Testing on 30s data (stable 2 from 15th Jan @ 13h54)

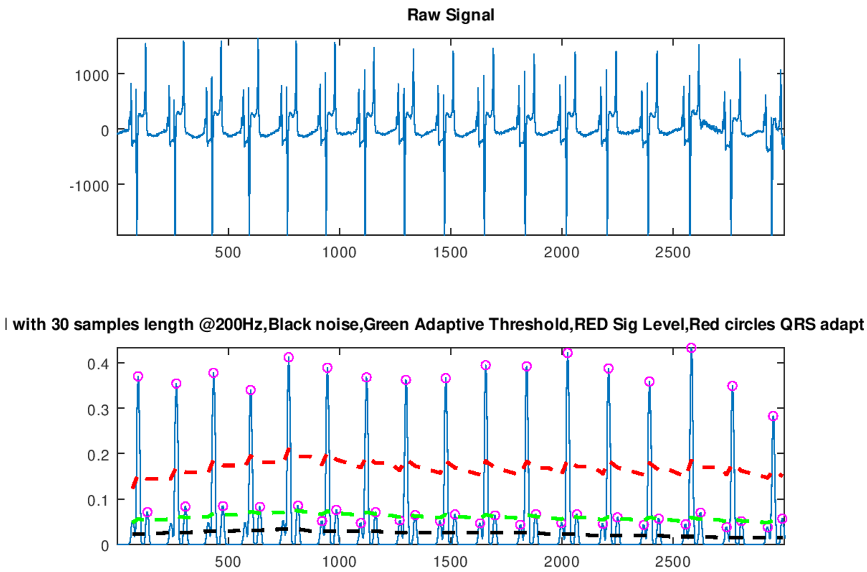
Tested with 15 sample moving average filter & updated s/n thresholds starting at:

Sig = 0.33 * max peak level over the first 2 seconds

Noise = 0. 5 * mean peak level over the first 2 seconds

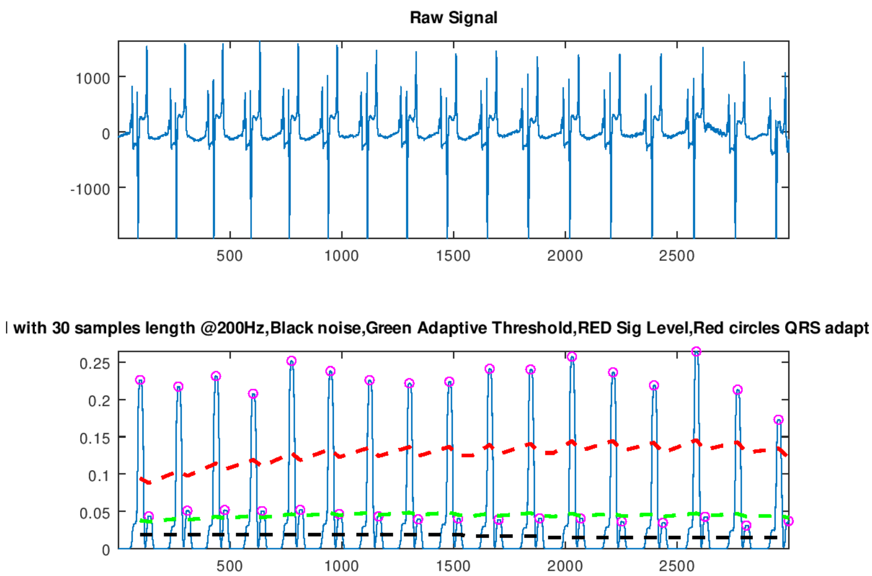
Moving average over 15 samples (per 100Hz of sample rate)

(DEFAULT FROM HUMAN ALGORITHM)



This signal is much 'spikier' than a clean horse ECG, note the false detections of the spikey T wave and the Q feature from time to time.

Same again, but with 30 sample moving average window (rather than 15). This will smooth out the spikes from the signal, and should help with the erroneous T wave and Q feature false detections:

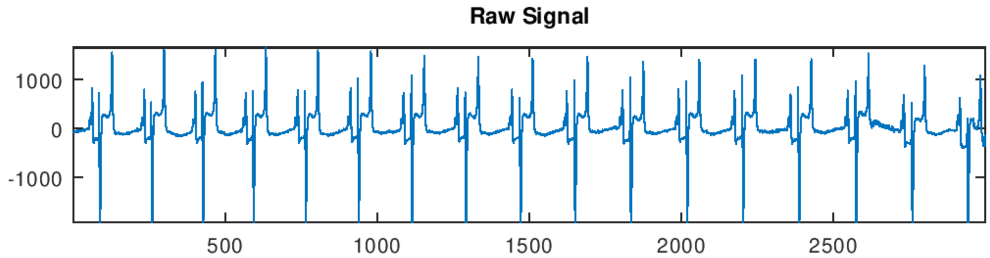


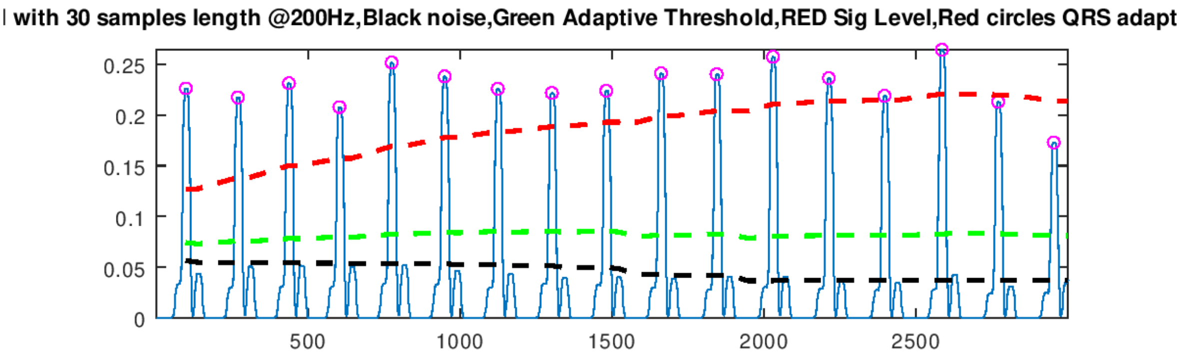
This does remove the Q-wave artefacts, but the T wave is still too spikey.

Next approach is to revert back to the 15 sample moving average window (we wish to preserve the original signal as much as possible) but change the starting Signal and Noise thresholds to the following values:

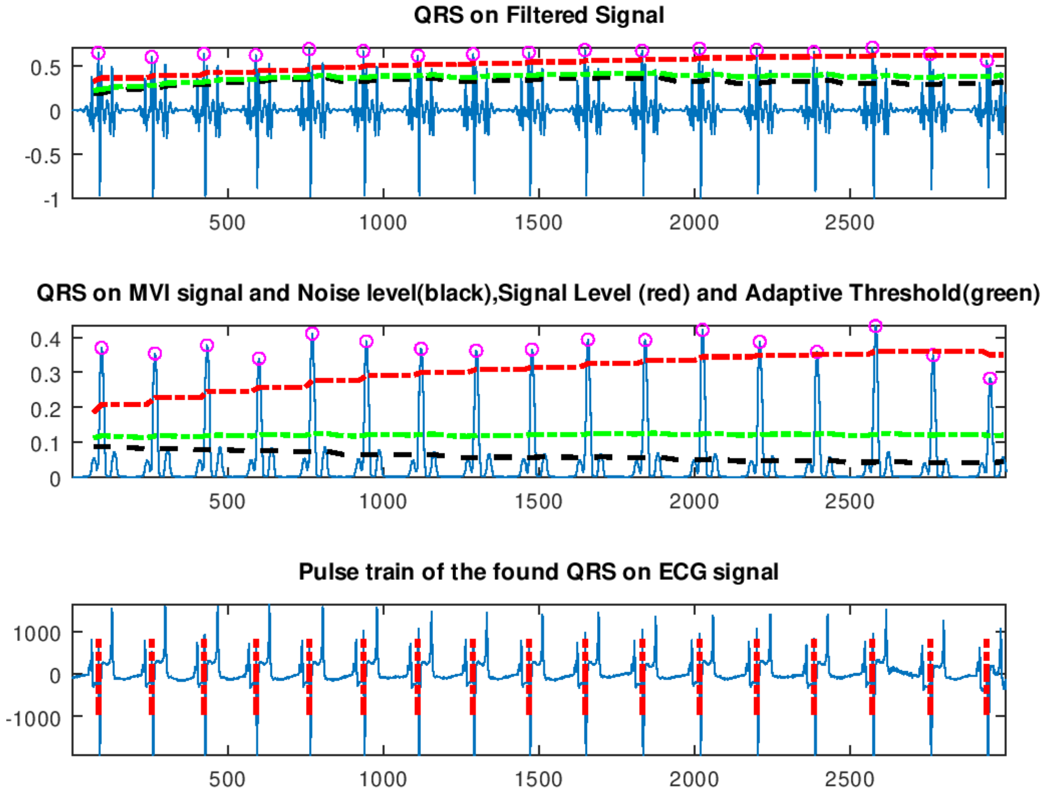
Sig = 0.5 * max peak level over the first 2 seconds
Noise = 0. 25 * mean peak level over the first 2 seconds
Moving average over 30 samples (per 100Hz of sample rate)
(DEFAULT FROM HUMAN ALGORITHM)

This will set the initial R detection threshold much higher than above, and similarly the noise floor will be much higher, at 25% of the peak signal magnitude.

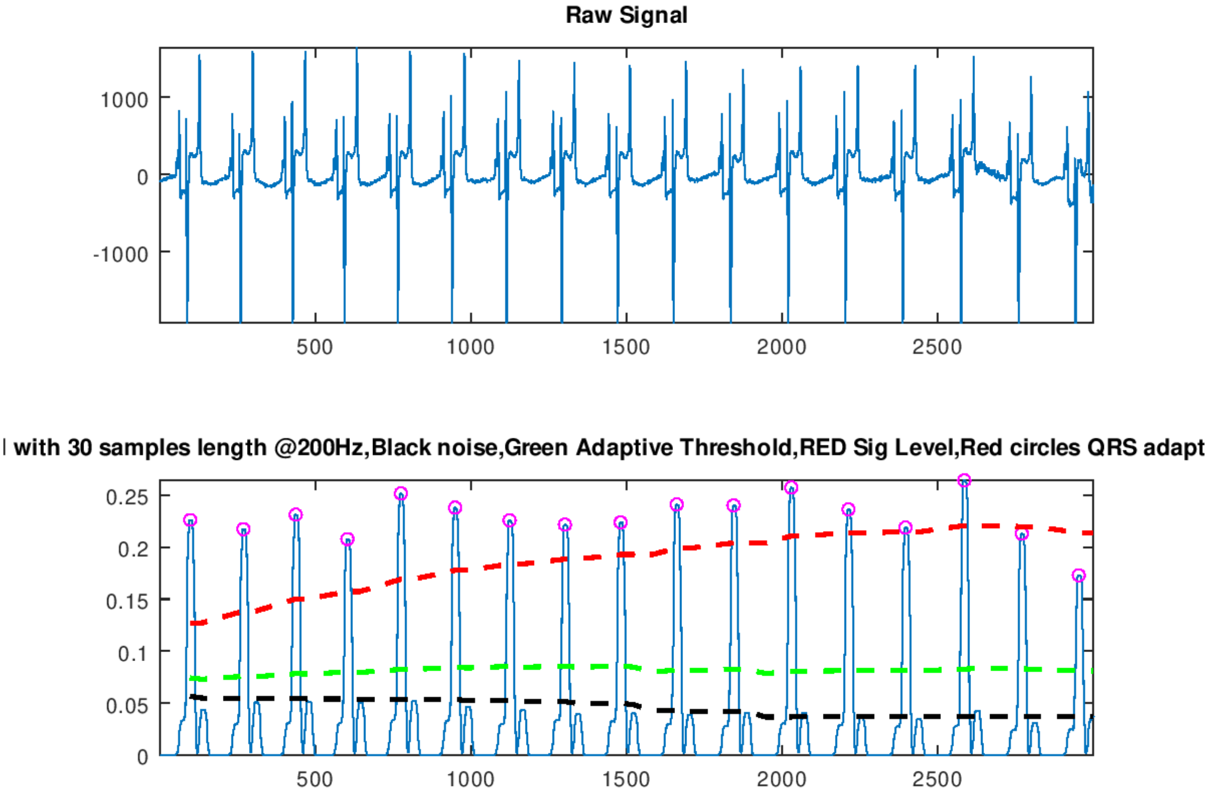




This works very well for this waveform now, with detections looking as per below:



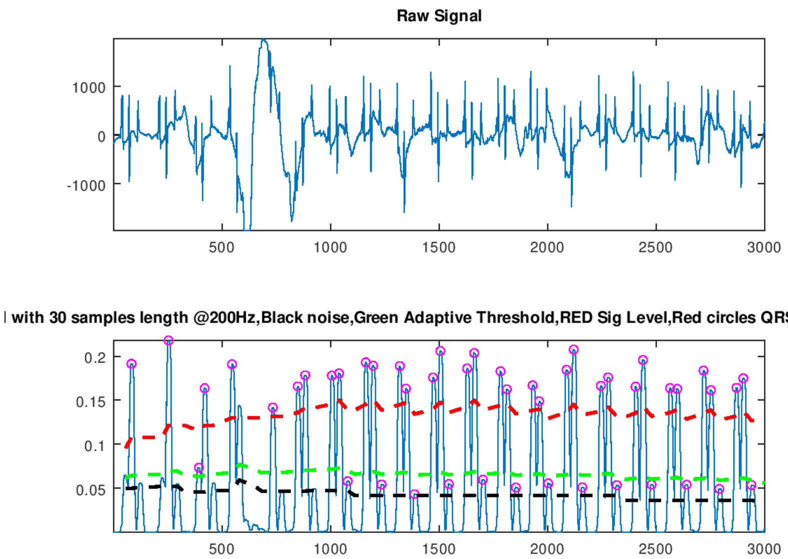
What effect does using a 30sample moving average filter have on the signal (it will introduce a 15 sample = 15ms @ 100hz delay)?



It has a number of effects:

- Reduces the overall peak amplitudes due to smoothing / averaging (including R and T features)
- Reduces the Q feature hugely
- Looks more robust to spikey/dirty data

Let's now try it on different data – also from Stable 2 – but on Jan 19th at 14h28



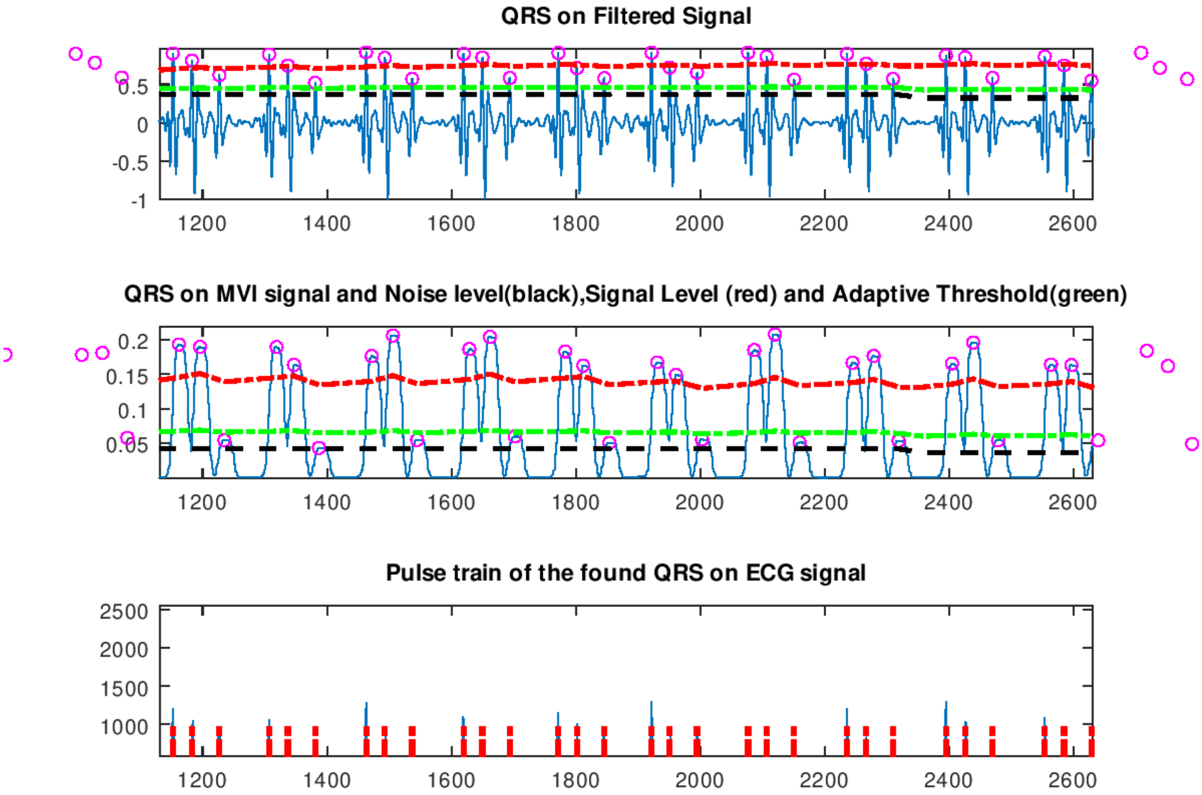
This uses the following settings:

Sig = 0.5 * max peak level over the first 2 seconds
Noise = 0. 25 * mean peak level over the first 2 seconds
Moving average over 15 samples (per 100Hz of sample rate)
(DEFAULT FROM HUMAN ALGORITHM)

Which worked well for spikey (but not noisy) data before.

This data is clearly much lower quality than that previously used. Features can be distinguished, but the HR detection is very poor.

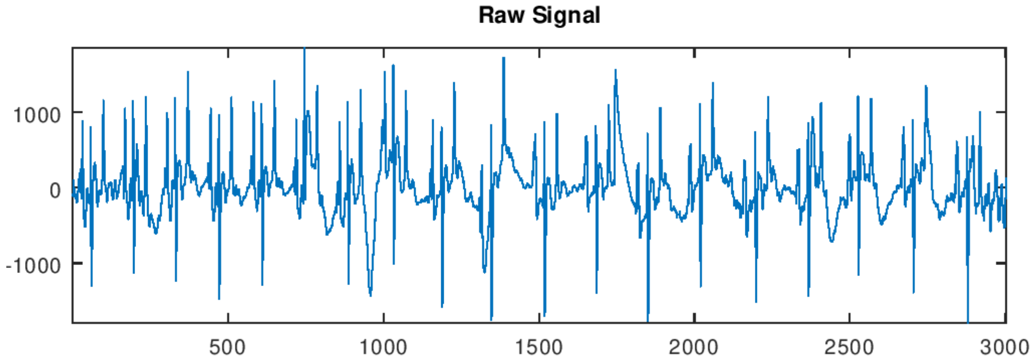
The problem is the size of the spikes which occur throughout the cardiac cycle:

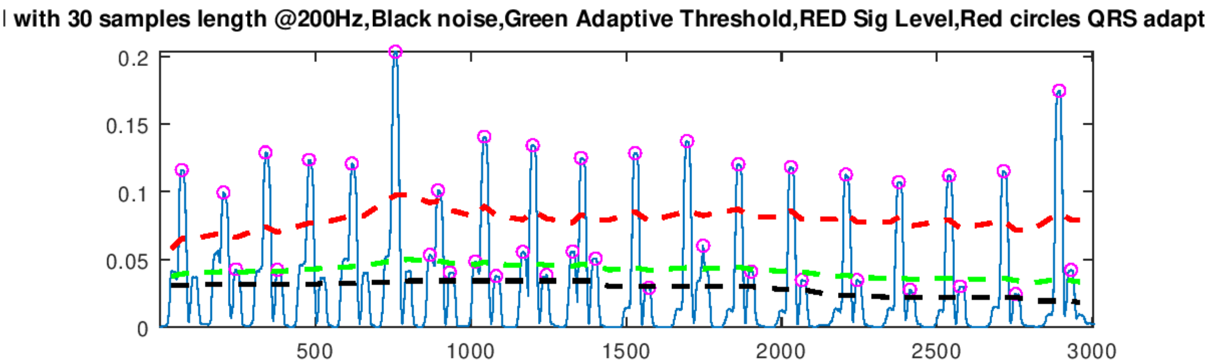


In this case, we end up with 3x the frequency of peaks detected per cardiac cycle.

An idea is whether we can apply physiological principles (refactory times in the cardiac cycle to eliminate any peaks which occur closely after a valid peak, to disregards noise peaks?)

Looking at other noisy data to see how it goes. Here is 30s of data from Stable 2 @23h26 on 20th Jan 2021 when I imagine the horse was resting.



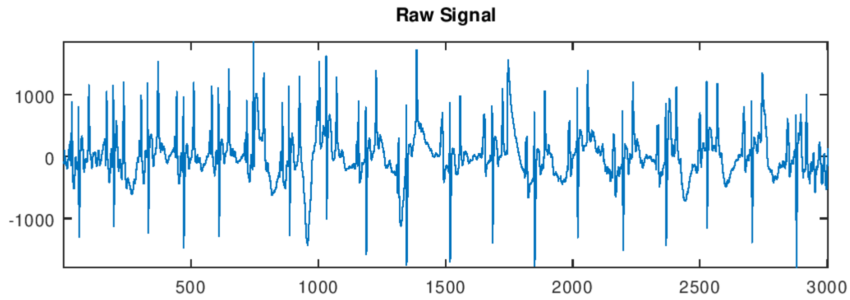


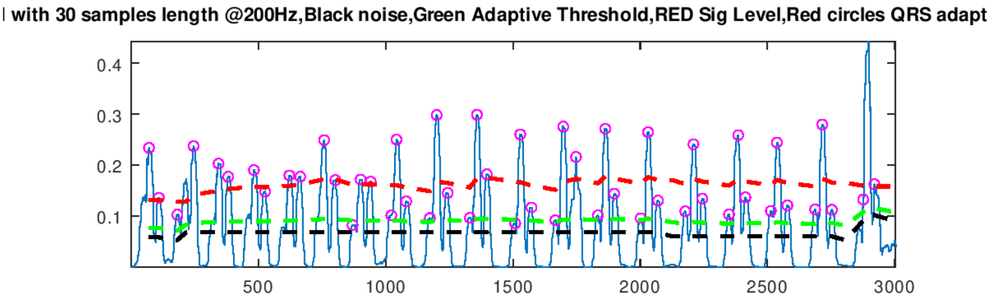
Again there are too many spikes to allow for reliable detection of the base HR.

Possible ideas:

- REALLY heavily low pass filter / moving average filter the dataset to remove all but the most prominent peaks from the signal ?
    - Options:
        - change the band pass filter from 5-15Hz down to 2-10Hz
        - keep the above, but moving average over a greater window (e.g. 50 samples ~= 0.5s worth of data) to really eliminate all but the basic pulse – this is independent of HR frequency so may be the better option than changing the BPF.

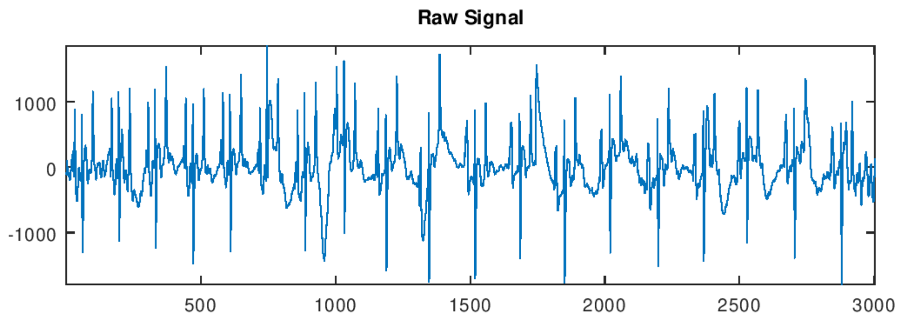
Effect of changing the BPF cutoffs on the same data as immediately before (30s data from 20th Jan at 23h26)

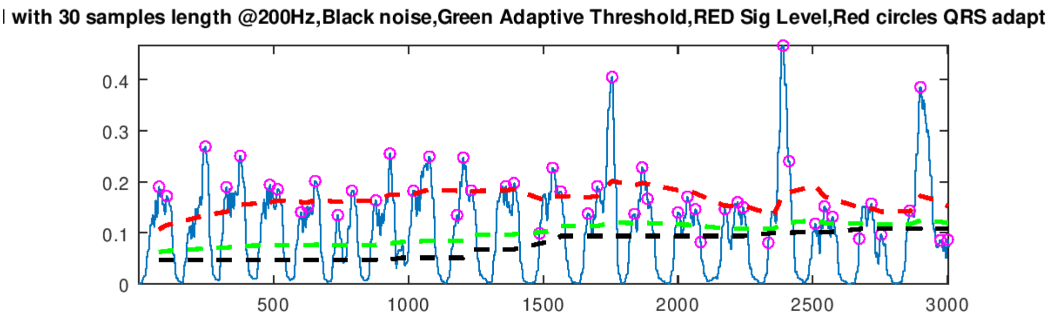




This is actually slightly worse than the previous configuration !! Another worry is that the R peak is much more filtered relative to the problematic Q feature and T wave.

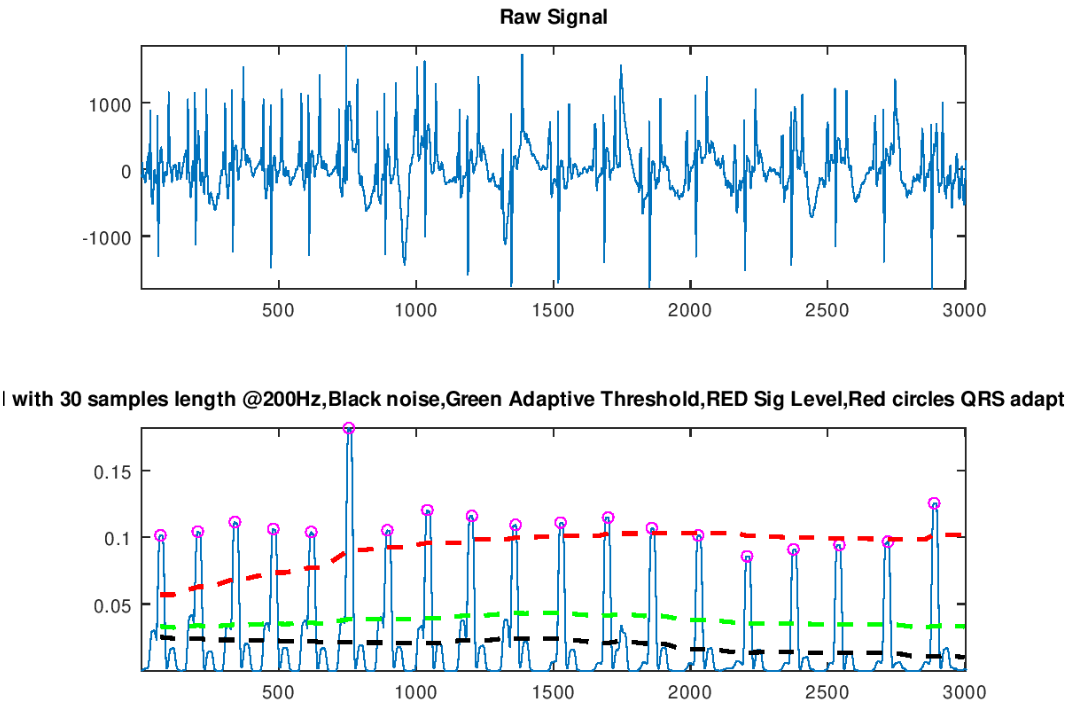
Trying 1Hz to 5Hz:

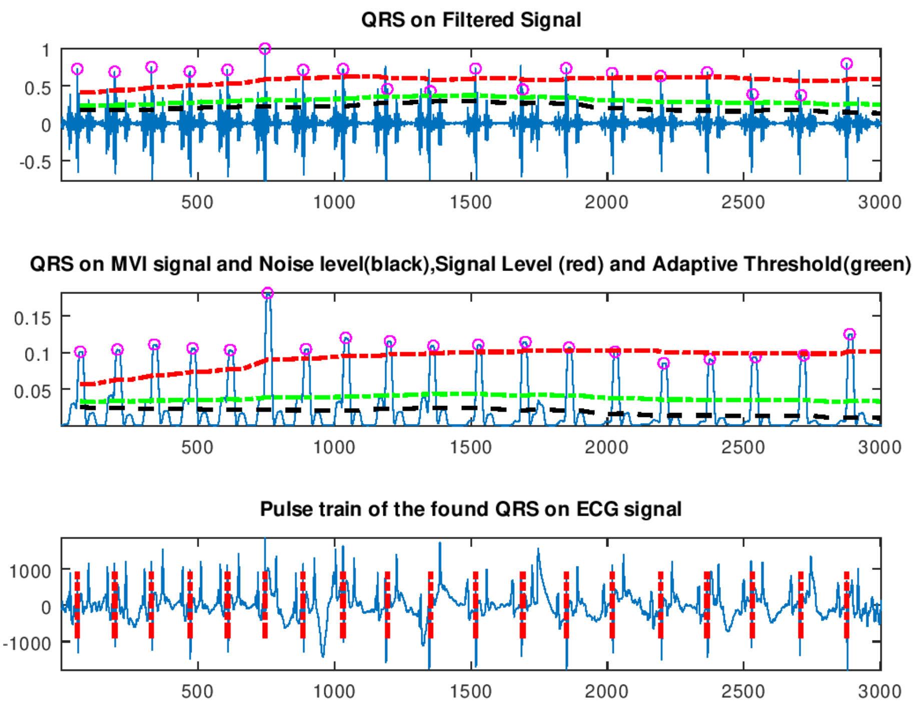




Its even worse!

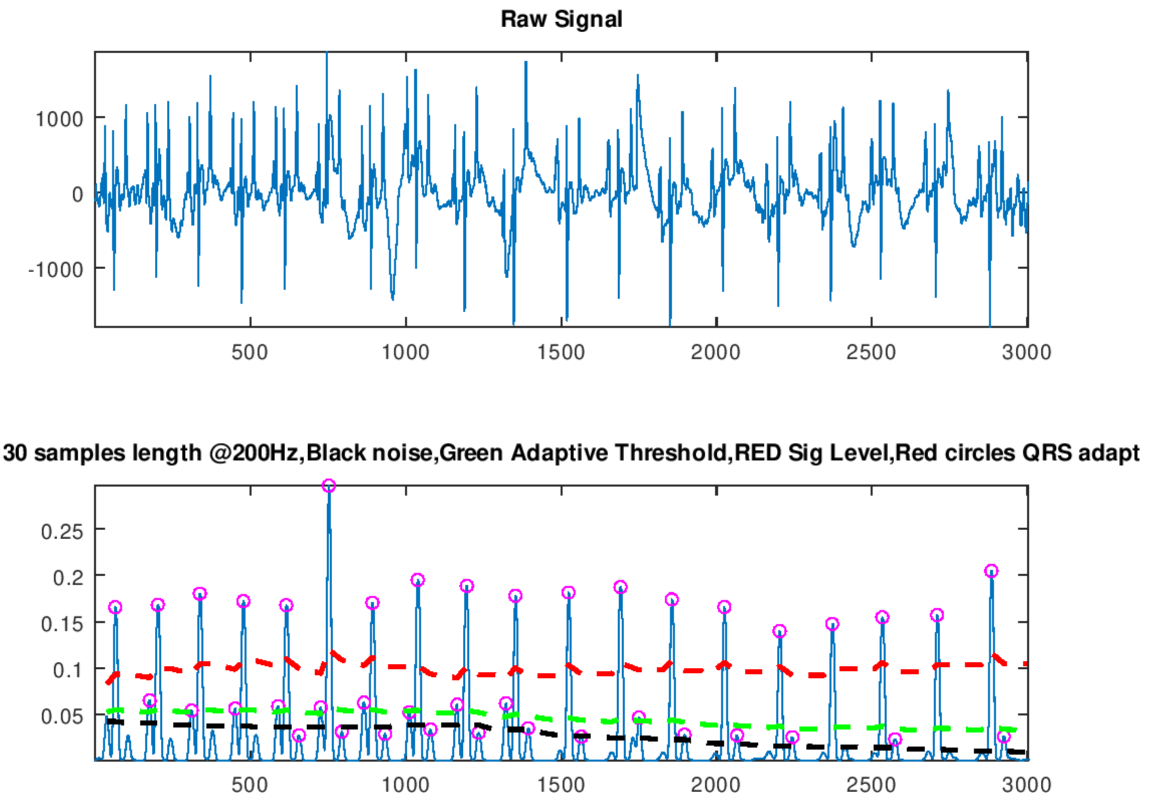
Skipping over intermediate steps, we move to using the BPF set with passband between 10 and 20Hz, which will filter out slower signals (P and T waves, Q and S features) and bias towards the R feature. As this noisy signal has lots of ‘spikes’, it will allow those through – but will set the signal up for more of a ‘battle of the spikes’ where the biggest will win. Let’s see how this work out in practise?



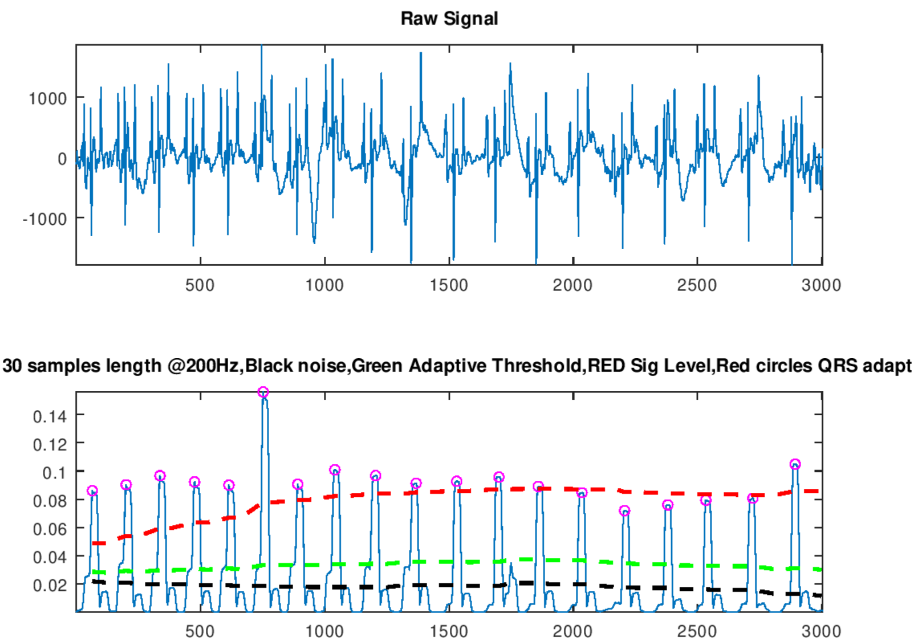


Exceptionally well would seem to be the answer !

Lets see if we can reduce the moving average buffer to 15 samples, what effect that will have?

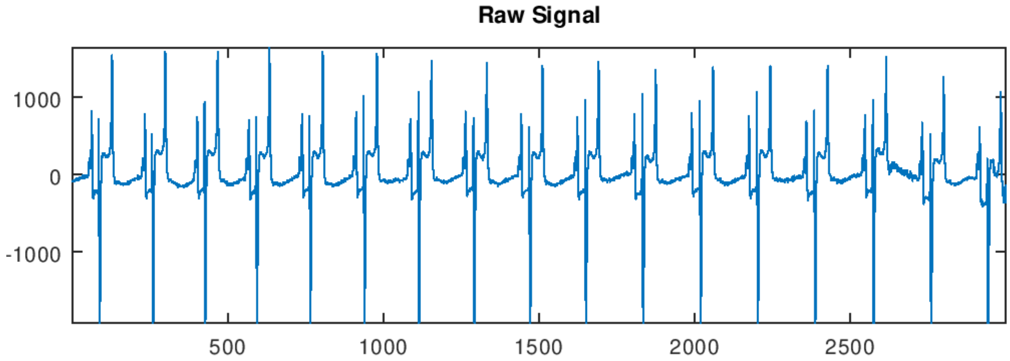


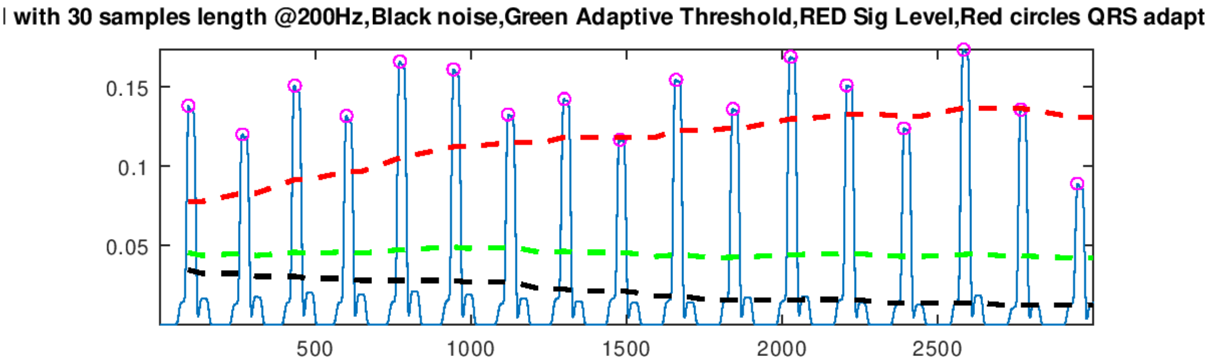
This doesn't work well – need a wider moving average filter. Will 30samples do any better than 25?



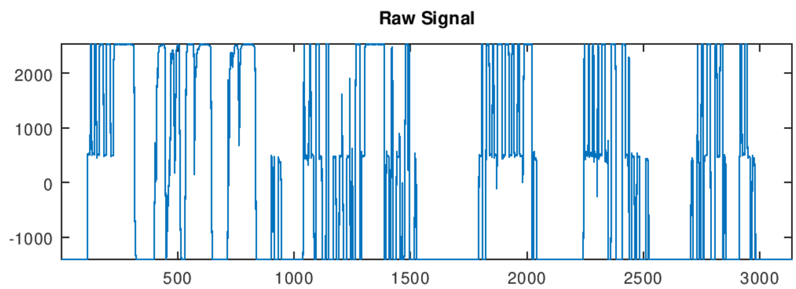
Both detect similarly successfully it would seem. Further trial to see if there are other advantages with dirtier data. But first – let's confirm this new regime:

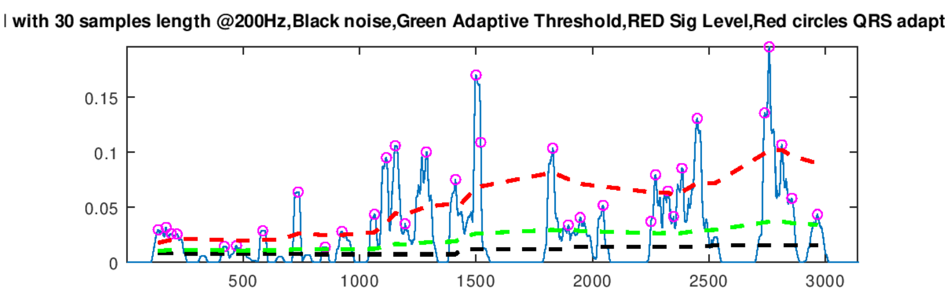
- BPF (10,20Hz) and 30sample moving average window with updated S/N thresholds works with the “cleaner” data from 15th Jan ? (where we started off with)





Yup – looks good. We’ll stick with that config and try to test it on further noisy data from Stable 2 @ 0h57 on 20th Jan 2021, the data is totally noisy.

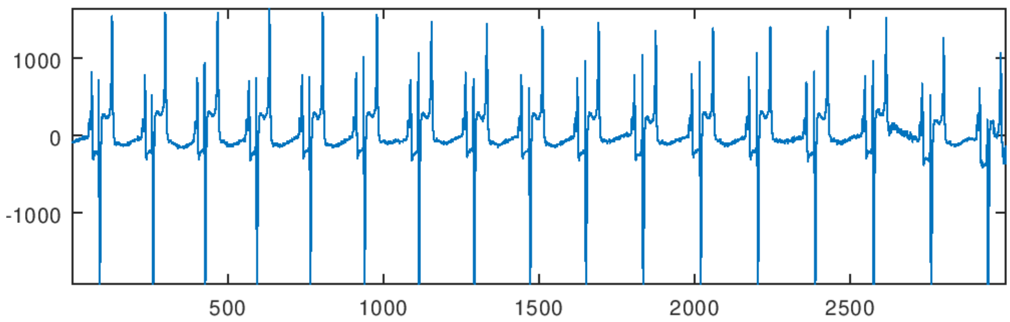




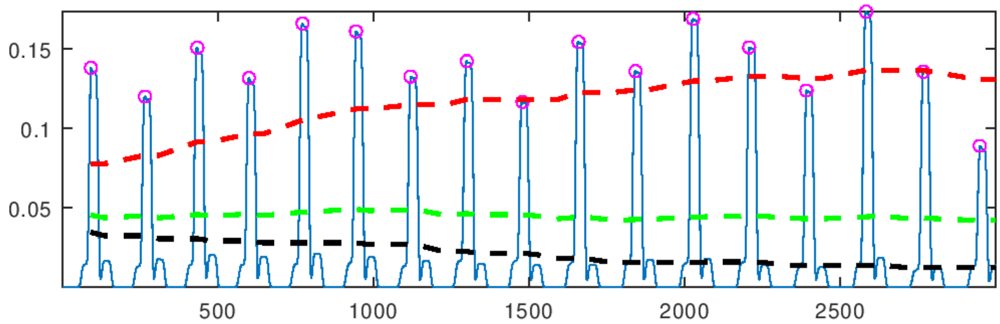
Some peaks are identified, but the HR signal is clearly unidentifiable from that sample set.

Testing this on the original clean, but spikey/distorted signal to ensure no regressive issues with that?

Raw Signal



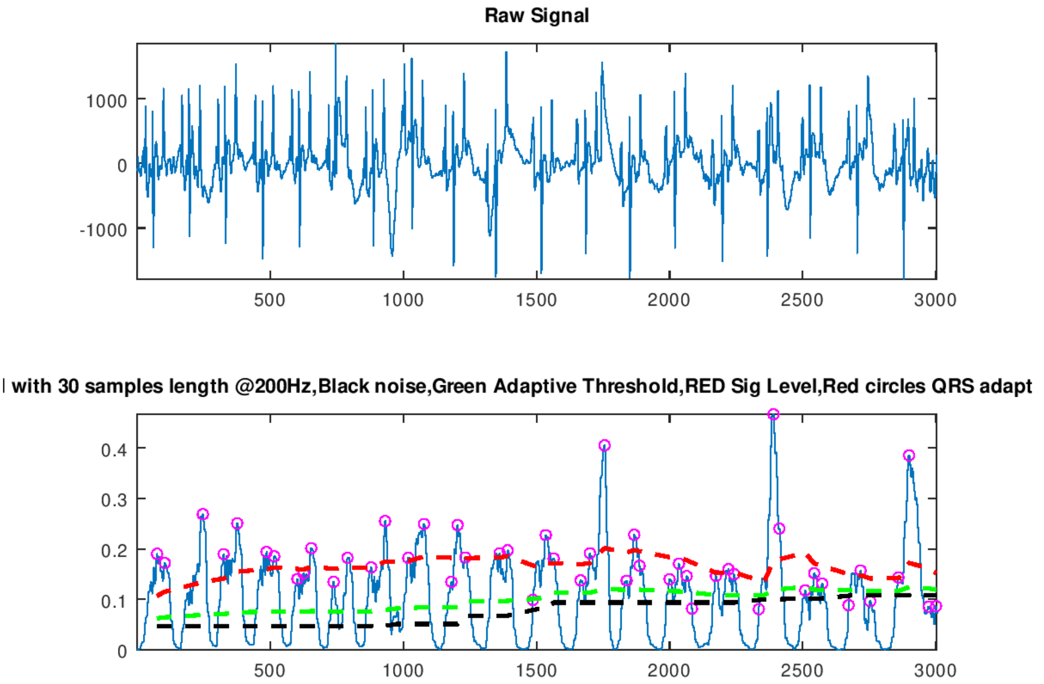
l with 30 samples length @200Hz,Black noise,Green Adaptive Threshold,RED Sig Level,Red circles QRS adapt



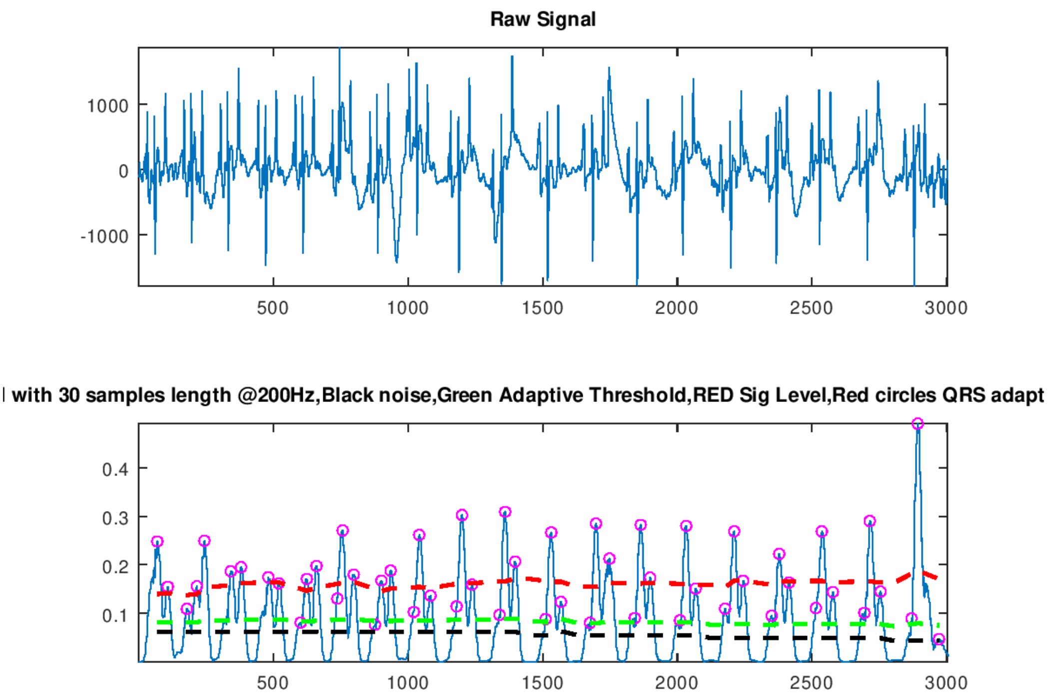
All good.

APPENDIX 1: EFFECT OF THE BPF CUTOFF

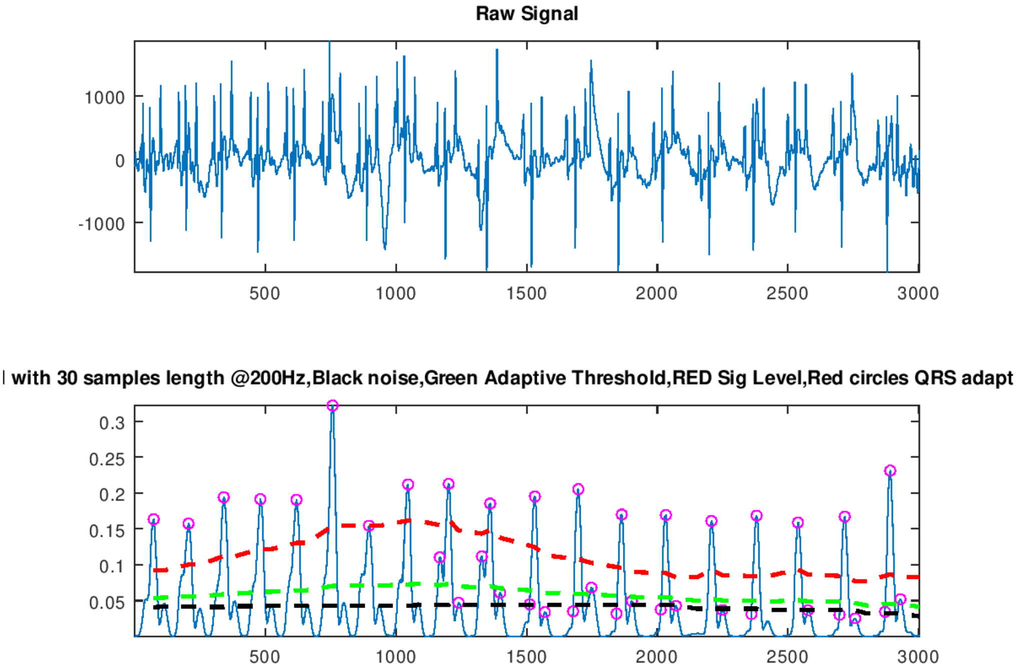
1. BPF=[5, 15]



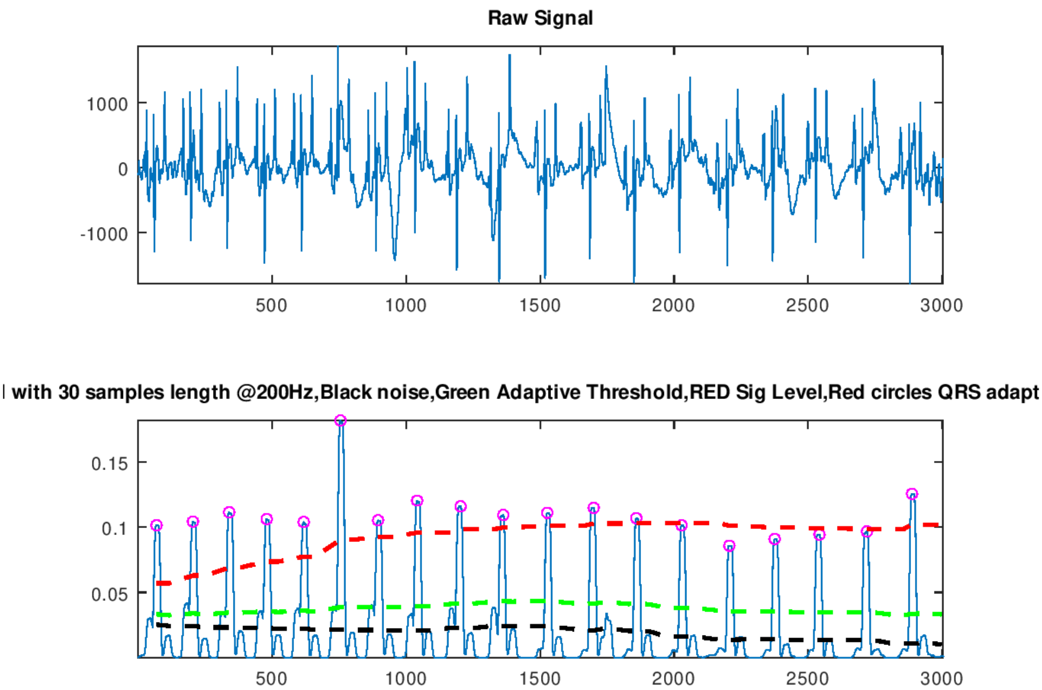
2. BPF=[5, 10]



3. BPF=[10, 15]

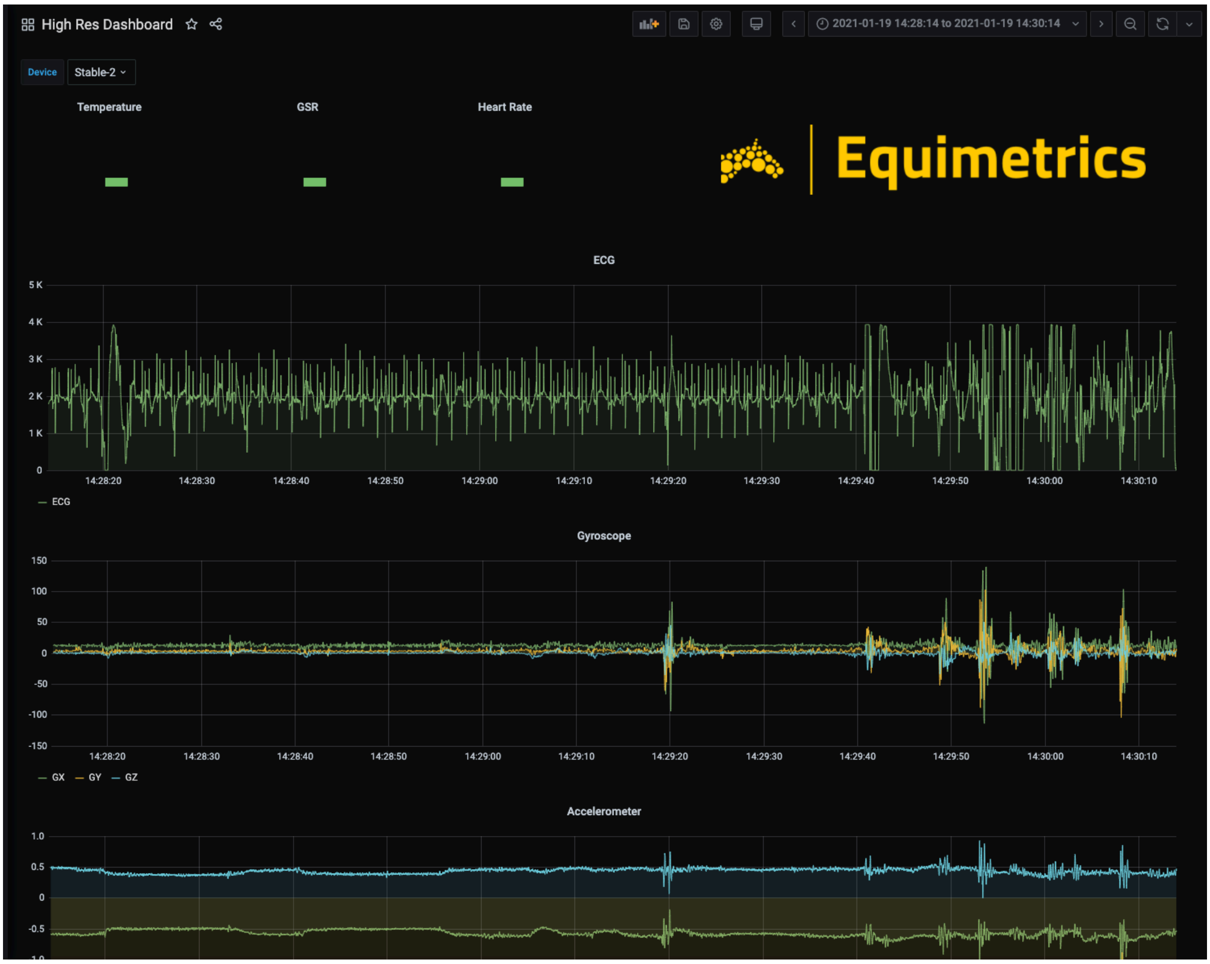


4. BPF=[10, 20]



This appears to work well on spikier data, as it allows the spikiest of all to come through, yet get averaged over 30samples once in…

STABLE 2 – Jan 19th at 14h28:14 to 14h30:14



Clean HR for first 1:45 approx then horse appears to move, with noise artefacts.

- Can we detect from the clean ECG signal?
- Can we detect from the clean GYRO signal?
- Ditto for the dirty signals?