Exp 5

Aim :  
  
Denoising the ECG signal using Discrete Wavelet Transform (DWT) .  
  
Theory :

ECG signal denoising using DWT decomposes the signal into frequency components, isolating noise primarily within detail coefficients. Thresholding these detail coefficients eliminates noise, and an inverse DWT reconstructs a cleaner ECG signal. The effectiveness hinges on selecting an appropriate wavelet, thresholding method, and decomposition level to optimally separate the signal from noise.

Code :

clc; clear all; close all;

% Load ECG signal from MIT-BIH Arrhythmia Database

[ecg\_signal, fs] = rdsamp('s0010\_re'); % Load signal (1st channel by default)

x = ecg\_signal(:,1); % Extract first channel

t = (0:length(x)-1) / fs; % Time vector

% Plot Original ECG Signal

figure;

subplot(3,1,1);

plot(t, x, 'b'); grid on;

title('Original Noisy ECG Signal (MIT-BIH Data)');

xlabel('Time (s)'); ylabel('Amplitude');

%% Step 1: Here we perform Wavelet Decomposition

waveletType = 'db12'; % Daubechies wavelet

level = 12; % Decomposition Level

[C, L] = wavedec(x, level, waveletType); % Wavelet decomposition

%% Step 2: We are doing Thresholding to Remove Noise

% Use higher threshold for better noise reduction

thr = median(abs(C)) / 0.6745 \* sqrt(2 \* log(length(C)));

% Now We Apply thresholding on \*\*only the detail coefficients\*\*

for i = 1:level

D = detcoef(C, L, i); % Extract detail coefficients

D = wthresh(D, 's', thr); % Soft thresholding

C(L(level + 2) + (1:length(D))) = D; % Update coefficients

end

%% Step 3: Baseline Wander Removal

A = appcoef(C, L, waveletType, level); % Extract approximation coefficients

C(1:length(A)) = 0; % Suppress baseline drift

%% Step 4: Reconstruct Clean ECG Signal

ecgDenoised = waverec(C, L, waveletType); % Reconstruct signal

% Plot Clean ECG Signal

subplot(3,1,2);

plot(t, ecgDenoised, 'r'); grid on;

title('Denoised ECG Signal using Wavelet Transform');

xlabel('Time (s)'); ylabel('Amplitude');

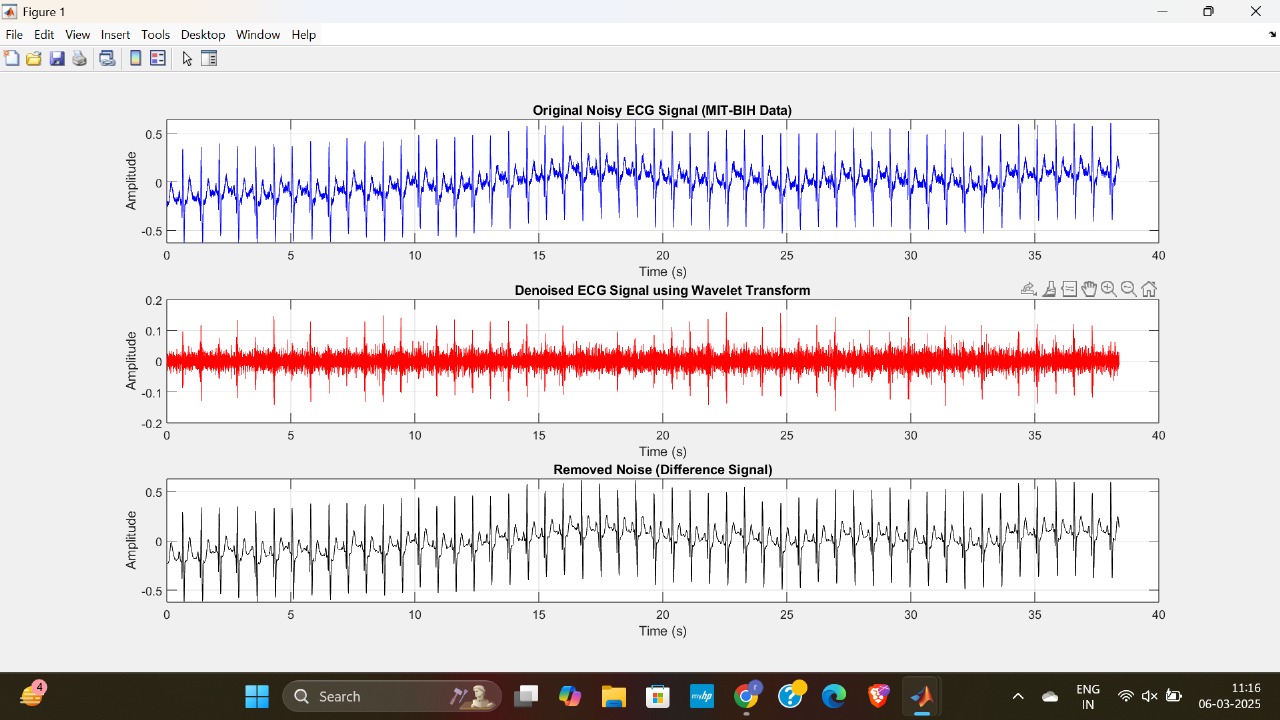
% Plot Difference (Noise Removed)

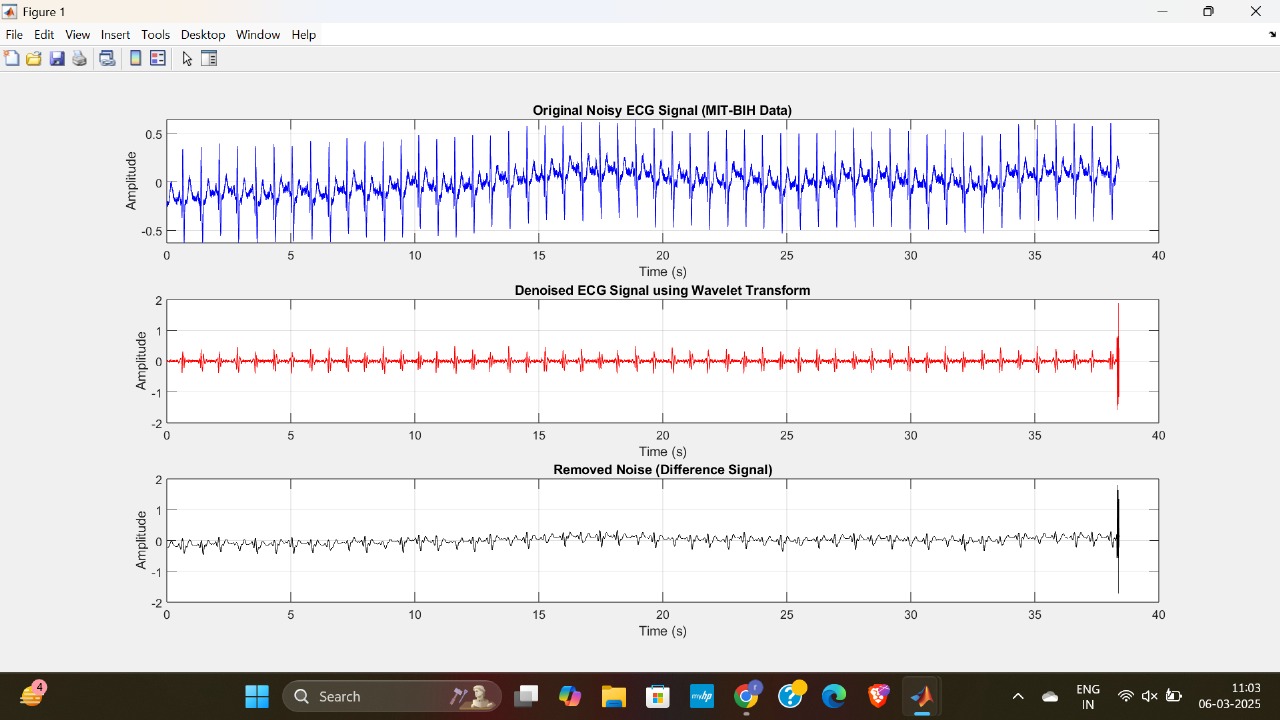
subplot(3,1,3);

plot(t, x - ecgDenoised, 'k'); grid on;

title('Removed Noise (Difference Signal)');

xlabel('Time (s)'); ylabel('Amplitude');t



  
  
Conclusion :-

This MATLAB code denoises an ECG signal using wavelet decomposition, thresholding, and baseline drift removal, resulting in a cleaner signal for analysis.