Term Project Report, Submission 5

Part 1:

Data Loading and Preprocessing:

- Loads raw ECG signal data from the '106m.mat' file.
- Extracts the raw signal, calculates the amplitude using the given gain, and creates a time vector.
- Extracts the data from 15 to 30 minutes.
- Plots ECG Waveform

Part 2:

Peak Detection for Heartbeat Count:

- Number of Heartbeats = QRS Peaks of the ECG Singal
- Uses the `findpeaks()` function to detect QRS peaks in the ECG signal.
- Sets thresholds for peak height and minimum peak distance to identify QRS peaks. (Used 0.2 (mV) and 0.4 sec respectively)
- Prints the number of heartbeats and calculates the average heartbeats per minute.

Part 3:

Calculating RR Intervals:

- Computes RR intervals by taking the difference between consecutive QRS peak indices.
- Used `diff()` function in MATLAB to get the difference between adjacent values

Part 4:

Plotting Histogram of RR Intervals:

- Plots a histogram of RR intervals using the histogram function.
- Number of bins = 100 (can be changed if need more distinctive)

Part 5:

Plotting RR Intervals Over Time:

- Cumulatively sums RR intervals to create a time as ‘x’ values for the plot.
- Plots the RR Intervals over time.

Part 6:

Interpolation of RR Intervals:

- Used in-built MATLAB function `interp1` to interpolate RR intervals.
- Uses cubic interpolation method to interpolate RR intervals.
- Uniformly samples the interpolated data for better representation. (Used 900 samples)
- Plots the interpolated RR interval (900 samples)

Part 7:

Sampling and Comparison of Original and Interpolated RR Intervals:

- Samples interpolated data at 128 Hz.
- Plots the original and interpolated RR intervals for the entire 15 minutes.
- Additionally, plots the comparison for the first 100 seconds to highlight differences.
- The interpolated RR intervals, sampled at 128 samples/second gives a smooth waveform, whereas the original waveform, without interpolation is not smooth.

Part 8:
Calculating Autocorrelation

- In this part, the code calculates the autocorrelation of a signal representing RR intervals.
- It first shifts the RR intervals by 100 samples and then uses the cross-correlation function to compute the autocorrelation.
- The result is then plotted to visualize the autocorrelation of the RR intervals.

Part 9:
Power Spectrum Calculation

- This section calculates the power spectrum of the RR intervals using the `pspectrum` function.
- The power spectrum is a representation of the signal's frequency content, and the resulting plot shows the power distribution across different frequency components.

Part 10:
FIR Filtering of ECG Signal

- This part focuses on filtering the raw ECG signal.
- It designs a finite impulse response (FIR) filter with a specified cutoff frequency of 50 Hz using the MATLAB's `fir1` function.
- The filter is then applied to the raw ECG signal to attenuate frequencies above the cutoff.
- The code plots both the raw and filtered ECG signals to visually compare their differences.
- The code also plots the frequency response of the designed filter using the `freqz` function.
- Additionally, the frequency response of both signals is computed and plotted to illustrate the impact of the FIR filter on the frequency content of the ECG signal.
- Calculated and printed the number of heartbeats and RR intervals based on the filtered ECG signal and compared with the raw ECG Signal

Part 11:
Calculating Cross-correlation

- In this part, the code calculates the Cross correlation of filtered ECG signal with raw ECG signal.
- It uses MATLAB's `xcross` function.
- The result is then plotted to visualize the Cross correlation of the RR intervals.

Part 12:
Repeated steps with filtered RR interval

- In this part, the code repeats the same executions with RR intervals as in parts 6 through 9, however, all the computations were post filtering of the signal.