Independent Project Report

Topic: Extraction of Diagnostic Features from simultaneously recorded physiological signals

---

Description: Four simultaneously recorded cardiovascular signals - ECG, PPG, PCG and SpO2, will be extensively studied and their correlation will be exploited to extract diagnostic information. At the end of this project, we expect a model/algorithm that will take these four signals as input and provide cardiovascular health index of the subject as output.

Independent Project details:

Instructor: Dr. Sujay Deb

Credits: 4

Duration: 4 months (Jan'15 – Apr'15)

Team members: Abhinav Aggarwal (2012121), Akshay Punhani (2012126)

Phonocardiogram signal acquisition using mobile phone

Phonocardiogram contains the heart sounds, usually called as lub-dub. They are technically called as the S1 and S2 waves, which are present in all PCG signals. For a cost-effective and quick recoding & analysis of heart beats, PCG signal is recorded from a smartphone.

Signal acquisition: For recording the heart-beats, smartphone's microphone is placed on chest, in contact with skin, and PCG signal is recorded using the phone's sounder recorder application. Smartphones usually save the audio file in .m4a or .wav format, at 44.1 kHz sampling frequency.

Signal processing and feature extraction: The recorded audio file, is directly imported into MATLAB, and passed through a low pass butterworth filter of cutoff frequency 100 Hz, and 20 dB gain. Further S1 and S2 peaks are extracted, along with heart rate calculation and cardiac arrhythmia detection. The algorithms involved and the process is explained in PCG section at page 12.

Following are the details for signal recorded though smartphone:

1. We took samples from 6 devices in total and were successful in capturing S1 wave from 4 devices.
2. Below are the devices used and corresponding findings:
   a) HTC Explorer (Entry level): Distinguishable S1 and S2 waves, low noise.
   b) Asus Zenfone 4 (Entry level): Only S1 wave, with high noise.
   c) Apple iPod 4g (Mid-level): Distinguishable S1 and S2 waves, low noise.
   d) Samsung Note (High level): Clear S1 wave, low noise.
   e) Micromax Magnus (Mid-level) and Nokia Lumia (Entry level), were not able to capture the signal.

Conclusion: The whole process of capturing heartbeat is a bit difficult, and required lot of hit and trials. Each device has different sensitivity of microphone, and required separate configuration of filtering during data processing. The process is best suited for heartbeat count, and S1 and S2 timings (for few devices). It also depends on the positioning of mic on the device. For microphone placed on the back of the device, results are good, as compared to the one's which has microphone on the bottom of device.
MATALB code is included in Appendix A.

Following are the filtered waveforms from four devices, plotted using MATLAB.

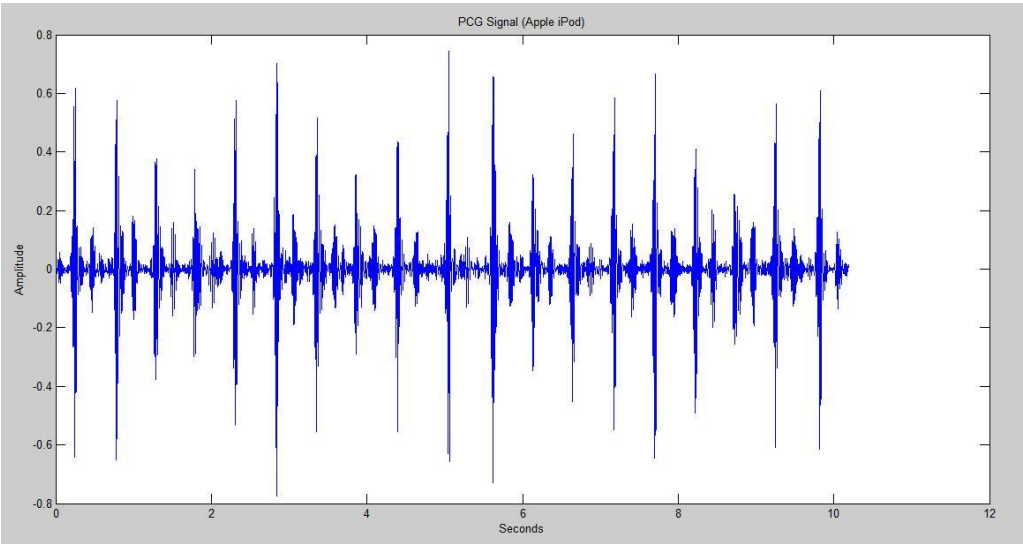


Fig 1

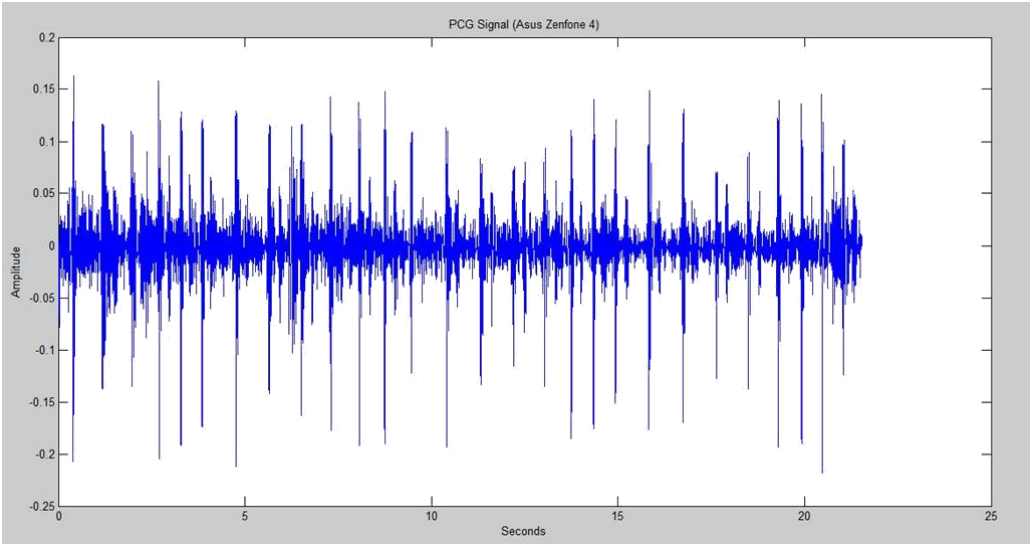


Fig 2

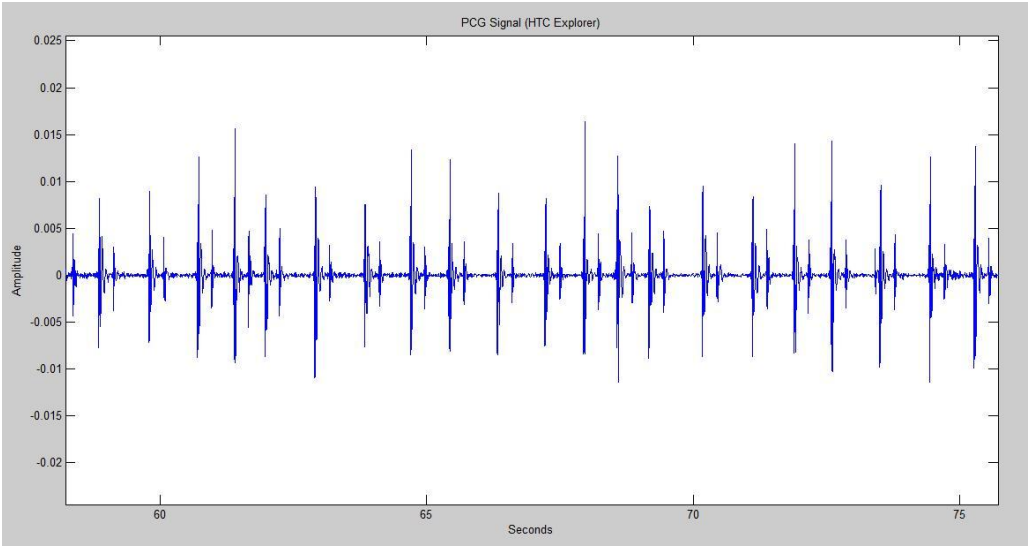


Fig 3

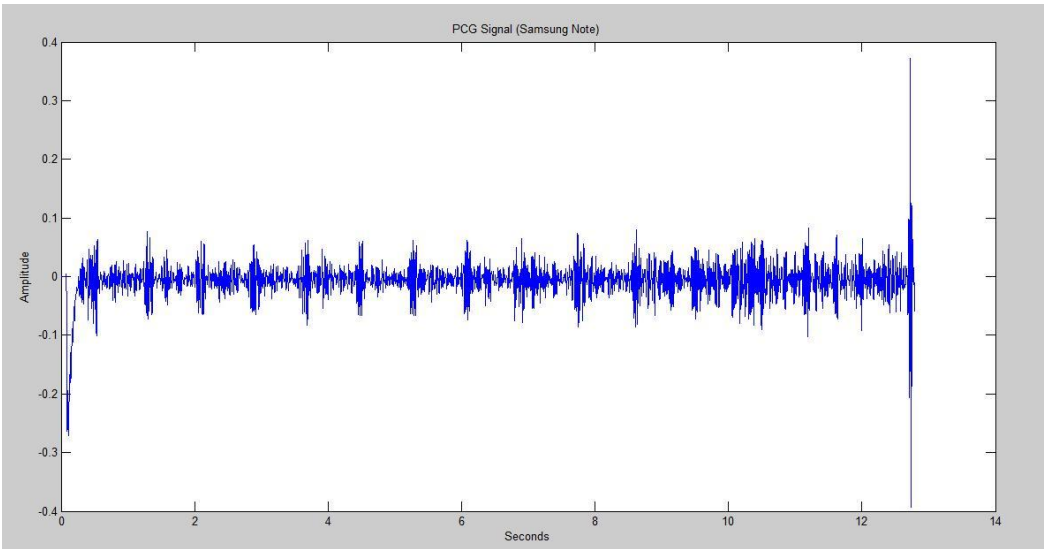


Fig 4

Vital signal set database of subjects

For the purpose of analysis and to test the algorithms, we created a database of 82 subjects. Two different online databases were used:

1. **MIMIC database**: "*The MIMIC II (**M**ultiparameter **I**ntelligent **M**onitoring in **I**ntensive **C**are) Databases contain physiologic signals and vital signs time series captured from patient monitors, and comprehensive clinical data obtained from hospital medical information systems, for tens of thousands of Intensive Care Unit (ICU) patients. Data were collected between 2001 and 2008 from a variety of ICUs (medical, surgical, coronary care, and neonatal) in a single tertiary teaching hospital. The MIMIC II Clinical Database contains clinical data from bedside workstations as well as hospital archives" [14].* The signals are available freely via PhysioNet ATM. We have handpicked the signals for 50 subjects, and separately made their ECG, PPG, BP, Heart Rate and SpO2 file for every subject. All signals are simultaneously recorded.

2. **The University of Queensland Vital Signs Dataset**: "*The University of Queensland Vital Signs Dataset contains a wide range of patient monitoring data and vital signs that were recorded during 32 surgical cases where patients underwent anaesthesia at the Royal Adelaide Hospital."[15].* Signals from this database are also picked, to ensure they are continuously available, and separate files for 32 subjects are made for ECG, PPG, BP, Heart Rate and SpO2, which are all simultaneously recorded.

Electrocardiography and Photoplethysmogram

Various features were extracted from ECG and PPG signals. First of all, the R-peaks in ECG and the peak & foot of PPG were detected in the following manner:

1. ECG and PPG signal was imported into MATLAB.
2. DC component is removed
3. Peaks are normalized to one
4. Baseline wandering is removed
5. Using 'findpeaks' function, R-peak of ECG and peak & foot of PPG are detected.

In order to ensure, all peaks were detected the following algorithm was coded:

R-peak detection

1. Find peaks with minimum peak distance = Fs/2 and minimum peak height = 0.5.
    a. Find peaks with minimum peak distance = Fs/5 and minimum peak height = 0.5.
    b. If Number of peaks from the two method, differ by more than 2, find new peaks with minimum peak distance = Fs/(x+1) and minimum peak height = 0.5., where x varies from 2 to 5.
        i. If the difference between two R-peaks vary greater than 10% of mean of R-peak interval, proceed with x = x+1

Once all the R-peaks are correctly located, we move to PPG peaks and take help of already located R-peaks.

PPG peak detection:

1. PPG peaks with minimum distance = Fs/2 and minimum peak height = 0.2
2. Check for only one PPG peak, in between two consecutive R-peaks of ECG
3. Check for equal number of PPG peak and R-peak, with maximum difference of 1 and equal number of PPG peak and foot, with maximum difference of 1
    a. If number of PPG peaks < Number of ECG peaks
        i. Find new PPG peaks with minimum distance = Fs/(x+1) and minimum peak height = 0.2, where x varies from 2 to 5
    b. If number of PPG peaks > Number of ECG peaks
        i. Find new PPG peaks with minimum distance = Fs/(x-1) and minimum peak height = 0.2, where x varies from 2 to 5
    c. If number of PPG Foot < Number of ECG peaks
        i. Find new PPG foot with minimum distance = Fs/(x+1) and minimum peak height = 0.2, where x varies from 2 to 5
    d. If number of PPG foot > Number of ECG peaks
        i. Find new PPG foot with minimum distance = Fs/(x-1) and minimum peak height = 0.2, where x varies from 2 to 5

4. If number of PPG peak and foot are greater than number of R-peaks +1
    a. PPG signal is wrong/ peaks cannot be found correctly

Throughout the algorithm, there are check points at which peaks are checked, and if they cannot be correctly detected, error message is displayed. Some peak detection plots are shown below.

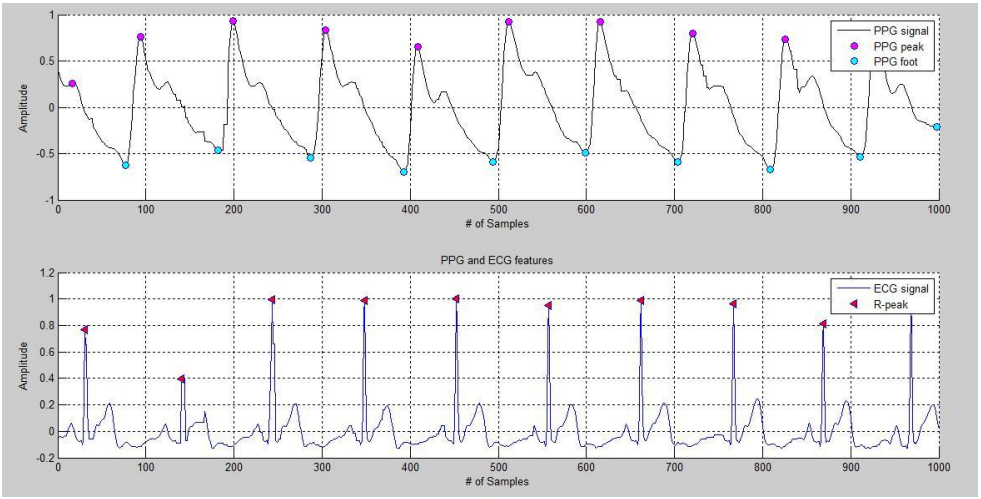


Fig 5 University of Queensland Vital Signs Dataset- Subject #1

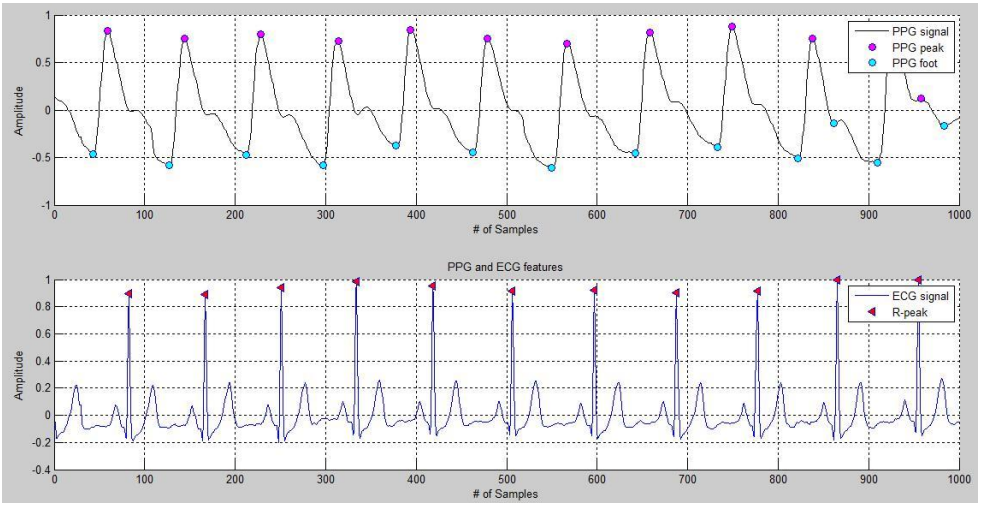


Fig 6 University of Queensland Vital Signs Dataset- Subject #8

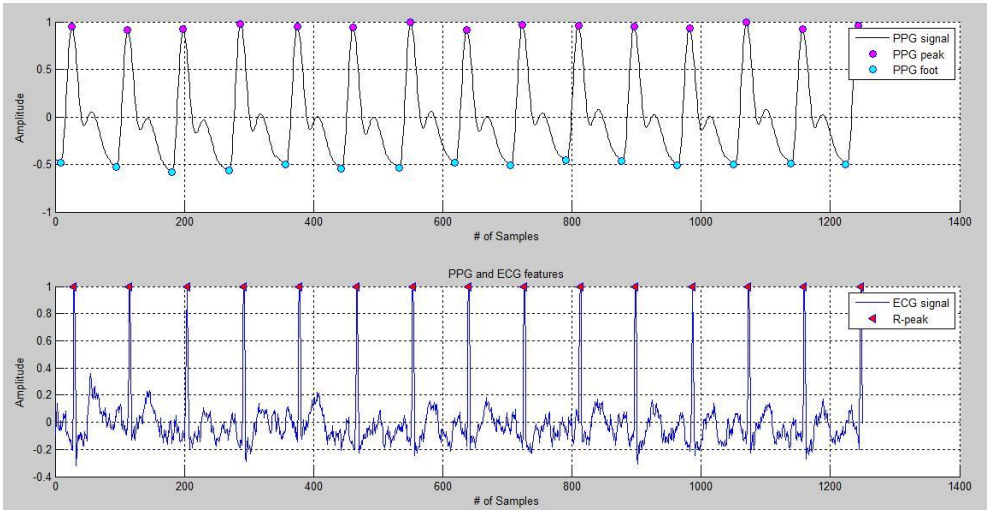


Fig 7 MIMIC database – Subject #4

After the three peaks are detected, the following parameters are calculated for 50 subjects

1. Pulse Transit Time (PTT) Max: Time difference between ECG R-peaks and PPG foot
2. Pulse Transit Time (PTT) Min: Time difference between ECG R-peaks and PPG peak
3. R-R interval: Time difference between two consecutive R-peaks of ECG
4. Systolic Upstroke (ST) : Time difference between PPG foot to PPG peak
5. Diastolic Time (DT) : Time difference between PPG peak to PPG foot
6. PP interval: Time difference between two consecutive peaks of PCG

Sometimes, a Dicrotic Notch is also observed in some PPG signals, for which an additional parameter of Time delay is calculated which is time difference between PPG peak and diastolic amplitude. Dicrotic Notch is located, where the amplitude of first derivative of PCG signal changes from positive to negative [4], shown in figure below.

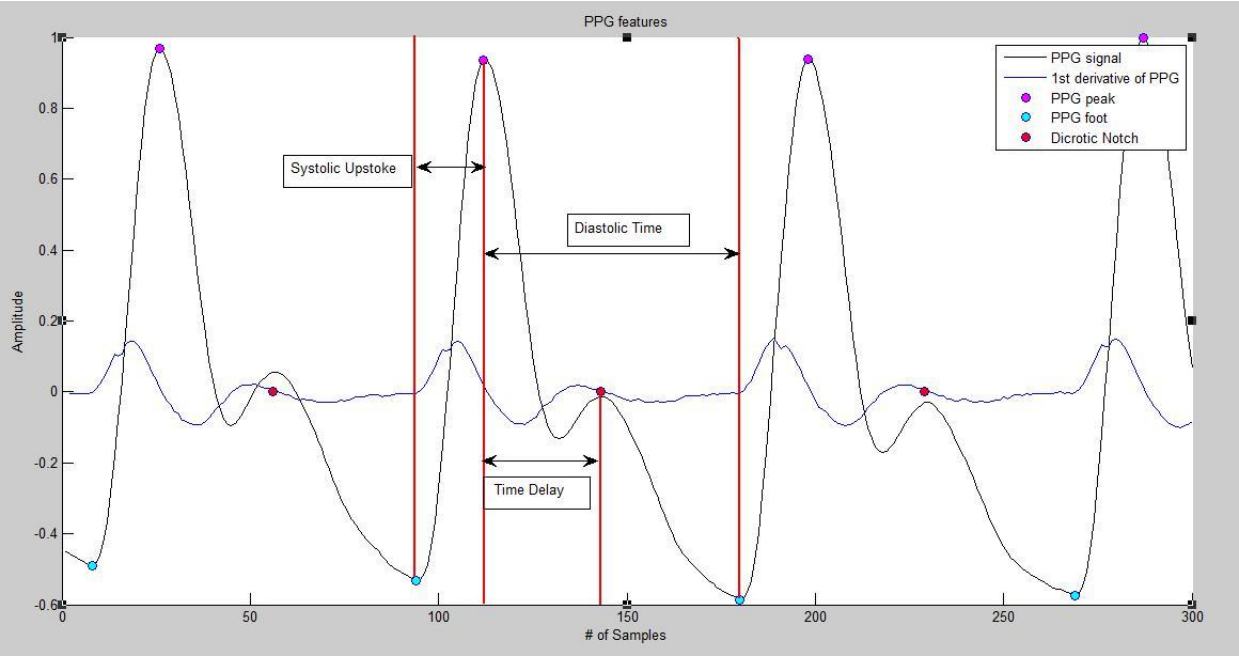


Fig 8

Blood Pressure Estimation using ECG and PPG features

Using the features and timings extracted from ECG and PPG features, various linear, as well as non-linear regression models were made, to estimate subject's Blood pressure. The following is the best model achieved:

Estimation of Diastolic Blood Pressure

3rd order polynomial Non-linear regression model with Diastolic time and Pulse transit time as variable

f(x,y) = p00 + p10*x + p01*y + p20*x^2 + p11*x*y + p02*y^2 + p30*x^3 +
p21*x^2*y + p12*x*y^2 + p03*y^3

Where f(x,y) is Diastolic BP and x = Diastolic Time & y = Pulse transit time

Coefficients (with 95% confidence bounds):

p00 = -3647 (-5316, -1979)
p10 = 652.1 (-1474, 2778)
p01 = 2.645e+04 (1.593e+04, 3.697e+04)
p20 = 1028 (-900.2, 2957)
p11 = -6962 (-1.347e+04, -456)
p02 = -5.851e+04 (-8.118e+04, -3.584e+04)
p30 = -622.1 (-1428, 183.6)
p21 = 717 (-1212, 2646)
p12 = 7503 (1054, 1.395e+04)
p03 = 4.253e+04 (2.59e+04, 5.916e+04)

R-sqr = 0.859403501

Adjusted R-sqr = 0.78031797

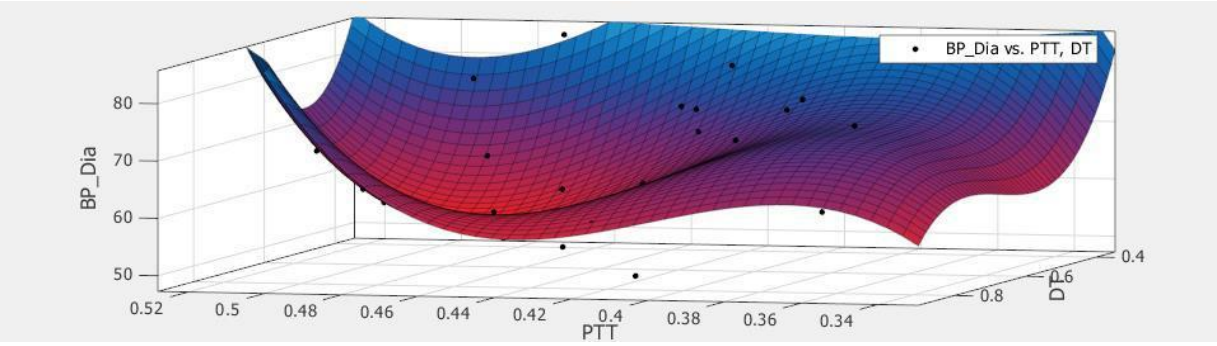


Fig 9

Regression model applied on training set

BP Diastolic	Cal	Error
60	63.43883	-3.43883
77	82.96186	-5.96186
77	74.11255	2.887449
61	65.45445	-4.45445
71	71.99895	-0.99895
82	79.19259	2.807406
79	75.83257	3.167431
62	63.82632	-1.82632
77	77.83065	-0.83065
61	65.20493	-4.20493
70	73.97341	-3.97341
76	74.8612	1.1388
81	74.88831	6.111687
73	72.62429	0.375712
63	61.63194	1.368058
70	69.86735	0.132653
73	77.0213	-4.0213
57	54.94111	2.058889
64	67.92363	-3.92363
61.7	58.38113	3.318872
61	63.61369	-2.61369
55.2	54.91546	0.284543
76	71.11683	4.88317
66	68.73264	-2.73264
52	54.93126	-2.93126
61	60.81099	0.18901

Regression model applied on full dataset

BP Diastolic	Cal	Error
60	63.43883	-3.43883
49	66.73502	-17.735
77	82.96186	-5.96186
77	74.11255	2.887449
61	65.45445	-4.45445
71	71.99895	-0.99895
70	59.30507	10.69493
82	79.19259	2.807406
79	75.83257	3.167431
44	66.14829	-22.1483
62	63.82632	-1.82632
77	77.83065	-0.83065
61	65.20493	-4.20493
76	67.25844	8.741559
49	59.86813	-10.8681
70	73.97341	-3.97341
50	56.21703	-6.21703
76	74.8612	1.1388
60	76.35087	-16.3509
81	74.88831	6.111687
73	72.62429	0.375712
63	61.63194	1.368058
70	69.86735	0.132653
84	72.82916	11.17084
44	64.33413	-20.3341
66	68.73264	-2.73264
52	54.93126	-2.93126
61	60.81099	0.18901
73	77.0213	-4.0213
57	54.94111	2.058889
64	67.92363	-3.92363
61.7	58.38113	3.318872
61	63.61369	-2.61369
55.2	54.91546	0.284543
76	71.11683	4.88317
52	59.06579	-7.06579

Conclusion: The BP regression model works for the range 49-82, beyond which, the error increases beyond acceptable level.

Estimation of Systolic Blood Pressure

3rd order polynomial Non-linear regression model with Heart rate and Systolic Time as variable.

f(x,y) = p00 + p10*x + p01*y + p20*x^2 + p11*x*y + p02*y^2 + p30*x^3 +
p21*x^2*y + p12*x*y^2 + p03*y^3

Where, f(x,y) is the Systolic Blood pressure, and x = Hear rate, y = Systolic Time

Coefficients

p00 =	221.6
p10 =	4.273
p01 =	-4200
p20 =	-0.2149
p11 =	141.1
p02 =	-3503
p30 =	0.001151
p21 =	-0.3982
p12 =	-187.7
p03 =	2.349e+04

R-sqr = 0.406624888
Adjusted R-sqr = 0.174434626

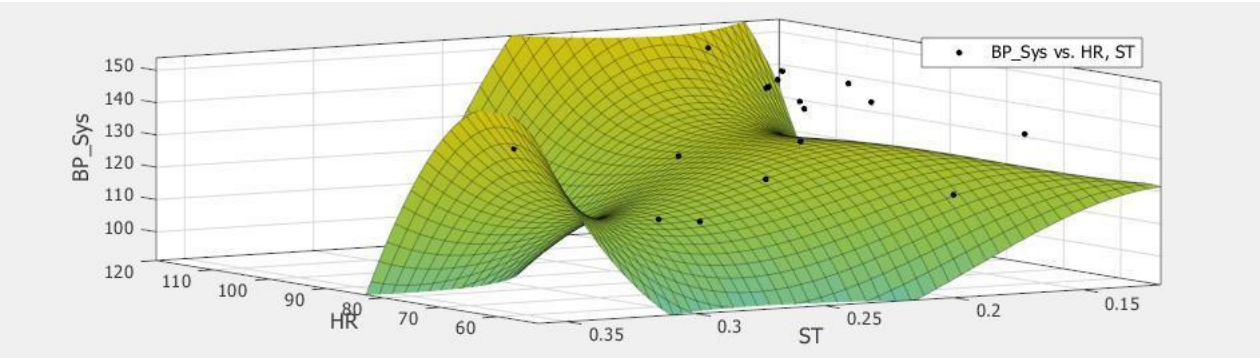


Fig 10

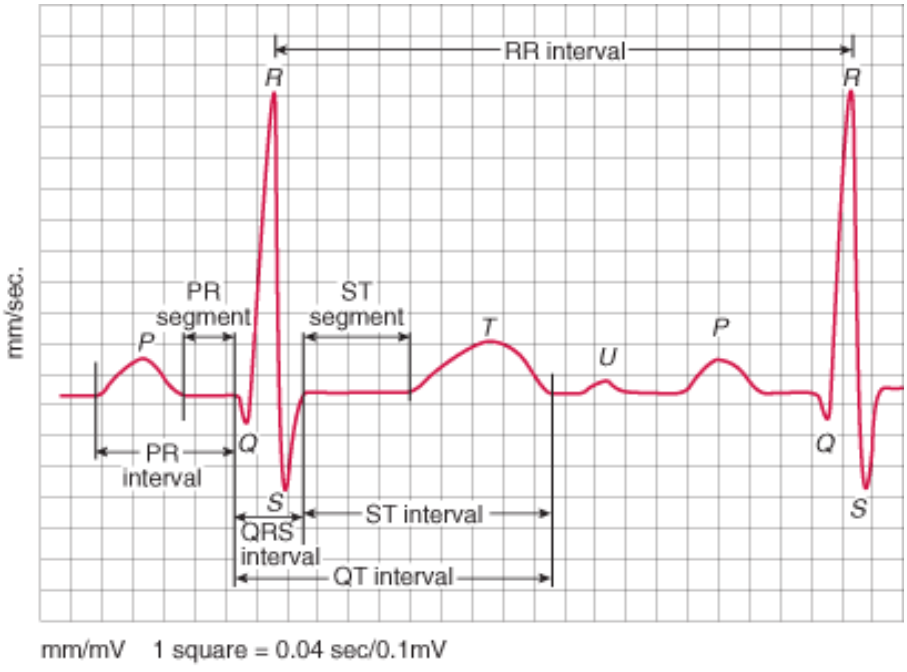
Regression model applied on training set

BP systolic	Cal	Error
113	116.4752	-3.47522
121	127.75	-6.75003
137	130.0716	6.928364
142	143.3302	-1.33019
121	115.4805	5.51954
118	129.3538	-11.3538
135	124.9417	10.05835
142	130.2243	11.77572
126	130.4861	-4.48614
101	105.2126	-4.21259
145	126.2576	18.74239
117	119.6832	-2.6832
144	130.5248	13.47521
105	118.644	-13.644
151	132.1318	18.8682
116	108.2212	7.778786
142	131.7678	10.23217
112	129.8132	-17.8132
94	103.0533	-9.05332
131	121.3044	9.695551
151	143.5011	7.498924
139	149.711	-10.711
103	120.167	-17.167
115	115.6877	-0.68773
148.3	131.7238	16.57615
125	127.622	-2.62197
144.8	132.1467	12.65333
140	147.5545	-7.55453
113	113.7447	-0.74471
134	130.319	3.680958
125	122.5268	2.473239

Conclusion: The BP regression model works for the range 95-142, beyond which, the error increases beyond acceptable level.

ECG SIGNAL

Electrocardiography (ECG) is the process of recording the electrical activity of the heart over a period of time using electrodes placed on a patient's body. In a conventional 12 lead ECG, ten electrodes are placed on the patient's limbs and on the surface of the chest. The overall magnitude of the heart's electrical potential is then measured from twelve different angles ("leads") and is recorded over a period of time (usually 10 seconds). It is a non invasive procedure. ECG can be used to measure the rate and rhythm of heartbeats, the size and position of the heart chambers, the presence of any damage to the heart's muscle cells or conduction system, the effects of cardiac drugs, and the function of implanted pacemakers.[1]



[2]

Algorithm Used for feature extraction and peak detection:

1. R peaks were detected using the earlier algorithm, already explained.
2. Using the R peaks, the max value of the graph was found between the two R peaks, and that gave us the T peak.
3. Next, the minimum value of the graph was found out between the R peak and the newly detected T peak. This gave us the S peak.

4. Now we have 3 peaks, R peak, T peak and S peak. Next step is to find the P peak, which was found out using max in between R peak (i+1) and T peak.
5. The Q peak was detected by using the minimum function in between the P peak detected previously and the R peak.
6. As all the six peaks have been detected, next step was to find the timing intervals.
7. The RR interval, the QRS complex, QT interval, PR interval, ST interval were found out using the peaks detected.
8. These intervals were found out for 35 samples, including data from MIMIC database and the Queensland data.
9. Systolic and Diastolic BP were also included to perform regression later on.

This is the data set created for 35 patients.

P-R interval: Time difference between R-peak and P peak of ECG.
QRS complex : Time difference between Q peak and S peak.
QT interval : Time difference between Q peak and T peak.
ST peak : Time difference between S peak and T peak.

DATA	QRS complex	QT Interval	PR Interval	ST Interval	RR interval	BP SYS	BP DIA
28	0.1309	0.3324	0.1869	0.2007	0.7532	129	52
32	0.1514	0.3451	0.1994	0.1943	0.6155	91	46
33	0.113	0.246	0.148	0.132	0.5723	149	84
34	0.1034	0.276	0.1131	0.1737	0.5895	112	58
35	0.103	0.275	0.1975	0.172	0.5089	184	109
36	0.0793	0.3855	0.0975	0.3062	0.6751	91	49
39	0.2709	0.2994	0.22	0.0246	0.605	97	71
42	0.1143	0.368	0.1897	0.2629	0.7093	109	76
44	0.0698	0.2451	0.1055	0.1738	0.6175	106	52
45	0.1105	0.2756	0.1135	0.1651	0.6	125	81
46	0.104	0.257	0.11	0.147	0.6256	121	75
48	0.0829	0.3076	0.1018	0.2262	0.6769	127	61
49	0.0503	0.2554	0.1377	0.2057	0.59	92	40
27	0.0895	0.3455	0.1069	0.2567	0.7428	109	53
2	0.116	0.2447	0.1107	0.1293	0.66	85	N.A
3	0.1011	0.28	0.1033	0.1789	0.7151	121	49
4	0.0684	0.1942	0.0807	0.1258	0.4093	137	77
5	0.1171	0.2749	0.0931	0.1542	0.7157	193	80
8	0.058	0.218	0.094	0.16	0.6577	118	71
9	0.0596	0.1956	0.0698	0.1353	0.6862	135	70
15	0.1136	0.2848	0.1496	0.1704	0.7987	108	44
16	0.0904	0.2032	0.056	0.1128	0.5415	101	62
18	0.1173	0.1995	0.1109	0.0875	0.4648	145	77

19	0.072	0.2291	0.1135	0.1578	0.5591	117	61
21	0.0732	0.2111	0.0831	0.1378	0.5204	97	49
22	0.101	0.242	0.107	0.141	0.6069	87	40
23	0.092	0.235	0.082	0.143	0.656	105	70
24	0.0833	0.2653	0.0807	0.1787	0.6429	86	50
25	0.1267	0.284	0.16	0.1567	0.6577	151	76
26	0.1142	0.2829	0.0989	0.1687	0.7175	116	60
28	0.0593	0.2113	0.0927	0.1515	0.5145	142	81
30	0.1072	0.32	0.1152	0.2112	0.7813	94	63
31	0.0815	0.204	0.075	0.1225	0.5209	154	102
32	0.0669	0.2709	0.0954	0.2046	0.5895	131	70

Regression Model for systolic BP using ECG parameters.

Diastolic Blood Pressure (DBP): Three variables are used to implement DBP namely low (LDBP) from 77 to 87, normal (NDBP) from 81 to 91and high (HDBP) from 81 to 91. DBP increases with anger, anxiety, and disgust, while it decreases with acute sadness.

Systolic Blood Pressure (SBP): Three linguistic variables are used to implement SBP, low (100–121), normal (110–134), and high (120–147). SBP is known to increase with fear and anxiety.

1. This regression model has been formed using QRS complex and RR interval.

p00 =163.1 ;
p10 =-51.86 ;
p01 =-63.4 ;
Equation:
r= p00 + p10.*x + p01.*y ;

Sys Bp	sys Bp from data	diff	% Error
108.5586	129	-20.4414	-15.84604651
116.2257	91	25.2257	27.72054945
120.956	149	-28.044	-18.82147651
120.3634	112	8.3634	7.467321429
116.1862	91	25.1862	27.67714286
110.6941	97	13.6941	14.11762887
112.2028	109	3.2028	2.938348624
120.3307	106	14.3307	13.5195283
119.3295	125	-5.6705	-4.5364

118.0435	121	-2.9565	-2.44338843
115.8853	127	-11.1147	-8.751732283
123.0854	92	31.0854	33.78847826
111.365	109	2.365	2.169724771
112.5196	121	-8.4804	-7.008595041
133.6032	137	-3.3968	-2.479416058
118.3939	118	0.3939	0.333813559
116.5041	135	-18.4959	-13.70066667
106.5711	108	-1.4289	-1.323055556
124.0808	101	23.0808	22.85227723
127.5485	145	-17.4515	-12.03551724
123.9191	117	6.9191	5.913760684
126.3105	97	29.3105	30.21701031
116.7385	105	11.7385	11.17952381
114.8312	151	-36.1688	-23.95284768
111.6881	116	-4.3119	-3.717155172
127.4054	142	-14.5946	-10.27788732
108.0062	94	14.0062	14.90021277
125.8484	154	-28.1516	-18.28025974
122.2563	131	-8.7437	-6.674580153

2.The second regression model was formed with the same variables using 3nd order degree(3x3).

```
p00 =   -111.7 ;
p10 =  1.744e+04 ;
p01 =   -1303 ;
p20 =  -8.423e+04  ;
p11 =  -2.953e+04 ;
p02 =    4039  ;
p30 =   2.131e+04  ;
p21 =   1.215e+05 ;
p12 =    5195  ;
p03 =    -2343;
r = p00 + p10*x + p01*y + p20.*x.^2 + p11.*x.*y + p02.*y.^2 + p30.*x.^3 +
p21.*x.^2.*y + p12.*x.*y.^2 + p03.*y.^3;
```

Sys BP	sys Bp from data	Difference	% error
112.0698	129	-16.9302	-13.12418605
121.4993	91	30.4993	33.51571429
125.5519	149	-23.4481	-15.73697987
122.965	112	10.965	9.790178571
113.9466	91	22.9466	25.21604396
96.0797	97	-0.9203	-0.948762887
114.6468	109	5.6468	5.180550459
115.8797	106	9.8797	9.320471698
122.5981	125	-2.4019	-1.92152
119.8485	121	-1.1515	-0.951652893
114.3046	127	-12.6954	-9.996377953
113.4689	92	21.4689	23.33576087
112.2059	109	3.2059	2.941192661
113.9527	121	-7.0473	-5.824214876
137.8114	137	0.8114	0.592262774
111.2661	118	-6.7339	-5.706694915
110.3575	135	-24.6425	-18.2537037
111.277	108	3.277	3.034259259
126.0858	101	25.0858	24.83742574
139.5839	145	-5.4161	-3.735241379
120.9657	117	3.9657	3.389487179
125.0419	97	28.0419	28.90917526
116.5313	105	11.5313	10.98219048
118.1954	151	-32.8046	-21.72490066
114.2103	116	-1.7897	-1.542844828
122.2789	142	-19.7211	-13.88809859
111.7594	94	17.7594	18.89297872
126.7221	154	-27.2779	-17.71292208
117.3484	131	-13.6516	-10.4210687

3.Another regression model is with QT and RR interval.(2x1)

p00 = 240.4 ;
p10 = 194.1 ;
p01 = -247 ;
p20 = 928.5 ;
p11 = 899.6;

Sys BP	Sys BP data	diff	% Error
112.4782	129	-16.5218	-12.8076
101.892	91	10.892	11.96923

121.7551	149	-27.2449	-18.28517
116.8592	112	4.8592	4.338571
94.962	91	3.962	4.353846
112.5712	97	15.5712	16.05278
102.8487	109	-6.1513	-5.643394
120.6787	106	14.6787	13.84783
116.9393	125	-8.0607	-6.44856
119.3036	121	-1.6964	-1.401983
112.9577	127	-14.0423	-11.05693
120.0887	92	28.0887	30.5312
109.9026	109	0.9026	0.828073
116.753	121	-4.247	-3.509917
138.0972	137	1.0972	0.800876
120.4917	118	2.4917	2.11161
118.1638	135	-16.8362	-12.47126
117.1616	108	9.1616	8.482963
127.8559	101	26.8559	26.59
133.3347	145	-11.6653	-8.045034
124.3296	117	7.3296	6.264615
128.3366	97	31.3366	32.30577
120.1604	105	15.1604	14.43848
115.968	151	-35.032	-23.2
116.558	116	0.558	0.481034
128.6488	142	-13.3512	-9.402254
115.1429	94	21.1429	22.49245
129.0956	154	-24.9044	-16.17169
117.7343	131	-13.2657	-10.12649

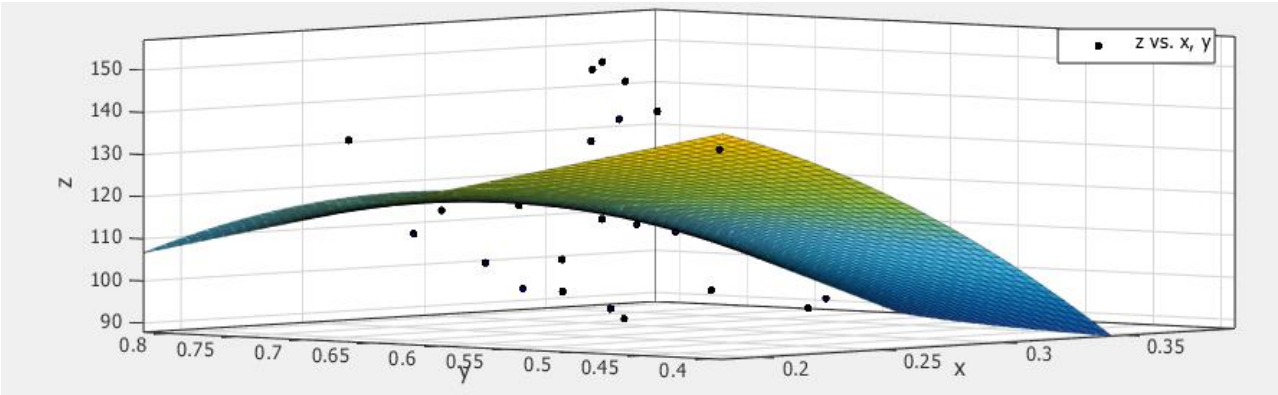


Fig 2x1(QT,RR)

4.Another model with QT and RR interval, this time both having degree 1.

p00 =163.1 ;
p10 =-51.86 ;
p01 =-63.4 ;

Sys Bp	sys Bp from data	diff	% Error
108.5586	129	-20.4414	-15.84605
116.2257	91	25.2257	27.72055
120.956	149	-28.044	-18.82148
120.3634	112	8.3634	7.467321
116.1862	91	25.1862	27.67714
110.6941	97	13.6941	14.11763
112.2028	109	3.2028	2.938349
120.3307	106	14.3307	13.51953
119.3295	125	-5.6705	-4.5364
118.0435	121	-2.9565	-2.443388
115.8853	127	-11.1147	-8.751732
123.0854	92	31.0854	33.78848
111.365	109	2.365	2.169725
112.5196	121	-8.4804	-7.008595
133.6032	137	-3.3968	-2.479416
118.3939	118	0.3939	0.333814
116.5041	135	-18.4959	-13.70067
106.5711	108	-1.4289	-1.323056
124.0808	101	23.0808	22.85228
127.5485	145	-17.4515	-12.03552
123.9191	117	6.9191	5.913761
126.3105	97	29.3105	30.21701
116.7385	105	11.7385	11.17952
114.8312	151	-36.1688	-23.95285
111.6881	116	-4.3119	-3.717155
127.4054	142	-14.5946	-10.27789
108.0062	94	14.0062	14.90021
125.8484	154	-	-

		28.1516	18.28026
122.2563	131	-8.7437	-6.67458

CONCLUSION :

From the above results we can conclude that for finding out systolic BP the Regression model with QT and RR interval works much better than the other QRS complex one. Also 2x1 model is working better than 1X1 and 3X3 of QRS complex. It can also be seen that the model works just fine for BP's in the range of 105 to 140(which can be considered as the normal BP), and if the SBP is very low, the model gives a very large error.

Some Screenshots of the ECG feature extraction

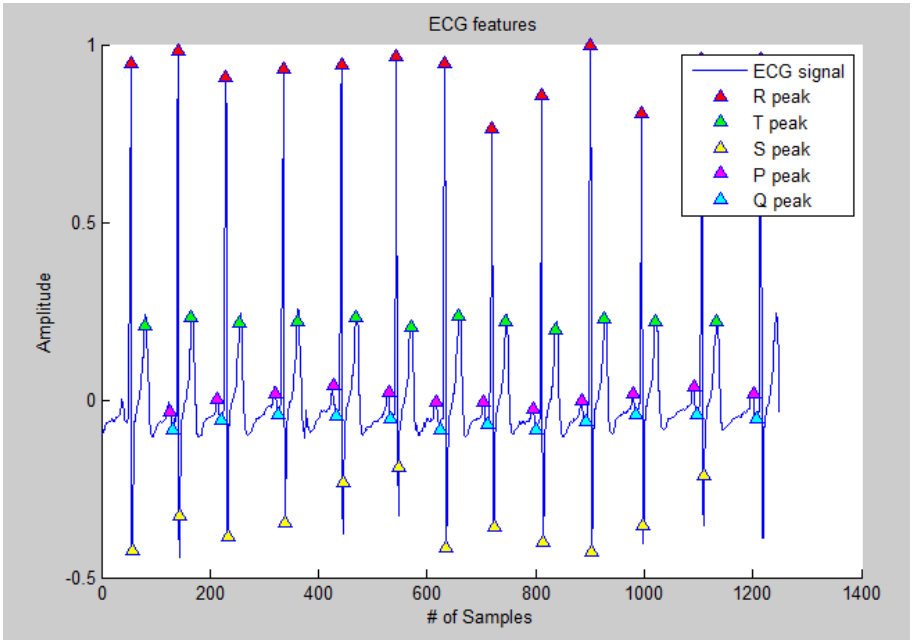
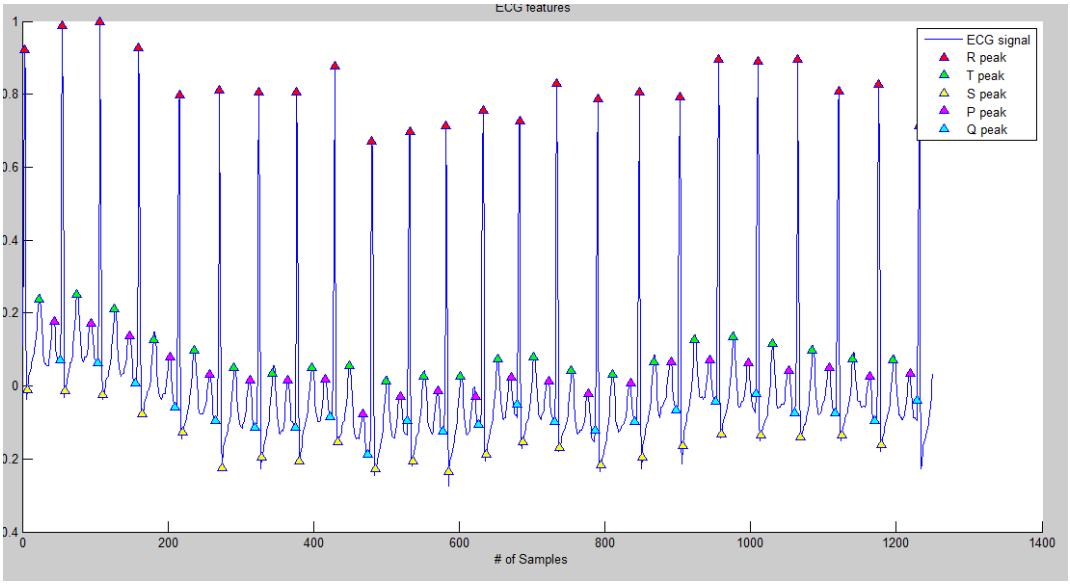


Fig : Peak detection in an ECG signal



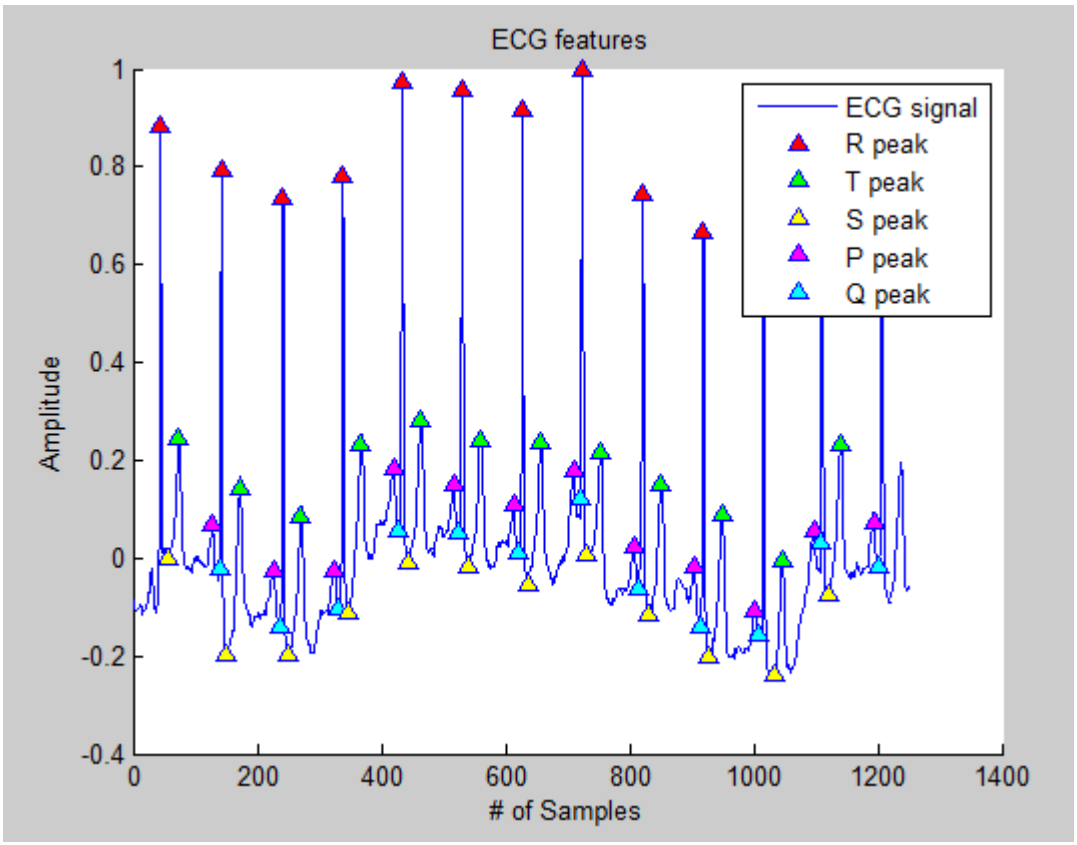


Fig : Detection in a bad ECG signal

SPo2 Signal

The amount of oxygen saturation (SpO2) in the blood is a very important parameter for detection and prevention of cardiovascular disease. SpO2 is the ratio of the oxyhaemoglobin present in blood to the total haemoglobin content of the blood. Normal SpO2 levels are considered to lie in the range of 95-99. If the level is below 90 percent, it is considered low, resulting in conditions like hypoxemia. Blood oxygen levels below 80 percent may compromise organ function, such as the brain and heart, and should be promptly addressed.[3]

Relative SpO2 measurements were made by comparing the red and blue bands, where the blue band is representative of the infrared wavelength used in traditional pulse oximeter SpO2 calculations. The standard deviations of the red and blue bands at each time point were used as the ac signals. The DC components were computed as the red and blue band mean intensities at each time point. A 10 s moving average window was then applied to the ac and dc components. Relative oxygen saturation was then computed as in the following[4] :

$$\mathrm{SpO}_2 = A - B\frac{\mathrm{AC}_{\mathrm{RED}}/\mathrm{DC}_{\mathrm{RED}}}{\mathrm{AC}_{\mathrm{BLUE}}/\mathrm{DC}_{\mathrm{BLUE}}}$$

Screenshots of the SPo2 app used:

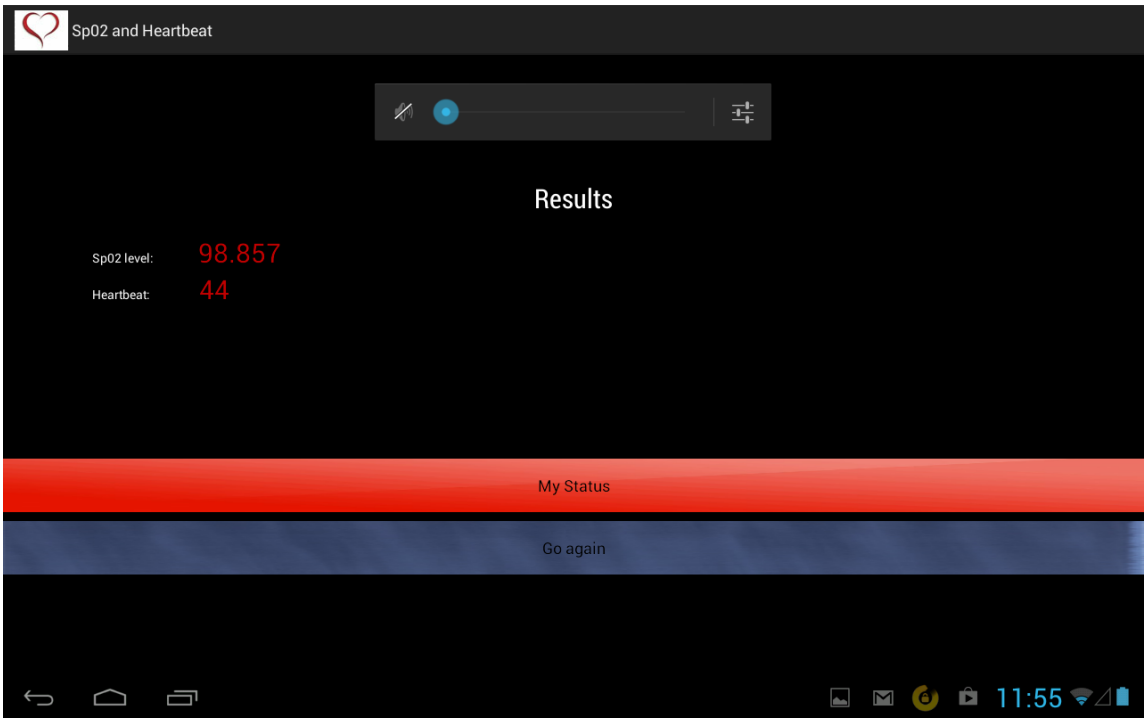


Fig : Spo2 for F, 47

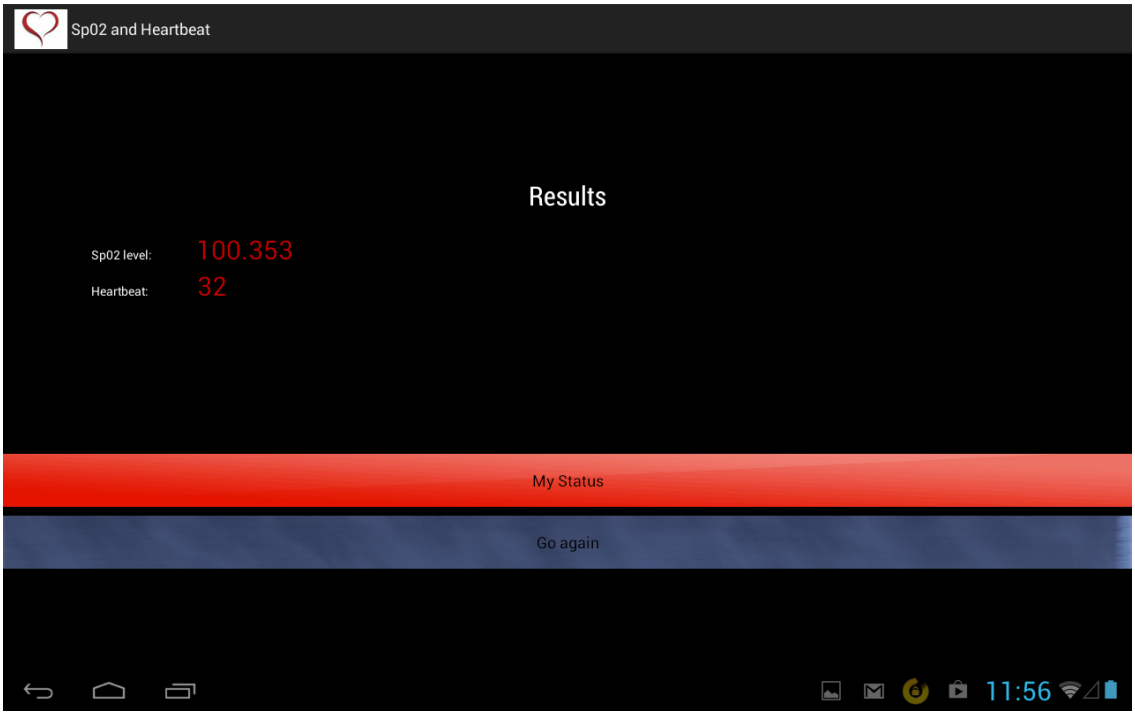
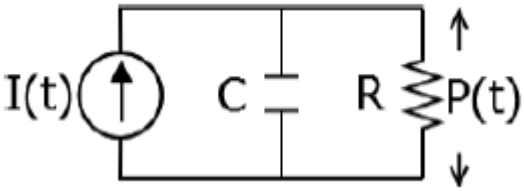


Fig: M,21(Wrong result)

PPG SIGNAL:

Instead of directly relating systolic and diastolic BP values with PPG features, our proposed methodology initially maps PPG features with some person specific intermediate latent parameters and later derives BP values from them. The 2-Element Windkessel model has been considered in the current context to estimate total peripheral resistance and arterial compliance of a person using PPG features, followed by linear regression for simulating arterial blood pressure.

Element Windkessel model : PPG features of a subject are used to derive various latent parameters of Windkessel model that controls human BP. The Windkessel model describes the human cardiovascular system in terms of an electrical circuit. The model can mathematically relate blood flow and blood pressure in arteries. In this analogy, arterial blood flow is described as the flow of fluid through a pipe. In the simplest form of Windkessel model (2-Element Windkessel model), the total peripheral resistance and arterial compliance are modeled as a resistance and capacitance respectively. The blood flow from ventricles to artery is analogous to a sinusoidal electrical current (I(t)) wave and arterial pressure wave is modeled as a time-varying electrical potential (P(t)).[5]



The Windkessel was applied on the available PPG signal and the features like the timing intervals between foot and peak of the ppg signal were extracted but we were unsuccessful in going forward with that approach.

Phonocardiogram:

Hardware Setup: Low cost, easy to assemble electronic stethoscope.

The hardware setup comprises of a 3.5 mm jack connected to a meter long wire, a small size condenser microphone, a stethoscope and a recording device, which can be either laptop or any other recording equipment.

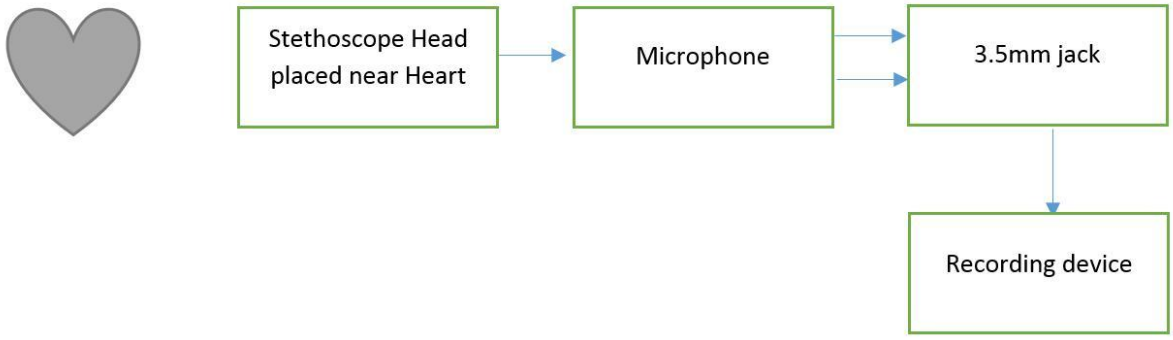


Fig 11

The recoding is done for a few seconds, and then imported into MATLAB to do further signal processing.

Software processing:

MATLAB is used to directly process the recorded wav format audio file. Following are the steps:

1. Audio file imported into MATLAB
2. Sampling frequency reduced to lower frequency (from 44.1kHz if recorded using laptop)
3. Cancel the DC components present in signal
4. Normalize the peaks to one
5. Filter signal with cutoff frequency of 100 Hz
6. Detect the envelope of PCG
7. Detect S1 peaks
8. Detect S2 peak between two consecutive S1 peaks
9. Calculate Heart rate from S1 peaks
10. Plot the Power spectral density of filtered signal
11. Check for irregular beats
12. If Heart rate > 100, subject has Tachycardia
13. If Heart rate < 60, subject has Brdycardia

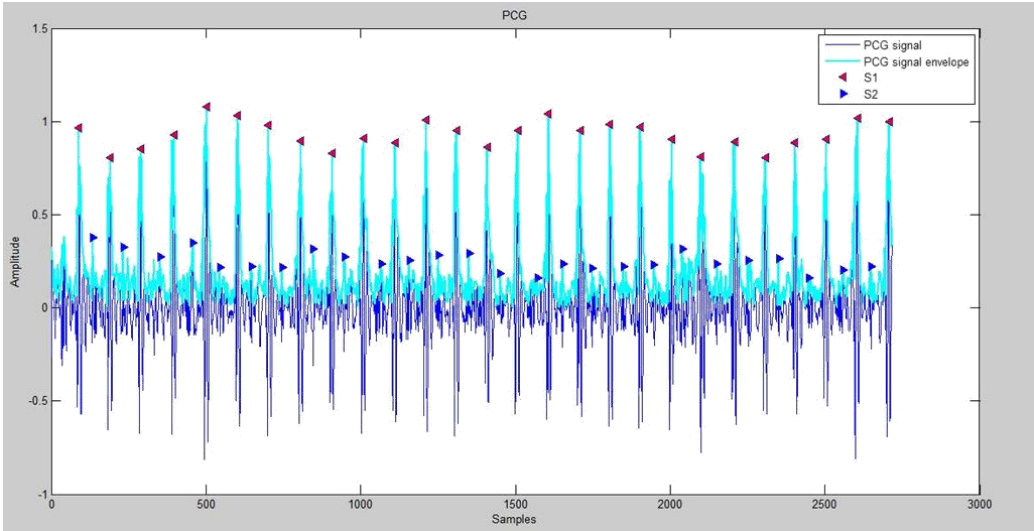


Fig 12 S1 and S2 peak detection from envelope of PCG

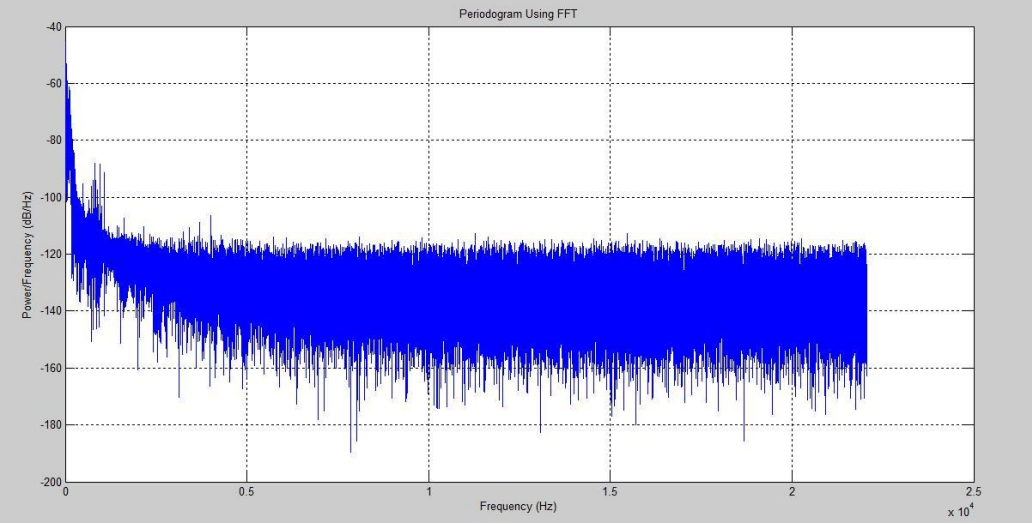


Fig 13 PSD of raw signal recorded

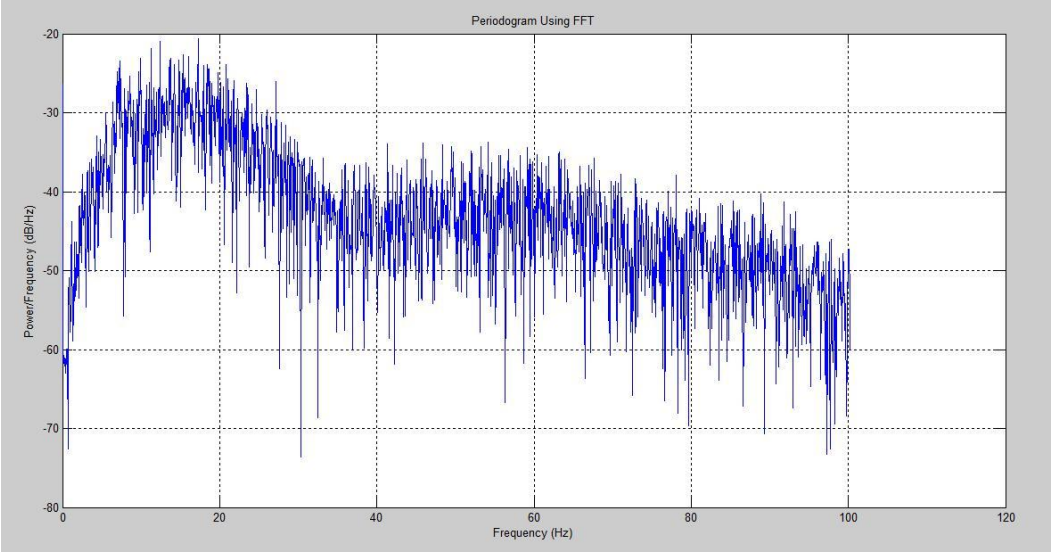


Fig 14 PSD of filtered signal.

Irregular heart beat detection (cardiac arrhythmia)[16]:

The following steps are taken to detect irreular heart beat:

1. Calculate the timing interval between each S1 adjacent wave.
2. From every S1 interval, calculate the instantaneous heart rate, ie. For 10 S1 waves, we have 10 instantaneous heart rate values.
3. Take the average of x and x+1 heart rate value, to reduce any error in calculation, therefore the average values reduces from 10 heart rate values to 9 average heart rate values.
4. Sort these heart rate values in accending order
5. Take the average of top 10 highest values of heart beat, and bottom 10 lowest values of heart beat.
6. If the average of high and low heart rate values differ by more than 30%, then there are some irregular heart beats detected from subject.

Assumptions:

1. The subject's phonocardiogram is recorded for aprrox 1 min.
2. Subject remains at rest and calm, ie. Subject has not undergone any physical exercise in past 30 minutes.
3. Since cardiac arrhythmia can occur anytime during the day, it can be detected with longer duration of signal recording. But since continous measurement is not very convininent, the system is currently designed for signals for few mintues, and detects any irregularity during that time frame only.

Points to be noted

1. The sampling frequency reduced doesn't have any significant impact on signal quality, and working with less number of data points also decreases the computational time.
2. Detecting the envelope of PCG signal also enhances the peak detection algorithms, as the S1 and S2 peaks are further enhanced and easier to detect.
3. The PSD plot of raw signal contains the power contribution from all freuqncies, although very small from frequncies above 100 Hz
4. In the PSD plot of filtered signal, only frequencies under 100 Hz have power and we can observe two approximate peaks centered around 20 Hz and 50 Hz corresponding to S1 and S2 wave, where the power of signal is more.
5. Irregular heart beat detection includes the cases of missed beat, or instances of sudden Tachycardia/ Brycardia, and
6. Detection of S3 peak and Heart Murmurs are very difficult to detect using an normal sthethoscope, mainly because of their low amplitide , and requires special electronic sthethosope, or other heart beat recording microhones built for the purpose.

MATLAB code for PCG signal processing, peak detection, and detection of irregular beats is included in Appendix A

Health Index of subject

By using features extracted from ECG, PPG, PCG and SpO2, we have prepared a 4 step health index of subject.

Levels	HR	PTT	Diastolic Time	BP diastolic	BP systolic	SpO2
Immediate Medical Attention	<45 or >135	<0.1 or >0.55	<0.33 or >0.9	<40 or >90	<70 or >180	<95
Needs to seek medical help in future	<55 or >100	nil	nil	40 - 60 85-90	140-180 70-90	95 or 96
Warning	55-60 or arrhythmia	nil	nil	60-70	130-140 90-110	96
Normal	60-100	nil	nil	70-85	110-130	96-100
Error* (in case of following)	<40 or >210	<0 or >1	<0	<0	<0	>100

*Error detection algorithms in place to detect signal errors or peak detection errors

Sources: [10], [11], [12], [13]

Assumptions:

- Subject is over 20 years old
- Subject is in rest position and relaxed while signals are recorded
- Subject has not undergone any physical exercise in past 30 minutes.
- Subject doesn't move/talk while signals are recorded for the purpose of signals being noise free
- Subject is not under the influence of BP lowering drug eg-Beta-Adrenergic Blocking Agents, which can lower the Heart rate.
- If subject is regular in athletics, his resting Heart rate can be lower than 40

References:

[1] X. F. Teng and Y. T. Zhang, "Continuous and Noninvasive Estimation of Arterial Blood Pressure Using a Photoplethysmographic Approach", *25** Annual International Conference of the IEEE EMBS, Cancun, Mexico, pg 3153-3156, September 17-21,2003

[2] A. D. Choudhury, R. Banerjee, A. Sinha and S. Kundu, "Estimating Blood Pressure using Windkessel Model on Photoplethysmogram" , 36th Annual International Conference of the Engineering in Medicine and Biology Society (EMBC), Chicago, IL, pg 4567- 4570, 26-30 Aug. 2014

[3] R. Samria, R. Jain, A. Jha, S. Saini4, S. Roy Chowdhury, "*Noninvasive Cuffless Estimation of Blood Pressure using Photoplethysmography without Electrocardiograph Measurement*", Region 10 Symposium, 2014 IEEE, pg 254 – 257, 14-16 April 2014

[4] M. Elgendi, "On the Analysis of Fingertip Photoplethysmogram Signals", *Current Cardiology Reviews,* 2012*, Vol. 8, No. 1* pg, 14-25

[5] Poon CCY, Teng XF, Wong YM, Zhang C, Zhang YT. Changes in the Photoplethysmogram waveform after exercise. Computer Architectures for Machine Perception, 2003 IEEE international Workshop on; 2004

[6] Lu S, Zaho H, Ju K, Shin K, Lee M, Shelly K, et al. Can Photoplethysmographic variability serve as an alternative approach to obtain heart rate variability information? Journal Clinical Monitoring and Computing. 2008;22(1):23-9.

[7] Takazawa K TN, Fujita M, Matsuoka O, Saiki T, Aikawa M, Tamura S, Ibukiyama C. Assessment of vasocative agents and vascular aging by the second derivative of Photoplethysmogram waveform. Hypertension. 1998:32:365-70.

[8] Millasseau S, Kelly R, Ritter J, Chowienczyk P. Determination of age-related increases in large artery stiffness by digital pulse contour analysis Clinical Science. 2002;103:371-7.

[9] Alty SR, Angarita-Jaimes N, Millasseau SC, Chowienczyk PJ. Predicting Arterial Stiffness from the Digital Volume Pulse Waveform. IEEE Transactions on Biomedical Engineering. 2007;54(12):2268-75.

[10] American Heart Association:
http://www.heart.org/HEARTORG/Conditions/More/MyHeartandStrokeNews/All-About-Heart-Rate-Pulse_UCM_438850_Article.jsp

[11] American Heart Association:
http://www.heart.org/HEARTORG/Conditions/HighBloodPressure/AboutHighBloodPressure/Understanding-Blood-Pressure-Readings_UCM_301764_Article.jsp

[12] http://www.disabled-world.com/artman/publish/bloodpressurechart.shtml

[13] Blood pressure association, UK:
http://www.bloodpressureuk.org/BloodPressureandyou/Thebasics/Bloodpressurechart

[14] MIMIC database: *http://physionet.org/mimic2*

[15] The University of Queensland Vital Signs Dataset
https://outbox.eait.uq.edu.au/uqdliu3/uqvitalsignsdataset/index.html

[16] Americal Heart Association
http://www.heart.org/HEARTORG/Conditions/Arrhythmia/Arrhythmia_UCM_002013_SubHomePage.jsp

Appendix A

MATLAB code for Phonocardiogram signal acquisition using mobile phone

```
clear
all clc
close all

%% Parameters
Sampling_Frequency = 44100;
Rp1=0.5;                % Passband ripple
Rs1=30;                 % Stop band ripple
freq =100;               % Cutoff Frequency (Hz)

n1 = 6;                 % Filter order

%%
[y,fs] = audioread('Heart4.wav'); % Read audio file y =
y(:,[1]);
%plot(y)
t=0:1/fs:(length(y)-
1)/fs; %plot(t,y);

CF_H = freq/(Sampling_Frequency/2);

Wn = [CF_H];                    % Stop band freq
[b1,a1] = butter(n1,Wn);        % Butterworth Filter
output = filter(20*b1,a1,y);    % Filter input signal
figure()
plot(t,y);
figure()
plot(t,output);
title ('PCG Signal (Samsung Note) ');
xlabel ('Seconds');
ylabel ('Amplitude');
%wavplay(output,Sampling_Frequency,'async');        % Output audio play
```

MATLAB code for for PCG signal processing, peak detection, and detection of irregular beats

```
clear
all clc
close all

%% Parameters
Sampling Frequency = 44100;                          %Sampling freq of .wav
recording
Rp1=0.5;                % Passband ripple
Rs1=30;                 % Stop band ripple
freq =100;               % Cutoff Frequency (Hz)

n1  = 6;                % Filter order
DS = 220;                                            % Decimation factor
FS = Sampling Frequency/DS;
min_seperation = round(Sampling_Frequency/(DS*3));  % Minimum seperation
b/w two S1 peaks
%%
[y,fs] = audioread('sir2.wav');        % Read audio file
%y = y(:,[1]);
y1  = decimate(y,DS);    % Reduce the sampling freq
y1  = y1 - mean (y1);    % cancel DC conponents
y1  = y1/ max( abs(y1 )); % normalize to one
%plot(y)
t=0:1/fs:(length(y)-1)/fs;
t1=0:1/(fs/DS):(length(y1)-1)/(fs/DS);
%plot(t,y);

%% Envelope detection

y2  = hilbert(y1);
env = abs(y2);
t11 = 0:length(env)-1;



%% S1 peak detection
[pks_PCG_S1,locs_PCG_S1] =
findpeaks(env,'MinPeakDistance',min_seperation,'MinPeakHeight',0.35);


%% Calculating S1 interval
for i=2:length(locs_PCG_S1)
    S1_interval(i-1) = locs_PCG_S1(i) - locs_PCG_S1(i-1);
end

S1_interval_mean = mean(S1_interval)/FS;

%% Finding S2 peak
%The algorithm finds the S2 as the maximum peak b/w two consecutive
S1 %peaks.
for (i=1: length(locs_PCG_S1)-1)
    temp1 = locs_PCG_S1(i);
    temp2 = locs_PCG_S1(i+1);
    z=1;
    temp3 = [];
    [pks_PCG_S2(i) locs_PCG_S2(i)] = max(env(temp1+30:temp2-
    30,:)); locs_PCG_S2(i) = locs_PCG_S2(i)+temp1+29;
```

```
end
%[pks_PCG_S2,locs_PCG_S2] =
findpeaks(env,'MinPeakDistance',round(0.4*mean(S1_interval)),'MinPeakHeight
',0.2);


%% PLOTTING    PCG    and
peaks plot(t11,y1)
hold on
plot(t11,abs(y2),'c','LineWidth',2)
plot(locs_PCG_S1,env(locs_PCG_S1),'<','MarkerFaceColor','r');
plot(locs_PCG_S2,env(locs_PCG_S2),'>','MarkerFaceColor', 'b');
legend('PCG signal', 'PCG signal envelope', 'S1', 'S2')
xlabel('Samples')
ylabel  ('Amplitude')
title ('PCG')
hold off;



%% Hear rate calculation
HR = 60 * (length(locs_PCG_S1))/(length(y1)/FS);
display (HR)




%% PSD [taken from http://in.mathworks.com/help/signal/ug/psd-
estimate-using-fft.html]
N = length(y1);
xdft = fft(y1);
xdft = xdft(1:N/2+1);
psdx = (1/(FS*N)) * abs(xdft).^2;
psdx(2:end-1) = 2*psdx(2:end- 1);
freq = 0:FS/length(y1):FS/2;

figure();
plot(freq,10*log10(psdx))
grid on
title('Periodogram Using FFT')
xlabel('Frequency (Hz)')
ylabel('Power/Frequency (dB/Hz)')



%% Irregular heart beat
for (i=1:length(S1_interval))
    Heart_rate(i) = 60/(S1_interval(i)/FS);
end

for (i=1:length(Heart_rate)-1)
    Heart_avg(i) = (Heart_rate(i) + Heart_rate(i+1))/2;
end
Heart_sort = sort(Heart_avg);
temp = length(Heart_sort);
Heart_avg_low = mean(Heart_sort(1:10));              % First 10 samples
Heart_avg_high = mean(Heart_sort((temp-9):temp));    % last 10 samples

if ((Heart_avg_high-Heart_avg_low)/Heart _avg_high) >
    0.3 display('Irregular beats detected')
end

if HR > 100
```

```
    display ('Tachycardia
detected') elseif HR < 60
    display ('Brdycardia detected')
end
```

MATLAB code for ECG and PPG

```
clear
all clc
close all


%% Importing signal into MATLAB
% ECG=importdata('C:\Users\Abhinav\Desktop\SEMESTER
6\IP\Files\Signal Bank\MIMIC databse- refined\Subject 20\ECG.txt');
% PPG_data=importdata('C:\Users\Abhinav\Desktop\SEMESTER
6\IP\Files\Signal Bank\MIMIC databse- refined\Subject 20\PPG.txt');
% HR=importdata('C:\Users\Abhinav\Desktop\SEMESTER
6\IP\Files\Signal Bank\MIMIC databse- refined\Subject 20\HR.txt');
% BP_Dia                    =importdata('C:\Users\Abhinav\Desktop\SEMESTER
6\IP\Files\Signal Bank\MIMIC databse- refined\Subject 20\BP_Dia.txt');
% BP_Sys=importdata('C:\Users\Abhinav\Desktop\SEMESTER
6\IP\Files\Signal Bank\MIMIC databse- refined\Subject 20\BP_Sys.txt');
% SPO2          =          importdata('C:\Users\Abhinav\Desktop\SEMESTER
6\IP\Files\Signal Bank\MIMIC databse- refined\Subject 20\SPO2.txt');

% ECG = ECG.data;
% PPG_data = PPG _data.data;
% HR = HR.data;
% BP_Dia = BP_Dia.data;
% BP_Sys = BP_Sys.data;
% SPO2= SPO2.data;

samples = 800;
data =csvread('C:\Users\Abhinav\Desktop\SEMESTER
6\IP\Queensland\32.csv', 1,3);
HR = data(1:samples/10,1);
SPO2= data(1:samples/10,2);
BP_Sys = data(1:samples/10,3);
BP_Dia = data(1:samples/10,4);
ECG = data(1:samples,5);
PPG_data = data(1:samples,6);


fs = 100;
min_seperation = round(fs);

%% Signal pre-processing
%PPG_data = PPG_data(1:length(PPG_data));
PPG_data = PPG_data - mean (PPG_data);    % cancel DC conponents
PPG_data = PPG_data/ max( abs(PPG_data )); % normalize to one
[p,s,mu] = polyfit((1:numel(PPG_data))',PPG_data,6);
f_y = polyval(p,(1:numel(PPG_data))',[],mu);

PPG = PPG_data - f_y;         % Detrend data
PPG_inv = -PPG';

%ECG = ECG(1:1250);
ECG = ECG - mean (ECG);     % cancel DC conponents
ECG = ECG/ max( abs(ECG )); % normalize to one

ECG = ECG;
FLAG = 0;        %% Flag bit for wrong R-peak detection
FLAG1 = 0;        %% Flag bit for wrong PPG
```

```
Spacing_factor = 1;


%% Locating R-peaks, PCG foot and peak


[pks_ECG_R,locs_ECG_R] =
findpeaks(ECG,'MinPeakDistance',min_seperation,'MinPeakHeight',0.5);


[pks_PPG,locs_PPG] =
findpeaks(PPG,'MinPeakDistance',min_seperation,'MinPeakHeight',0.2);
[pks_PPG_inv,locs_PPG_inv] =
findpeaks(PPG_inv,'MinPeakDistance',min_seperation,'MinPeakHeight',0.2);



%% R-peak verification
[pks_ECG_Rn,locs_ECG_Rn] =
findpeaks(ECG,'MinPeakDistance',round(min_seperation/5),'MinPeakHeight',0.5 );


for (j=2:4)
for i=1:length(locs_ECG_R)-1
    diff_ECG(i) = locs_ECG_R(i+1) - locs_ECG_R(i);
end

counter =0;
for i=1:length(diff_ECG)
    if ((mean(diff_ECG)-diff_ECG(i))/mean(diff_ECG)*100 >
        10) counter = counter +1;
    end
end



 if abs (length(pks_ECG_Rn)-(length(pks_ECG_R))) >
        2 [pks_ECG_R,locs_ECG_R] =
findpeaks(ECG,'MinPeakDistance',round((min_seperation)/(j+1)),'MinPeakHeigh
t',0.5);
        %Spacing_factor = j+1;
        display ('R-peak detection algorithm revised
        2') FLAG = 1;


 elseif (counter/length(diff_ECG))*100 >
    10 [pks_ECG_R,locs_ECG_R] =
findpeaks(ECG,'MinPeakDistance',round((min_seperation)/(j+1)),'MinPeakHeigh
t',0.5);
    %Spacing_factor = j+1;
    display ('R-peak detection algorithm revised
    1') FLAG = 1;

 else
    FLAG = 11;
    Spacing_factor = j;
    display('Breaking
    out') break;
end
end

%% Peak vector position checking
```

```
if locs_ECG_R(1) > locs_PPG(1)
     locs_PPG_check = locs_PPG(2:length(locs_PPG));
end



for (i=1: min(length(locs_ECG_R),length(locs_PPG))-1)


    if (((locs_ECG_R(i) < locs_PPG(i)) && (locs_PPG(i) < locs_ECG_R(i+1))))
       continue;


    elseif (((locs_ ECG_R(i) < locs_PPG(i)) && (locs_PPG(i)
> locs_ECG_R(i+1))))
       FLAG1 =
    12; else
        %FLAG1 = 12;
    end
end


if (FLAG1 == 12)
    %display ('Error in PTT: Location error of PTT foot/peak wrt R-
    peak') display('FLAG1 active')
end


%% PPG verification


locs_PPG_inv =
locs_PPG_inv'; pks_PPG_inv =
pks_PPG_inv'; for (i=2:6)
if (abs(length(locs_PPG) - length(locs_PPG_inv))) >
    1 %[pks_PPG,locs_PPG] =
findpeaks(PPG,'MinPeakDistance',round((min_seperation)/(i+1)),'MinPeakHeigh
t',0.1);
    %[pks_PPG_inv,locs_PPG_inv] = findpeaks(PPG
_inv,'MinPeakDistance',round((min_seperation)/(i+1)),'MinPeakH
eight',0.1);
    display ('PPG peak detection algorithm revised
    1') %display ('Wrong PTT')
end
if (length(pks_ECG_R) - length(pks_PPG)) >
    1 [pks_PPG,locs_PPG] =
findpeaks(PPG,'MinPeakDistance',round((min_seperation)/(i-
1)),'MinPeakHeight',0.1);
    %[pks_PPG_inv,locs_PPG_inv] = findpeaks(PPG
_inv,'MinPeakDistance',round((min_seperation)/(i+1)),'MinPeakH
eight',0.1);
    display ('PPG peak detection algorithm PPG -1')


 elseif (length(pks_PPG_inv)- length(pks_ECG_R)) >
   1 % [pks_PPG,locs_PPG] =
findpeaks(PPG,'MinPeakDistance',round((min_seperation)/(i+1)),'MinPeakHeigh
t',0.1);
    [pks_PPG_inv,locs_PPG_inv] = findpeaks(PPG
_inv,'MinPeakDistance',round((min_seperation)/(i+1)),'MinPeakH
eight',0.1);
    display ('PPG peak detection algorithm PPG_inv +1')
end


 if (length(pks_ECG_R) - length(pks_PPG_inv)) > 1
```

```
   % [pks_PPG,locs_PPG] =
findpeaks(PPG,'MinPeakDistance',round((min_seperation)/(i+1)),'MinPeakHeigh
t',0.1);
    [pks_PPG_inv,locs_PPG_inv] =
findpeaks(PPG_inv,'MinPeakDistance',round((min_seperation)/(i-
1)),'MinPeakHeight',0.1);
    display ('PPG peak detection algorithm PPG_inv -1')



elseif (length(pks_PPG)- length(pks_ECG_R)) >
   1 [pks_PPG,locs_PPG] =
findpeaks(PPG,'MinPeakDistance',round((min_seperation)/(i+1)),'MinPeakHeigh
t',0.1);
   % [pks_PPG_inv,locs_PPG_inv] = findpeaks(PPG
_inv,'MinPeakDistance',round((min_seperation)/(i+1)),'MinPeakH
eight',0.1);
    display ('PPG peak detection algorithm PPG +1')


end
end


%% PTT verfication
if (abs(length(locs_ECG_R)- length(locs_PPG)) > 2)
    display ('Signal error: Wrong PTT calculations')
elseif (abs((length(locs_ECG_R)- length(locs_PPG_inv)) >
    2)) display ('Signal error: Wrong PTT calculations')
end


%% Plotting graphs
t_ECG=0:1/fs:(length(ECG)-
1)/fs; t = 1:length(ECG);
figure();
%hold on;




t_PPG=0:1/fs:(length(PPG)-
1)/fs; subplot(2,1,1)
hold on;
plot (t, PPG, '-k')


plot(locs_PPG,PPG(locs_PPG),'o','MarkerFaceColor','m');
plot(locs_PPG_inv,PPG(locs_PPG_inv),'o','MarkerFaceColor','c');
legend('PPG signal','PPG peak', 'PPG foot');
xlabel('# of Samples');
ylabel('Amplitude'); grid on;
hold off;



subplot(2,1,2)
hold on;
plot (t,ECG);
%plot(locs_ECG_Q,ECG(locs_ECG_Q),'rv','MarkerFaceColor','b');
plot(locs_ECG_R,ECG(locs_ECG_R),'<','MarkerFaceColor','r');

%plot(locs_ECG_S,ECG(locs_ECG_S),'>','MarkerFaceColor','g');


%plot(locs_ECG_T,ECG(locs_ECG_T),'^','MarkerFaceColor','y');
legend('ECG signal','R-peak');
```

```
xlabel('# of Samples'); ylabel('Amplitude');
title ('PPG and ECG features');
grid on
hold off;
pks_PPG_inv = -pks_PPG_inv;

%% Calculate PTT Max
if (locs_PPG(1)>locs_ECG_R(1))
    for i=1:min(length(locs_ECG_R),length(locs _PPG))
        ptt(i) = locs_PPG(i) - locs_ECG_R(i);
    end

elseif (locs_PPG(2)>locs_ECG_R(1))
    locs_PPG = locs_PPG(2:length(locs_PPG));
    for i=1:min(length(pks_ECG_R),length(pks_PPG))-1
        ptt(i) = locs_PPG(i) - locs_ECG_R(i);
    end

 elseif (locs_PPG(3)>locs_ECG_R(1))
    locs_PPG = locs_PPG(3:length(locs_PPG));
    for i=1:min(length(pks_ECG_R),length(pks_PPG))-1
        ptt(i) = locs_PPG(i) - locs_ECG_R(i);
    end
end

%% Caculate PTT Min
if (locs_PPG_inv(1)>locs_ECG_R(1))
    for i=1:min(length(locs_ECG_R),length(locs_PPG_inv))
        ptt_min(i) = locs_PPG_inv(i) - locs_ECG_R(i);
    end

elseif (locs_PPG_inv(2)>locs_ECG_R(1))
     locs_PPG_inv = locs_PPG_inv(2:length(locs_PPG_inv));
     for i=1:min(length(pks_ECG_R), length(pks_PPG_inv))-
       1 ptt_min(i) = locs_PPG_inv(i) - locs_ECG_R(i);
    end

 elseif (locs_PPG_inv(3)>locs_ECG_R(1))
     locs_PPG_inv = locs_PPG_inv(3:length(locs_PPG_inv));
     for i=1:min(length(pks_ECG_R), length(pks_PPG_inv))-
       1 ptt_min(i) = locs_PPG_inv(i) - locs_ECG_R(i);
    end
end

%% Calculate Heart Beat
Heart_Beat = 60 * (length(pks_ECG_R))/(length(ECG)/fs);
display(Heart_Beat);
ptt_mean = mean(ptt);
ptt_time_max = ptt_mean/fs


%% Calculate PTT
ptt_mean_min = mean(ptt_min);
ptt_time_min = ptt_mean_min/fs

%% Calculate R-R interval
for i=2:length(locs_ECG_R)
    RR_interval(i-1) = locs_ECG_R(i) - locs_ECG_R(i-1);
end
```

```
RR_interval_mean = mean(RR_interval)/fs


%% 3-pint cross check on caculated values


%% HR verification
HR_mean = mean(HR);

if ((abs(HR_mean-Heart_Beat)/HR_mean)*100 < 5)
    display ('Heart rate calculation correct')

elseif   ((abs(HR_mean-Heart_Beat)/HR_mean)*100   <
    10) display ('Heart rate calculation slightly
    off') %(abs(HR_mean-Heart_Beat)/HR_mean)*100
else
    display ('Heart rate calculation completely wrong')
end

%% R-peak verification
for i=1:length(locs_ECG_R)-1
    diff_ECG(i) = locs_ECG_R(i+1) - locs_ECG_R(i);
end

counter =0;
for i=1:length(diff_ECG)
    if ((mean(diff_ECG)-diff_ECG(i))/mean(diff_ECG)*100 >
        5) counter = counter +1;
    end
end

if ((counter/length(diff_ECG))*100 > 20 && FLAG ==
    1) display ('ECG signal detected irregular' )
elseif (counter/length(diff_ECG))*100 > 20
    display ('Possible R-peak detection error')
end

%% PTT verfication
if (length(locs_ECG_R)- length(locs_PPG) > 2) display
    ('Signal error: Wrong PTT calculations')
elseif ((length(locs_ECG_R)- length(locs_PPG_inv) > 2))
    display ('Signal error: Wrong PTT calculations')
end

if (length(locs_PPG) - length(locs_PPG_inv) > 1)
    display ('PCG signal error: Wrong PTT calculations')
end

%% HR,    SpO2,    BP    from
signals heart = mean(HR)
SpO2 = mean(SPO2)
Bp_Dia = mean(BP_Dia)
Bp_Sys = mean(BP_Sys)

%% ST and DT calculation
if (locs_PPG_inv(1) < locs_PPG(1))
    for (i=1:min(length(locs_PPG_inv),length(locs_PPG))-1)
```

```
        ST(i) = locs_PPG_inv(i) - locs_PPG(i);
        DT(i) = locs_PPG_inv(i+1) - locs_PPG(i);
    end
elseif (locs_PPG_inv(1)> locs_PPG(1))
    for (i=1:min(length(locs_PPG_inv),length(locs _PPG))-
        1) ST(i) = locs_PPG_inv(i) - locs_PPG(i+1);
        DT(i) = locs_PPG_inv(i) - locs_PPG(i);
    end
end
ST = ST(1:length(ST)-1);
DT = DT(1:length(DT)-1);
Systolic_upstroke = abs(mean(ST))/fs
Diastolic_time = mean(DT)/fs


%% Calculate P-P interval
for i=2:length(locs_PPG)
    PP_interval(i) = locs_PPG(i) - locs_PPG(i-1);
end

PP_interval_mean = mean(PP_interval)/fs
```

Code for ECG Feature Extraction:

```
clear all
clc
close all
%ECG=importdata('C:\Users\Akshay punhani\Desktop\SEM6\MIMIC database Akshay\data26\ECG.txt');
ECG=importdata('C:\Users\Akshay punhani\Desktop\SEM6\MIMIC database Akshay\University of Queensland
Vital Signs Dataset\ECG32.txt');
z1=[];
z2=[];
z3=[];
z4=[];
fs = 125;
min_seperation = fs;
ECG = ECG(1:1250);

ECG = ECG - mean (ECG);    % cancel DC conponents
ECG = ECG/ max( abs(ECG )); % normalize to one

%ECG_inv = -ECG;
FLAG = 0;      %% Flag bit for wrong R-peak detection
FLAG1 = 0;      %% Flag bit for wrong PPG
%% Locating peaks
[pks_ECG_R,locs_ECG_R] = findpeaks(ECG,'MinPeakDistance',round(min_seperation/2),'MinPeakHeight',0.5);

% ECG_inv = -ECG;
% [pks_ECG_S,locs_ECG_S] = findpeaks(ECG_inv,'MinPeakDistance',50,'MinPeakHeight',0.12);
display(locs_ECG_R);
%display(locs_ECG_S);

%[pks_ECG_S,locs_ECG_Q_raw] = findpeaks(ECG_inv,'MinPeakDistance',1,'MinPeakHeight',0.05);
%[pks_ECG_Raw,locs_ECG_Raw] = findpeaks(ECG,'MinPeakDistance',10,'MinPeakHeight',0.01);

%locs_ECG_T = locs_ECG_Raw(ECG(locs_ECG_Raw)>0.1 & ECG(locs_ECG_Raw)<0.5);
% /*
% x=0.9;
% y=1.1;
```

```
%
%
%    if (locs_ECG_R>locs_ECG_S)
%       [pks_ECG_S,locs_ECG_S] = findpeaks(ECG_inv,'MinPeakDistance',50,'MinPeakHeight',0.15*x);
%       if (locs_ECG_R==locs_ECG_S)
%          break;
%       end
%       else
%          [pks_ECG_S,locs_ECG_S] = findpeaks(ECG_inv,'MinPeakDistance',50,'MinPeakHeight',0.15*(x-.1));
%          if (locs_ECG_R==locs_ECG_S)
%          break;
%          end
%    end
%    if (locs_ECG_R<locs_ECG_S)
%       [pks_ECG_S,locs_ECG_S] = findpeaks(ECG_inv,'MinPeakDistance',50,'MinPeakHeight',0.15*y);
%       if (locs_ECG_R==locs_ECG_S)
%          break;
%       else
%          [pks_ECG_S,locs_ECG_S] = findpeaks(ECG_inv,'MinPeakDistance',50,'MinPeakHeight',0.15*(y+.1));
%          if (locs_ECG_R==locs_ECG_S)
%          break;
%          end
%       end
%    end
%
%    */
% x=length(locs_ECG_S);
% y=length(locs_ECG_R);
%   if(locs_ECG_S(1)<locs_ECG_R(1))
%      locs_ECG_S(1)=[];
%   end
%   if (length(locs_ECG_S)>length(locs_ECG_R))
%      locs_ECG_S(x)=[];
%   end
%
%  for j=1:length(min(locs_ECG_S,locs_ECG_R))
%      if (locs_ECG_S(j)-locs_ECG_R(j) >5)
%         locs_ECG_S(1)=[];
%      end
%  end
%
%  disp(locs_ECG_S);
%

% indexmin = find(min(y) == y);
% xmin = x(indexmin);
% ymin = y(indexmin);
%
% indexmax = find(max(y) == y);
% xmax = x(indexmax);
% ymax = y(indexmax);
%
%T peak Speak Qpeak P peak


%% R-peak verification
[pks_ECG_Rn,locs_ECG_Rn] =
findpeaks(ECG,'MinPeakDistance',round(min_seperation/5),'MinPeakHeight',0.5);
```

```
for (j=2:4)
for i=1:length(locs_ECG_R)-1
   diff_ECG(i) = locs_ECG_R(i+1) - locs_ECG_R(i);
end

counter =0;
for i=1:length(diff_ECG)
   if ((mean(diff_ECG)-diff_ECG(i))/mean(diff_ECG)*100 > 10)
      counter = counter +1;
   end
end



 if abs (length(pks_ECG_Rn)-(length(pks_ECG_R))) > 2
       [pks_ECG_R,locs_ECG_R] =
findpeaks(ECG,'MinPeakDistance',round((min_seperation)/(j+1)),'MinPeakHeight',0.5);

       display ('R-peak detection algorithm revised 2')
       FLAG = 1;

 elseif (counter/length(diff_ECG))*100 > 10
   [pks_ECG_R,locs_ECG_R] =
findpeaks(ECG,'MinPeakDistance',round((min_seperation)/(j+1)),'MinPeakHeight',0.5);

   display ('R-peak detection algorithm revised 1')
   FLAG = 1;

 else
   FLAG = 11;
   display('Breaking out')
   break;
end
end

%% Plotting graphs
t_ECG=0:1/fs:(length(ECG)-1)/fs;
t = 1:length(ECG);
figure(1);
%hold on;

hold on;
plot (t,ECG);
%plot(locs_ECG_Q,ECG(locs_ECG_Q),'rv','MarkerFaceColor','b');
%plot(locs_ECG_R,ECG(locs_ECG_R),'<','MarkerFaceColor','r');

%plot(mat,ECG(mat),'>','MarkerFaceColor','g');

%plot(locs_ECG_T,ECG(locs_ECG_T),'^','MarkerFaceColor','y');
legend('ECG signal','R-peak');
xlabel('# of Samples'); ylabel('Amplitude');
title ('PPG and ECG features');

grid on
hold off;



% k= 1;
```

```
% for i=1:length(locs_ECG_Q_raw)
%     for j=1:length(locs_ECG_S)
%
%         if locs_ECG_Q_raw(i)== locs_ECG_S(j)
%             if (i>=length(locs_ECG_Q_raw))
%                 break;
%             else
%             i=i+1;
%             break;
%             end
%         else
%             j = j+1;
%         end
%     end
%     locs_ECG_Q(k) = locs_ECG_Q_raw(i);
%     k = k+1;
%
% end
% locs_ECG_Q = locs_ECG_Q';


%% Calculate R-R interval
for i=2:length(locs_ECG_R)
   RR_interval(i) = locs_ECG_R(i) - locs_ECG_R(i-1);

end

RR_interval_mean = mean(RR_interval)/fs;


%% R-peak verification
for i=1:length(locs_ECG_R)-1
   diff_ECG(i) = locs_ECG_R(i+1) - locs_ECG_R(i);
end

counter =0;
for i=1:length(diff_ECG)
   if ((mean(diff_ECG)-diff_ECG(i))/mean(diff_ECG)*100 > 10)
      counter = counter +1;
   end
end

if ((counter/length(diff_ECG))*100 > 10 && FLAG == 11)
   display ('ECG signal error')
elseif (counter/length(diff_ECG))*100 > 10
   display ('Possible R-peak detection error')
end

t_ECG=0:1/fs:(length(ECG)-1)/fs;
t = 1:length(ECG);
figure(2);
hold on;
plot (t,ECG);
%plot(locs_ECG_Q,ECG(locs_ECG_Q),'rv','MarkerFaceColor','b');
plot(locs_ECG_R,ECG(locs_ECG_R),'^','MarkerFaceColor','r');

for j=1:length(locs_ECG_R)-1
   mean1=round((locs_ECG_R(j)+locs_ECG_R(j+1))/2);
```

```
    ECG1=ECG(locs_ECG_R(j)+5:mean1);
    %figure(1);
    %plot(ECG1);
    xmax = find(max(ECG1) ==ECG1);
    A=min(xmax);
    p1=A+locs_ECG_R(j)+3;
    z1=[z1 p1];

    %disp(p1);  %tpeak
    ECG2=ECG(locs_ECG_R(j):p1);
    %figure(2);
    %plot(ECG2);
    xmin = find(min(ECG2) ==ECG2);
    B=min(xmin);
     p2=B+locs_ECG_R(j);
    %disp(p2);    %speak
    z2=[z2 p2];
    ECG3=ECG(mean1+5:locs_ECG_R(j+1)-7);
    %figure(4);
    %plot(ECG3);
    xmax2 = find(max(ECG3) ==ECG3);
    C=min(xmax2);
    p3=C+mean1+7;      %p peak
    %disp(p3);
    z3=[z3 p3];
    ECG4=ECG(C+mean1+7:locs_ECG_R(j+1));
    %figure(5);
    %plot(ECG4);
    %xmin2 = find(min(ECG4) ==ECG4);
    %disp(xmin2+xmax2+mean1+5);
    ECG5=ECG(p3:locs_ECG_R(j+1)); %qpeak
    qmin=find(min(ECG5)==ECG5);
    D=min(qmin);
    p4=p3+D;
    z4=[z4 p4];
   % disp(p4);

    %p4=xmin2+xmax2+mean1+5;
    %p5=qmin+xmax+locs_ECG_R(j)+3;
    %mat=[p1,p2,p3,p5];
    %disp(mat);
    plot(p1,ECG(p1),'^','MarkerFaceColor','g');
    plot(p2,ECG(p2),'^','MarkerFaceColor','y');
    plot(p3,ECG(p3),'^','MarkerFaceColor','m');
    plot(p4,ECG(p4),'^','MarkerFaceColor','c');
end

disp('T peak locations');
disp(z1);
disp('S peak locations');
disp(z2);
disp('P peak locations');
disp(z3);
disp('Q peak locations');
disp(z4);
disp('R peak locations');
disp(locs_ECG_R.');

for j=1:length(locs_ECG_R)-2
```

```
  qrs(j)=z2(j+1)- z4(j);
  qt(j)= z1(j+1)-z4(j);
  pr(j)= locs_ECG_R(j+1)-z3(j);
  st(j)=z1(j)-z2(j);

end
disp('QRS complex locations');
disp(qrs);
disp('QT  locations');
disp(qt);
disp('PR  locations');
disp(pr);
disp('ST  locations');
disp(st);


qrsmean = mean(qrs)/fs;
disp('QRS complex')
disp(qrsmean);
qtmean = mean(qt)/fs;
disp('QT Interval')
disp(qtmean);

prmean = mean(pr)/fs;
disp('PR Interval')
disp(prmean);
stmean = mean(st)/fs;
disp('ST Interval')
disp(stmean);
disp('RR interval')
disp(RR_interval_mean);
```

Code for Regression used (in Cftool) :

```
p00 =149.8 ;
p10 =370.9 ;
p01 =-87.84 ;
p20 =-1536 ;
p11 =70.08 ;
x=ECG1( :,1); y=ECG1( :,5);
p=x.';
q=y.';
z=x.*y;
disp(z);

r= p00 + p10.*x + p01.*y + p20.*x.^2+ p11.*z; ;
disp(r);

p00 =164.1 ;
p10 =-138.5 ;
p01 =-14.06 ;

x=ECG1( :,2); y=ECG1( :,5);
p=x.';
q=y.';
```

```
z=x.*y;
% disp(z)
r= p00 + p10.*x + p01.*y ;
disp(r);

p00 =      189.2;
p10 =      679.2  ;
p01 =       -267  ;
p20 =      -1464  ;
p11 =     -468.2  ;
p02 =      187.4;
x=ECG1( :,1); y=ECG1( :,5);
p=x.';
q=y.';
z=x.*y;
r = p00 + p10*x + p01*y + p20.*x.^2 + p11.*x.*y + p02.*y.^2;
disp(r);

p00 =     -111.7 ;
p10 =   1.744e+04 ;
p01 =      -1303 ;
p20 =  -8.423e+04  ;
p11 =  -2.953e+04 ;
p02 =       4039  ;
p30 =   2.131e+04  ;
p21 =   1.215e+05 ;
p12 =       5195  ;
p03 =      -2343;

x=ECG1( :,1); y=ECG1( :,5);
p=x.';
q=y.';
z=x.*y;
r = p00 + p10*x + p01*y + p20.*x.^2 + p11.*x.*y + p02.*y.^2 + p30.*x.^3 + p21.*x.^2.*y
+ p12.*x.*y.^2 + p03.*y.^3;
disp(r);

p00 =      240.4  ;
p10 =     -194.1 ;
p01 =       -247  ;
p20 =     -928.5 ;
p11 =      899.6  ;

x=ECG1( :,2); y=ECG1( :,5);
p=x.';
q=y.';
z=x.*y;
disp(z);
r= p00 + p10.*x + p01.*y + p20.*x.^2+ p11.*z; ;
disp(r);
```