Real-time Heart Monitoring and ECG Signal Processing

Fatima Bamarouf, Claire Crandell, and Shannon Tsuyuki

Advisors: Drs. Yufeng Lu and Jose Sanchez

Department of Electrical and Computer Engineering
Bradley University
November 24, 2015

Contents

Introduction and Overview

- Problem Description
- Project Objectives
- System Block Diagram
- Division of Labor

Problem Description

- Arrhythmias
  - Are irregular heartbeats caused by defective electrical signals in the heart [1]
  - Include premature ventricular contractions (PVCs)
  - PVCs may lead to ventricular tachycardia (VT)

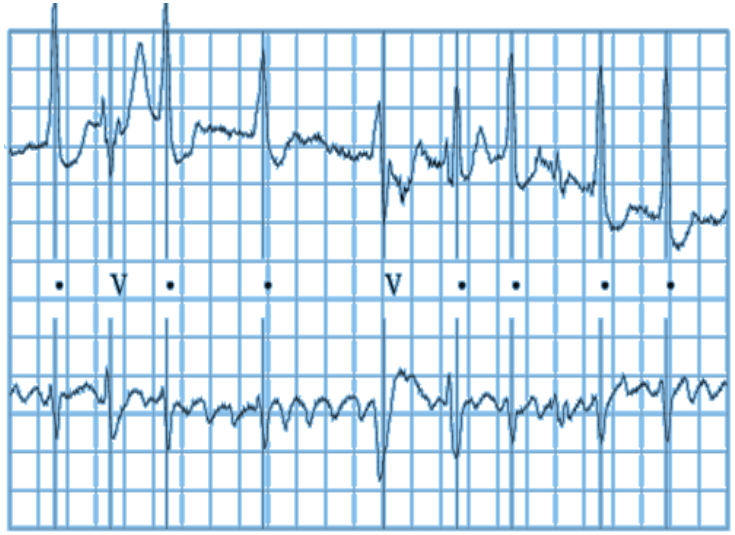


Figure 1. Electrocardiogram with “V” labels for PVCs [2]

Problem Description

- An electrocardiogram (ECG) describes the heart's electrical activity
- An ECG can be recorded using a Holter monitor or event monitor

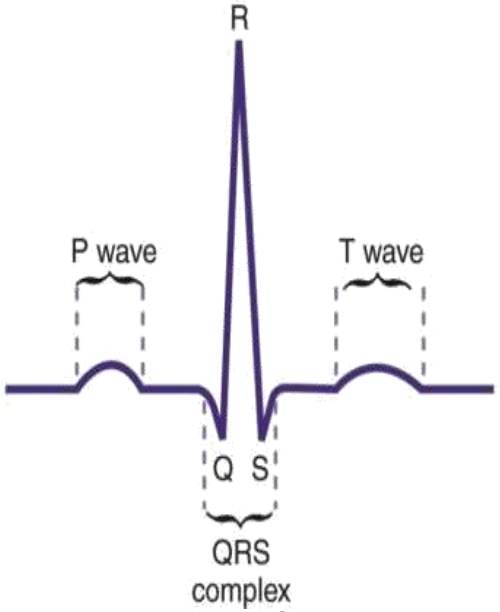


Figure 2. Features of a normal ECG [3]

Problem Description

- Holter and event monitors are limited in functionality
  - Utilize some in-platform signal processing for diagnostic assistance
  - Must perform some signal processing offline
  - Are unable to address medical issues in real time

System Block Diagram

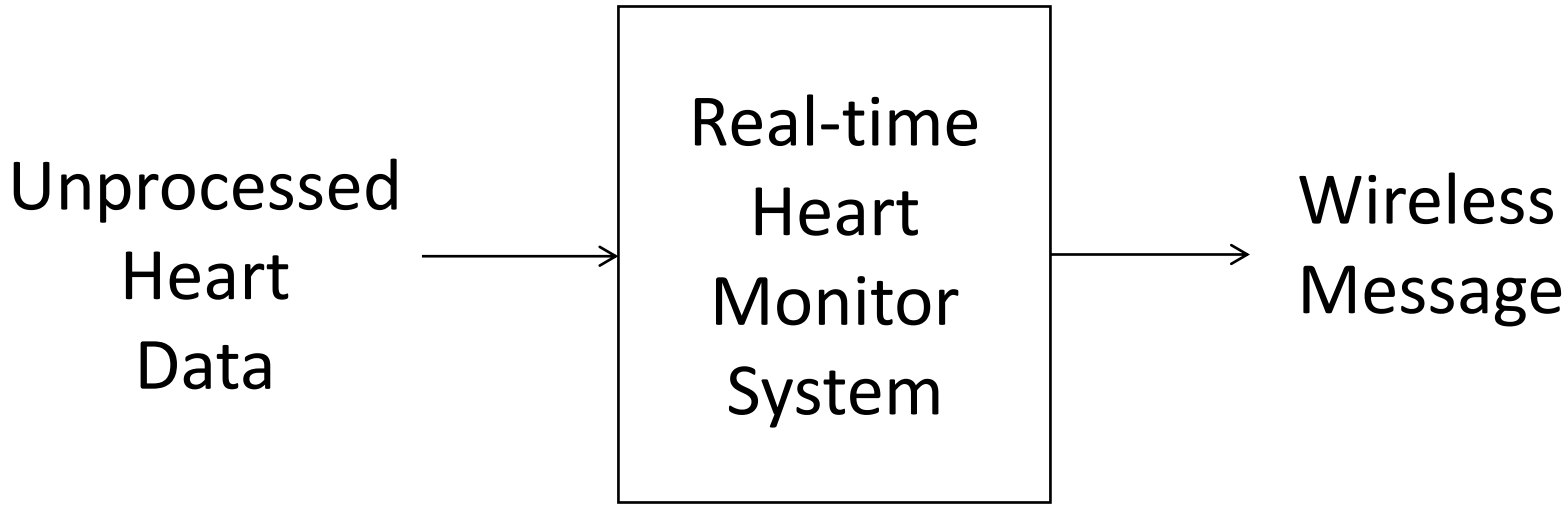


Figure 3. Overall heart monitoring system diagram

Project Objectives

- Develop a low-power, stand-alone embedded system for continuous heart monitoring that will
  - Process ECG data in real time
  - Detect PVCs accurately and consistently
  - Alert the patient's doctor wirelessly of ventricular tachycardia

Project Objectives

- Real-time ECG signal processing
- On-board signal processing computations
- Battery-powered functionality

Division of Labor

- MATLAB Simulation (PVC detection)
  - Shannon/Fatima
- C Programming (PVC detection)
  - Claire/Shannon
- Wi-Fi Communication
  - Fatima/Claire/Shannon

Contents

Problem Approach

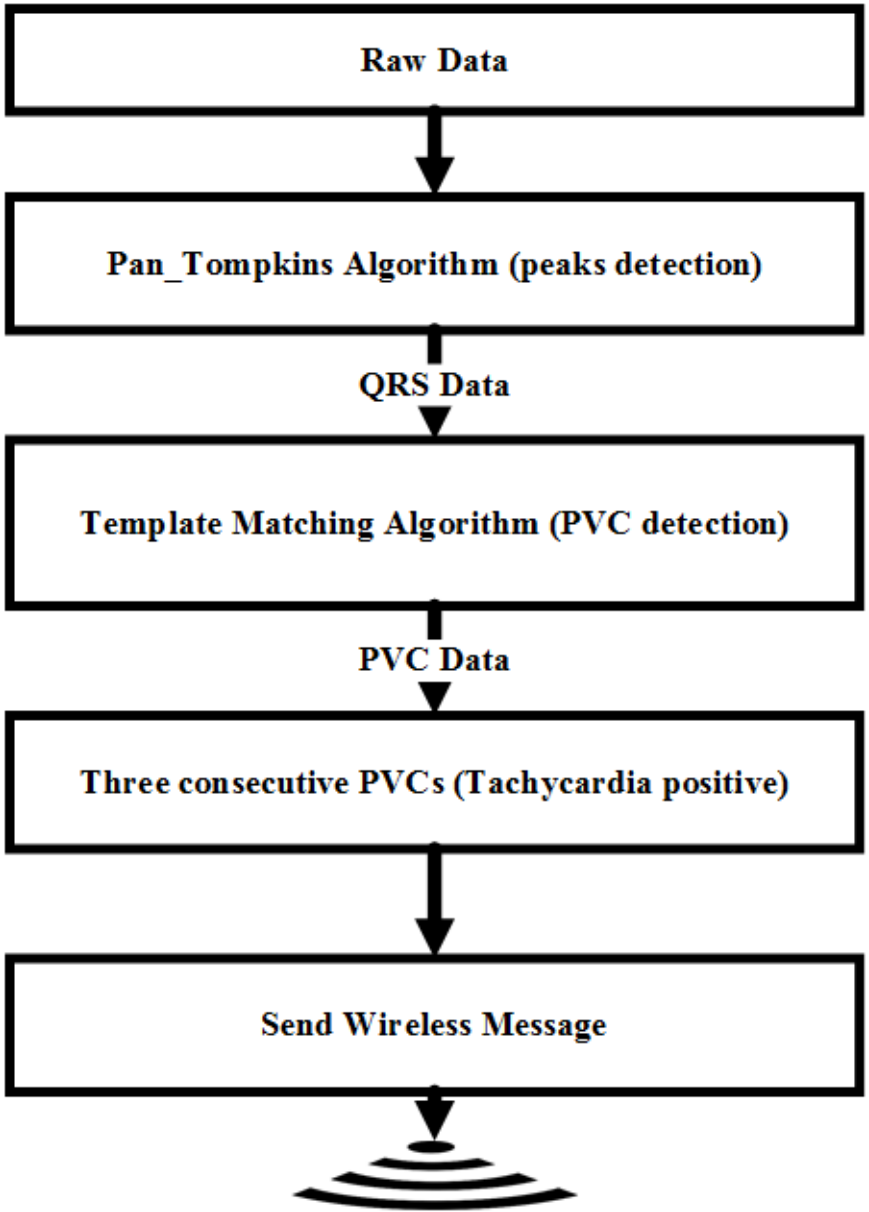


Figure 4. High-level flowchart for algorithms

Template Generation Algorithm [4]

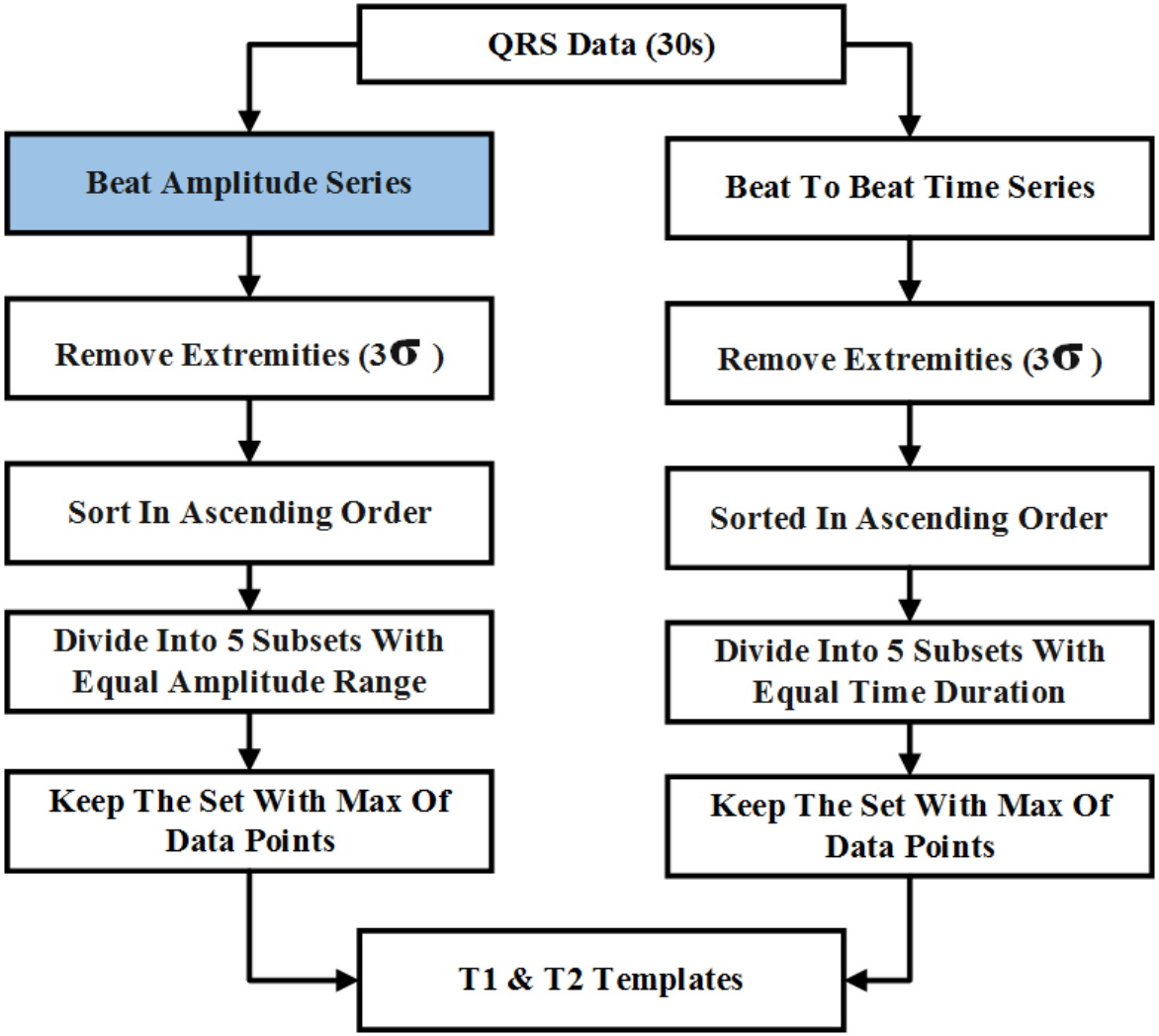


Figure 5. Flowchart for T1 and T2 generation

Beat Amplitude Series

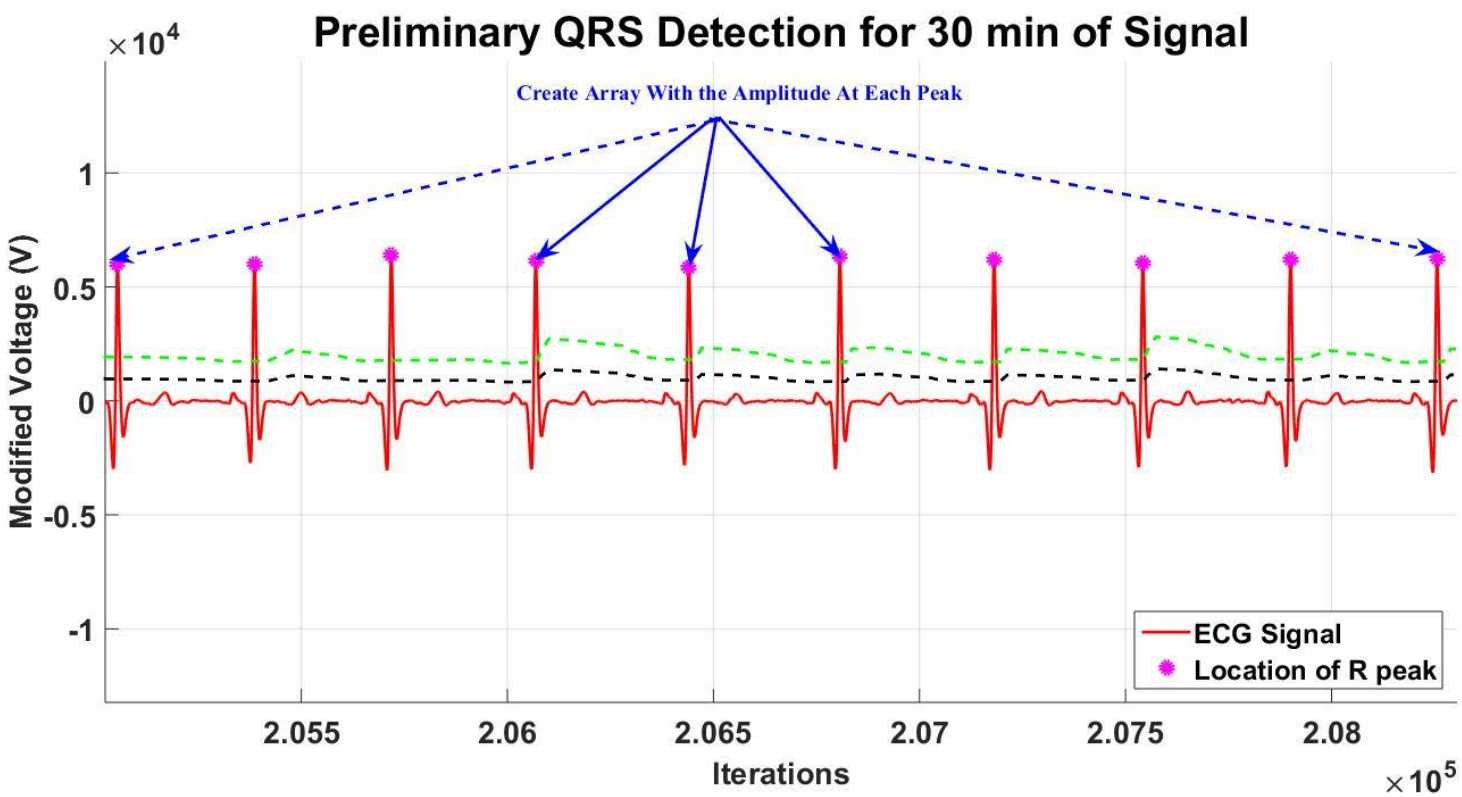


Figure 6. MATLAB plot of QRS detection results. The marked peaks were placed into an array.

Template Generation Algorithm

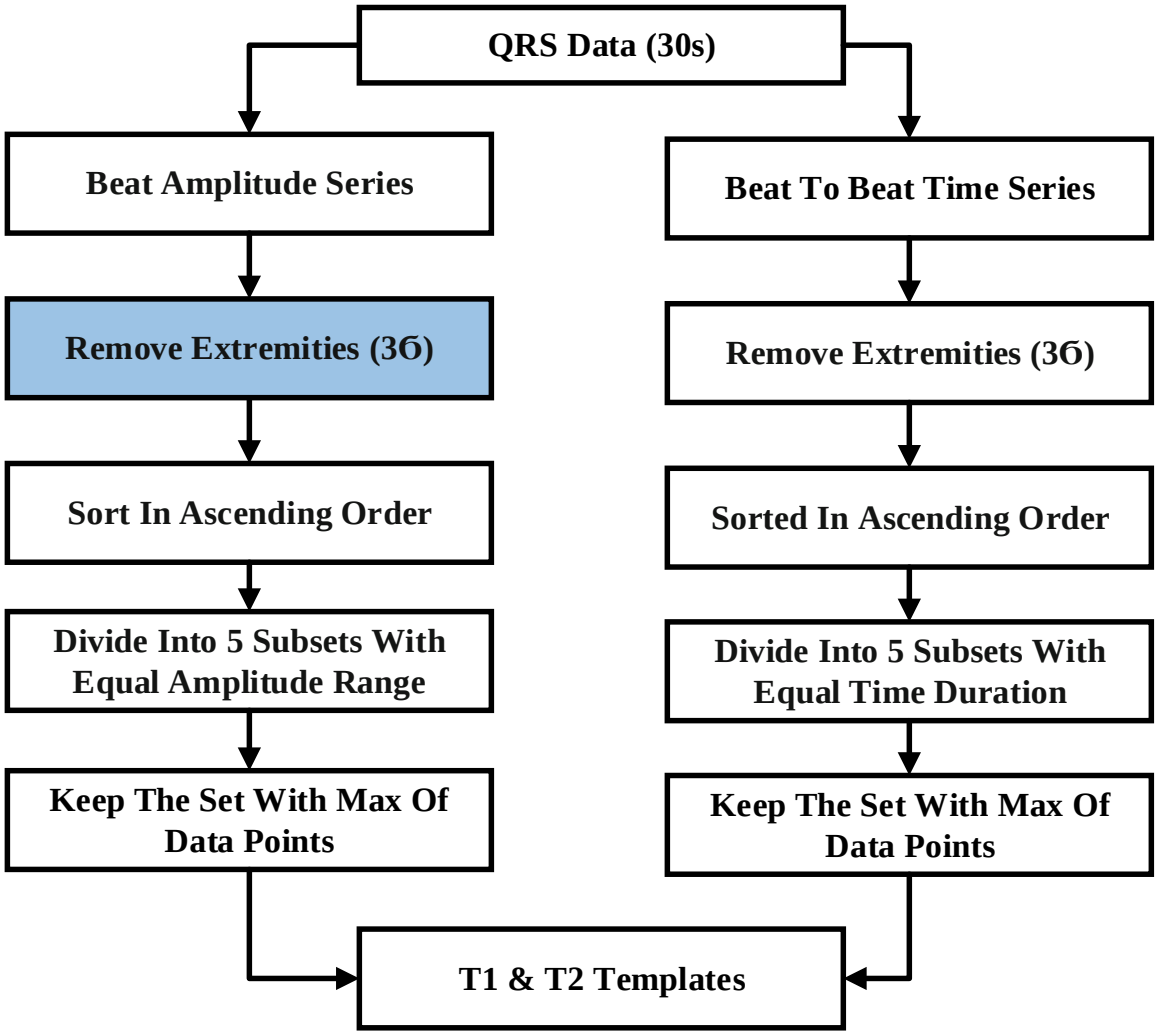


Figure 7. Flowchart for T1 and T2 generation

Removing Extremities (3σ)

- An amplitude *A* is kept if it is fulfills the criterion

$$\bar{A} - 3\sigma \leq A \leq \bar{A} + 3\sigma$$

where $\bar{A}$ is the mean of the amplitude range

Template Generation Algorithm

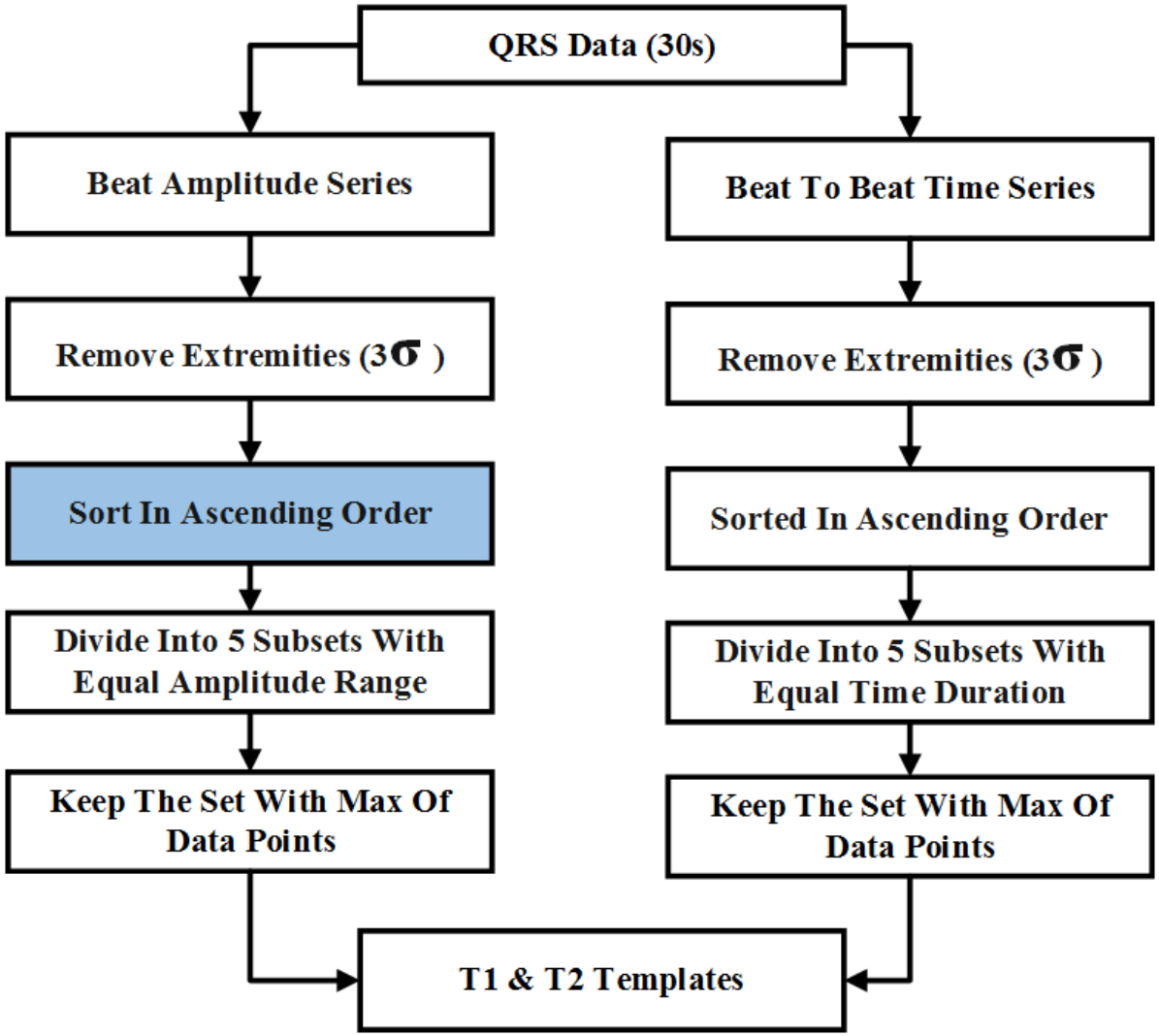


Figure 8. Flowchart for T1 and T2 generation

Template Generation Algorithm

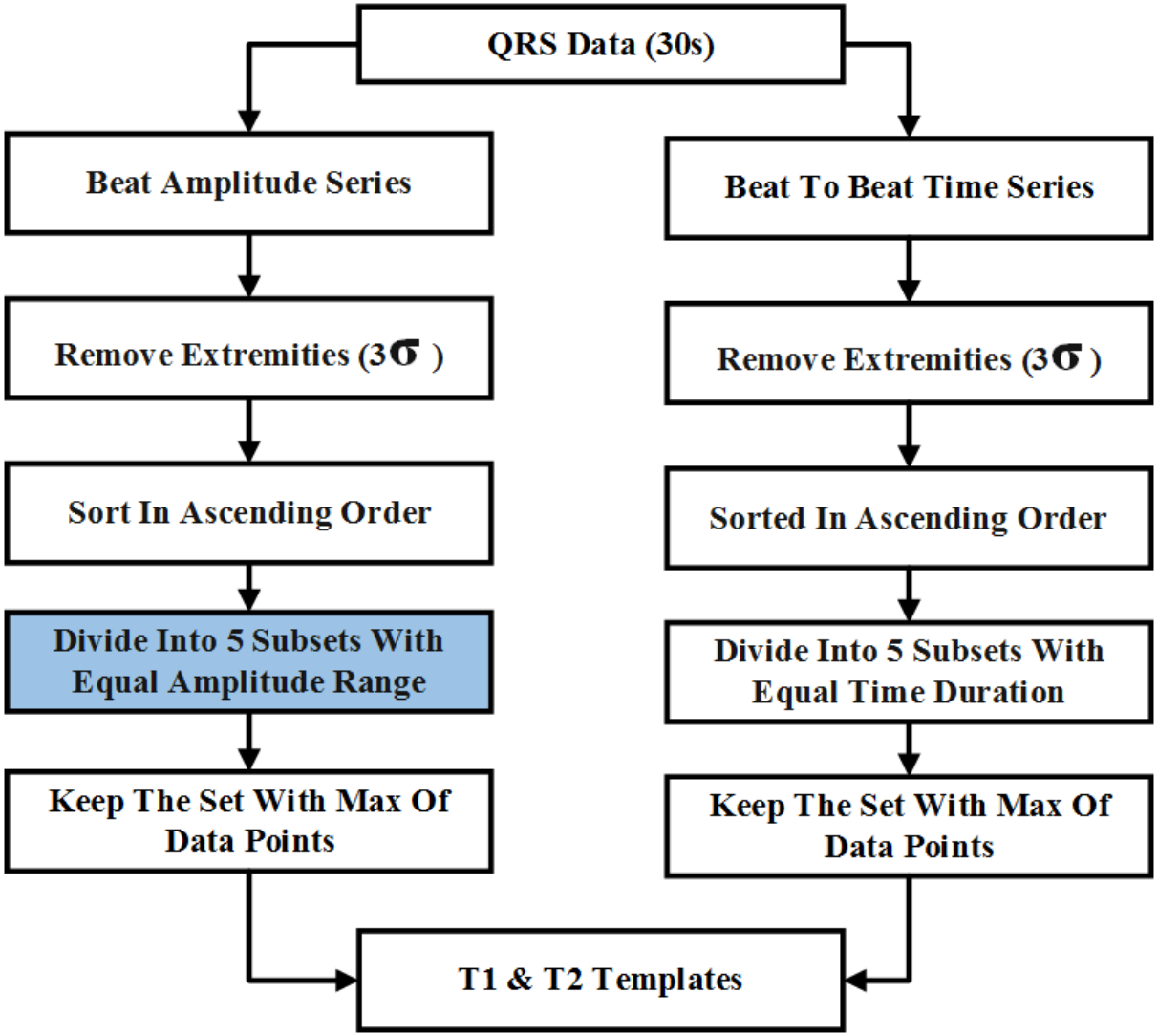


Figure 9. Flowchart for T1 and T2 generation

Equal Sets

- sort_series=[0 1 1.5 1.7 2 3 5 5 7 9 10]
- Step=(Max-Min)/5=2
- Five equal sets: [0 1 1.5 1.7 2] [3] [5 5] [7] [9 10]
- Set one has maximum data points

Template Generation Algorithm

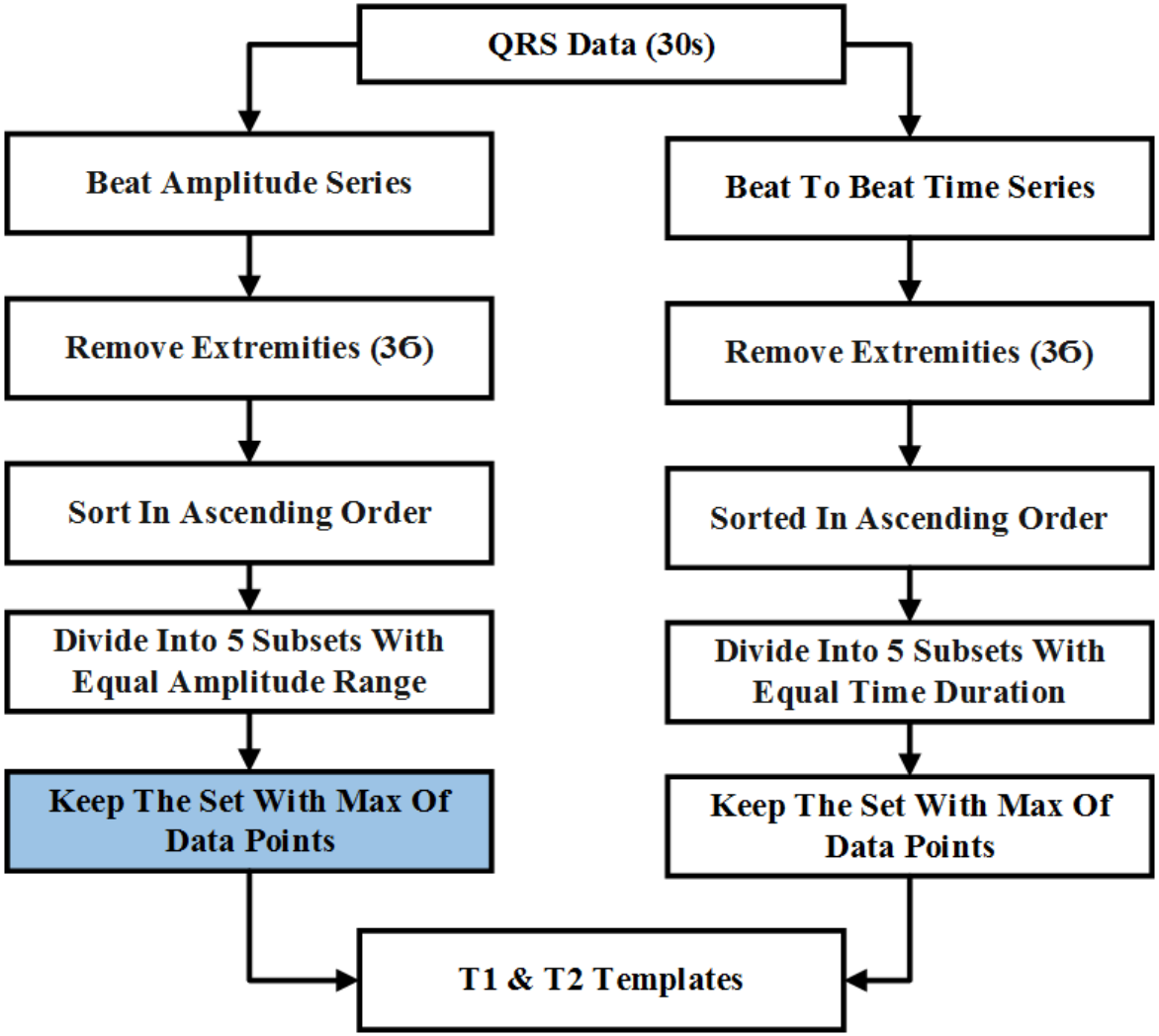


Figure 10. Flowchart for T1 and T2 generation

Template Generation Algorithm

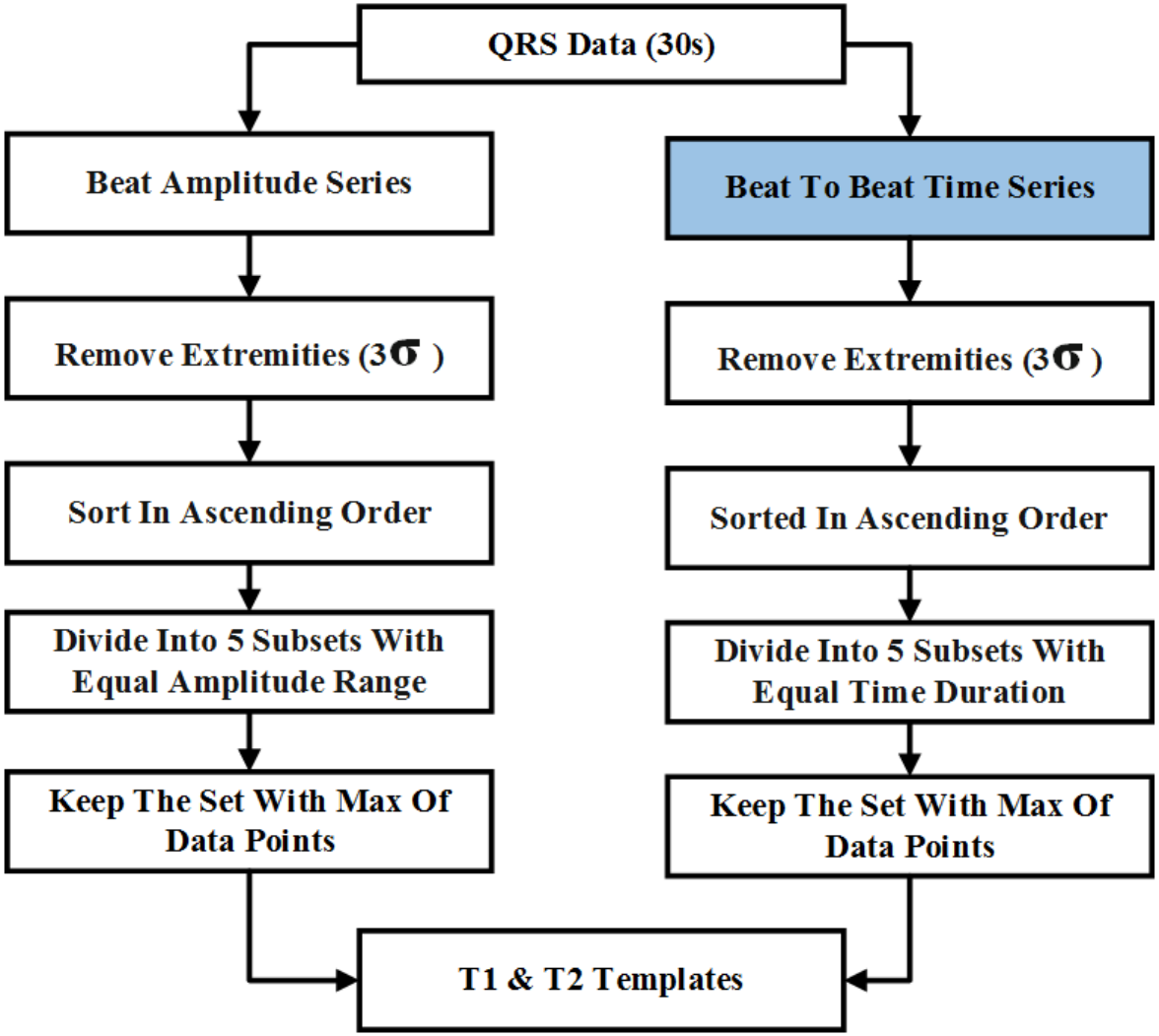


Figure 11. Flowchart for T1 and T2 generation

Data between Two Fiducial Points

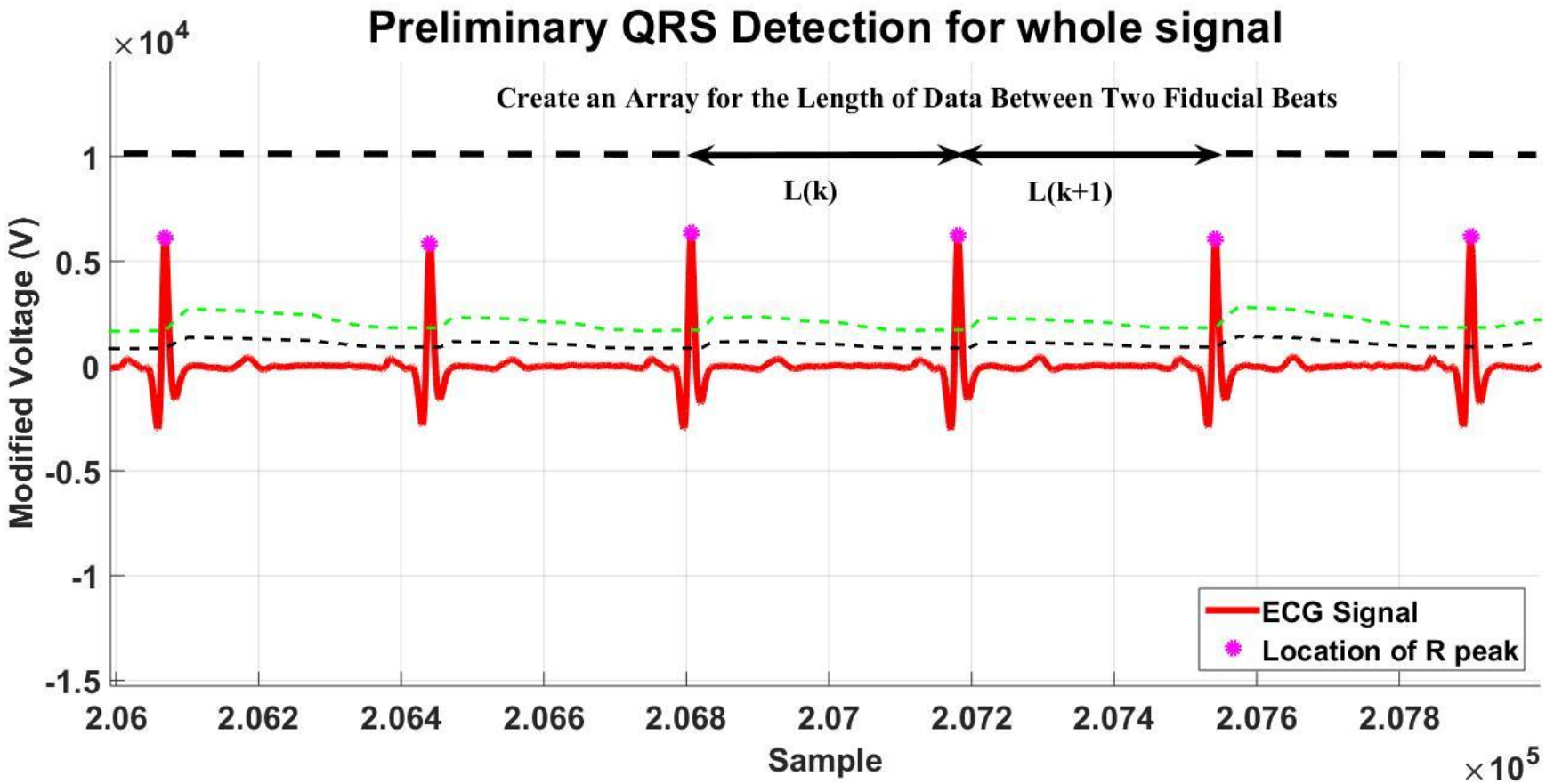


Figure 12. MATLAB plot of QRS detection results. The marked RR-intervals were placed into an array.

Template Generation Algorithm

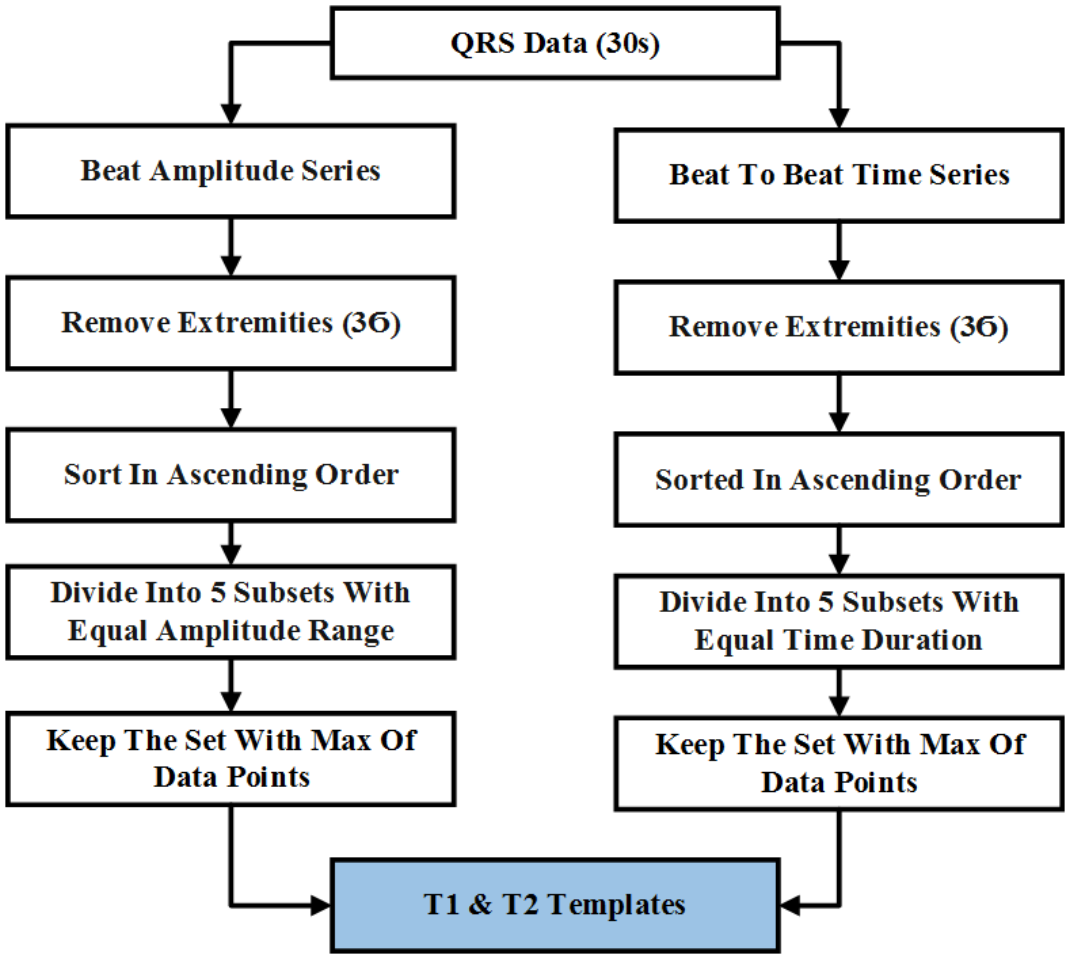


Figure 13. Flowchart for T1 and T2 generation

Templates T1 and T2

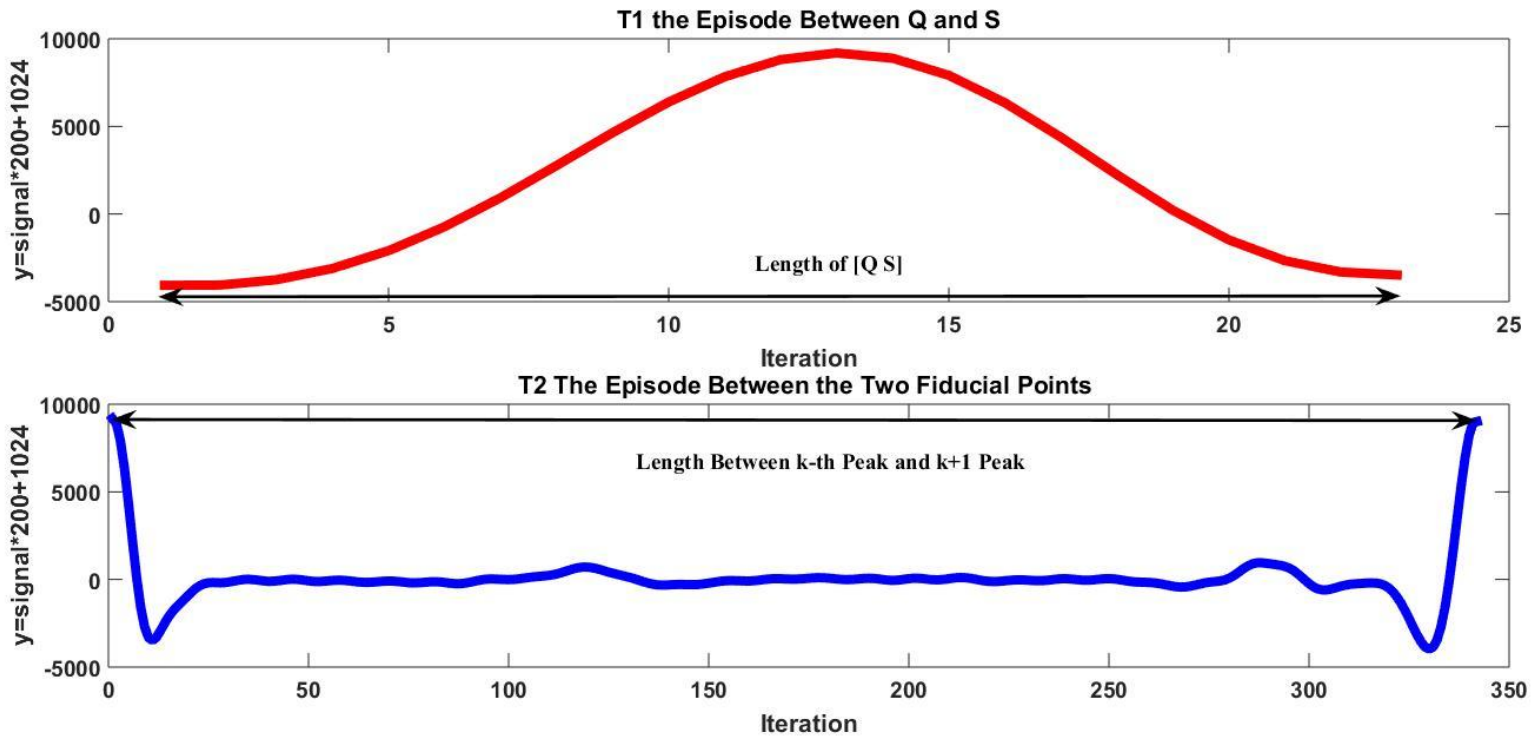


Figure 14. T1 and T2 generation

Template-Matching Algorithm

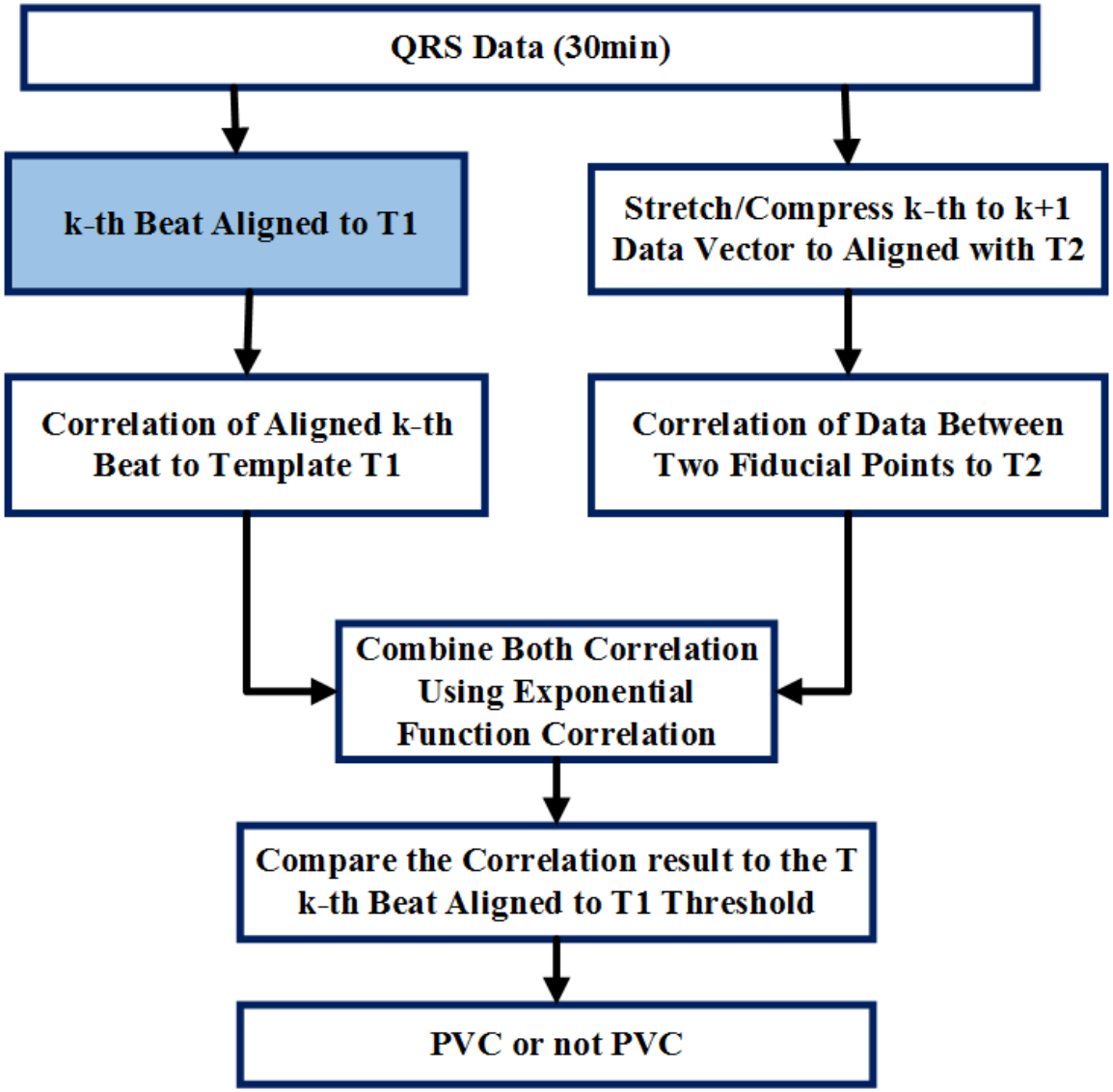


Figure 15. Flowchart for template-matching algorithm

T1 Alignment

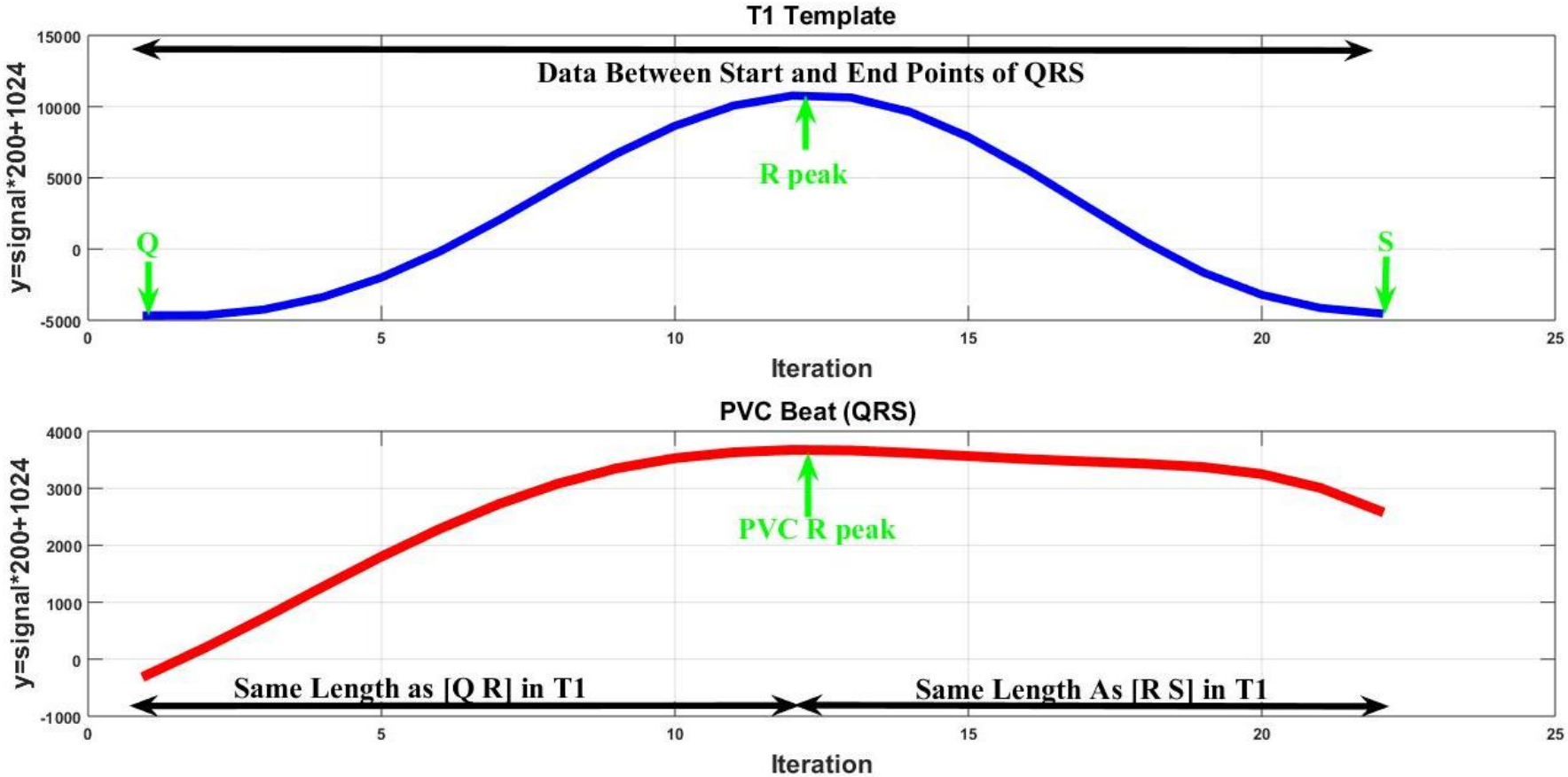


Figure 16. T1 template and PVC QRS complex aligned to T1's R peak index

Template-Matching Algorithm

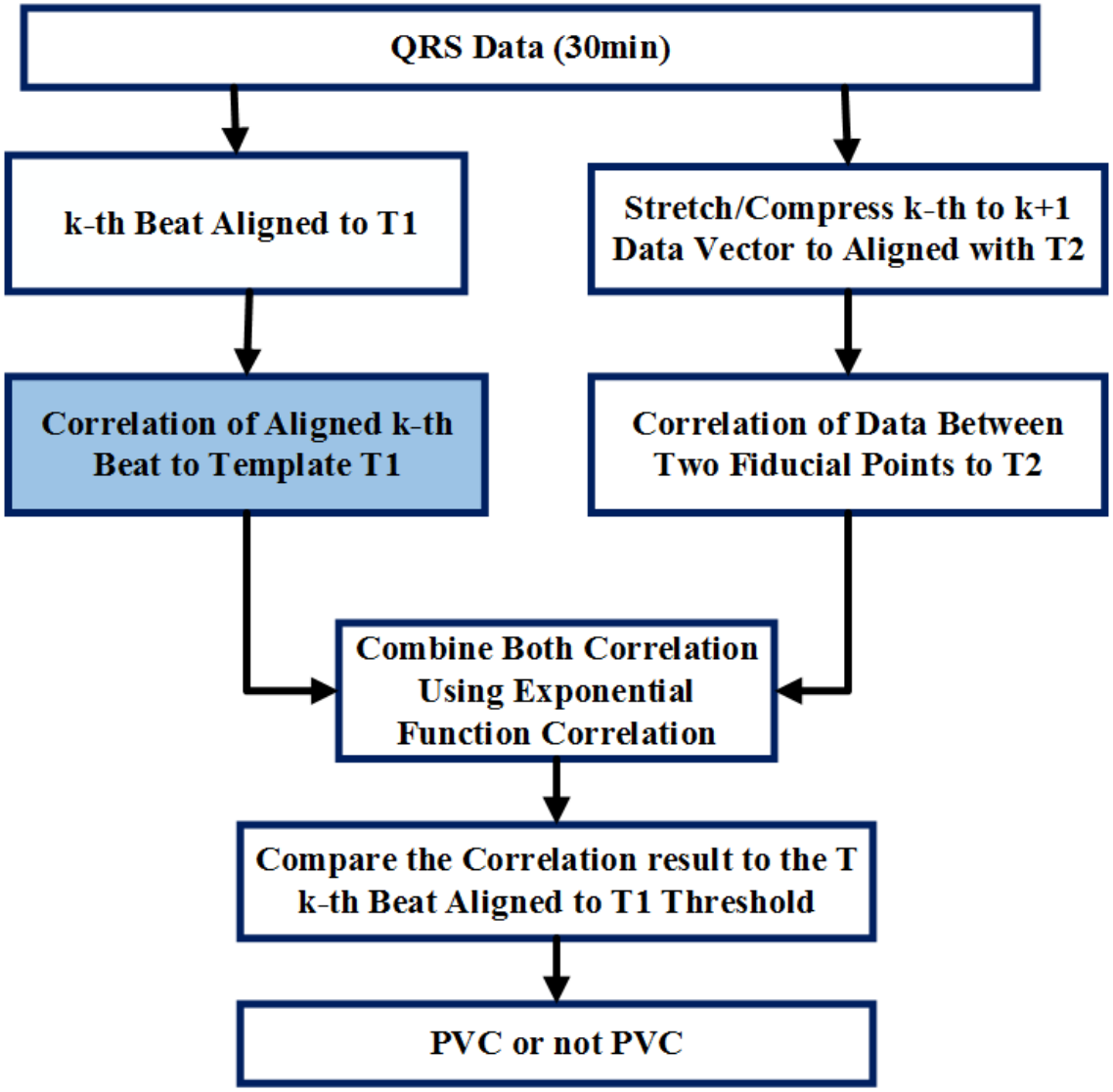


Figure 17. Flowchart for template-matching algorithm

Normalized Correlation Coefficient

$$x_k = \frac{\sum_{n=0}^{L-1}\left[b_k(n) - \bar{b}_k\right]\left[N(n) - \bar{N}\right]}{\sqrt{\sum_{n=0}^{L-1}\left[b_k(n) - \bar{b}_k\right]^2\left[N(n) - \bar{N}\right]^2}}$$

Where $b_k(n)$ is the QRS complex in the kth beat and L is the length of the predetermined

T1 (N(n))

Template-Matching Algorithm

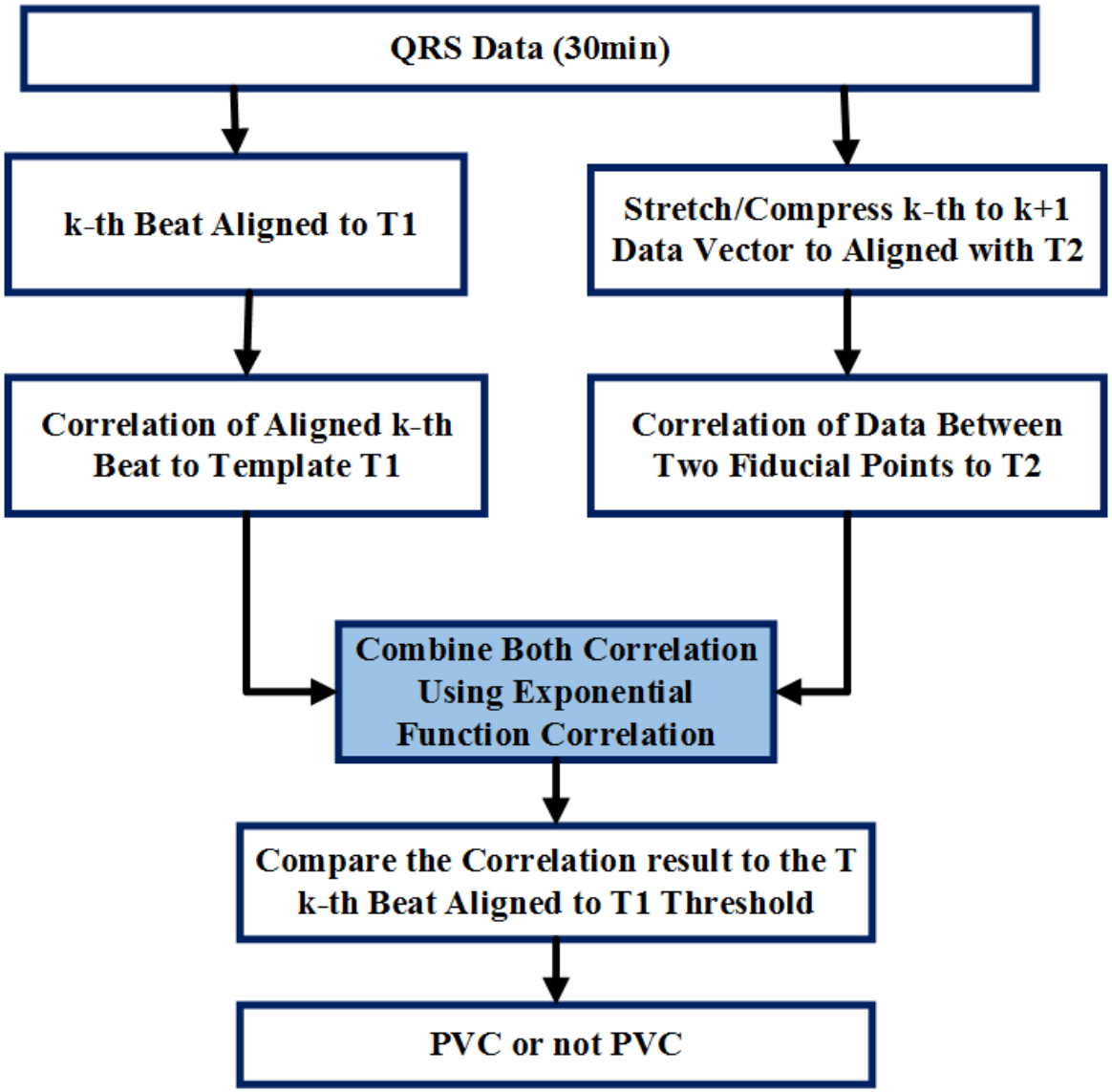


Figure 18. Flowchart for template-matching algorithm

Exponential Function Correlation

$$z_k = f\left(x_{k,}y_k\right) = \frac{\left(e^{{x_k}^r} + e^{{y_k}^r}\right)}{2e}$$

Where r determines the increasing rate of the slope and $z_k \geq z_{thre}$.

Schedule

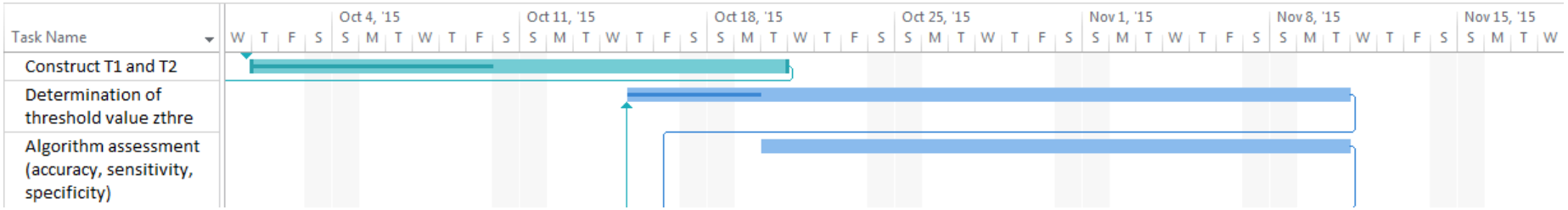


Figure 19. Gantt chart for fall semester

Future Schedule

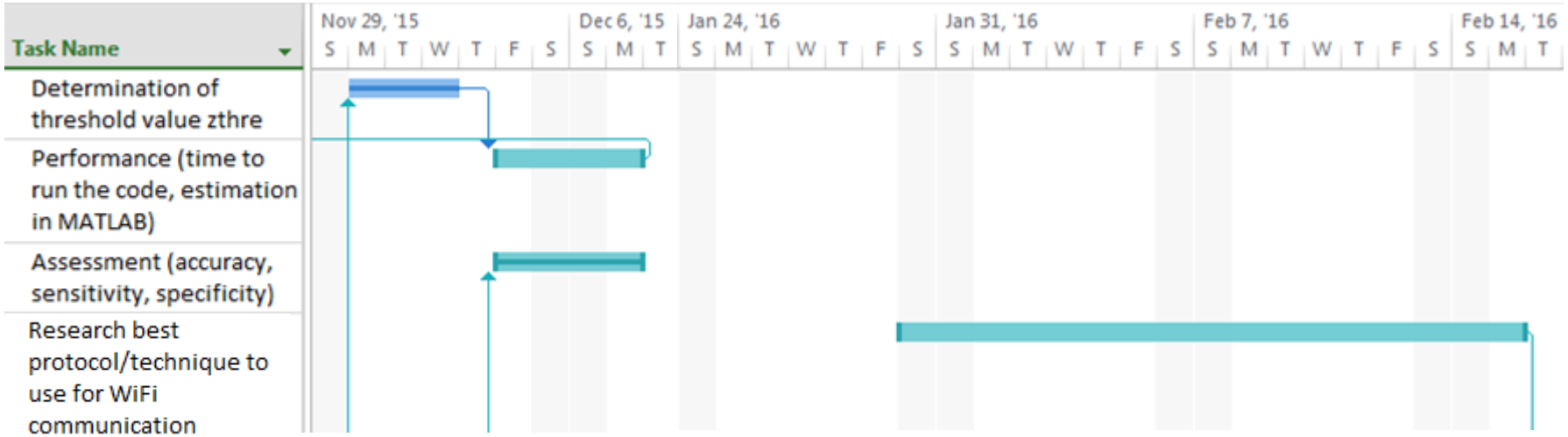


Figure 20. Gantt chart up to the second progress presentation

Contents

Template-Matching Algorithm

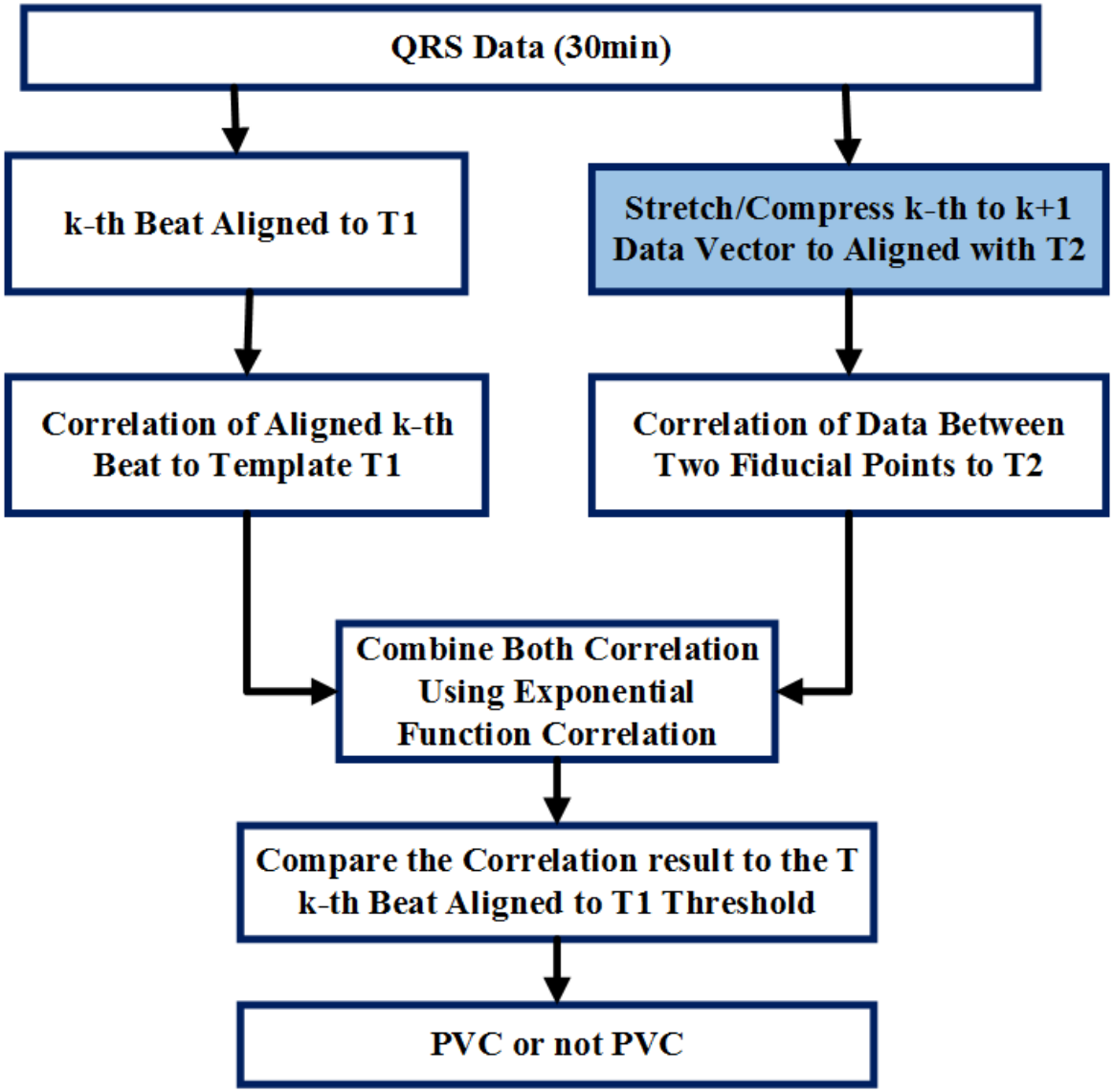


Figure 21. Flowchart for template-matching algorithm

Data between Two Fiducial Points

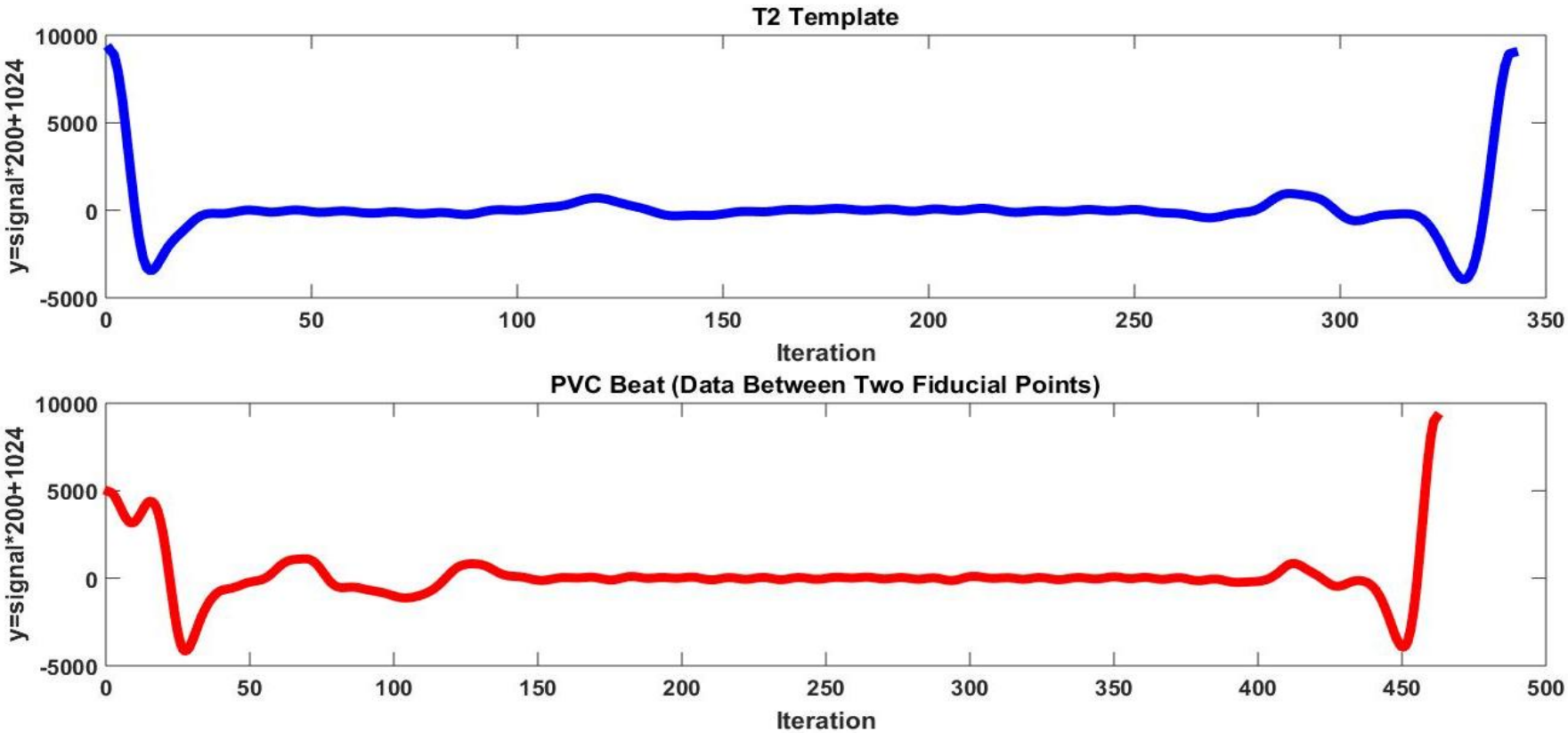


Figure 22. T2 (interval between R peaks) and RR-interval signal for PVC beat

Interpolation and Extrapolation

- Let b_k be the data between k^{th} beat and $k+1$ beat.
- If length(b_k)$\leq$ Length($T2$), we do extrapolation based on

$$new\_b_k = b_k[1 + \alpha(n-1)]\ (n = 1,2, \dots L)$$

$$\alpha = \frac{length(b_k)}{length(T_2)}$$

Example (Extrapolation)

$$L_k = 10, \quad L = 21, \quad \alpha_k = \frac{10}{21} \approx 0.476$$

TABLE I. INDICES AFTER EXTRAPOLATION

n value	Calculated Index
1	1
2	1.48
3	1.95
4	2.43
5	2.90

Algorithm Efficacy, 100s

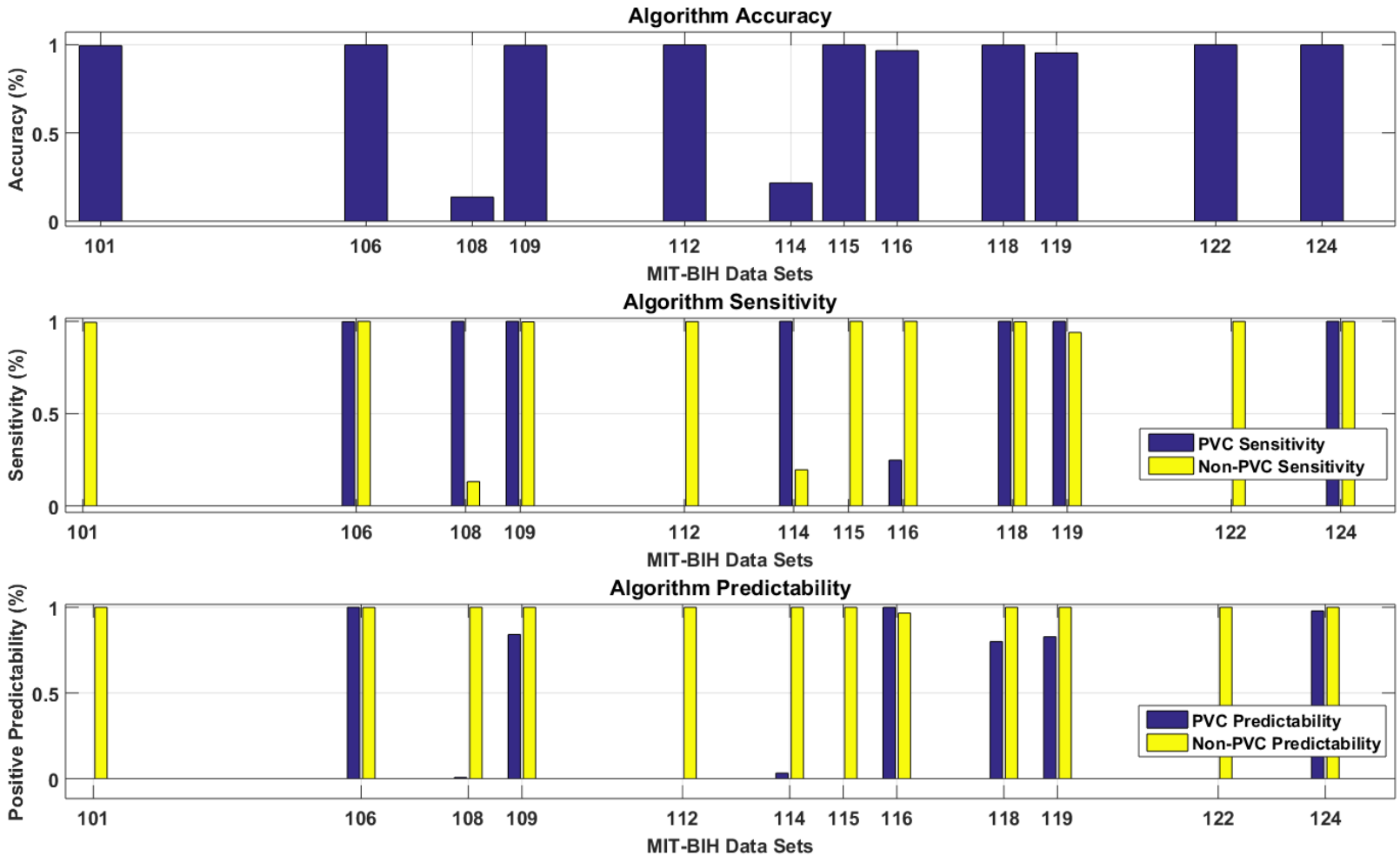


Figure 23. Performance of template-matching algorithm MATLAB simulation

Alternative QRS Detection [5]

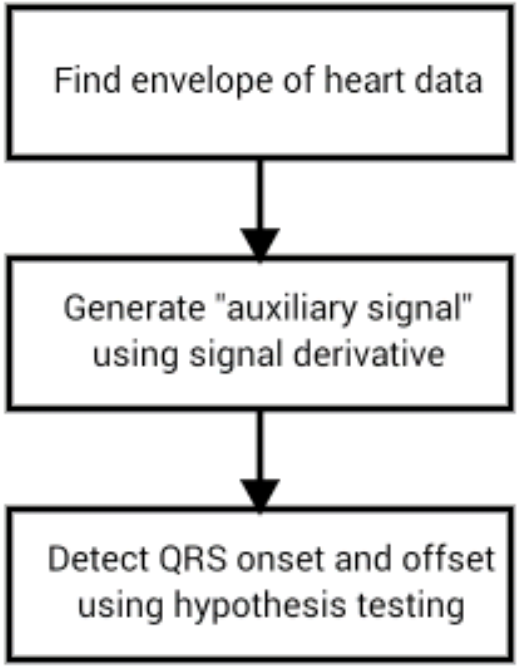


Figure 24. Flowchart for alternative QRS detection method

Converting MATLAB to C

- Template-Matching Functions
  - `normIntRange.m`
  - `normal_amplt.m`
  - `Corr_Coeff.m`
  - `Z_Corr.m`

Schedule

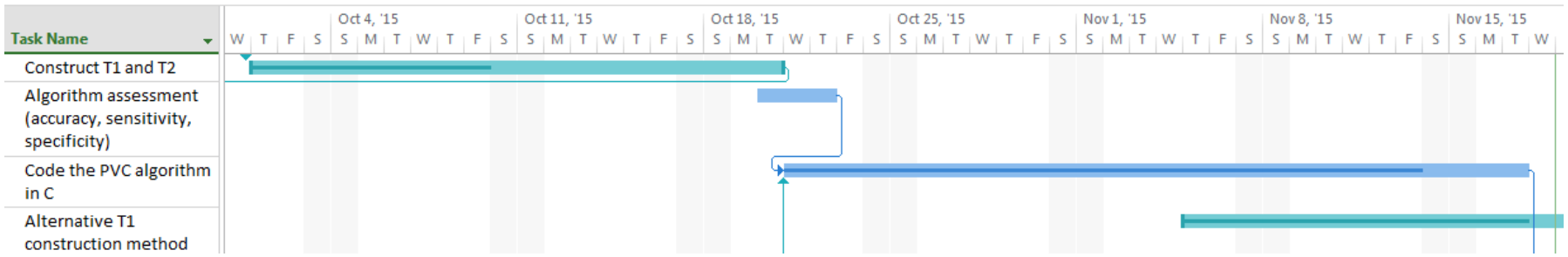


Figure 25. Gantt chart for fall semester

Future Schedule

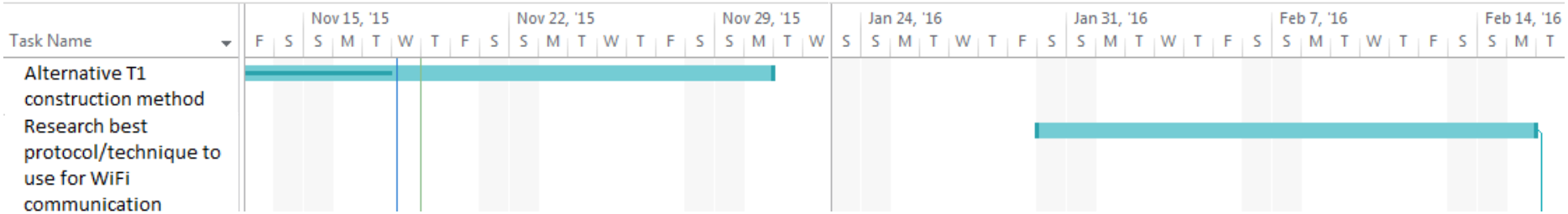


Figure 26. Gantt chart up to the second progress presentation

Contents

Template Generation Algorithm

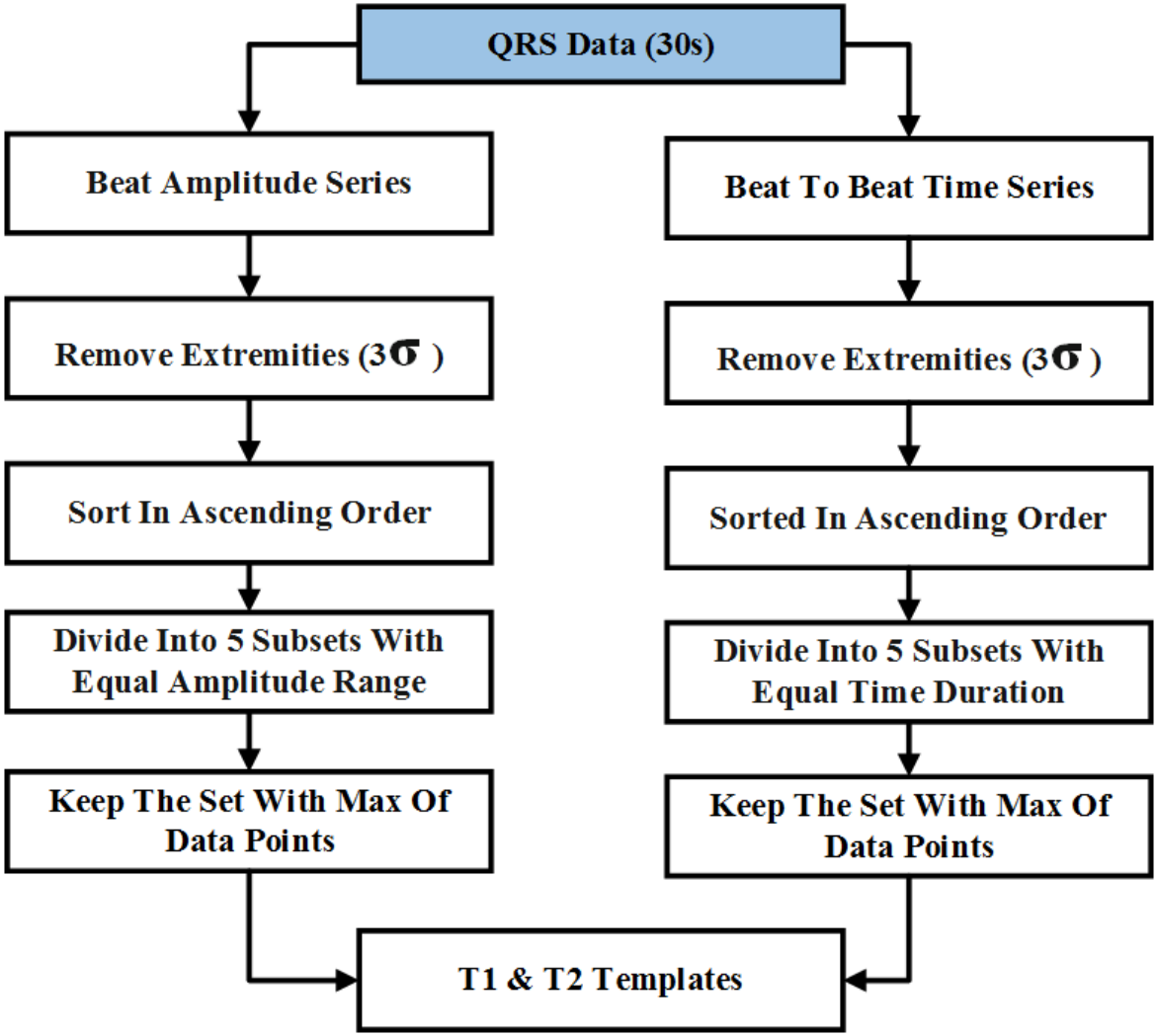


Figure 27. Flowchart for T1 and T2 generation

Testing of QRS Detection Phase

- Pan-Tompkins algorithm implemented on CC3200

Figure 28. CC3200 LaunchPad [6]

UDMA Diagram

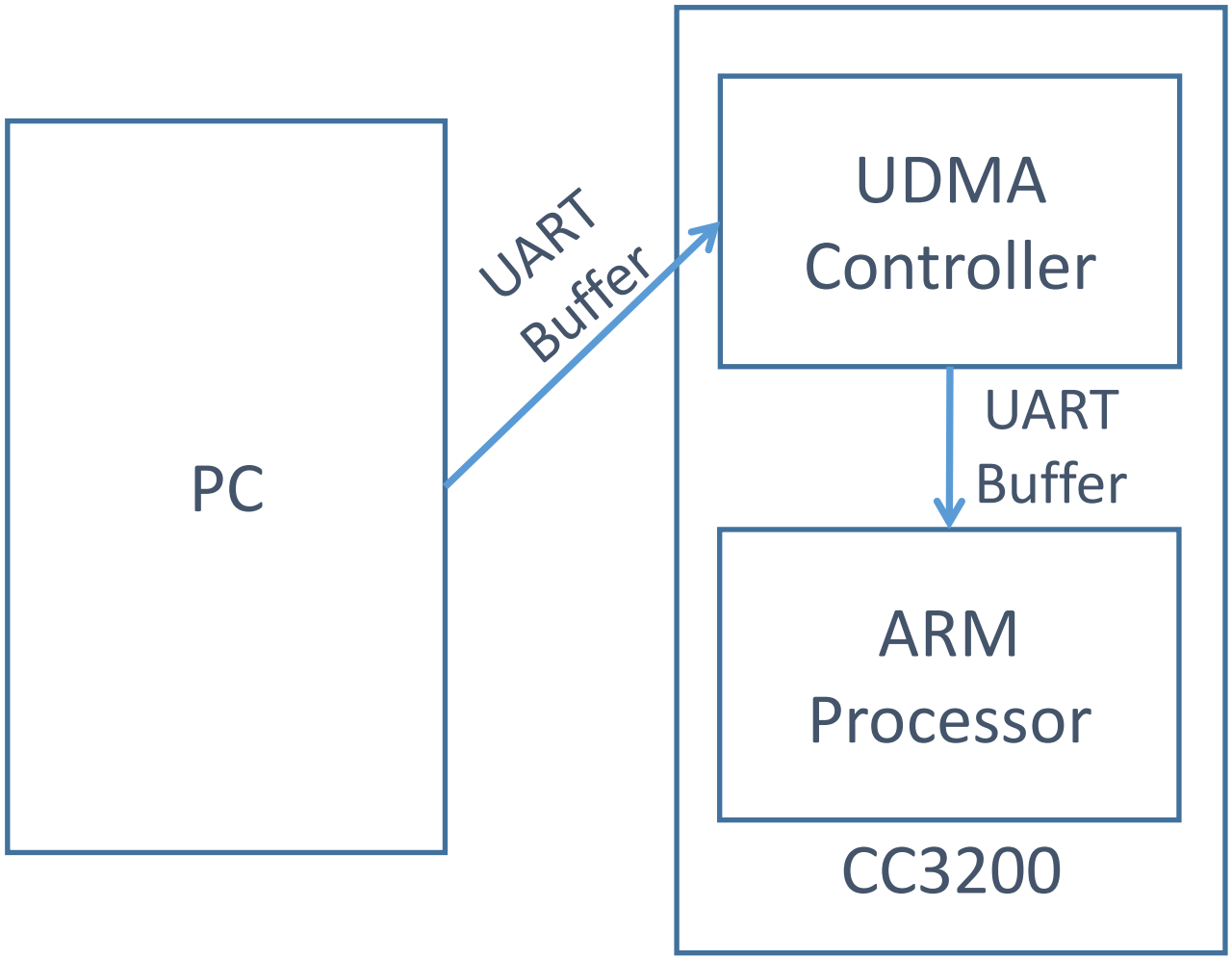


Figure 29. UART data flow between the PC and the CC3200

UART Data Transfer

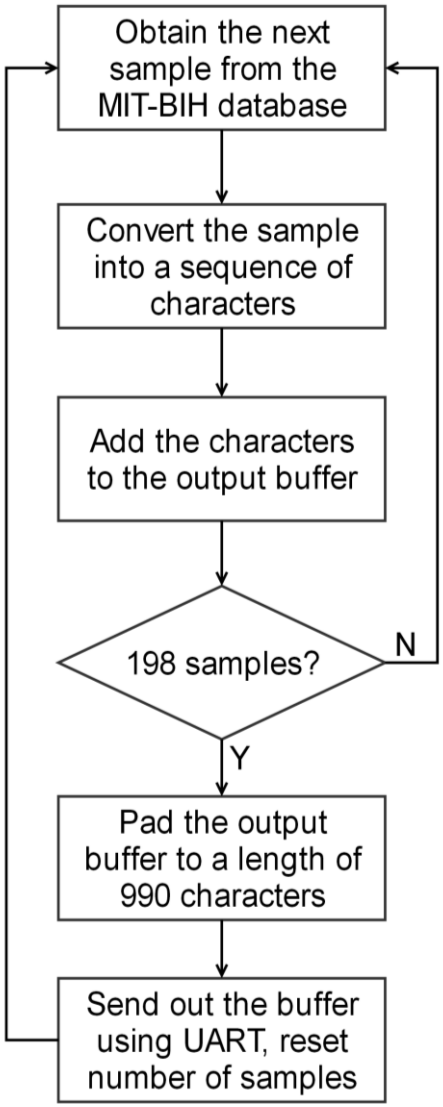
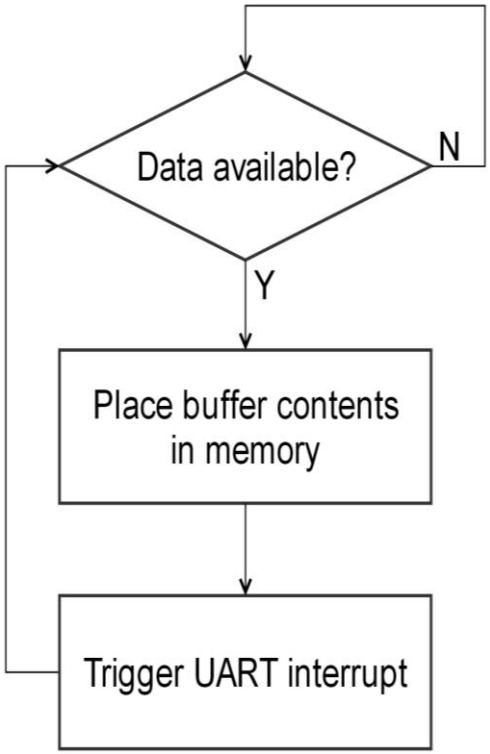


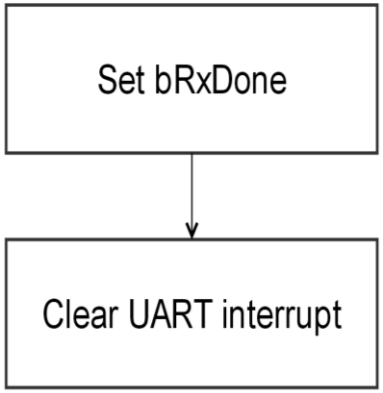
Figure 30. UART data transfer flowchart (PC)

UART Data Transfer

UDMA Controller



UART Interrupt



Main Function

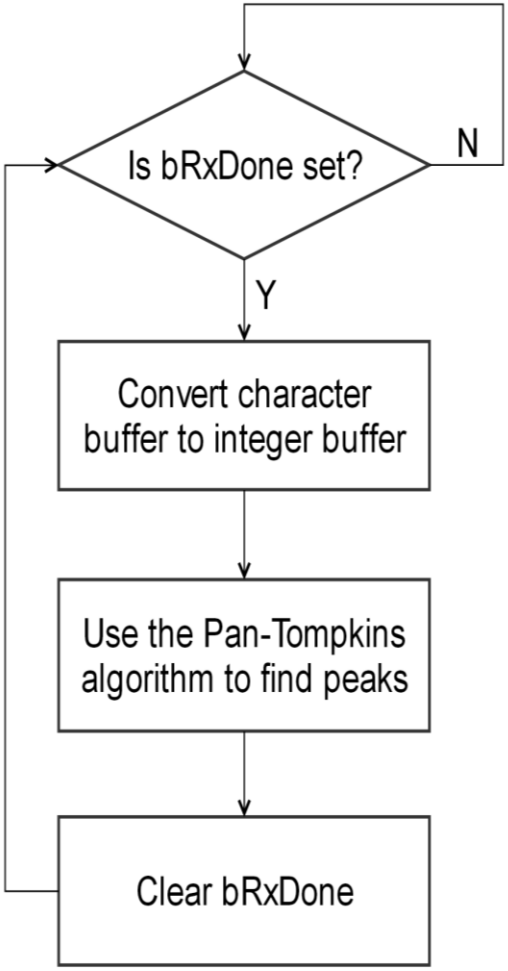


Figure 31. UART data transfer flowcharts (CC3200)

Pan-Tompkins Algorithm

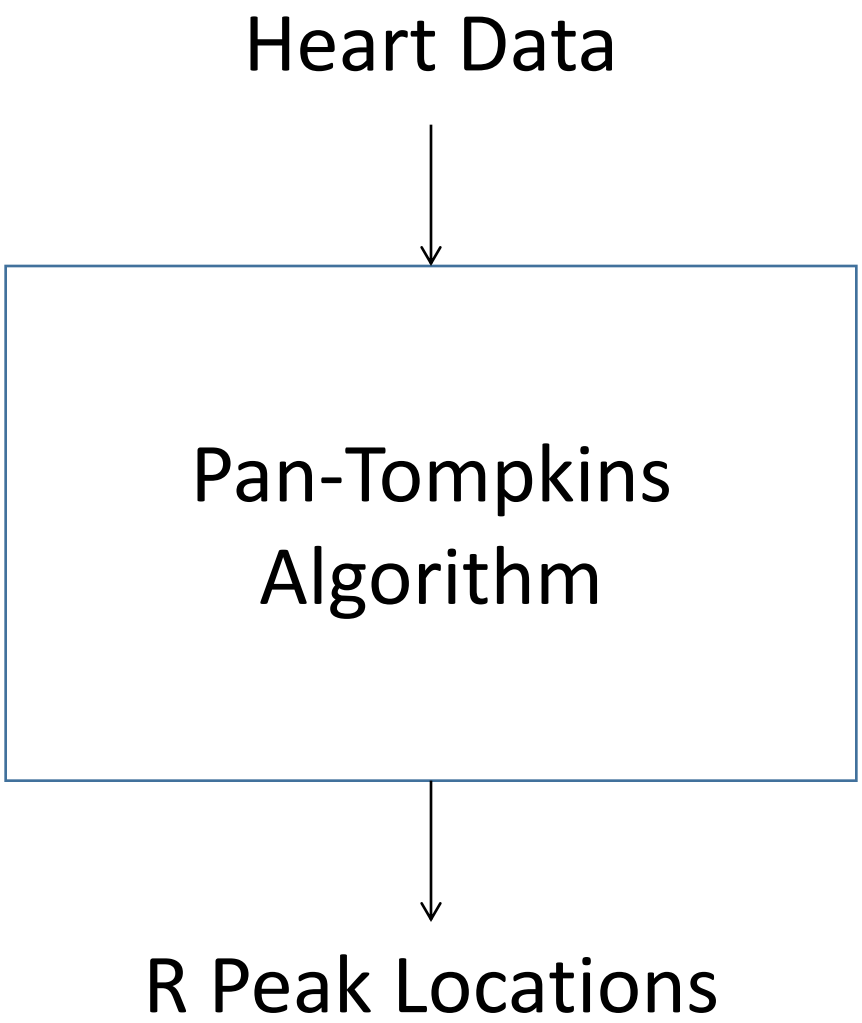


Figure 32. Pan-Tompkins block diagram

Pan-Tompkins Algorithm

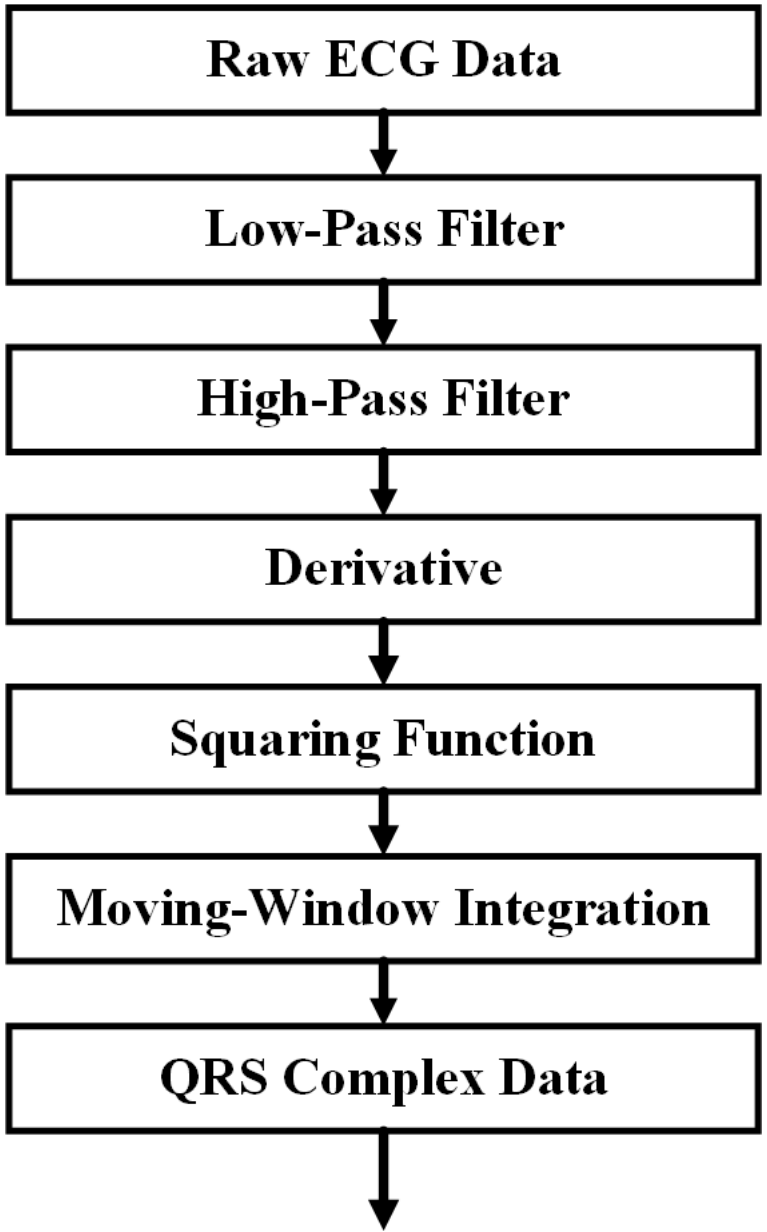


Figure 33. Flowchart for Pan-Tompkins algorithm

Results

TABLE II. PERFORMANCE OF PAN-TOMPKINS C IMPLEMENTATION

Record	QRS Sensitivity (%)	QRS Positive Predictivity (%)
100	97.83	100.00
102	96.69	100.00
103	99.43	100.00
105	99.76	99.76
106	96.67	99.69

Conversion of MATLAB Code

- Conversion of template-matching MATLAB code to C code
  - `Normal_amplt.m`
  - Integrated code project

C Implementation Results

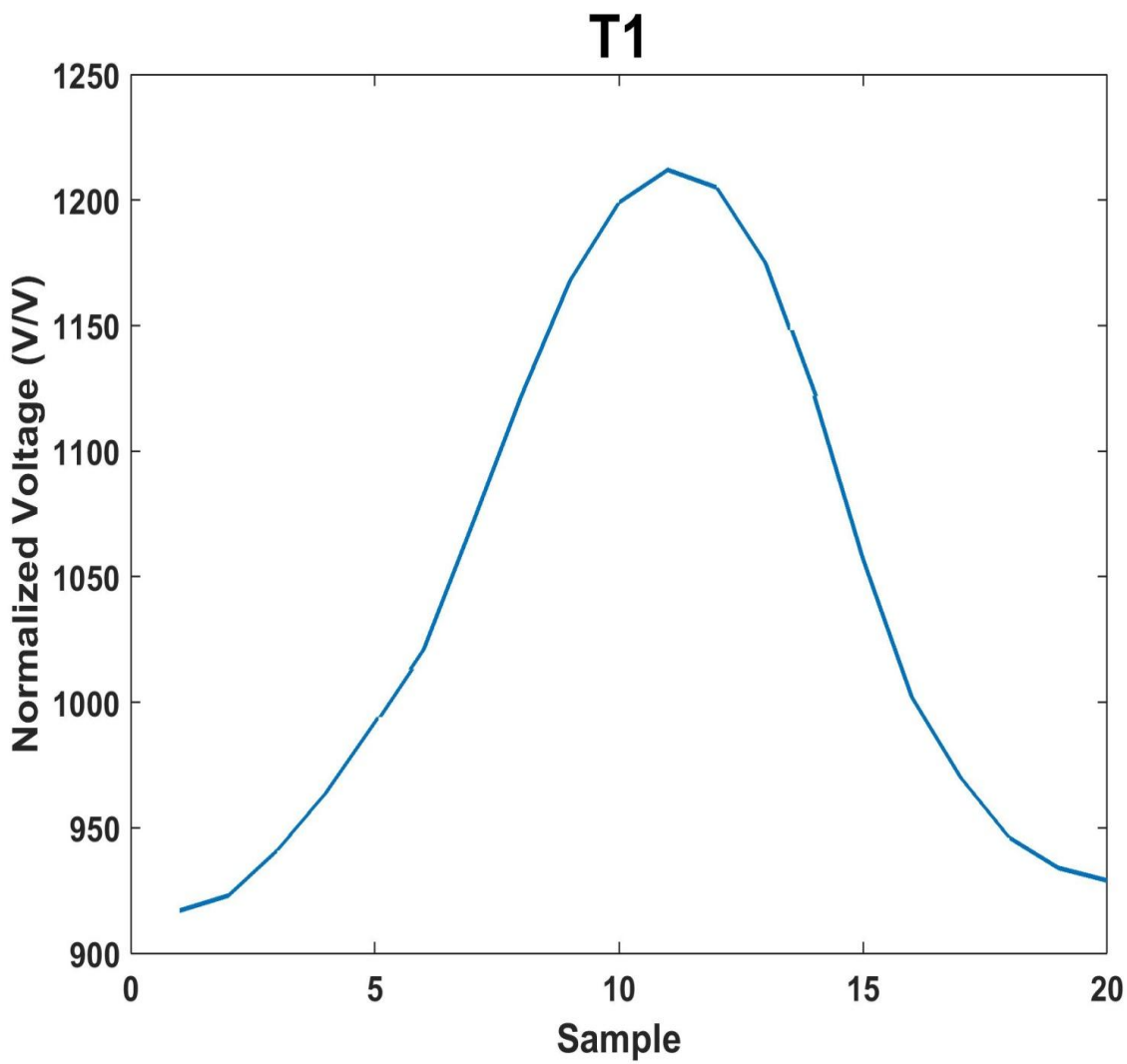


Figure 34. T1 (QRS complex) template generated on the CC3200

C Implementation Results

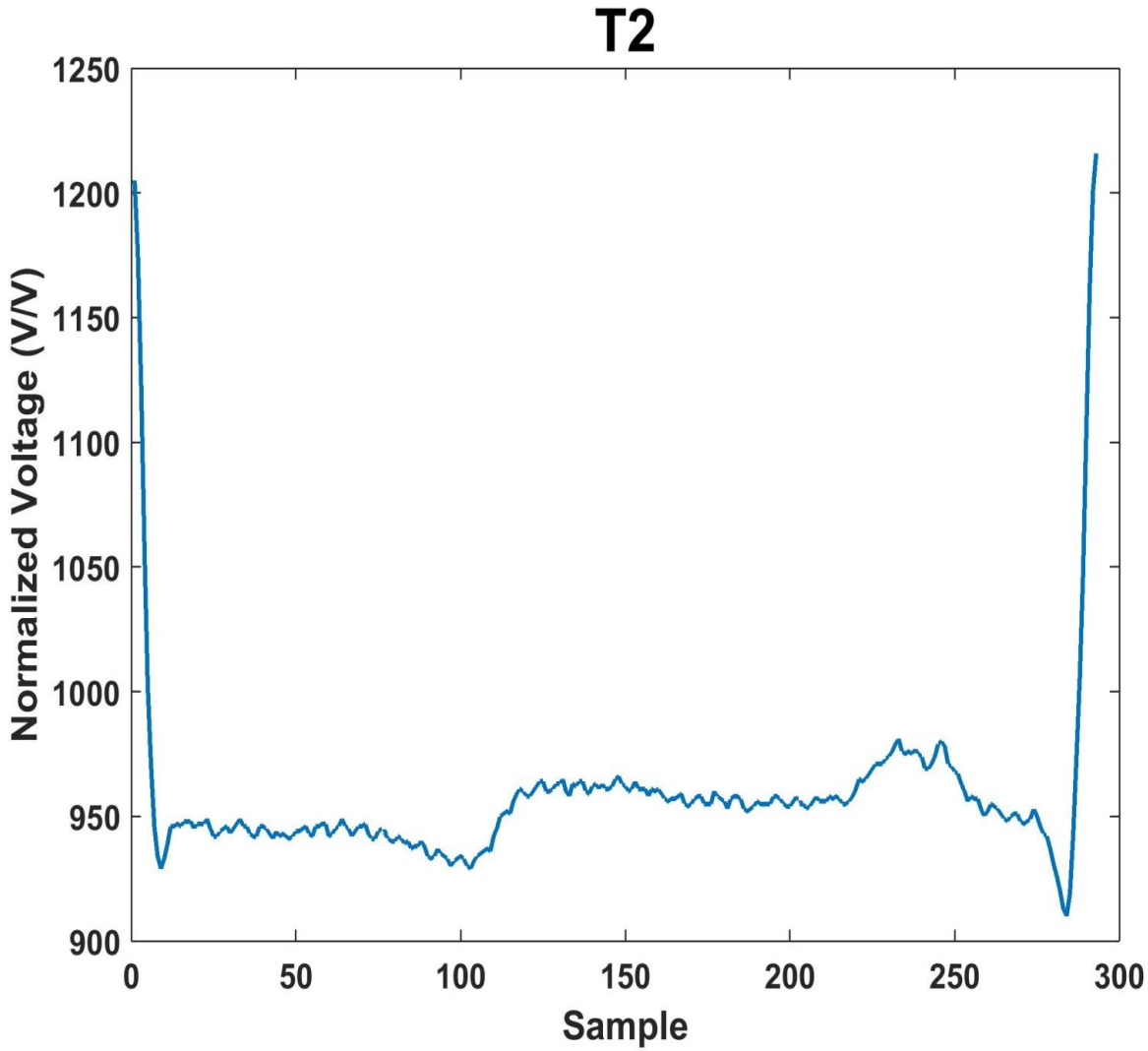


Figure 35. T2 (RR interval) template generated on the CC3200

Schedule

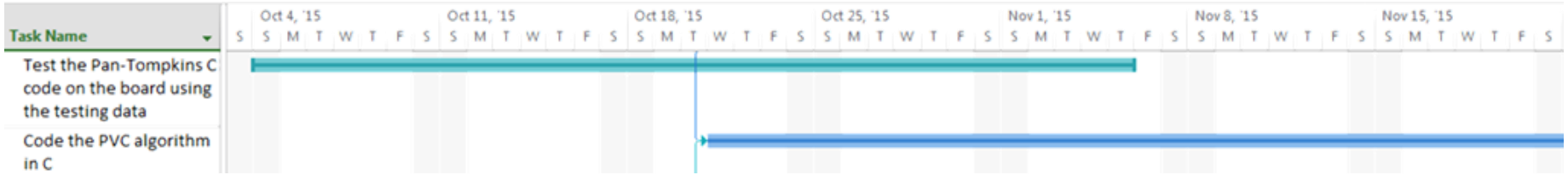


Figure 36. Gantt chart for fall semester

Future Schedule

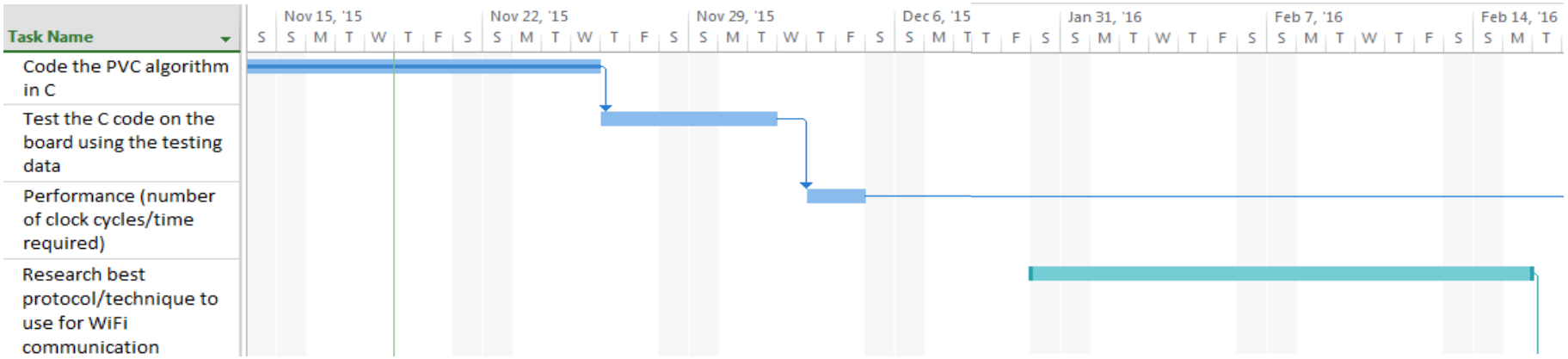


Figure 37. Gantt chart up to the second progress presentation

Contents

Summary and Conclusions

- PVCs are irregular heartbeats that may lead to VT
- An embedded device is proposed that will detect PVCs in real time and wirelessly alert the patient’s doctor of VT

Summary and Conclusions

- Design must include real-time ECG signal processing, on-board signal processing computations, and battery-powered functionality

Summary and Conclusions

- Current Tasks
  - MATLAB simulation of the PVC detection method
  - C implementation of completed PVC detection code

Summary and Conclusions

- Future Tasks
  - Complete simulation and testing of all MATLAB code
  - Test the integrated template-matching/Pan-Tompkins code
  - Research wireless communication methods

Real-time Heart Monitoring and ECG Signal Processing

Fatima Bamarouf, Claire Crandell, and Shannon Tsuyuki

Advisors: Drs. Yufeng Lu and Jose Sanchez

Department of Electrical and Computer Engineering
Bradley University
November 24, 2015

References

- [1] *Arrhythmias.* [Online] Available: http://watchlearnlive.heart.org/CVML_Player.php?moduleSelect=arrhyt
- [2] *MIT-BIH Arrhythmia Database.* [Online] Available: http://www.physionet.org/physiobank/database/mitdb/
- [3] *Cardiovascular System Assessments.* [Online] Available: http://media.lanecc.edu/users/driscolln/RT116/softchalk/Cardia_Assessment/Cardia_Assessment_print.html
- [4] P. Li, et al., "A low-complexity data-adaptive approach for premature ventricular contraction recognition," *Signal, Image and Video Processing*, vol. 8, no. 1, pp. 111-120, 2013. [Online] Available: http://link.springer.com/article/10.1007%2Fs11760-013-0478-6
- [5] A. Illanes-Manriquez and Q. Zhang, "An Algorithm for Robust Detection of QRS Onset and Offset in ECG Signals," [Online] Available: http://www.cinc.org/Proceedings/2008/pdf/0857.pdf
- [6] *CC3200-LAUNCHXL.* [Online] Available: http://www.ti.com/ww/en/launchpad/launchpads-connected-cc3200-launchxl.html?DCMP=cc3100cc3200&HQS=cc3200launchpad-oob
- [7] B. Ribeiro, et al., "Choosing Real-Time Predictors for Ventricular Arrhythmia Detection," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 21, no. 08, pp. 1249-1263, 2007. [Online] Available: https://eden.dei.uc.pt/~bribeiro/FCT_files_2006/LNCS_ICONIP2006.pdf
- [8] *QRS Detection Algorithm Using Savitzky-Golay Filter. [Online] Available: https://hal.archives-ouvertes.fr/file/index/docid/747954/filename/550.pdf*

Envelope Signal [5]

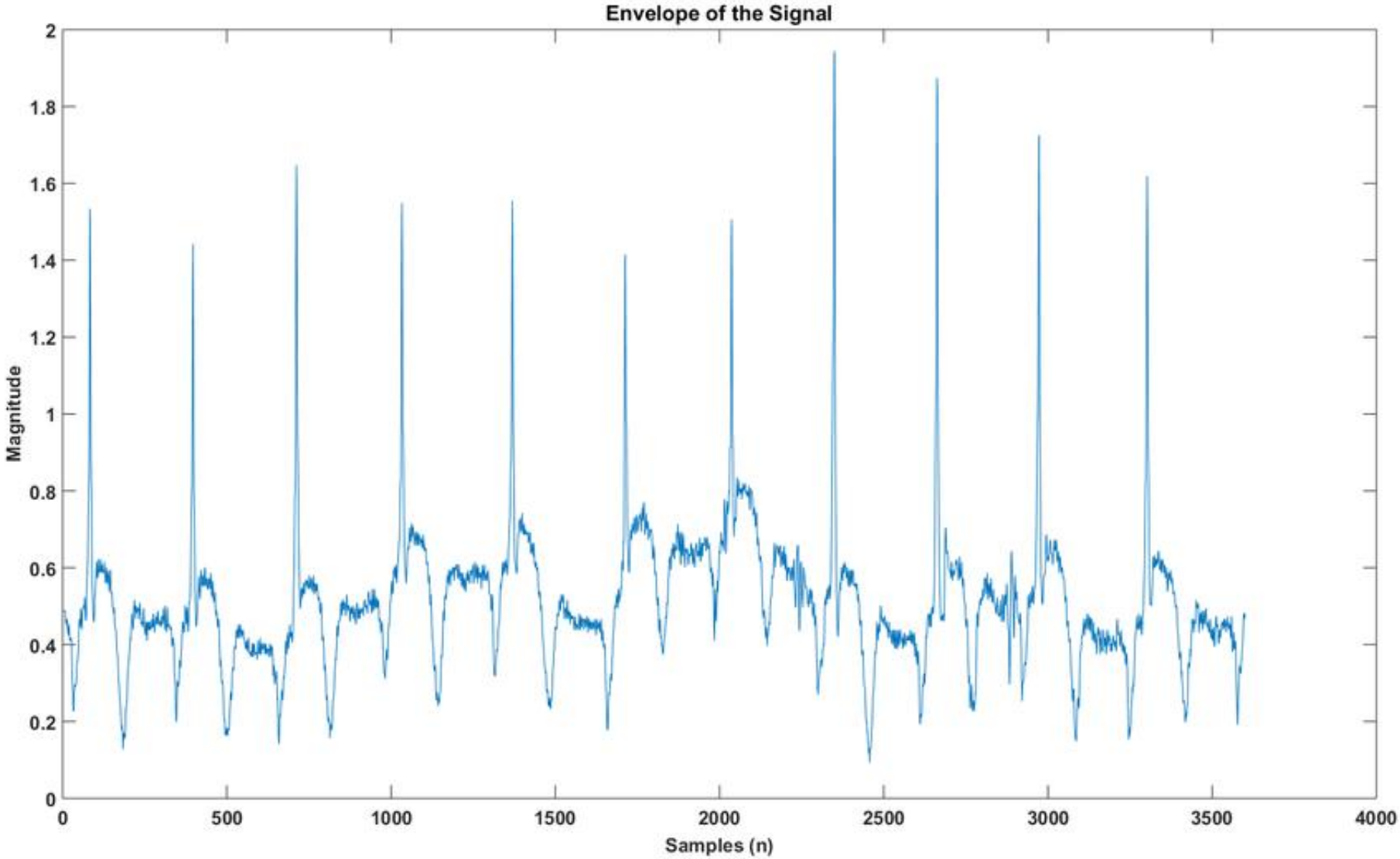


Figure 38. Envelope of the initial ECG signal

Auxiliary Signal [5]

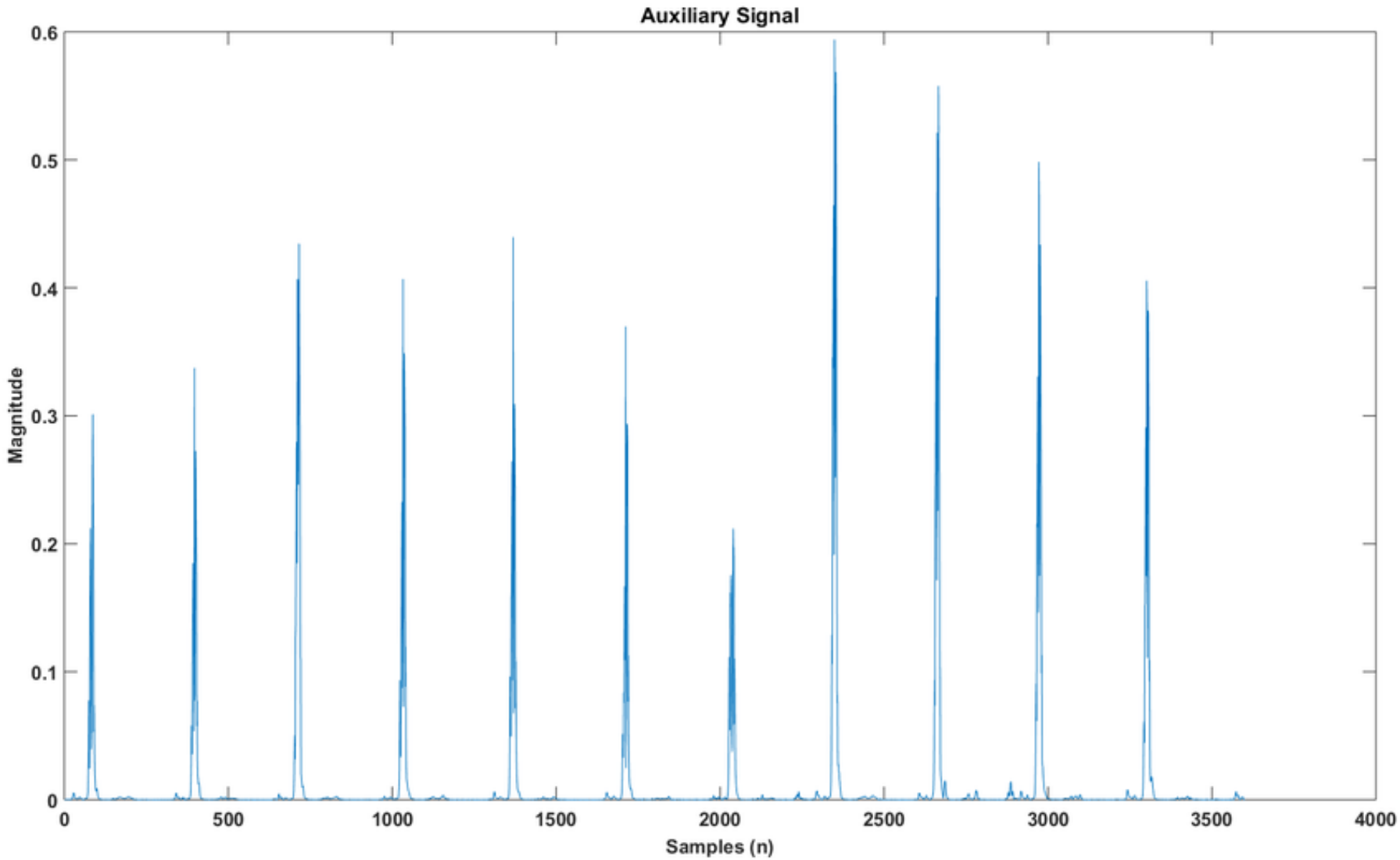


Figure 39. Auxiliary signal of the envelope

State Diagram

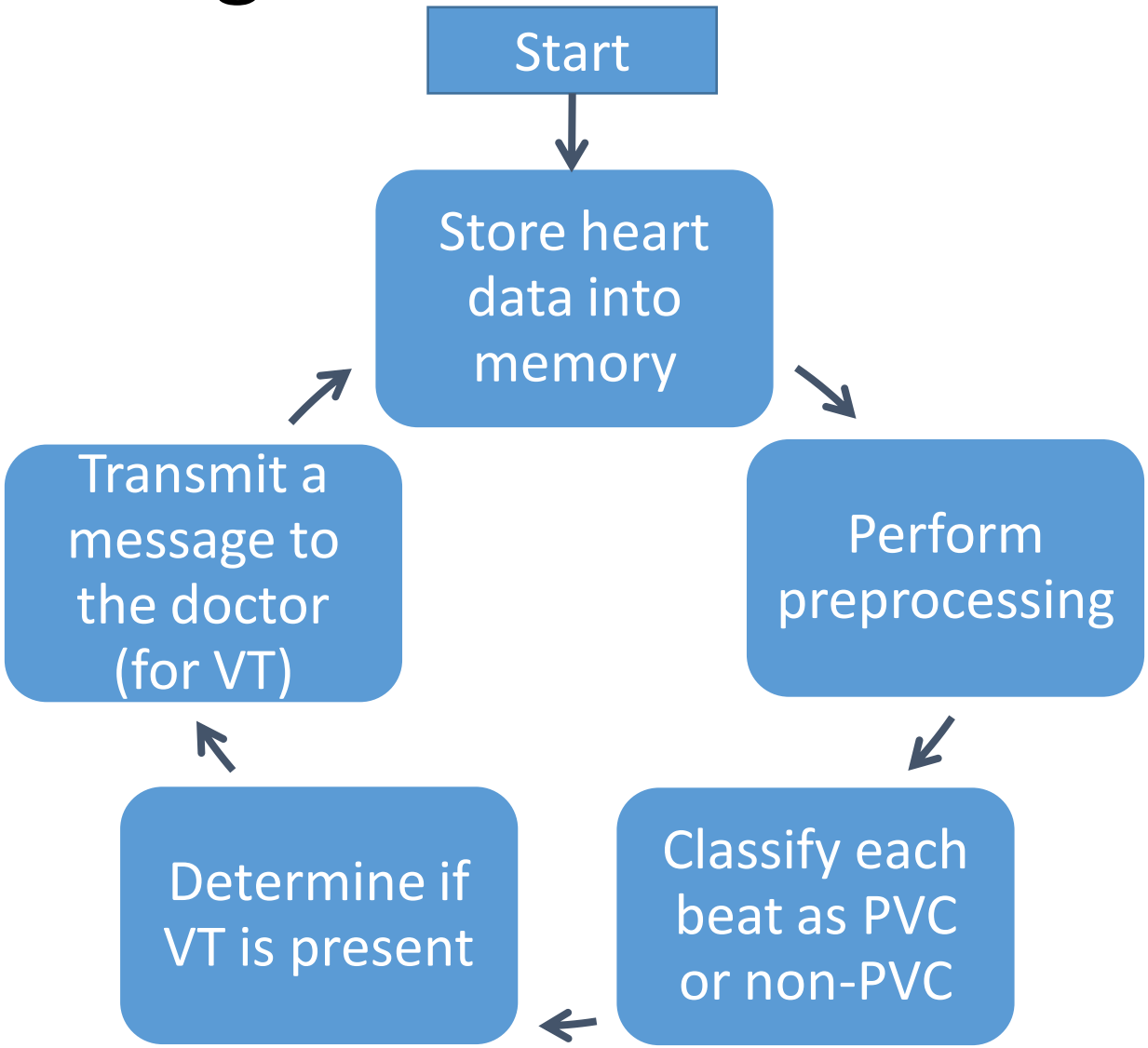


Figure 40. State diagram for heart monitoring system

Detailed Gantt Chart (1)

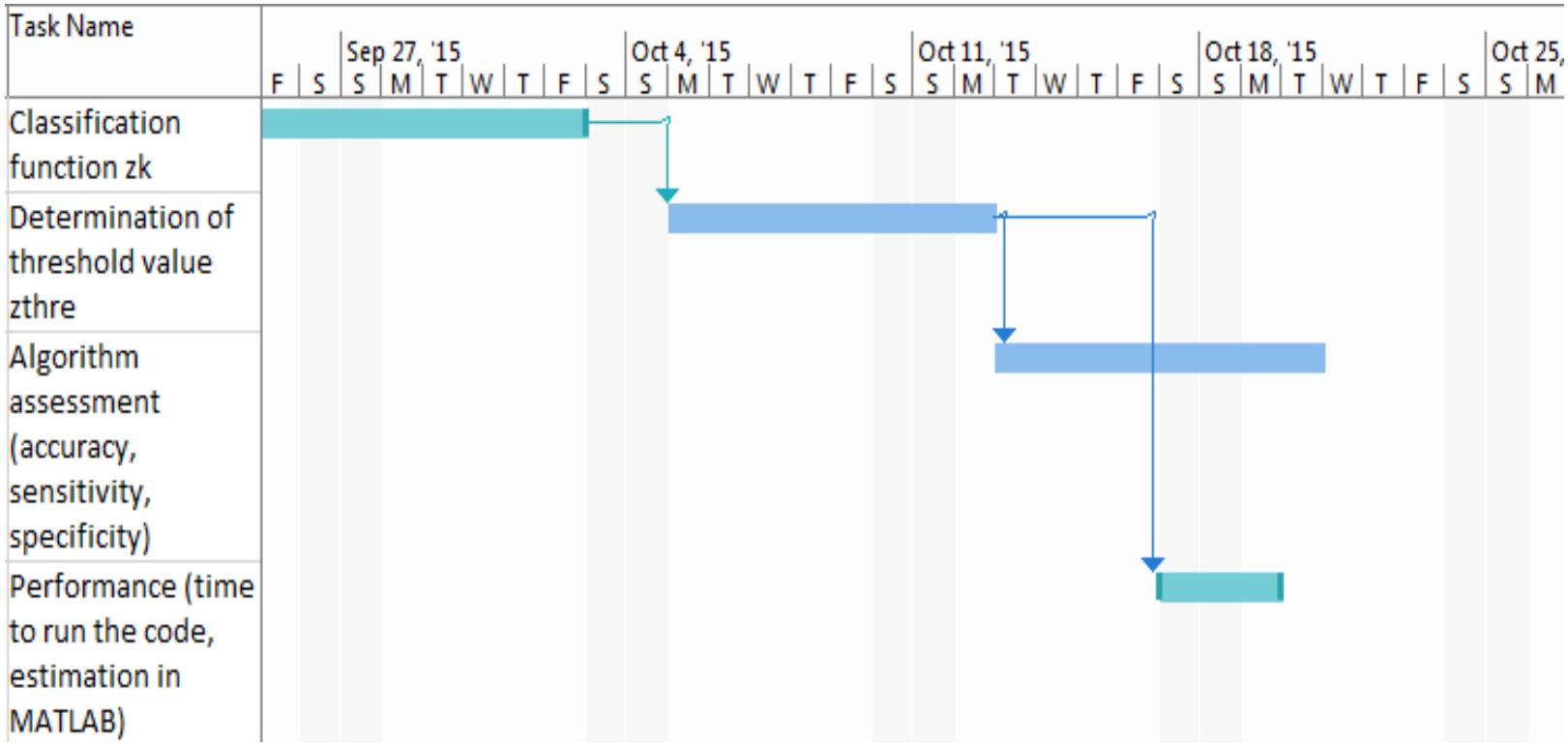


Figure 41. Gantt chart for the MATLAB simulation (PVC algorithm) phase of the project

Detailed Gantt Chart (2)

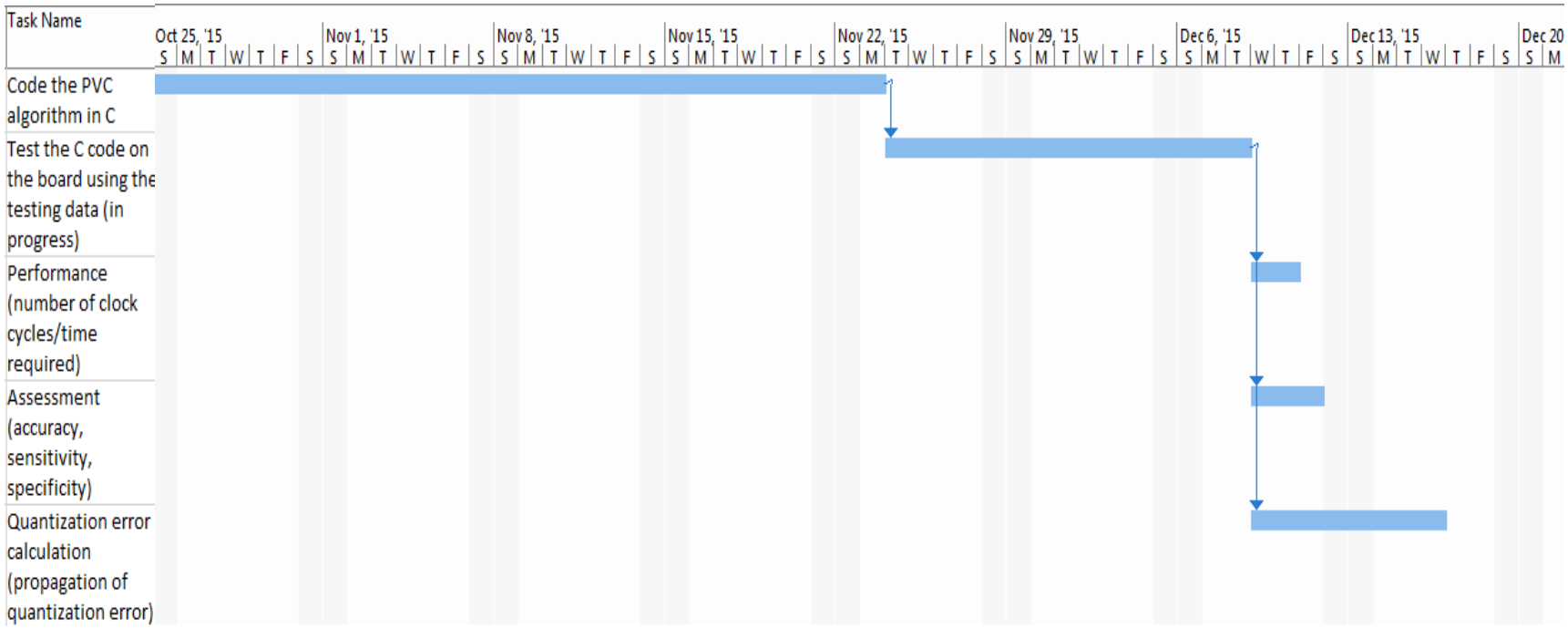


Figure 42. Gantt chart for the C implementation (PVC algorithm) phase of the project

Detailed Gantt Chart (3)

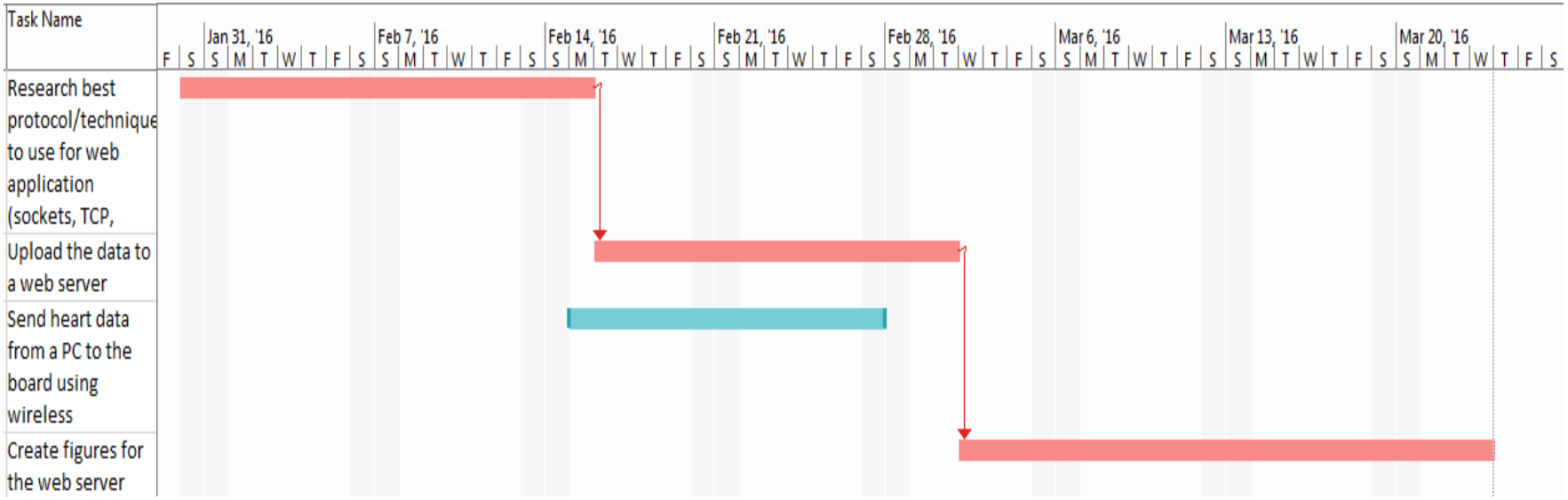


Figure 43. Gantt chart for the wireless development phase of the project

Specificity and Sensitivity [7]

$$SP = \frac{TN}{TN + FP} \qquad SE = \frac{TP}{TP + FN}$$

- TP (True Positive): detected QRS complex that is present in the signal
- TN (True Negative): data point between QRS complexes that does not contain a QRS peak
- FP (False Positive): incorrect identification of QRS peak
- FN (False Negative): QRS peak that was not detected by the algorithm

Memory Requirements

- Sampling rate for ECG signal (MIT-BIH arrhythmia database): 360 Hz
- Number of samples required for 30 seconds of ECG data: 10,800
- Amount of memory required: 21 kB

Schedule

TABLE III. PROJECT SCHEDULE

Task	Duration (hours)
PVC Algorithm (MATLAB)	65
PVC Algorithm (C)	100
Wi-Fi Communication	150
Progress Report I	80
Progress Report II	80
Final Presentation	80
Final Report	80

Schedule

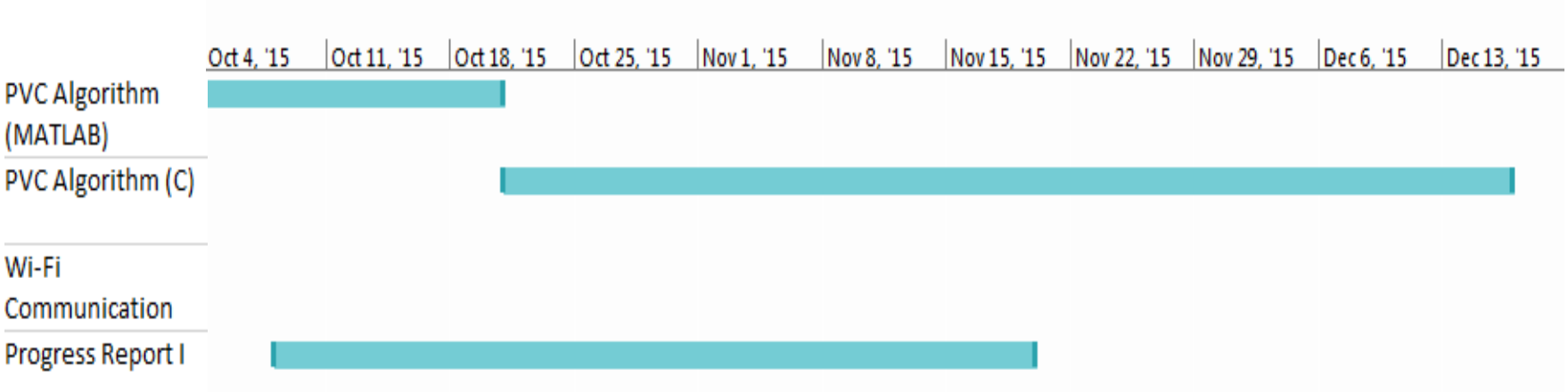


Figure 44. Gantt chart for the fall semester

Schedule

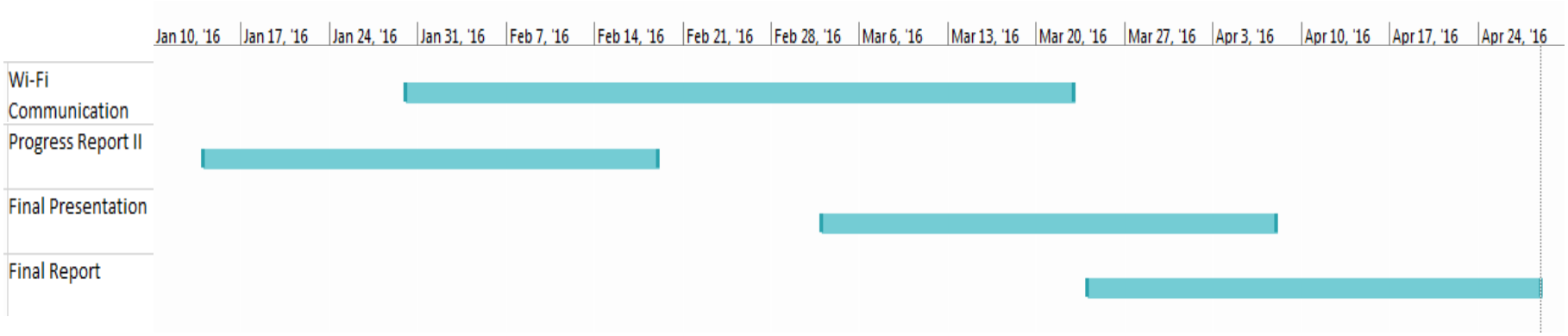


Figure 45. Gantt chart for the spring semester

Pan-Tompkins Algorithm [8]

1. Low-pass Filter
   - 11 Hz cut-off frequency
   - 5-sample delay
   - Gain of 36

$$y(n) = 2y(n-1) - y(n-2) + x(n) - 2x(n-6) + x(n-12)$$

2. High-pass Filter
   - 5 Hz cut-off frequency
   - 29-sample delay
   - Gain of 1

$$y(n) = y(n-1) - \frac{1}{32}x(n) + x(n-16) - x(n-17) + \frac{1}{32}x(n-32)$$

Pan-Tompkins Algorithm

3. Derivative
   - Provides information about QRS slope
   - Approximates derivative from 0-30 Hz
   - Has a 4-sample delay

$$y(n) = \frac{1}{8}[2x(n) + x(n-1) - x(n-3) - 2x(n-4)]$$

4. Squaring Function
   - Emphasizes higher frequencies of the ECG (caused by QRS complexes)

$$y(n) = x^2(n)$$

Pan-Tompkins Algorithm

5. Moving-Window Integration
   - Detects long-duration and large-amplitude QRS complexes

$$y(nT) = \frac{1}{N}[x(nT - (N-1)T) + x(nT - (N-2)T + \cdots + x(nT)]$$

Algorithm Efficacy, 200s

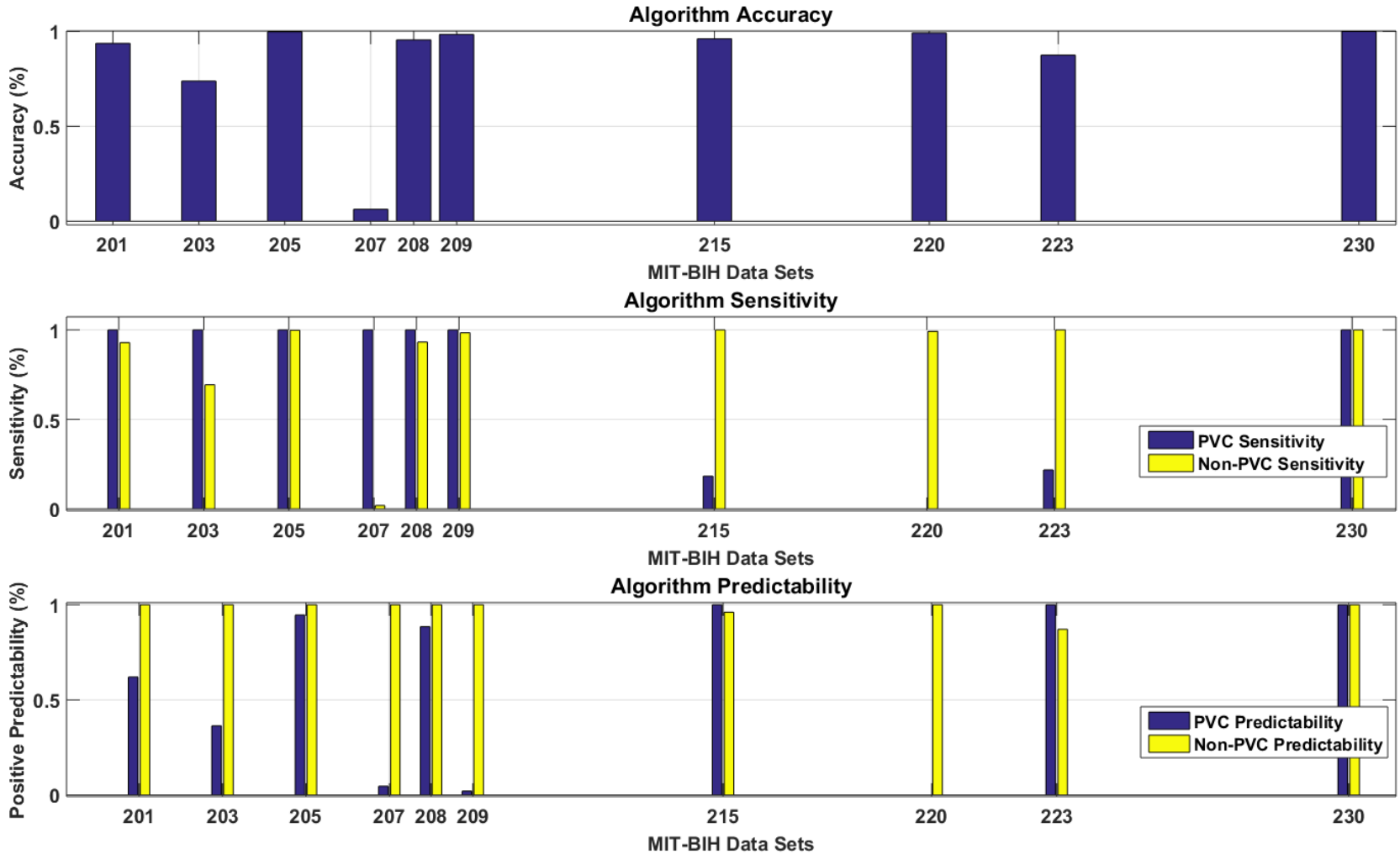


Figure 47. Performance of template-matching algorithm MATLAB simulation

WFDB Library (PC side)

- `isigopen()`: open a specific WFDB record
- `getvec()`: get the next sample in the record

WFDB Library (PC side)

- `isigopen()`: open a specific WFDB record
- `getvec()`: get the next sample in the record

WFDB Toolbox (MATLAB)

- `rdsamp()`: place samples from a WFDB record into a vector
- `rdann()`: place annotations (characters) from a WFDB record into a vector

WFDB Toolbox (MATLAB)

- `wrsann()`: write experimental annotations into a vector
- `bxb()`: generate a report (with accuracy and positive predictivity data) using experimental annotations

Sample BXB Report

```
Beat-by-beat comparison results for record mitdb/100
Reference annotator: atr
     Test annotator: test

                    Algorithm
          n     s     v     f     q     o     x
     ------------------------------------------
 N  |   356     0     0     0     0     8     0
 S  |     4     0     0     0     0     0     0
 V  |     0     0     0     0     0     0     0
 F  |     0     0     0     0     0     0     0
 Q  |     0     0     0     0     0     0     0
 O  |     0     0     0     0     0
 X  |     0     0     0     0     0

            QRS sensitivity:  97.83% (360/368)
 QRS positive predictivity: 100.00% (360/360)
            VEB sensitivity:      -  (0/0)
 VEB positive predictivity:       -  (0/0)
           SVEB sensitivity:   0.00% (0/4)
SVEB positive predictivity:       -  (0/0)
     RMS RR interval error: 168.87 ms
```

Figure 48. Text file generated using the `bxb()` function in the WFDB toolbox

DSP library for CC3200

- `Corr_Coeff.c`
  - `arm_sub_f32()`
  - `arm_mult_f32()`
- `Normal_amplt.c, normIntRange.c`
  - `arm_mean_f32()`
  - `arm_std_f32()`

Sorting the Amplitude Series

- Let S_amp=[2 1 10 5 9 1.5 3 0 1.7 5 7]
- After sorting,

  sort_series=[0 1 1.5 1.7 2 3 5 5 7 9 10]

QRS Detection

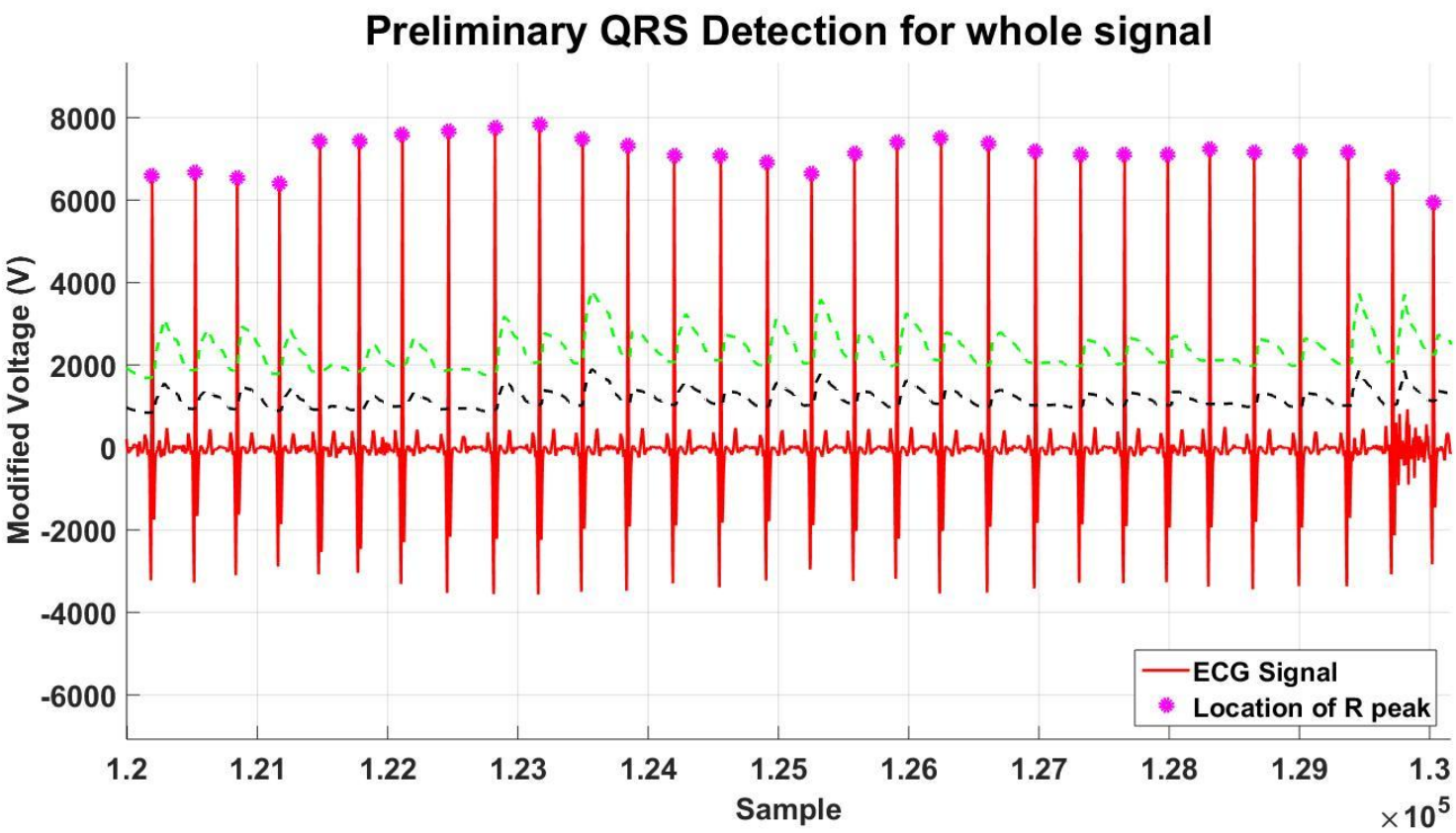


Figure 49. MATLAB plot of QRS detection results

Alternative QRS Detection [5]

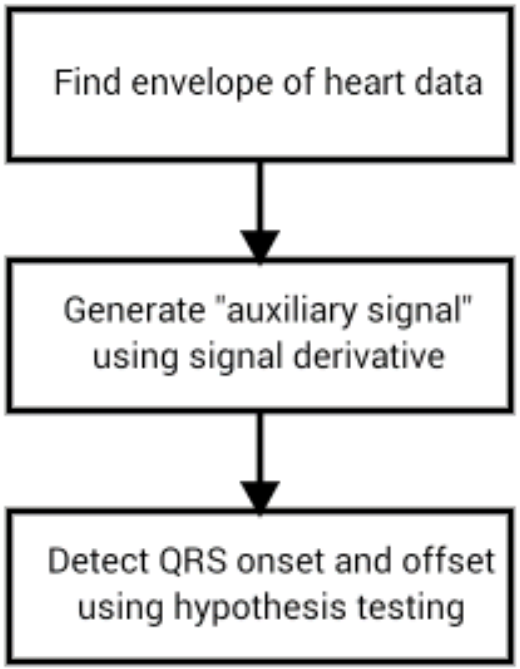


Figure 50. Flowchart for alternative QRS detection method

- Uses statistics to accurately locate QRS onset and offset
- Can be used to determine abnormal QRS complexes

Alternative QRS Detection

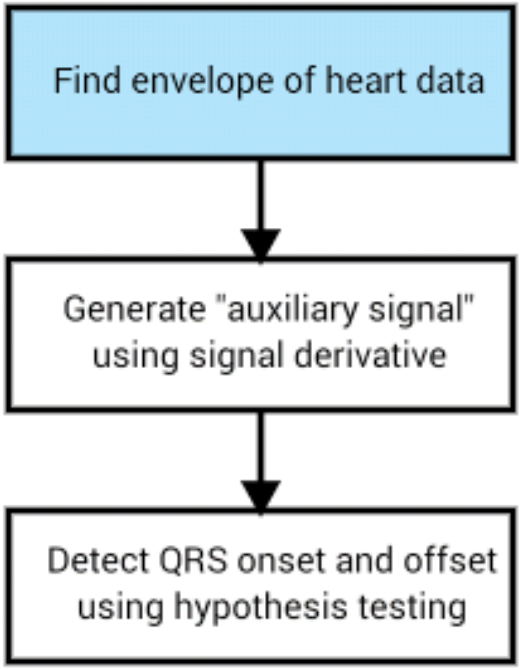


Figure 51. Flowchart for alternative QRS detection method

- Use the Hilbert transform to obtain the envelope

Alternative QRS Detection

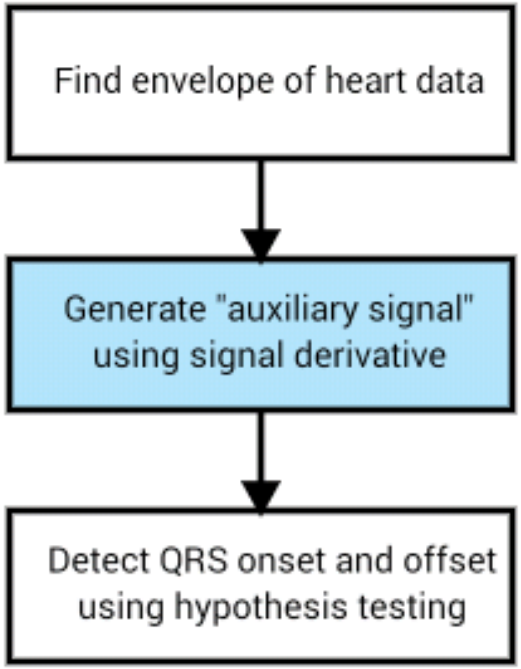


Figure 52. Flowchart for alternative QRS detection method

- Estimate the derivative using a parabola:

$$h'(k) = \frac{1}{10}\left(2\big(h(k+2r) - h(k-2r)\big) + h(k+r) - h(k-r)\right)$$

Alternative QRS Detection

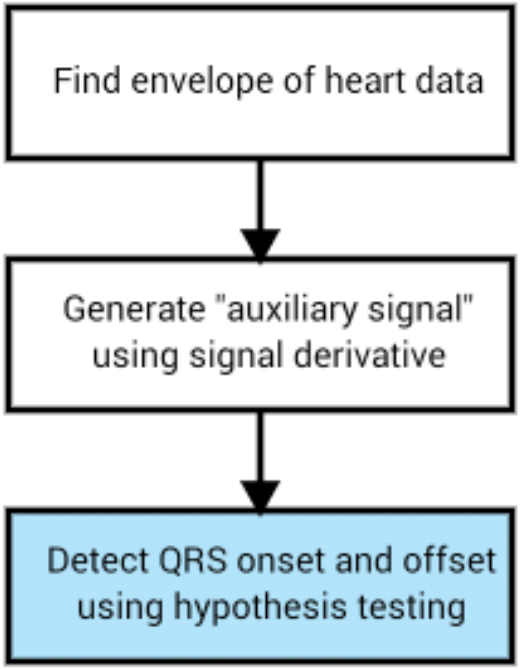


Figure 53. Flowchart for alternative QRS detection method

- Calculate a cumulative mean for the QRS onset and offset windows
- Determine the probability density functions