**ECG Processing**

**15.Jan.2021**

I wrote some matlab script to process the ECG data according to the Pan Tompkins algorithm, i.e. the accepted state of the art for human ECG processing in real time systems.

Here are the results.

Raw ECG from ‘ERROR’ channel on Grafana, dated: 12/01/2021 @16h37:59 through to 16h38:04. This data is from Shay, i.e. is human ECG @ 100sps.

Here is the raw signal, a beautifully clean ECG:

Chart, box and whisker chart

Description automatically generated

Although already band pass filtered in the AD8232 analog front end, I further processed it to be consistent with the Pan-Tompkins algorithm (i.e. state of the art for real-time human ECG analysis), who do a 2nd order band pass with pass band between 5Hz and 15Hz.

After this filtering, the signal looks as follows:

Chart

Description automatically generated

We then differentiate the signal to identify the large signal changes, i.e. to accentuate the R peak.

Chart, histogram

Description automatically generated

The beat frequency becomes clearer here intuitively. Actually it starts to look more like the gyro signal – which may play in our favour ;-)

Now, we square it up, to accentuate the larger peaks further from the smaller ones, and rectify the signal

Chart

Description automatically generated

In terms of determining the “BEAT” threshold (i.e. the level above which we consider it to be a beat), I try out a couple approaches using a moving average window, as you can see in the sequence of images below:

Chart

Description automatically generated

The Green line shows the threshold for considering the pulse to be an R peak, and when the signal is above the Green threshold, we then search for the peak in that signal, which is labelled by the red circles in the graph.

Note the black dashed line indicates the noise in the signal – which is VERY low, especially compared to the dashed red line, which is the signal level.

Chart, line chart, box and whisker chart

Description automatically generated

Using this approach, I detected peaks in the ECG interspaced at the following time intervals in the 5s segment: ﻿0.91683s, 1.00000s, 0.88084s, 0.92014s, 0.98062

Thus, the average beat time in this segment is 0.93968s, equivalent to a heart rate of 63.9beats per minute.

On a beat by beat basis, Shay’s HR was: 65bpm, 60bpm, 68bpm, 65bpm, 61bpm

**Conclusions:**

1. This is very encouraging. It means the hardware is working extremely well as designed, and our focus is now really to achieve this on a horse. This needs the following approach
   1. What can we achieve now – with nonoptimal electrode mounting?
   2. What can we achieve with improved electrode mounting?
2. I need to do the above with the Gyro signal to see if I can repeat this reliably (and consistently with the ECG signal) to determine the HR.
3. I want to prove out this algorithm on horse ECG, which may involve some changes in filtering to what I’ve used here, due to the slower HR.
4. This algorithm is relatively easy to implement in real time on the Arduino, we would want to try this out as a separate experiment. No hardware changes required, only software.