## MATLAB CODE FOR PAPR REDUCTION USING CLIPPING AND FILTERING TECHNIQUE

clc; clear; close all;

%% Parameters

N = 64; % Number of subcarriers

M = 16; % 16-QAM modulation

CP = N/4; % Cyclic prefix length

numSymbols = 1000; % Number of OFDM symbols

clipRatio = 0.8; % Clipping ratio (A/sigma)

%% Generate 16-QAM OFDM Signal

data = randi([0 M-1], N, numSymbols);

modData = qammod(data, M, 'UnitAveragePower', true);

ifftData = ifft(modData, N); % IFFT operation

%% Compute PAPR Before Clipping

originalOFDM = [ifftData(end-CP+1:end, :); ifftData]; % Add CP

PAPR\_original = 10 \* log10(max(abs(originalOFDM(:)).^2) / mean(abs(originalOFDM(:)).^2));

%% Clipping

sigma = sqrt(mean(abs(ifftData(:)).^2)); % RMS value

A = clipRatio \* sigma; % Clipping threshold

clippedSignal = ifftData;

clippedSignal(abs(clippedSignal) > A) = A .\* exp(1j \* angle(clippedSignal(abs(clippedSignal) > A)));

%% Filtering (Removing Out-of-Band Components)

clippedFFT = fft(clippedSignal, N); % FFT

filteredFFT = clippedFFT .\* (abs(clippedFFT) > 0.01 \* max(abs(clippedFFT))); % Simple Thresholding

filteredSignal = ifft(filteredFFT, N); % IFFT to get time-domain signal

%% Compute PAPR After Clipping & Filtering

clippedOFDM = [filteredSignal(end-CP+1:end, :); filteredSignal]; % Add CP

PAPR\_clipped = 10 \* log10(max(abs(clippedOFDM(:)).^2) / mean(abs(clippedOFDM(:)).^2));

%% Display Results

disp(['Original PAPR (dB): ', num2str(mean(PAPR\_original))]);

disp(['PAPR After Clipping & Filtering (dB): ', num2str(mean(PAPR\_clipped))]);

%% Plot CCDF

papr\_original\_db = 10 \* log10(abs(originalOFDM(:)).^2 / mean(abs(originalOFDM(:)).^2));

papr\_clipped\_db = 10 \* log10(abs(clippedOFDM(:)).^2 / mean(abs(clippedOFDM(:)).^2));

[ccdf\_original, papr\_values] = ecdf(papr\_original\_db);

[ccdf\_clipped, ~] = ecdf(papr\_clipped\_db);

figure;

semilogy(papr\_values, 1-ccdf\_original, 'b', 'LineWidth', 2); hold on;

semilogy(papr\_values, 1-ccdf\_clipped, 'r', 'LineWidth', 2);

xlabel('PAPR (dB)'); ylabel('CCDF');

legend('Original', 'Clipped & Filtered');

title('PAPR Reduction using Clipping and Filtering');

grid on;

# MATLAB CODE FOR REDUCTION OF PAPR USING SLM TECHNIQUE

clc; clear; close all;

% Parameters

N = 64; % Number of subcarriers

U = 4; % Number of phase sequences for SLM

num\_symbols = 1000; % Number of OFDM symbols

M = 16; % 16-PAM Modulation Order

% Generate 16-PAM Modulated Data

pam\_levels = -15:2:15; % 16-PAM levels

data = pam\_levels(randi([1 M], num\_symbols, N)); % Random 16-PAM symbols

% Perform IFFT to get the OFDM signal

ofdm\_symbols = ifft(data, [], 2);

% Compute PAPR without SLM

papr\_original = zeros(1, num\_symbols);

for i = 1:num\_symbols

peak\_power = max(abs(ofdm\_symbols(i, :)).^2);

avg\_power = mean(abs(ofdm\_symbols(i, :)).^2);

papr\_original(i) = 10 \* log10(peak\_power / avg\_power);

end

% Generate Random Phase Sequences for SLM

phase\_sequences = exp(1j \* 2 \* pi \* rand(U, N));

% Apply SLM and Compute PAPR

papr\_slm = zeros(1, num\_symbols);

for i = 1:num\_symbols

min\_papr = inf;

for u = 1:U

% Apply Phase Sequence

modified\_signal = ifft(data(i, :) .\* phase\_sequences(u, :));

peak\_power = max(abs(modified\_signal).^2);

avg\_power = mean(abs(modified\_signal).^2);

papr\_value = 10 \* log10(peak\_power / avg\_power);

% Select the signal with the lowest PAPR

if papr\_value < min\_papr

min\_papr = papr\_value;

end

end

papr\_slm(i) = min\_papr;

end

% Plot CCDF of PAPR

figure;

hold on;

grid on;

cdfplot(papr\_original);

cdfplot(papr\_slm);

set(gca, 'YScale', 'log'); % Convert to log scale for CCDF representation

xlabel('PAPR (dB)');

ylabel('CCDF');

title('PAPR Reduction using SLM in 16-PAM OFDM');

legend('Without SLM', 'With SLM');

hold off;