**Matlab code for "Noise Reduction Using Optimal Estimation”**

% Set up recording parameters

fs = 44100; % Sampling frequency (e.g., CD quality)

bitsPerSample = 16; % Bits per sample

numChannels = 1; % Number of audio channels (mono)

recordingDuration = 5; % Duration of recording in seconds

% Create an audio recorder object

recObj = audiorecorder(fs, bitsPerSample, numChannels);

% Record audio

disp('Start speaking.')

recordblocking(recObj, recordingDuration);

disp('End of recording.');

% Store recorded audio data

audioData = getaudiodata(recObj);

% Play back recorded audio

play(recObj);

% Define Kalman filter parameters

processNoise = 1e-5;

measurementNoise = 0.1;

% Apply Kalman filter for noise reduction

kalmanFilteredData = zeros(size(audioData));

x\_est = audioData(1);

P\_est = 1;

for i = 1:length(audioData)

% Prediction

x\_pred = x\_est;

P\_pred = P\_est + processNoise;

% Update

K = P\_pred / (P\_pred + measurementNoise);

x\_est = x\_pred + K \* (audioData(i) - x\_pred);

P\_est = (1 - K) \* P\_pred;

kalmanFilteredData(i) = x\_est;

end

% Plot the recorded audio waveform before and after noise reduction

subplot(2,1,1);

plot(audioData);

xlabel('Sample');

ylabel('Amplitude');

title('Recorded Audio Waveform (Before Noise Reduction)');

subplot(2,1,2);

plot(kalmanFilteredData);

xlabel('Sample');

ylabel('Amplitude');

title('Recorded Audio Waveform (After Noise Reduction)');

% Save the recorded audio with noise reduction to a file

audiowrite('recorded\_audio\_noise\_reduced.wav', kalmanFilteredData, fs); % Create an audio player object for the Kalman-filtered audio

player = audioplayer(kalmanFilteredData, fs);

% Play the Kalman-filtered audio

play(player);

% Time domain plot of recorded audio signal

figure;

subplot(2,2,1);

plot((1:length(audioData)) / fs, audioData);

xlabel('Time (s)');

ylabel('Amplitude');

title('Recorded Audio Message (Time Domain)');

% Frequency domain plot of recorded audio signal

subplot(2,2,2);

frequencies = linspace(-fs/2, fs/2, length(audioData));

audioData\_fft = fftshift(fft(audioData));

plot(frequencies, abs(audioData\_fft));

xlabel('Frequency (Hz)');

ylabel('Magnitude');

title('Recorded Audio Message (Frequency Domain)');

xlim([-fs/2, fs/2]);

% Time domain plot of Kalman filtered audio signal

subplot(2,2,3);

plot((1:length(kalmanFilteredData)) / fs, kalmanFilteredData);

xlabel('Time (s)');

ylabel('Amplitude');

title('Kalman Filtered Audio Message (Time Domain)');

% Frequency domain plot of Kalman filtered audio signal

subplot(2,2,4);

kalmanFilteredData\_fft = fftshift(fft(kalmanFilteredData));

plot(frequencies, abs(kalmanFilteredData\_fft));

xlabel('Frequency (Hz)');

ylabel('Magnitude');

title('Kalman Filtered Audio Message (Frequency Domain)');

xlim([-fs/2, fs/2]);