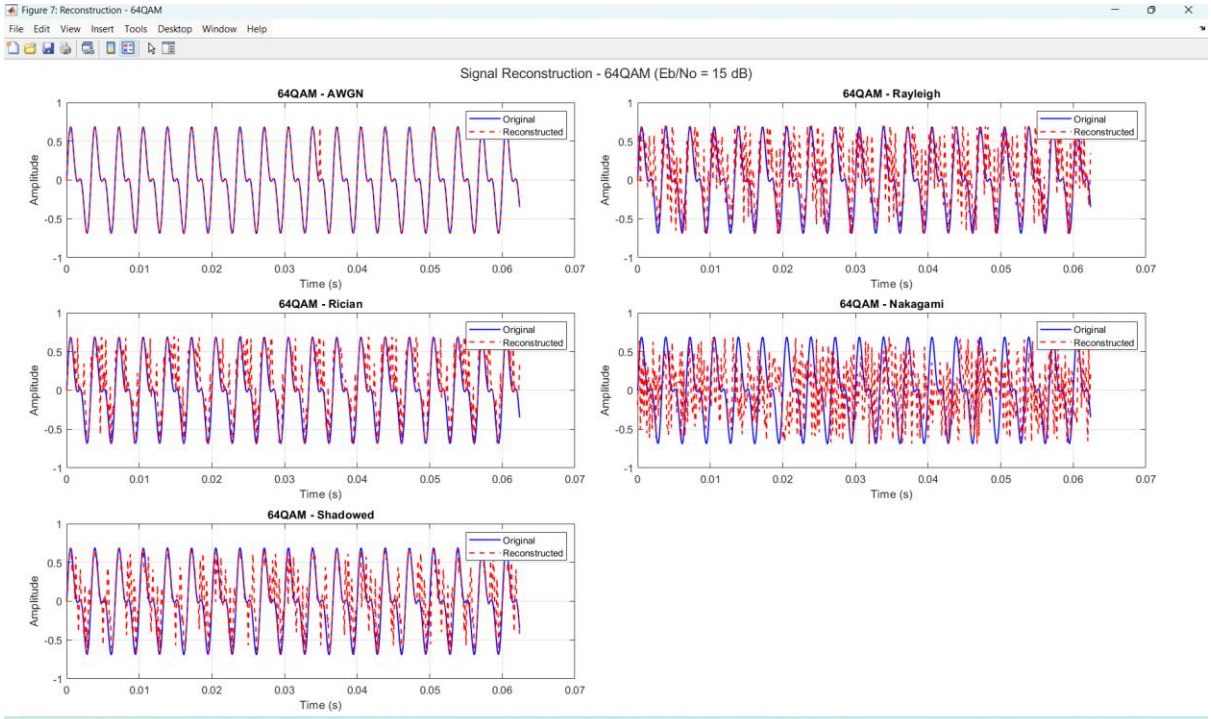


Figure 7: Reconstruction - 64QAM
File Edit View Insert Tools Desktop Window Help
Signal Reconstruction - 64QAM (Eb/No = 15 dB)
64QAM - AWGN
64QAM - Rayleigh
64QAM - Rician
64QAM - Nakagami
64QAM - Shadowed
Original
Reconstructed
Amplitude
Time (s)

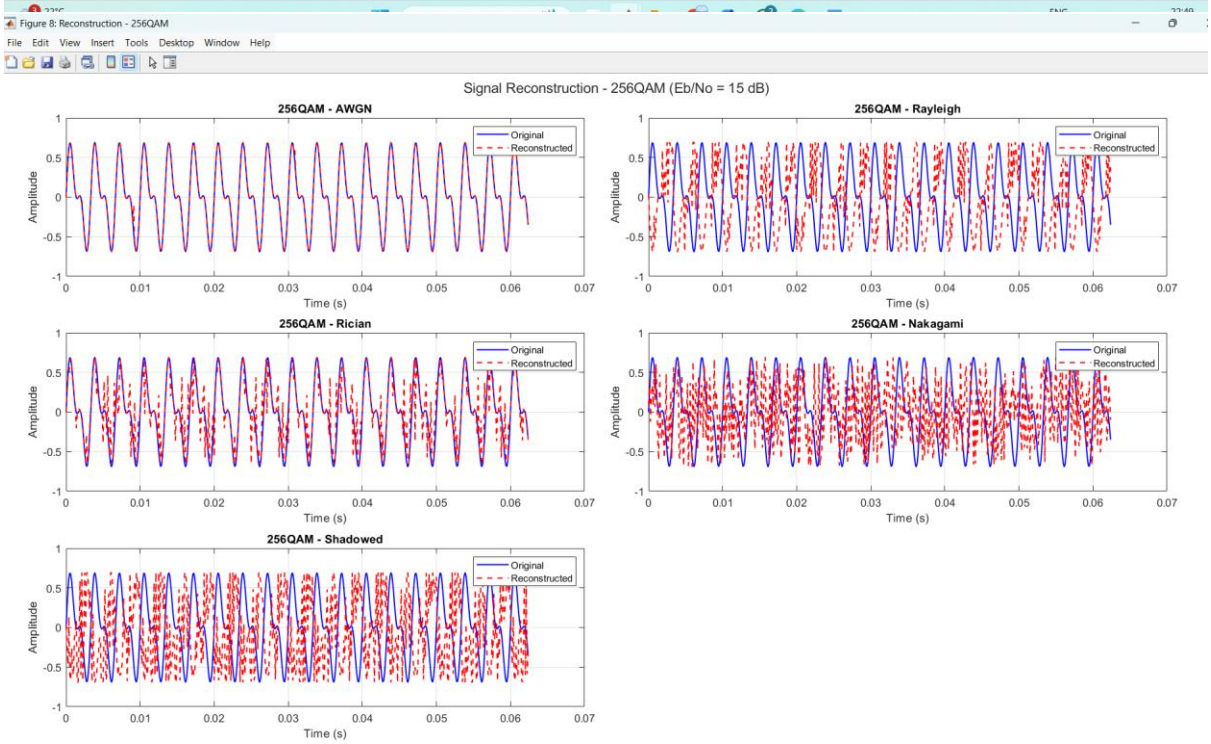


Figure 8: Reconstruction - 256QAM
File Edit View Insert Tools Desktop Window Help
Signal Reconstruction - 256QAM (Eb/No = 15 dB)
256QAM - AWGN
256QAM - Rayleigh
256QAM - Rician
256QAM - Nakagami
256QAM - Shadowed
Original
Reconstructed
Amplitude
Time (s)

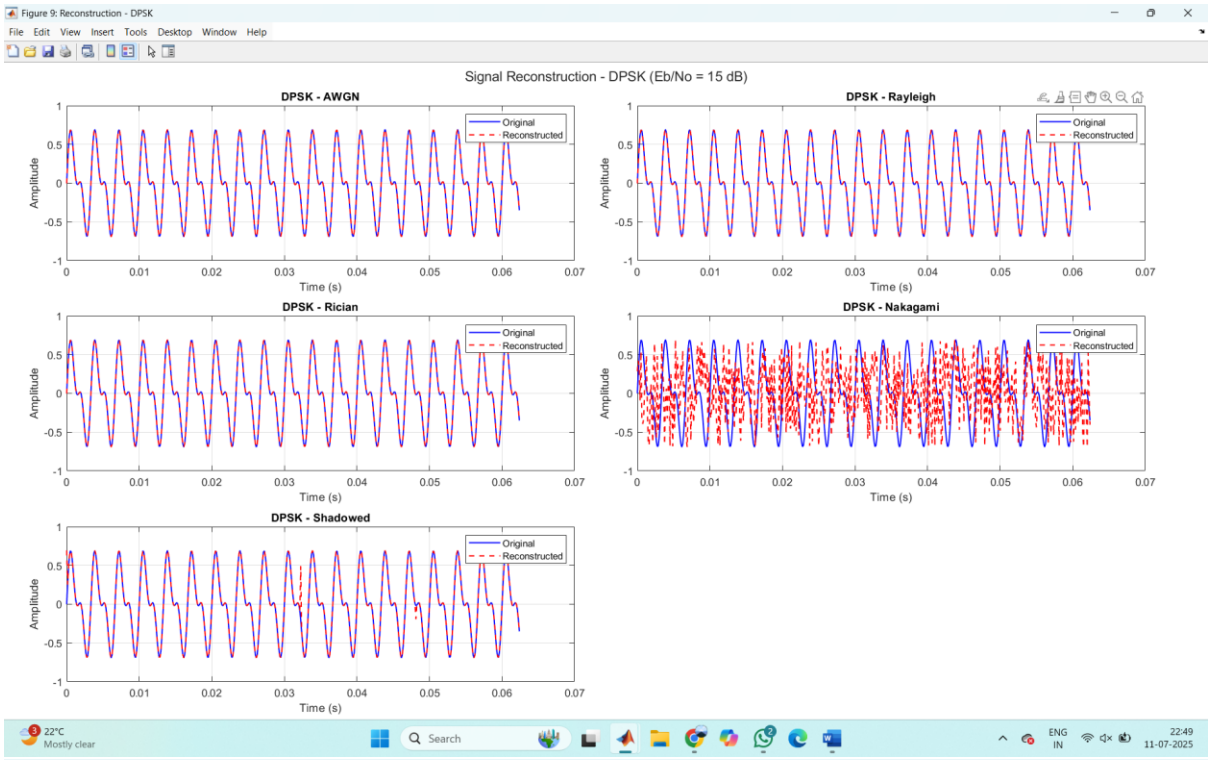


Figure 9: Reconstruction - DPSK
File Edit View Insert Tools Desktop Window Help
Signal Reconstruction - DPSK (Eb/No = 15 dB)
DPSK - AWGN
DPSK - Rayleigh
DPSK - Rician
DPSK - Nakagami
DPSK - Shadowed
Original
Reconstructed
Amplitude
Time (s)
22°C
Mostly clear
Search
ENG
IN
22:49
11-07-2025

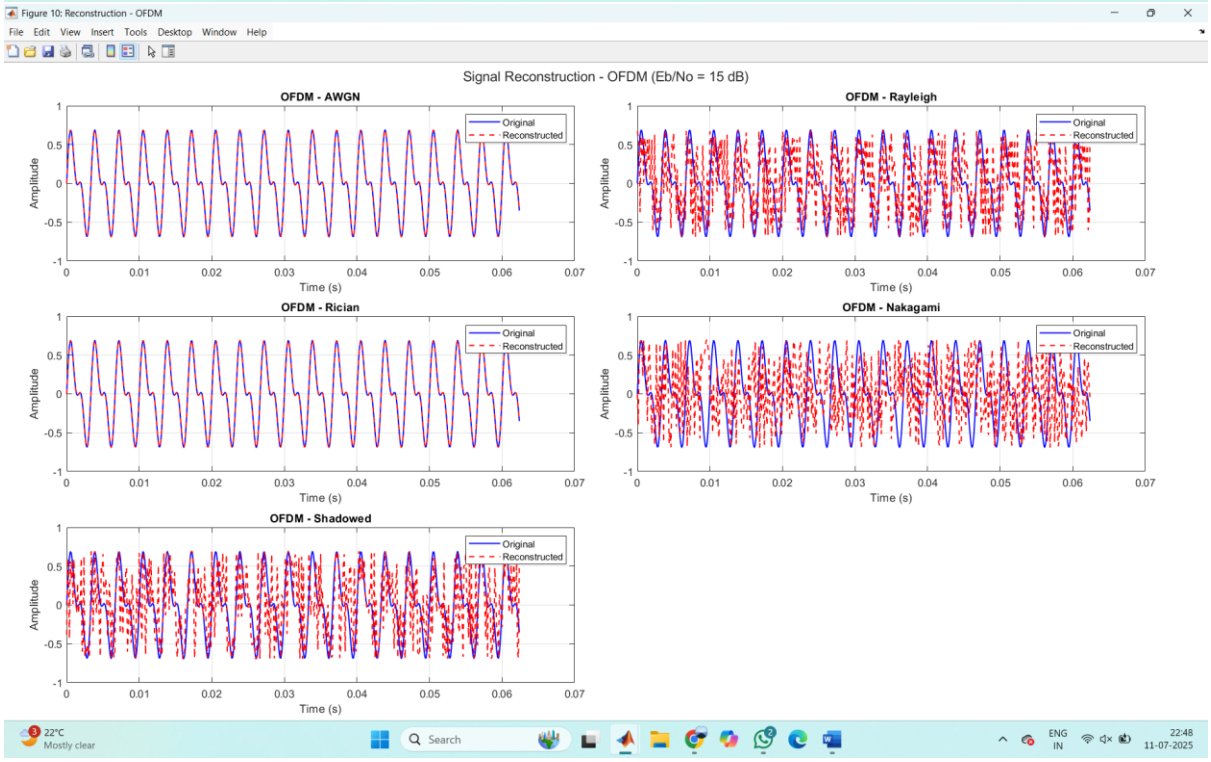


Figure 10: Reconstruction - OFDM
File Edit View Insert Tools Desktop Window Help
Signal Reconstruction - OFDM (Eb/No = 15 dB)
OFDM - AWGN
OFDM - Rayleigh
OFDM - Rician
OFDM - Nakagami
OFDM - Shadowed
Original
Reconstructed
Amplitude
Time (s)
22°C
Mostly clear
Search
ENG
IN
22:48
11-07-2025

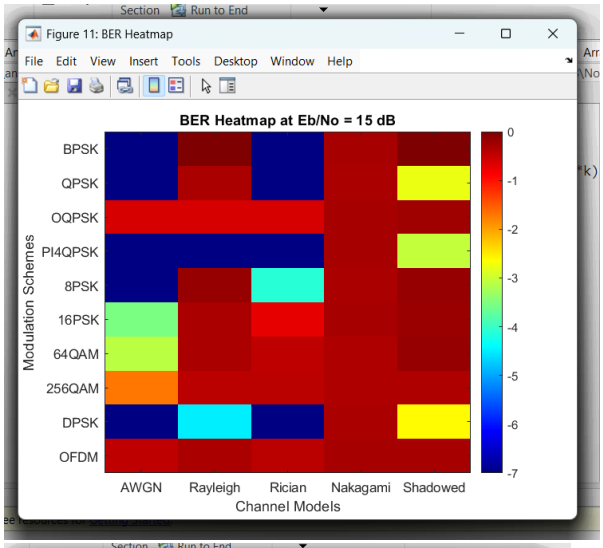


Figure 11: BER Heatmap
File Edit View Insert Tools Desktop Window Help
BER Heatmap at Eb/No = 15 dB
BPSK
QPSK
OQPSK
PI4QPSK
8PSK
16PSK
64QAM
256QAM
DPSK
OFDM
Modulation Schemes
AWGN Rayleigh Rician Nakagami Shadowed
Channel Models
0 -1 -2 -3 -4 -5 -6 -7

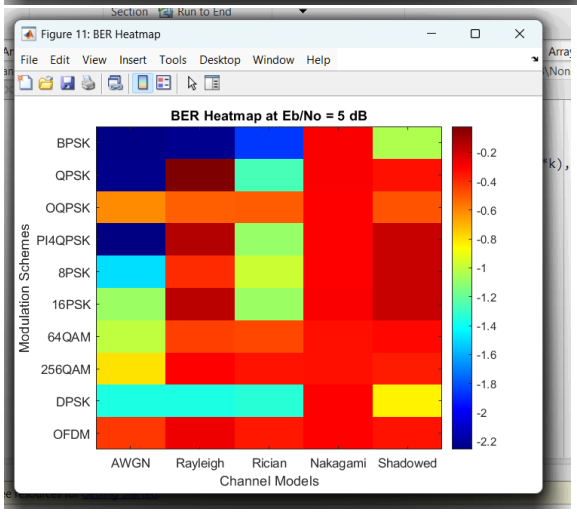


Figure 11: BER Heatmap
File Edit View Insert Tools Desktop Window Help
BER Heatmap at Eb/No = 5 dB
BPSK
QPSK
OQPSK
PI4QPSK
8PSK
16PSK
64QAM
256QAM
DPSK
OFDM
Modulation Schemes
AWGN Rayleigh Rician Nakagami Shadowed
Channel Models
-0.2 -0.4 -0.6 -0.8 -1 -1.2 -1.4 -1.6 -1.8 -2 -2.2

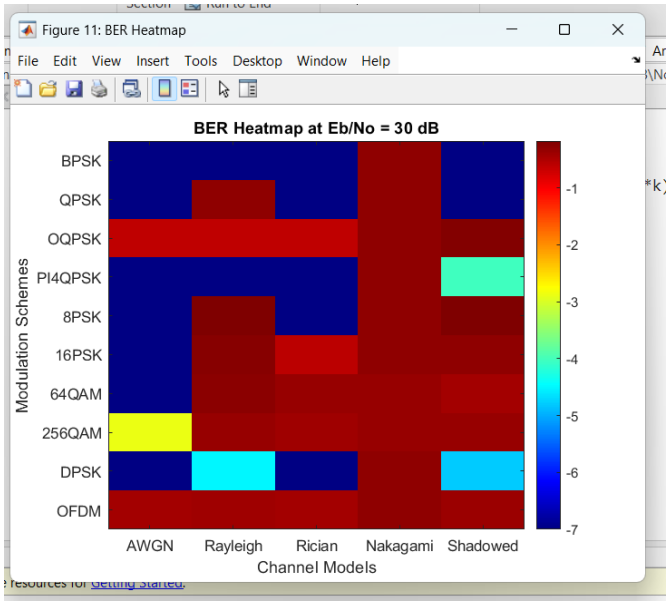
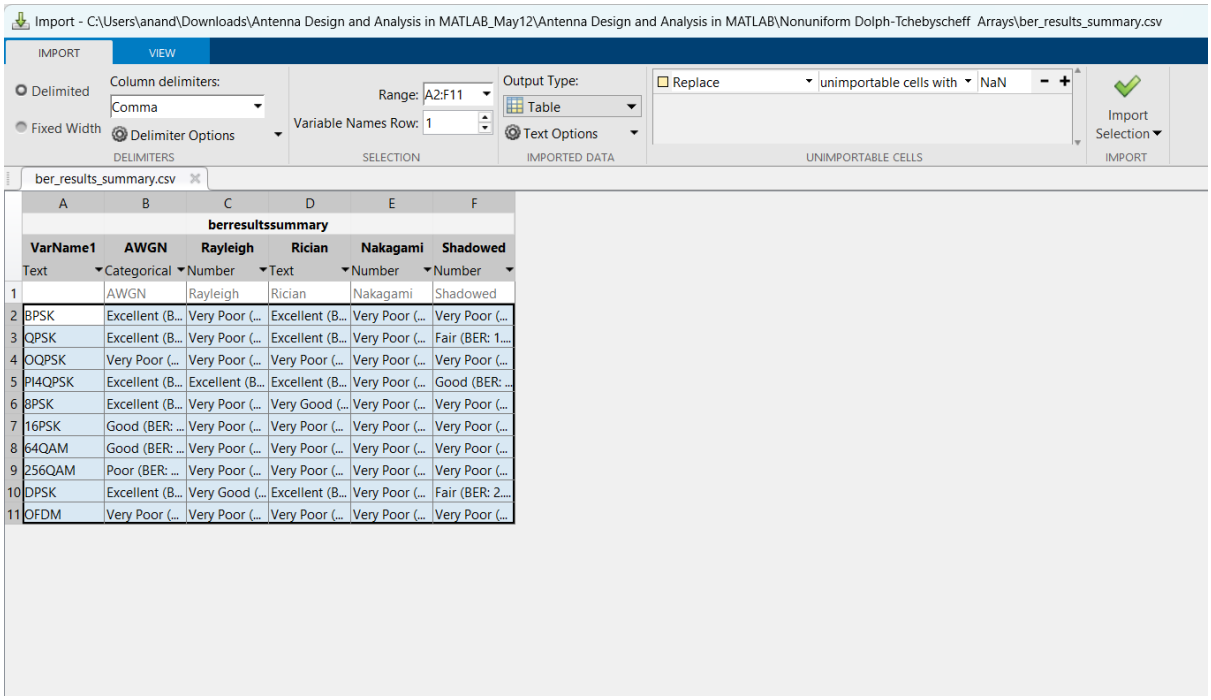


Figure 11: BER Heatmap
File Edit View Insert Tools Desktop Window Help
BER Heatmap at Eb/No = 30 dB
BPSK
QPSK
OQPSK
PI4QPSK
8PSK
16PSK
64QAM
256QAM
DPSK
OFDM
Modulation Schemes
AWGN Rayleigh Rician Nakagami Shadowed
Channel Models
-1 -2 -3 -4 -5 -6 -7



	A	B	C	D	E	F
	berresultssummary					
	VarName1	AWGN	Rayleigh	Rician	Nakagami	Shadowed
	Text	Categorical	Number	Text	Number	Number
1		AWGN	Rayleigh	Rician	Nakagami	Shadowed
2	BPSK	Excellent (B...	Very Poor (...	Excellent (B...	Very Poor (...	Very Poor (...
3	QPSK	Excellent (B...	Very Poor (...	Excellent (B...	Very Poor (...	Fair (BER: 1....
4	OQPSK	Very Poor (...	Very Poor (...	Very Poor (...	Very Poor (...	Very Poor (...
5	PI4QPSK	Excellent (B...	Excellent (B...	Excellent (B...	Very Poor (...	Good (BER: ...
6	8PSK	Excellent (B...	Very Poor (...	Very Good (...	Very Poor (...	Very Poor (...
7	16PSK	Good (BER: ...	Very Poor (...	Very Poor (...	Very Poor (...	Very Poor (...
8	64QAM	Good (BER: ...	Very Poor (...	Very Poor (...	Very Poor (...	Very Poor (...
9	256QAM	Poor (BER: ...	Very Poor (...	Very Poor (...	Very Poor (...	Very Poor (...
10	DPSK	Excellent (B...	Very Good (...	Excellent (B...	Very Poor (...	Fair (BER: 2....
11	OFDM	Very Poor (...	Very Poor (...	Very Poor (...	Very Poor (...	Very Poor (...

CODE

```
clc; clear; close all;


%% Parameters
fs = 8000;
t = 0:1/fs:1;
signal = 0.5*sin(2*pi*300*t) + 0.3*sin(2*pi*600*t); % Synthetic analog signal
signal_norm = (signal - min(signal)) / (max(signal) - min(signal));
quantized = uint8(signal_norm * 255);
bits_orig = reshape(de2bi(quantized, 8, 'left-msb')', [], 1);


% GUI prompt with error handling
defs = {'15','yes'};
answer = inputdlg({'Enter SNR value (in dB):','Export BER to CSV? (yes/no)'}, 'Simulation Parameters', [1 40], defs);
if isempty(answer) || numel(answer) < 2
    errordlg('Simulation canceled or invalid input.','Input Error');
    return;
end
EbNodB_plot = str2double(answer{1});
if isnan(EbNodB_plot)
    errordlg('Invalid Eb/No value. Please enter a numeric value.','Input Error');
    return;
end
saveCSV = strcmpi(answer{2},'yes');


EbNodB = EbNodB_plot;
channels = {'AWGN','Rayleigh','Rician','Nakagami','Shadowed'};
modSchemes = {'BPSK','QPSK','OQPSK','PI4QPSK','8PSK','16PSK','64QAM','256QAM','DPSK','OFDM'};
ber = zeros(length(modSchemes), length(channels), length(EbNodB));
```

```
rayleigh = comm.RayleighChannel('SampleRate',fs);
rician = comm.RicianChannel('SampleRate',fs,'KFactor',4);
nakagami_m = 2; shadow_sigma = 4;

summary = strings(length(modSchemes)+1, length(channels)+1);
summary(1,2:end) = string(channels);
summary(2:end,1) = string(modSchemes);

for m = 1:length(modSchemes)
    modType = modSchemes{m};
    useOFDM = strcmp(modType, 'OFDM');

    switch modType
        case 'BPSK', M = 2; phaseOffset = 0;
        case 'QPSK', M = 4; phaseOffset = pi/4;
        case 'OQPSK', M = 4;
        case 'PI4QPSK', M = 4; phaseOffset = pi/8;
        case '8PSK', M = 8; phaseOffset = 0;
        case '16PSK', M = 16; phaseOffset = 0;
        case '64QAM', M = 64;
        case '256QAM', M = 256;
        case 'DPSK', M = 4;
        case 'OFDM', M = 16; numSubcarriers = 64; cpLen = 16;
    end

    k = log2(M);
    snr = EbNodB + 10*log10(k);
    numBits = floor(length(bits_orig)/k)*k;
    if numBits == 0
        warning(['Skipping modulation scheme ' modType ' due to invalid k.']); continue;
```

```
end
bits = bits_orig(1:numBits);
symbols = bi2de(reshape(bits, [], k), 'left-msb');

if useOFDM
    modData = qammod(symbols, M, 'gray');
    padLen = mod(numel(modData), numSubcarriers);
    if padLen ~= 0
        modData(end+1:numel(modData)+numSubcarriers-padLen) = 0;
    end
    modData = reshape(modData, numSubcarriers, []);
    ifftSig = ifft(modData, numSubcarriers);
    modSig = [ifftSig(end-cpLen+1:end,:); ifftSig];
    modSig = modSig(:);
elseif ismember(modType, {'64QAM','256QAM'})
    modSig = qammod(symbols, M, 'gray');
elseif ismember(modType, {'BPSK','QPSK','8PSK','16PSK','PI4QPSK'})
    modSig = pskmod(symbols, M, phaseOffset, 'gray');
elseif strcmp(modType, 'DPSK')
    modSig = dpskmod(symbols, M);
elseif strcmp(modType, 'OQPSK')
    I = 2*bits(1:2:end)-1;
    Q = 2*bits(2:2:end)-1;
    Q = [0; Q(1:end-1)];
    modSig = double(I) + 1j*double(Q);
end

figure('Name',['Reconstruction - ' modType],'Color','w');
tiledlayout(3,2,'Padding','compact','TileSpacing','compact');

for c = 1:length(channels)
```

```
chType = channels{c};
switch chType
    case 'AWGN'
        rx = awgn(modSig, snr, 'measured');
    case 'Rayleigh'
        release(rayleigh);
        faded = rayleigh(modSig);
        faded = faded / sqrt(mean(abs(faded).^2));
        rx = awgn(faded, snr, 'measured');
    case 'Rician'
        release(rician);
        faded = rician(modSig);
        faded = faded / sqrt(mean(abs(faded).^2));
        rx = awgn(faded, snr, 'measured');
    case 'Nakagami'
        m_int = round(nakagami_m);
        g = sum(randn(m_int*2, numel(modSig)).^2, 1) / (2 * m_int);
        h = sqrt(g) .* exp(1j * 2 * pi * rand(1, numel(modSig)));
        faded = modSig .* h.';
        rx = awgn(faded, snr, 'measured');
    case 'Shadowed'
        release(rayleigh);
        shadow_dB = shadow_sigma * randn(size(modSig));
        faded = rayleigh(modSig) .* 10.^(shadow_dB/20);
        faded = faded / sqrt(mean(abs(faded).^2));
        rx = awgn(faded, snr, 'measured');
end

if useOFDM
    try
        rx = reshape(rx, numSubcarriers+cpLen, []);
```

```
        rx = rx(cpLen+1:end, :);
        rxFFT = fft(rx, numSubcarriers);
        demod = qamdemod(rxFFT(:), M, 'gray');
    catch
        continue;
    end
elseif ismember(modType, {'64QAM','256QAM'})
    demod = qamdemod(rx, M, 'gray');
elseif ismember(modType, {'BPSK','QPSK','8PSK','16PSK','PI4QPSK'})
    demod = pskdemod(rx, M, phaseOffset, 'gray');
elseif strcmp(modType, 'DPSK')
    demod = dpskdemod(rx, M);
elseif strcmp(modType, 'OQPSK')
    I_rx = real(rx) > 0;
    Q_rx = imag(rx) > 0;
    Q_rx = [Q_rx(2:end); 0];
    oqpskBits = reshape([I_rx'; Q_rx'], [], 1);
    demod = bi2de(reshape(oqpskBits(1:floor(length(oqpskBits)/k)*k), [], k), 'left-msb');
end

if strcmp(modType,'OQPSK')
    rxBits = oqpskBits;
else
    rxBits = de2bi(demod, k, 'left-msb');
    rxBits = reshape(rxBits, [], 1);
end

minLen = min(length(bits), length(rxBits));
berVal = mean(bits(1:minLen) ~= rxBits(1:minLen));
if berVal == 0, berVal = 1e-7; end
ber(m,c,1) = berVal;
```

```
        if berVal < 1e-5
            level = "Excellent";
        elseif berVal < 1e-4
            level = "Very Good";
        elseif berVal < 1e-3
            level = "Good";
        elseif berVal < 1e-2
            level = "Fair";
        elseif berVal < 1e-1
            level = "Poor";
        else
            level = "Very Poor";
        end
        summary(m+1,c+1) = level + " (BER: " + sprintf('%.2e', berVal) + ")";

        rxBits = rxBits(1:floor(length(rxBits)/8)*8);
        if isempty(rxBits), continue; end
        rxQuant = bi2de(reshape(rxBits, 8, [])', 'left-msb');
        signalRec = double(rxQuant)/255;
        signalRec = signalRec*(max(signal)-min(signal)) + min(signal);
        L = min(500,length(signalRec));
        nexttile;
        plot(t(1:L),signal(1:L),'b-', 'LineWidth',1.2); hold on;
        plot(t(1:L),signalRec(1:L),'r--','LineWidth',1.2);
        title([modType ' - ' chType],'Interpreter','none','FontWeight','bold');
        xlabel('Time (s)'); ylabel('Amplitude');
        legend('Original','Reconstructed'); grid on;
    end
    sgtitle(['Signal Reconstruction - ' modType ' (Eb/No = ' num2str(EbNodB_plot) ' dB)']);
end
```

```
if saveCSV
    writecell(cellstr(summary), 'ber_results_summary.csv');
end

figure('Name','BER Heatmap','Color','w');
logBER = log10(squeeze(ber(:,:,1)));
imagesc(logBER);
colormap(jet); colorbar;
xticks(1:length(channels)); xticklabels(channels);
yticks(1:length(modSchemes)); yticklabels(modSchemes);
xlabel('Channel Models'); ylabel('Modulation Schemes');
title(['BER Heatmap at Eb/No = ' num2str(EbNodB_plot) ' dB']);
```