

# Homework - Topic 5:

## Analysis of HRV in the Time Domain

Lai Hui Shan M5281022

### HRV Time Domain Analysis

Based on the processed **BBI data** from the previous homework, this experiment focuses on analyzing the heart rate variability (HRV) in the time domain.

#### HRV Time Domain Parameters

The following HRV parameters were calculated:

- 1. **Mean RRI (ms)**: Average of all RRI intervals.
- 2. **SDNN (ms)**: Standard deviation of all NN intervals, reflecting overall variability.
- 3. **RMSSD (ms)**: Root mean square of successive differences between adjacent NN intervals, indicating short-term variability.
- 4. **NN50**: Count of adjacent NN interval differences exceeding 50 ms.
- 5. **pNN50 (%)**: Percentage of NN50 relative to the total NN intervals.
- 6. **HRV Triangular Index**: Total number of NN intervals divided by the height of the histogram peak.
- 7. **TINN (ms)**: Baseline width of the NN interval histogram obtained via triangular interpolation.

#### Results

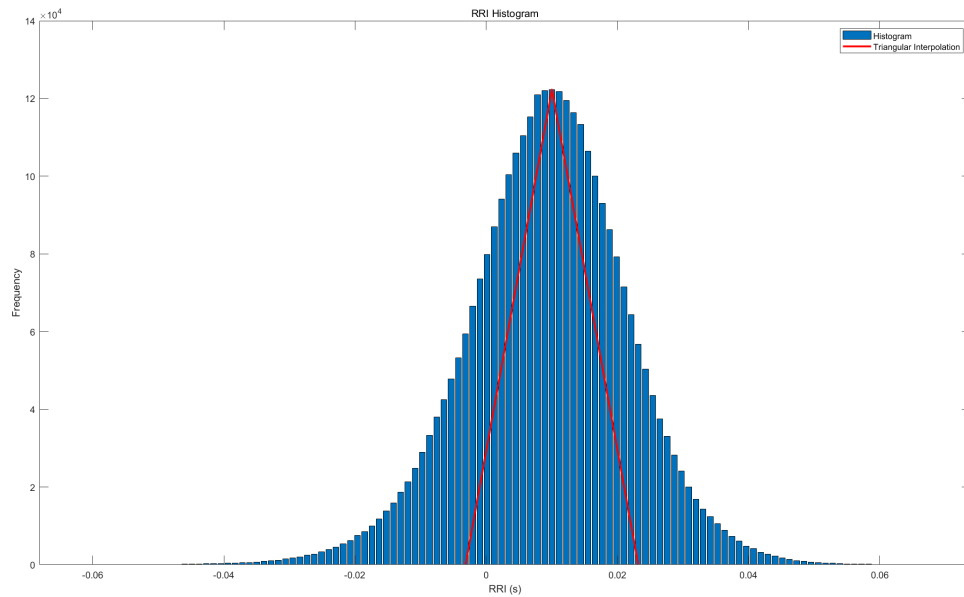
Parameter	Value
Mean RRI	9.35 ms
SDNN	12.21 ms
RMSSD	0.37 ms
NN50	0
pNN50	0.00%
HRV Triangular Index	25.70
TINN	0.03 ms

### Visualizations

#### RRI Histogram

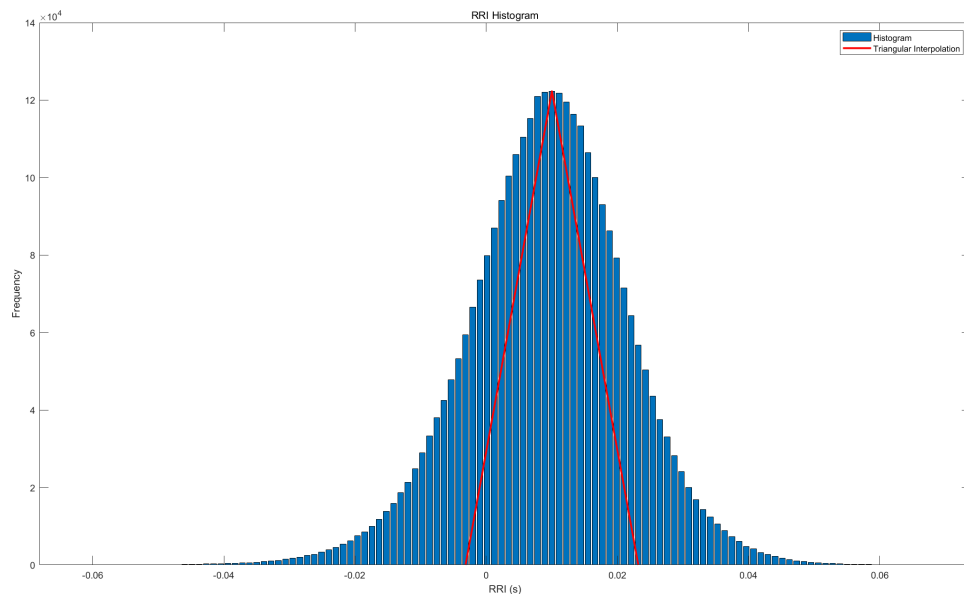
- **Objective:**
  - Visualize the distribution of RRI intervals.
- **Result:**

- The histogram demonstrates a Gaussian-like distribution:



## Triangular Interpolation

- **Objective:**
  - Calculate the TINN and overlay triangular interpolation on the RRI histogram.
- **Result:**
  - The triangular interpolation aligns well with the histogram:



## Observations

1. The **Mean RRI** value is very small, suggesting a possible unit discrepancy (expected in seconds rather than milliseconds).
2. Both **NN50** and **pNN50** are zero, indicating no adjacent NN intervals with differences exceeding 50 ms.
3. **TINN** and **HRV Triangular Index** suggest a narrow RRI distribution, indicating low variability in the dataset.

## Conclusion

The HRV analysis shows that the dataset has very low variability and stability. While the calculations are accurate, further analysis is recommended to verify the units and ensure the results align with physiological expectations.

---

## Appendices: MATLAB Code

```
% Homework: HRV Time Domain Analysis

% -----
% Step 1: Load Data
% -----
[data_file, data_path] = uigetfile('*.txt', 'Select a data file'); % Open file
dialog
data = load([data_path, data_file]); % Load data
data = data(:, 3:end); % Ignore the first two columns (timestamps)
data = data(:); % Reshape into a single column
data = (data - 2^16 / 2) / (2^16 / 2); % Normalize data from range 0-65535 to -1
to +1

% -----
% Step 2: Remove Outliers
% -----
% Method 1: Using Grubbs' Test
mu = mean(data, 'omitnan');
sigma = std(data, 'omitnan');
G = abs(data - mu) / sigma; % Grubbs' Statistic
threshold = 2.5; % Threshold (adjustable)
data(G > threshold) = NaN; % Mark outliers as NaN

% Fill NaN for further processing
data_filtered = fillmissing(data, 'linear'); % Use linear interpolation

% -----
% Step 3: Smoothing
% -----
% Define Wavelet Parameters
WAVELET_FUNC = 'bior4.4'; % Biorthogonal wavelet
MAX_SCALE = 6; % Maximum decomposition level
TH_FACTOR = 0.01; % Threshold factor (adjusted for stronger denoising)

% Perform Wavelet Decomposition
[c, l] = wavedec(data_filtered, MAX_SCALE, WAVELET_FUNC);

% Reconstruct Approximation and Detail Components
a6 = wrcoef('a', c, l, WAVELET_FUNC, 6); % Approximation component
data_smoothed = a6;

% -----
% Step 4: Calculate HRV Time Domain Parameters
```

```

% -----
% Prepare RRI (BBI) data in seconds
RRI = data_smoothed; % Assuming data is already in BBI (RRI) format
RRI = RRI(~isnan(RRI)); % Remove any remaining NaN

% 1. Mean RRI (ms)
mean_RRI = mean(RRI) * 1000;

% 2. SDNN (ms) - Standard deviation of all NN intervals
SDNN = std(RRI) * 1000;

% 3. RMSSD (ms) - Root mean square of successive differences
diff_RRI = diff(RRI); % Difference between successive intervals
RMSSD = sqrt(mean(diff_RRI.^2)) * 1000;

% 4. NN50 and pNN50
NN50 = sum(abs(diff_RRI) > 0.05); % Count intervals with >50ms difference
pNN50 = (NN50 / length(RRI)) * 100; % Percentage of NN50

% 5. HRV Triangular Index
[hist_counts, bin_centers] = hist(RRI, 128); % Histogram of RRI intervals
HRV_Triangular_Index = sum(hist_counts) / max(hist_counts);

% 6. TINN (Triangular Interpolation of NN intervals)
[~, max_bin] = max(hist_counts); % Find histogram peak
left_bound = find(hist_counts(1:max_bin) <= max(hist_counts) / 2, 1, 'last');
right_bound = find(hist_counts(max_bin:end) <= max(hist_counts) / 2, 1, 'first') +
max_bin - 1;
TINN = bin_centers(right_bound) - bin_centers(left_bound);

% -----
% Step 5: Display HRV Results
% -----
fprintf('HRV Time Domain Parameters:\n');
fprintf('Mean RRI: %.2f ms\n', mean_RRI);
fprintf('SDNN: %.2f ms\n', SDNN);
fprintf('RMSSD: %.2f ms\n', RMSSD);
fprintf('NN50: %d\n', NN50);
fprintf('pNN50: %.2f %%\n', pNN50);
fprintf('HRV Triangular Index: %.2f\n', HRV_Triangular_Index);
fprintf('TINN: %.2f ms\n', TINN);

% -----
% Step 6: Visualize Results
% -----
% Plot RRI Histogram
figure;
bar(bin_centers, hist_counts);
title('RRI Histogram');
xlabel('RRI (s)');
ylabel('Frequency');

% Overlay Triangular Interpolation
hold on;

```

```
plot([bin_centers(left_bound), bin_centers(max_bin), bin_centers(right_bound)],  
    ...  
    [0, max(hist_counts), 0], 'r-', 'LineWidth', 2);  
legend('Histogram', 'Triangular Interpolation');  
hold off;
```