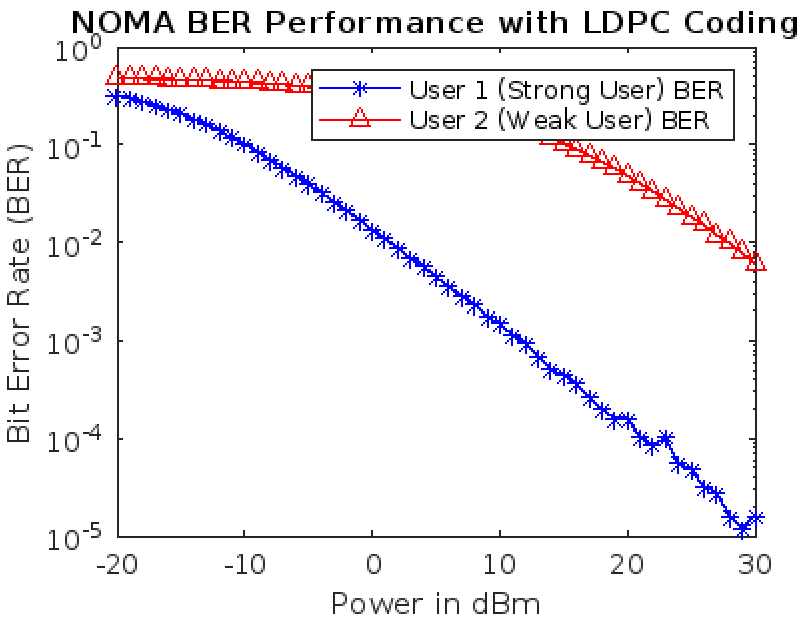
```
N = 1000; % Number of bits per user
D = [200 1000]; % Distance matrix for two users
C = [0.3 0.7]; % Power coefficients for two users
PT = -20:1:30; % Transmitted power in dBm
pt = (10^-3) * 10.^(PT / 10); % Transmitted power in linear scale
BW = 10^6; % System bandwidth
No = -174 + 10 * log10(BW); % Noise power (dBm)
no = (10^-3) * 10.^(No / 10); % Noise power in linear scale
BER1 = []; % Will contain all BER for User 1 for all power values
BER2 = []; % Will contain all BER for User 2 for all power values
R = 500; % Number of iterations for all SNR values

% Binary data creation for two users
DATA1 = randi([0 1], 1, N); % Data for User 1
DATA2 = randi([0 1], 1, N); % Data for User 2

% Modulation
MOD1 = pskmod(DATA1, 2); % Modulated signal for User 1
MOD2 = pskmod(DATA2, 2); % Modulated signal for User 2

% Loop over the transmitted power levels
for i = 1:length(PT)
    ber1 = 0;
    ber2 = 0;

    % Iterations for each SNR value
    for j = 1:R
        % Channel design for two users
        h1 = sqrt(D(1)^(-4)) * (randn(1, N) + 1j * randn(1, N)) / sqrt(2);
        h2 = sqrt(D(2)^(-4)) * (randn(1, N) + 1j * randn(1, N)) / sqrt(2);
        w1 = sqrt(no) * (randn(1, N) + 1i * randn(1, N)) / sqrt(2);
        w2 = sqrt(no) * (randn(1, N) + 1i * randn(1, N)) / sqrt(2);

        % Superposition coding
        T = sqrt(pt(i)) * (MOD1 * sqrt(C(1)) + MOD2 * sqrt(C(2))); % Ensure
size matches h1 and w1

        % Received signals at User 1 and User 2
        y1 = T .* h1 + w1;
        y2 = T .* h2 + w2;

        % User 1 reception
        newy11 = y1 ./ h1; % Channel equalization
        dec11 = pskdemod(newy11, 2); % Demodulation
        newmod11 = pskmod(dec11, 2); % Remodulation for SIC (Successive
Interference Cancellation)
        newy12 = newy11 - sqrt(C(2) * pt(i)) * newmod11; % SIC
        decy1 = pskdemod(newy12, 2); % Final demodulation
        ber1 = ber1 + nnz(decy1 ~= DATA1); % Count bit errors for User 1

        % User 2 reception
        y2 = y2 ./ h2; % Channel equalization
        dec21 = pskdemod(y2, 2); % Demodulation for User 2
        ber2 = ber2 + nnz(dec21 ~= DATA2); % Count bit errors for User 2
    end

    % Compute BER for User 1 and User 2
    BER1 = [BER1, ber1 / (R * N)];
    BER2 = [BER2, ber2 / (R * N)];
end

% Plot results
figure;
semilogy(PT, BER1, '-*b', 'LineWidth', 1);
hold on;
semilogy(PT, BER2, '-^r', 'LineWidth', 1);
xlabel('Power in dBm');
ylabel('Bit Error Rate (BER)');
legend('User 1 (Strong User) BER', 'User 2 (Weak User) BER');
title('NOMA BER Performance with LDPC Coding');
```



Published with MATLAB® R2024b