

Engn 395 Project 02: *Straight from the Heart*

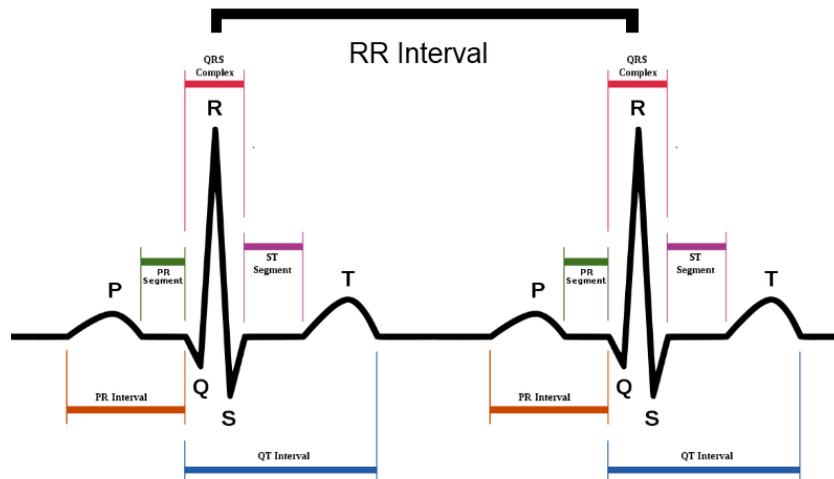


Figure 1. Top: Typical ECG machine seen in a hospital setting (Image credit: [medicinenet.com](https://www.medicinenet.com)): Bottom: Prototype ECG signal showing P wave, QRS complex and T wave (Image credit: https://www.researchgate.net/figure/The-illustration-of-QRS-complex_fig1_260157313). The QRS complex represents ventricular contraction, and marking successive complexes is a standard way to extract the heart rate. For a quick overview of how this signal is generated by the cardiac events, see <https://www.youtube.com/watch?v=RZY4daFwMa8>

Background/Introduction:

According to the CDC, heart disease is the leading cause of death in the United States, accounting for about 1 in every 4 deaths (<https://www.cdc.gov/heartdisease/facts.htm>). The electrocardiogram (ECG) is a standard non-invasive test that measures the heart's electrical activity, commonly recommended when patients present with chest pain, trouble breathing, or a feeling of pounding or fluttering heart. The test "only" requires applying stick-on electrodes to the torso region. Only is in quotes because there's a technological marvel that occurs thereafter. Firstly, in Circuits-land the signal has to be appropriately amplified, filtered, and possibly recorded for later analysis. In Signal Processing land lies another technological marvel, wherein various techniques identify the heart rate in automated fashion and report a number like 74 bpm on a screen for the doctors and nurses to interpret in the clinical/patient context. Even more sophisticated signal processing techniques can be used to identify ECG signal abnormalities such as atrial fibrillation, prolonged QT waves, etc. For now, the task at hand is to do the first best thing—identify successive QRS complexes and thereby determine the heart rate (and/or heart rate variability).

Problem Statement:

Design and implement a signal processing technique that detects the QRS complex of successive heart beats in the ECG signal. Furthermore, determine the instantaneous heart rate (R-R interval) vs. time.

The signal processing you develop should be as robust as possible against typical sources of noise, such as baseline wander and movement artifacts.

Data sets:

We will be using data sets available from [PhysioNet](#), an open-access database for biological signal research and analysis.

Specifically, we will use ECG recordings from the [MIT-BIH Noise Stress Test Database](#), which were converted into Matlab-compatible format and downloaded using the [PhysioBank ATM](#). These files are posted in a box folder.

This ECG database includes six ½ hour recordings (Table 1) with various signal-to-noise-ratios (SNRs). Various levels of noise were added to “clean” recordings to simulate an ambulatory setting. These include baseline wander, muscle contraction artifacts and electrode motion artifacts. Increasing levels of noise were added to the originally “clean” recordings.

Table 1. Data file summary in order of decreasing SNR

Filename	SNR (dB)
118e24m.mat	24
118e18m.mat	18
118e12m.mat	12
118e6m.mat	6
118e00m.mat	0
118e_6m.mat	-6

Take particular note that for these recordings: “Noise was added beginning after the first 5 minutes of each record, during two-minute segments alternating with two-minute clean segments.” So, make sure to analyze at least the first 15 minutes of the recording, which contain alternating clean and noisy segments summarized here:

Time (min)	Data Clean/Noisy	Type of Noise
0-5	Clean	
5-7	Noisy 1	Baseline wander
7-9	Clean	
9-11	Noisy 2	Muscle contraction artifact
11-13	Clean	
13-15	Noisy 3	Electrode movement artifact

Each data file contains 2 ECG signal recordings from a [standard 12-lead ECG](#) placement. We will use the “V1” electrode for analysis. Recordings were made at a sampling frequency of $f_s = 360$ Hz.\

Take particular note that

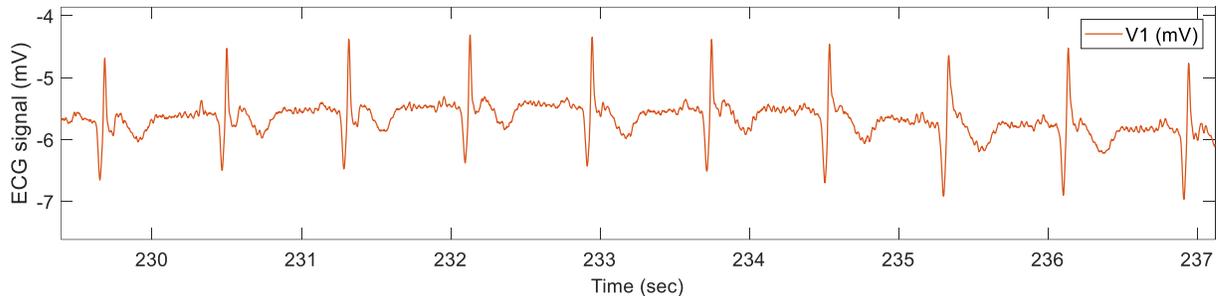


Figure 2 ECG recording from the MIT-BIH Noise Stress Test Database. This is a 'clean' segment with no added noise from the 118e24m.mat file. The QRS complex looks different than the 'standard' one you see due to lead placement at the V1 location, placed medially just to the right side of the sternum.

Getting ECG data into Matlab:

1. Download the data files from box. Take note of the folder into which you save the files. Make that folder your working directory in matlab. For instance, on my machine I do:

```
>> cd ('C:\Users\ericksonj\Box\WandL\Courses_2019\SignalProcessing\projects\ecg\MITstressTest')
```
2. Loading data can be done with the `load()` function. Alternatively, use the Import Data wizard.

```
>> load('118e24m.mat')
```

This should load a variable 'val' into the workspace
3. Remember, there are two leads of ECG data, we want just the V1 electrode, which is in the second row

```
>> V1 = val(2,:);
```
4. Make a time vector. We are told the sampling frequency is 360 Hz, so:

```
>> fs = 360; dt = 1/fs; t = dt*[1:size(V1,2)];
```
5. Make a plot

```
>> plot(t,V1)  
>> plot(t,V1)  
>> xlabel('Time (s)'), ylabel('ECG signal (mV)'), title('118e24.m'),
```

You will NOT immediately see a plot like the ECG trace in Figure 2 because you have now plotted 30 min of heartbeats crammed into a very small window (~2500 total ECG waveforms). You have to zoom in time-wise to see individual ECG waveforms.

6. If you want to crop a section of your waveform, e.g. you want to work with just the clean waveform segment during the first 5 minutes, you can crop a time segment by computing integer indices (these are based on sampling rate):

```
>> tstart = 0; tend = 5*60; %units of s
>> Istart = 1+(fs*tstart); Iend = fs*tend; %integer sample
numbers
>> Vclean = V1(Istart:Iend);
>> tclean = t(Istart:Iend);
>> figure ;plot(tclean, Vclean)
>> xlabel('Time (s)'), ylabel('ECG signal (mV)')
>> title('Clean segment only t= 0-5 min')
```

Other matlab functions you will likely want to use:

Statistics: mean, median, std

Filtering: butter, filter, firlfilt

Other: diff, snoop

Your mission:

1. First start analysis of clean segments. (We will tackle noisy segments in the coming days...) The objective is to write a signal processing technique which optimizes true positives, while minimizing false negatives and false positives. In view of the ECG signals write in “simple, but no simpler” terms possible what your plan is to detect the QRS complex. Be mathematically precise in your terminology. Some thoughts for considerations: Do you apply smoothing? Are derivatives helpful? 1st better than 2nd? Or vice-versa? Does detection with the NEO signal help aid detection?
2. Write matlab code to implement your signal processing technique. See a list of possibly helpful functions above. Consult the Mathworks website documentation as needed:
<https://www.mathworks.com/help/matlab/>
3. This data is available in this box drive folder:
<https://wlu.box.com/s/tif699hvnf1kfuu9om2ehmnq3lumtkv2>
4. Visualize the results of your signal processing technique tagging QRS waveforms. Also, plot the heart-rate vs. time. This is typically done by computing the inter-beat interval and taking the reciprocal thereof.
5. Evaluate the successes (and failures) of your first algorithm. In what specific instances did it correctly tag a QRS complex (true positive = TP)? In what specific instances did you observe the

algorithm incorrectly tagging a non-QRS part of the ECG traces (false positive = FP), or when it missed an actual event (false negative = FN)?

6. Are your results generalizable—i.e., will your signal processing technique work well for voltage vs. time ECG data that isn't exactly like the data set you worked with.
7. How would you revise your methodology to optimize (minimize FPs and FNs, and maximize TPs)?
8. Report your findings with sufficient illustrative figures and explanatory text. 4 pages max! Also, submit your matlab code .m file and copy and paste text thereof into an appendix.