EXPERIMENT 2

AIM: IMPLEMENTATION OF FIR AND IIR FILTER OVER ECG SIGNAL

SOFTWARE: MATLAB

THOERY:

Electrocardiography (ECG) is a critical tool for monitoring heart activity, but recorded signals often contain noise from various sources such as powerline interference (50 Hz in many regions) and high-frequency disturbances. To ensure accurate analysis, filtering techniques are applied to enhance signal quality.

1. ECG Signal Acquisition

ECG signals can be generated synthetically or acquired from real datasets. In this experiment, a synthetic ECG signal is modeled using sinusoidal components, while a real ECG signal is read using rdsamp().

2. Noise Sources in ECG Signals

* High-Frequency Noise: Caused by muscle movements and external electronic interference.
* Powerline Interference (50 Hz): Introduced from surrounding electrical devices.

3. Filtering Techniques

To remove noise, two filtering techniques are employed:

1. FIR Low-Pass Filtering:
   * A Finite Impulse Response (FIR) low-pass filter is designed with a cutoff frequency of 55 Hz.
   * It effectively removes high-frequency noise while preserving the ECG waveform.
2. IIR Notch Filtering:
   * An Infinite Impulse Response (IIR) notch filter is designed at 50 Hz with a bandwidth of 5 Hz.
   * It eliminates powerline interference while maintaining signal integrity.

4. Frequency Domain Analysis

The Fast Fourier Transform (FFT) is applied to analyze the signal before and after filtering. It helps visualize noise reduction in the frequency spectrum.

CODES:

Code 1:

clc

clear aal

close all

% Load or Generate ECG Signal

fs = 500; %Sampling Frequency (Hz)

t = 0:1/fs:5; %Time vector (5 Seconds)

f\_ecg = 1.5; % Approximate ECG frequency (Hz)

% Generate a synthetic ECG signal (sum of sinusoidals)

ecg\_signal = sin(2\*pi\*f\_ecg\*t) + 0.2\*sin (2\*pi\*50\*t) + 0.1\*randn(size(t));

%% Plot Original Signal

figure

subplot(3,1,1);

plot(t, ecg\_signal, 'b');

xlabel('Time(s)');

ylabel('Amplitude');

title('Original Noisy ECG Signal')

xlim([0 2]);

%% FIR Low Pass Filtered Signal

fc = 55; % Cutoff frequency (Hz)

order = 50; % Filter order

fir\_coeffs = fir1(order, fc/(fs/2), 'low'); % FIR filter design

ecg\_fir\_filtered = filter(fir\_coeffs, 1, ecg\_signal);

% Plot FIR Filtered Signal

subplot(3,1,2);

plot(t, ecg\_fir\_filtered, 'g')

xlabel('Time(s)');

ylabel('Amplitude');

title('FIR Low Pass Filtered ECG Signal (Remove HF Noise)')

xlim([0 2]);

%% IIR Notch Filter (Renove Powerline Interference)

f\_notch = 50; %Notch frequency (Hz)

bw = 5; % Bandwidth (Hz)

wo = f\_notch/(fs/2); %Normalized frequency

[b, a] = iirnotch(wo, bw/(fs/2)); %IIR Notch filter design

ecg\_iir\_filtered = filter(b, a, ecg\_fir\_filtered);

% Plot IIR Filtered Signal

subplot(3,1,3);

plot(t, ecg\_iir\_filtered, 'r');

xlabel('Time(s)');

ylabel('Amplitude');

title('IIR Notch Filtered ECG Signal (Remove HF Noise)')

xlim([0 2]);

%% Frequency Analysis

figure;

frequencies = linspace(0, fs/2, length(t)/2);

% Compute FFT

fft\_orig = abs(fft(ecg\_signal));

fft\_fir = abs(fft(ecg\_fir\_filtered));

fft\_iir = abs(fft(ecg\_iir\_filtered));

% Plot FFT before and after filering

plot(frequencies, fft\_orig(1: length(frequencies)), 'b', 'LineWidth', 1);

hold on

plot(frequencies, fft\_fir(1: length(frequencies)), 'g', 'LineWidth', 1);

plot(frequencies, fft\_iir(1: length(frequencies)), 'r', 'LineWidth', 1);

xlabel('frequency (Hz)');

ylabel('Magnitude');

Code 2:

clc

clear all;

close all;

[signal, fs] = rdsamp('s0010\_re',1,500); %Read 500 samples from the signal

%Extract the first channel (assuming a multi-lead ECG)

ecg\_signal = signal(:,1);

%Time vector based on signal length and sampling frequency

t = (0:length(ecg\_signal)-1) / fs;

%% Plot Original Signal

figure;

subplot(3,1,1);

plot(t, ecg\_signal, 'b');

xlabel('Time(s)');

ylabel('Amplitude');

title('Original Noisy ECG Signal')

% xlim([0 2]);

%% FIR Low Pass Filtered Signal

fc = 55; % Cutoff frequency (Hz)

order = 50; % Filter order

fir\_coeffs = fir1(order, fc/(fs/2), 'low'); % FIR filter design

ecg\_fir\_filtered = filter(fir\_coeffs, 1, ecg\_signal);

% Plot FIR Filtered Signal

subplot(3,1,2);

plot(t, ecg\_fir\_filtered, 'g')

xlabel('Time(s)');

ylabel('Amplitude');

title('FIR Low Pass Filtered ECG Signal (Remove HF Noise)')

% xlim([0 2]);

%% IIR Notch Filter (Renove Powerline Interference)

f\_notch = 50; %Notch frequency (Hz)

bw = 5; % Bandwidth (Hz)

wo = f\_notch/(fs/2); %Normalized frequency

[b, a] = iirnotch(wo, bw/(fs/2)); %IIR Notch filter design

ecg\_iir\_filtered = filter(b, a, ecg\_fir\_filtered);

% Plot IIR Filtered Signal

subplot(3,1,3);

plot(t, ecg\_iir\_filtered, 'r');

xlabel('Time(s)');

ylabel('Amplitude');

title('IIR Notch Filtered ECG Signal (Remove HF Noise)')

% xlim([0 2]);

%% Frequency Analysis

figure;

frequencies = linspace(0, fs/2, length(t)/2);

% Compute FFT

fft\_orig = abs(fft(ecg\_signal));

fft\_fir = abs(fft(ecg\_fir\_filtered));

fft\_iir = abs(fft(ecg\_iir\_filtered));

% Plot FFT before and after filering

plot(frequencies, fft\_orig(1: length(frequencies)), 'b', 'LineWidth', 1);

hold on

plot(frequencies, fft\_fir(1: length(frequencies)), 'g', 'LineWidth', 1);

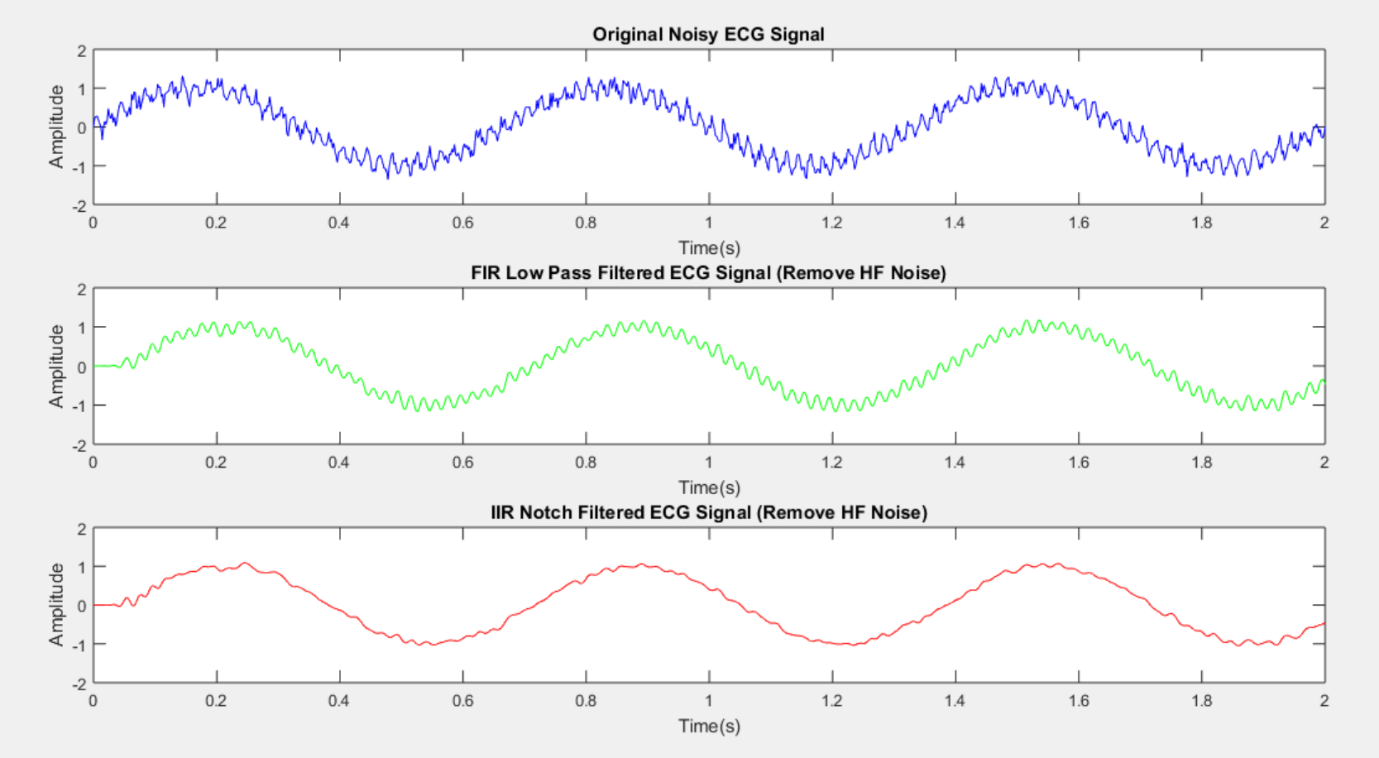
plot(frequencies, fft\_iir(1: length(frequencies)), 'r', 'LineWidth', 1);

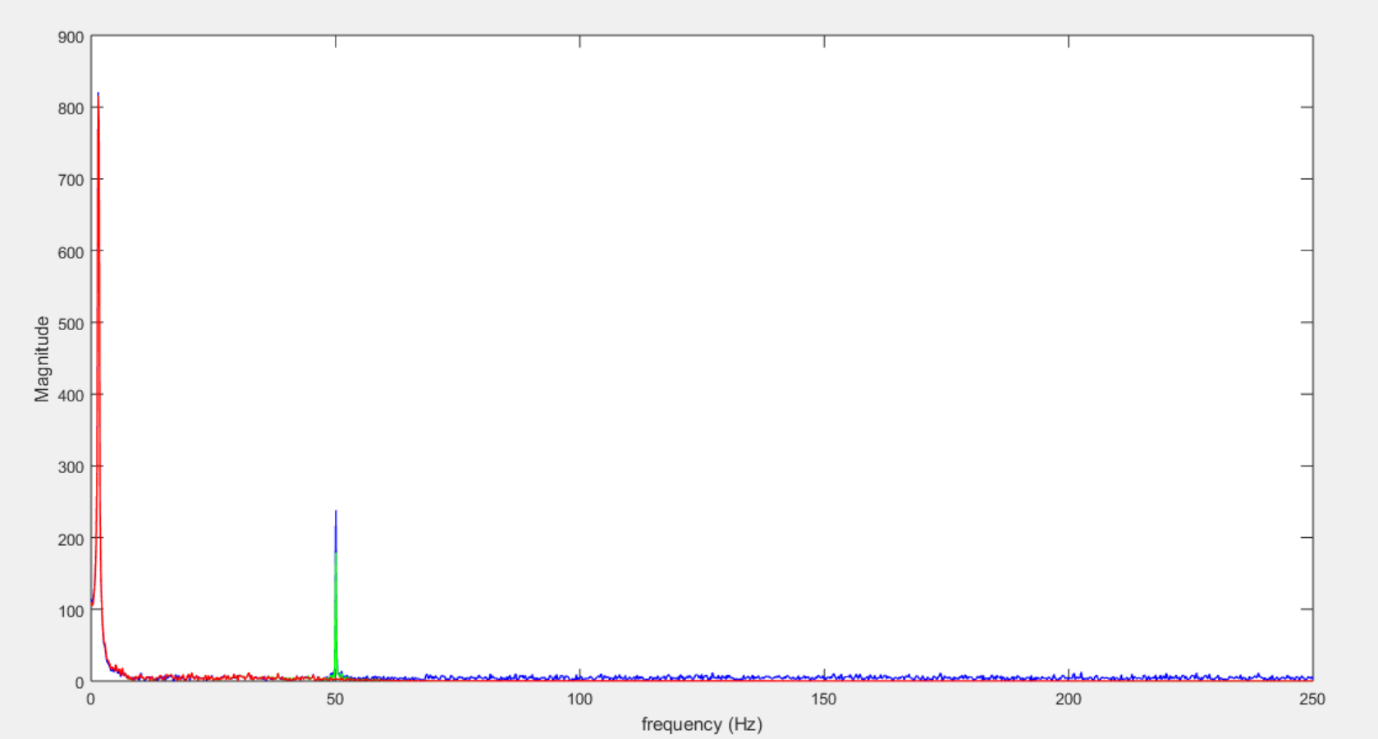
xlabel('frequency (Hz)');

ylabel('Magnitude');

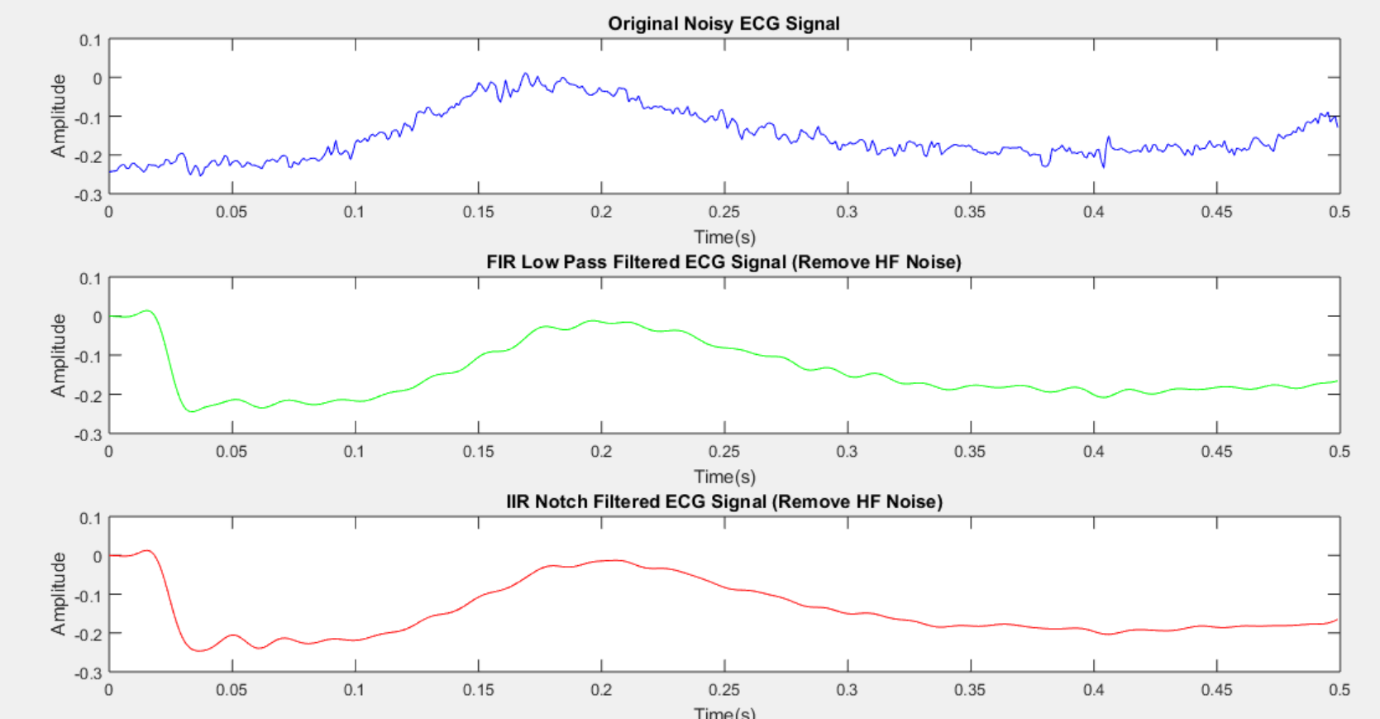
OUTPUT:

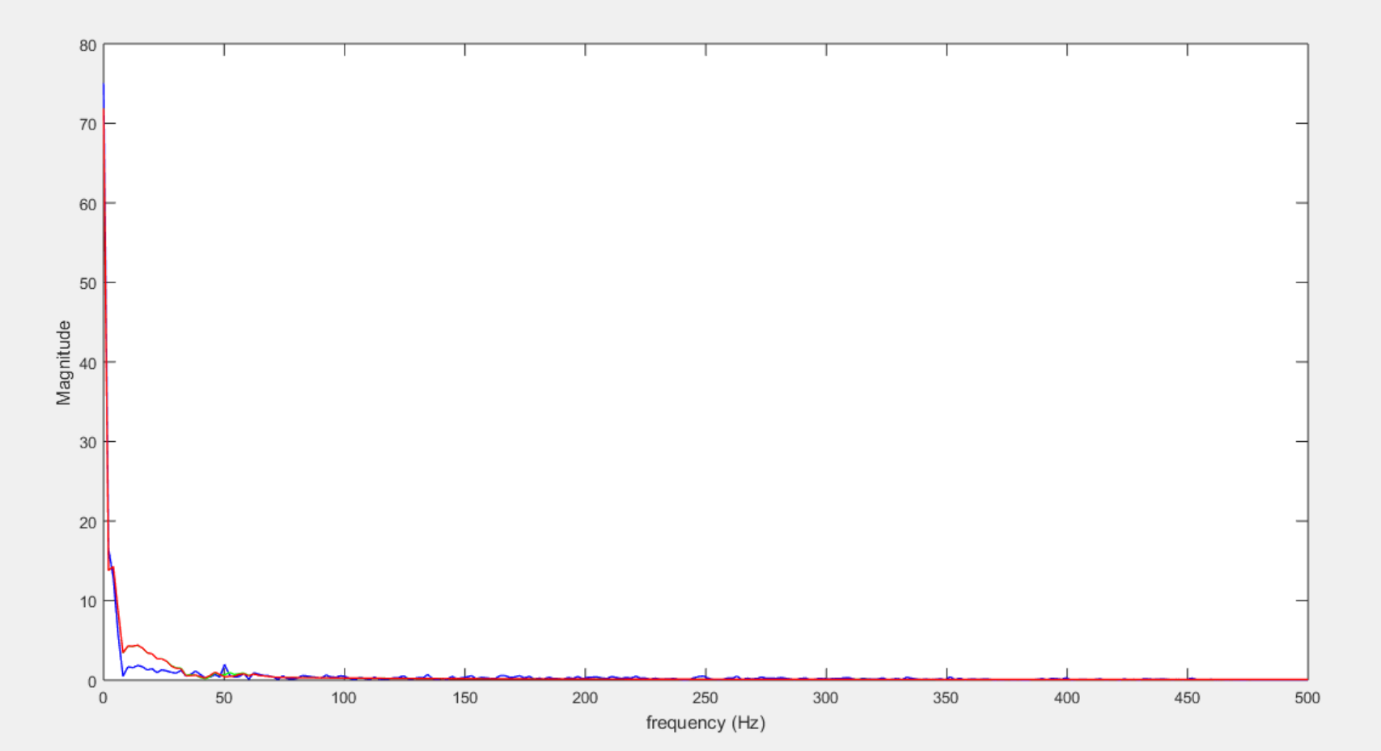
Code 1:





Code 2:





CONCLUTION:

By applying FIR and IIR filtering techniques, ECG signal quality is significantly improved, allowing for better interpretation and analysis of heart activity. This experiment demonstrates the effectiveness of digital filtering in biomedical signal processing.