Expt3  
  
% Load ECG signal

data = load('100m.mat');

fs = 1000; % Sampling frequency (adjust if needed)

% Check field names in the loaded data structure

fieldnames(data)

% Extract ECG signal (replace 'val' with the correct field name if different)

ecg\_signal = data.val;

% Ensure the signal is a row vector for processing

% ecg\_signal = ecg\_signal(:);

% Design and apply a high-pass filter

hp\_cutoff = 0.5; % High-pass cutoff frequency in Hz

hp\_order = 4; % Filter order

[b\_hp, a\_hp] = butter(hp\_order, hp\_cutoff / (fs / 2), 'high');

ecg\_highpass = filtfilt(b\_hp, a\_hp, ecg\_signal);

% Design and apply a low-pass filter

lp\_cutoff = 40; % Low-pass cutoff frequency in Hz

lp\_order = 4; % Filter order

[b\_lp, a\_lp] = butter(lp\_order, lp\_cutoff / (fs / 2), 'low');

ecg\_lowpass = filtfilt(b\_lp, a\_lp, ecg\_highpass);

% Time vector for plotting

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

% Plot original and processed ECG signals

figure;

% Original ECG signal

subplot(3, 1, 1);

plot(time, ecg\_signal);

title('Original ECG Signal');

xlabel('Time (s)');

ylabel('Amplitude');

% High-pass filtered ECG signal

subplot(3, 1, 2);

plot(time, ecg\_highpass);

title('ECG Signal After High-Pass Filtering');

xlabel('Time (s)');

ylabel('Amplitude');

% Low-pass filtered ECG signal

subplot(3, 1, 3);

plot(time, ecg\_lowpass);

title('ECG Signal After Low-Pass Filtering');

xlabel('Time (s)');

ylabel('Amplitude');

EXpt4  
clear;

%load ecg signal

fid = fopen('rec\_1.dat','r');

data = fread(fid, 'int16');

fclose(fid);

ecg\_signal = data(1:2:end);

% define parameters

fs = 500;

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

%high pass filtering

h\_cutoff = 0.5;

[b\_h, a\_h] = butter(2, h\_cutoff/(fs/2), 'high');

h\_filtered = filtfilt(b\_h, a\_h, ecg\_signal);

%low pass filtering

l\_cutoff = 50;

[b\_l, a\_l] = butter(2, l\_cutoff/(fs/2), 'low');

l\_filtered = filtfilt(b\_l, a\_l, h\_filtered);

%detect peaks

[peaks, locs] = findpeaks(l\_filtered, 'MinPeakHeight', 0.5, 'MinPeakDistance', fs/2);

%plot each fckin thing

figure;

subplot(4,1,1);

plot(t, ecg\_signal);

title("Original signal");

subplot(4,1,2);

plot(t, h\_filtered);

title("High pass filtered signal");

subplot(4,1,3);

plot(t, l\_filtered);

title("Low pass filtered signal");

subplot(4,1,4);

plot(t, l\_filtered);

hold on;

plot(t(locs), peaks, 'ro');

title("Ecg signal with peaks");

% calculate heart rate

rr\_interval = diff(locs)/fs;

heart\_rate = 60 / mean(rr\_interval);

disp("Heart rate is: ");

disp(heart\_rate);

EXpt6  
clear;

% read the ecg signal

fid = fopen('rec\_1.dat', 'r');

data = fread(fid, 'int16');

fclose(fid);

ecg\_signal = data(1:2:end);

%define parameters

fs = 250;

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

N = length(ecg\_signal);

ecg\_fft = fft(ecg\_signal);

figure;

subplot(2,1,1);

plot(t,ecg\_signal);

title("Original Signal");

subplot(2,1,2);

plot((1:floor(N/2)), abs(ecg\_fft(1:floor(N/2))));

title("FFT of Signal");

expt7  
clear;

%read ecg

% fid = fopen('rec\_1.dat','r');

% data = fread(fid, 'int16');

% fclose(fid);

% ecg\_signal = data(1:2:end);

data = load('100m.mat');

ecg\_signal = data.val;

%define parameters and awgn

fs = 500;

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

noise\_var = 0.5; % You can adjust this value

noisy\_signal = awgn(ecg\_signal, 10\*log10(1/noise\_var), 'measured');

%cutoff frequencies

l\_cutoff = 30;

h\_cutoff = 0.5;

b\_cutoff = [30, 249];

%low pass

[b\_l, a\_l] = butter(2, l\_cutoff/(fs/2), 'low');

l\_filtered = filtfilt(b\_l, a\_l, noisy\_signal);

%high pass

[b\_h, a\_h] = butter(2, h\_cutoff/(fs/2), 'high');

h\_filtered = filtfilt(b\_h, a\_h, noisy\_signal);

%notch filter (band stop)

[b\_n, a\_n] = butter(2, b\_cutoff/(fs/2), 'stop');

n\_filtered = filtfilt(b\_n, a\_n, noisy\_signal);

%plot all figures

figure;

subplot(5,1,1);

plot(t, ecg\_signal);

title('Original Signal'); xlabel('Time (s)'); ylabel('Amplitude');

subplot(5,1,2);

plot(t, noisy\_signal);

title('Noisy Signal'); xlabel('Time (s)'); ylabel('Amplitude');

subplot(5,1,3);

plot(t, l\_filtered);

title('Low Pass filtered Signal'); xlabel('Time (s)'); ylabel('Amplitude');

subplot(5,1,4);

plot(t, h\_filtered);

title('High Pass filtered Signal'); xlabel('Time (s)'); ylabel('Amplitude');

subplot(5,1,5);

plot(t, n\_filtered);

title('Notch Filtered Signal'); xlabel('Time (s)'); ylabel('Amplitude');

exp5  
% Load the EEG data

data = load('subject00\_1\_edfm.mat');

% Extract fieldnames and EEG signal

fields = fieldnames(data);

disp(fields); % Display the field names to confirm structure

eeg\_sig = data.eeg\_signal; % Assuming the signal is stored under the 'val' field

num\_ch = size(eeg\_sig, 1); % Number of channels

% Determine the grid size for subplots

rows = ceil(num\_ch / 2); % Dynamically set rows to fit all channels

col = 2; % Fixed number of columns

% Plot each EEG channel

figure; % Create a new figure

for channel = 1:num\_ch

subplot(rows, col, channel); % Dynamically adjust subplot indices

plot(eeg\_sig(channel, :), 'LineWidth', 1); % Plot the EEG signal for each channel

title(['EEG Channel ', num2str(channel)], 'FontSize', 10);

grid on;

xlabel('Sample Points');

ylabel('Amplitude');

end