

ENG10004

Digital and Data Systems

Project Report

Name : Joshua Lillington-Moore

Student ID : 103666887

Self Assessment Details

	Few (0-49)	Minor (50-59)	Important (60-69)	Major (70-79)	Major and Leadership (80-100)
Self - Assessment					

- Assuming this was a Group Project, so obviously technically 100% but, I marked it on the quality of my project as a whole, I believe my report is much better than my code. 55-65 for the code, 70-80 for the report

Declaration

I declare that this report is my individual work. I have not copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part of this submission been written for me by another person.

Signature: 

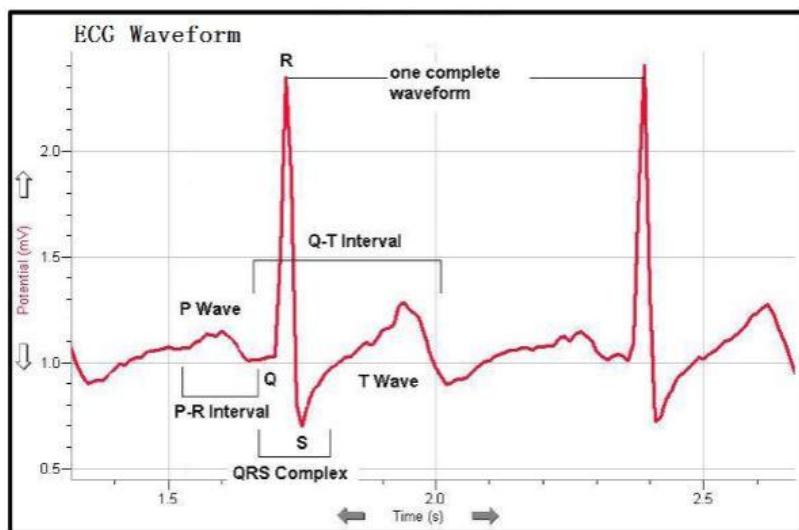
Design Tasks Details

Project tasks

The Project goal for the final assignment of Digital and Data Systems involved creating a software solution in MATLAB for a Heart Rate Monitor and Analyzer, part A and part B respectively. Given a multitude of resources to look at, consisting of MATHWORKS, Labs and Lecture notes, the goal was to use these in order to explore as many different ideas for the software solution as possible.

Starting with Part A, the Heart Rate Monitor. This involved using signal data, which was originally measured using an ECG sensor and developing a software program that was able to, in the most accurate, creative and reliable way possible, keep track of the users heart rate and their BPM. The solution needed to work for multiple samples and needed to be able to detect the peak values of the heart rate and use these values to calculate the BPM and display it in the command window.

Moving on to Part B, the Heart Rate Analyzer is a continuation of Part A. Using similar aspects of Part B required using a filter on the data. This was done by a number of steps shown in Tutorial 9 involving analysing the sensor signal to find the unwanted noise, then designing the filter before applying it and adjusting it till it was right. This filtered data needed to be presented in a figure. Then the software solution needed to be able to detect the peak R values of the graph, as well as the Q,S,T,P points which are shown in the graph below.



Swinburne : ENG10004 : Software Project Guidelines

As can be seen the code needs to be able to detect these specific points, as accurately as possible, and then use markers to show where they are on the graph. Using different markers and colours in order to differentiate between the points. Once these values are found the software needs to be able to calculate the QT and PR intervals as well as the QRS duration. These values are then required to show the user whether the values calculated are in or out of

the normal ranges which were given. Similar to Part A it needs to be able to work for multiple different samples. On a more broader scale, both tasks are required to be completed in Matlab, they must be completed in the time frame given and only samples that are provided can be used.

Background research

The Final Assignment involved using the raw signal data collected by an Electrocardiogram (ECG or EKG). This machine is used in medical situations, mainly hospitals. To read an individual's heart rate by tracking the heart's electrical activity when it beats and then presenting this in the form of a graph. This graph shows the heart's activity, where waveforms will arise when the heart beats. When the graph line is flat, this baseline is called the Isoelectric line and indicates there is no heartbeat. The graph is very easy to read and is helpful when trying to diagnose someone who may have heart problems.

There are five major deflections in which are represented by the letters P,Q,R,S and T and represent the heart's cycle. This starts with the P wave, then Q which goes down, then it peaks at the R wave before going back down to the S wave, then finally it goes back up to the T wave. This heart cycle is quite complex and involves polarization and depolarization. The intervals seen in between the different points can indicate whether a heart is healthy at that moment or not.

When the ECG signal is processed unwanted signals can interfere with the graph. This is why for the solution it is required to use a filter in order to try and get rid of this unwanted noise. These unwanted noises can be broken down into Artifacts and Movement/Motion artefacts. Artifacts can create issues by merging with the ECG signal and include things such as powerline interference and baseline wander. Movement/Motion artefacts only occur at very low frequencies so high pass filters can be used to prevent their interference.

When choosing between high or low pass filters, a low pass filter will provide smooth control removing high frequency noises and jitters, while a high pass filter will remove low frequency noises. There are also bandpass and bandstop filters which provide their own functions however won't be very useful for this software solution.

Final design solution

For my Final Design Solution I did not transition very far from the base code that was shown in Lab 9, as I found this to be easiest to adapt from. Starting with Part A, I kept the solution as simple as possible. In the beginning I attempted to first filter the data in order to try and get a more accurate reading for the beats per minute however once I learned about Prominence and how it could be used in the findpeaks to narrow down the peaks found I no longer needed

filtering for part A. I opted to use `findpeaks` as it was definitely the easiest option compared to hard coding in actual math equations to solve for the derivative. It is a function for a reason and allowed for more freedom, if it required only looking for one peak than it could've been justifiable but using `findpeaks` made it so it would find all the peaks for a given criteria that could easily be adjusted and was shown in a very simple example in tutorial 9. To find the bpm I made a slight change to the equation I was shown in the Week 10 Lab, as originally I had this shown below :

```

PartA.m | TryingPeaks.m | PeaksAttempt3.m | attempt4.m | PartB.m | tryingstuff5.m | tryingpeaks6.m
1 -   fileName = 'Sample_1.mat'; %choose which sample
2 -   Loaded = load(fileName);
3 -   VariableNames = fieldnames(Loaded);
4 -   x = Orig_Sig; %sample is loaded than the original signal is made into x
5 -
6 -   Sec = 10;
7 -   rate = 360;
8 -
9 -   figure(1) %plots original data
10 -  subplot(2,1,1)
11 -  plot(x,'b-')
12 -  title('Time domain');
13 -  xlabel('Time');
14 -  ylabel('Voltage');
15 -  %need to smooth in order to get rid of multiple peaks in similar location
16 -  [peaks, locs] = findpeaks(x,'MinPeakHeight',1000,'MinPeakProminence',150);
17 -  hold on; plot(t(locs(1)),peaks(1),'s','MarkerSize',7)
18 -
19 -  numpeaks = length(locs);
20 -  avedist = (max(locs) - min(locs))/(numpeaks-1);
21 -  bpm = (avedist/rate)*60;
22 -  fprintf(bpm);
23 -  fprintf(bpm);

Command Window
>> tryingpeaks6
Unrecognized function or variable 'fprintf'.
Error in tryingpeaks6 (line 23)
fprintf(bpm);

>> bpm
bpm =
55.3333

```

This was only working for sample 1, however when I readjusted the equation and made it equal to this as shown below the bpm was correct for all samples except 8 :

```

PartA.m | TryingPeaks.m | PeaksAttempt3.m | attempt4.m | PartB.m | tryingstuff5.m | tryingpeaks6.m
1 -   fileName = 'Sample_2.mat'; %choose which sample
2 -   Loaded = load(fileName);
3 -   VariableNames = fieldnames(Loaded);
4 -   x = eval(['Loaded.',VariableNames{1}]); %sample is loaded than the original
5 -
6 -   Sec = 10;
7 -   rate = 360;
8 -
9 -   figure(1) %plots original data
10 -  subplot(2,1,1)
11 -  plot(x,'b-')
12 -  title('Time domain');
13 -  xlabel('Time');
14 -  ylabel('Voltage');
15 -  %need to smooth in order to get rid of multiple peaks in similar location
16 -  [peaks, locs] = findpeaks(x,'MinPeakHeight',1000,'MinPeakProminence',150);
17 -  hold on; plot(t(locs(1)),peaks(1),'s','MarkerSize',7)
18 -
19 -  numpeaks = length(locs);
20 -  avedist = (max(locs) - min(locs))/(numpeaks-1);
21 -  bpm = (avedist*rate)*60;
22 -  fprintf(bpm);

Command Window
>> tryingpeaks6
Unrecognized function or variable 'fprintf'.
Error in tryingpeaks6 (line 23)
fprintf(bpm);

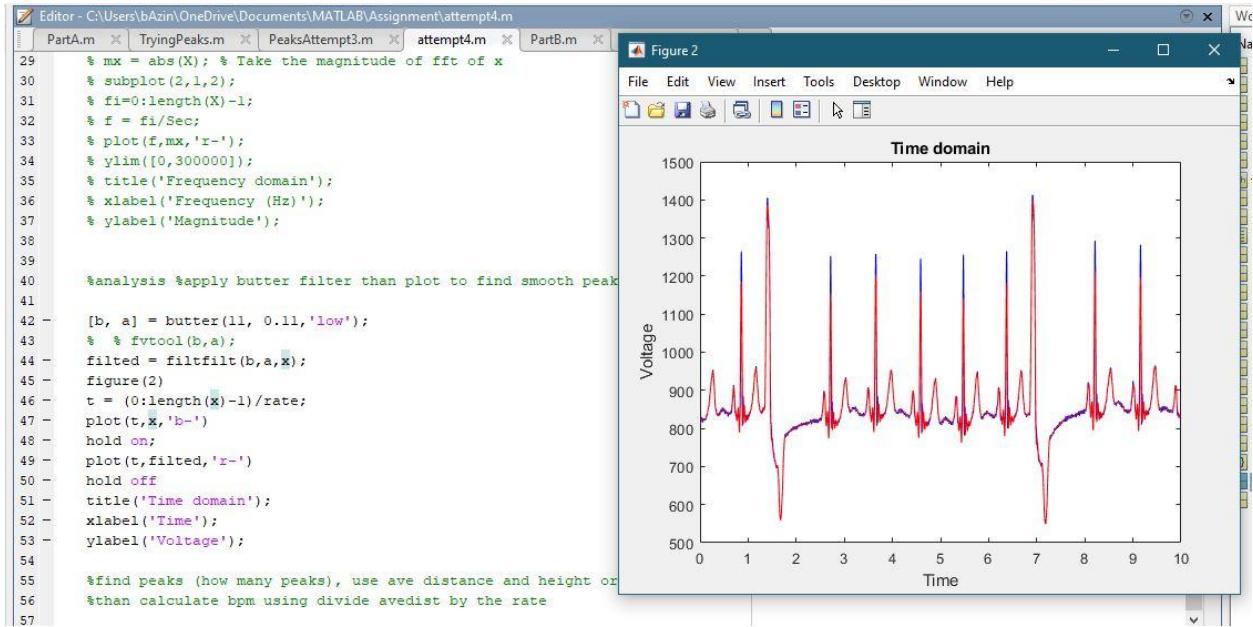
>> bpm
bpm =
60.4478

```

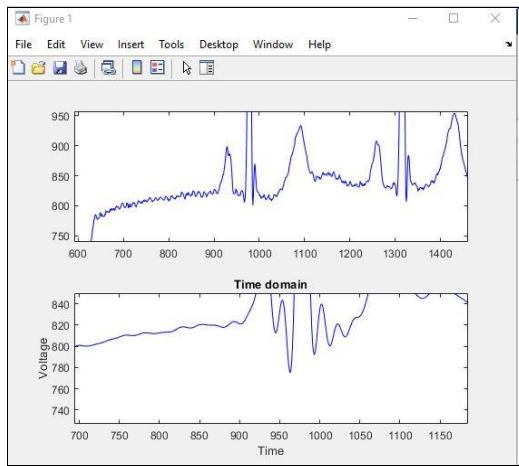
I had to make adjustments to the min peak prominence as well as the min peak height in order to get all the correct calculated values. I made a spreadsheet comparing them to the true values that were told by a lecturer. Once I was able to get all of these correct I moved on to part B.

Sample :	True Results :	Measured Results		
1	65	65.06		
2	60	60.4478		
3	90	90.48		
4	89	89.8396		
5	49	49.3473		
6	53	53.3169		
7	85	84.8082		
8	103	173.3135	outlier	can't fix without reworking others

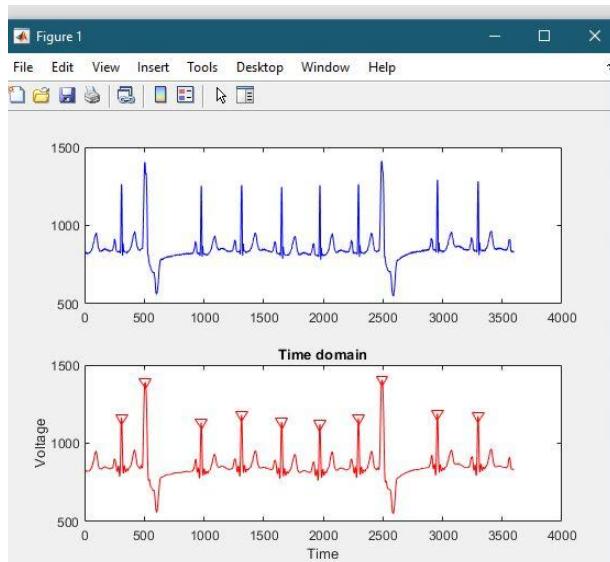
Firstly I needed to backtrack, originally I had filtered my data using a butter filter for part A. However I didn't need to use this until Part B. I followed similar steps to what was shown in tutorial 9 where I analysed the signal to look for unwanted noise. Firstly plotting it in a figure and then looking for the section that I needed to look at. Once I did that I zoomed in to look for any bumps. I took out the constant using detrend like in tutorial 9. I found a few small bumps which can be seen below. However nothing super helpful for the filter. I tried to replicate the segment and it showed a bit more but this was still off.



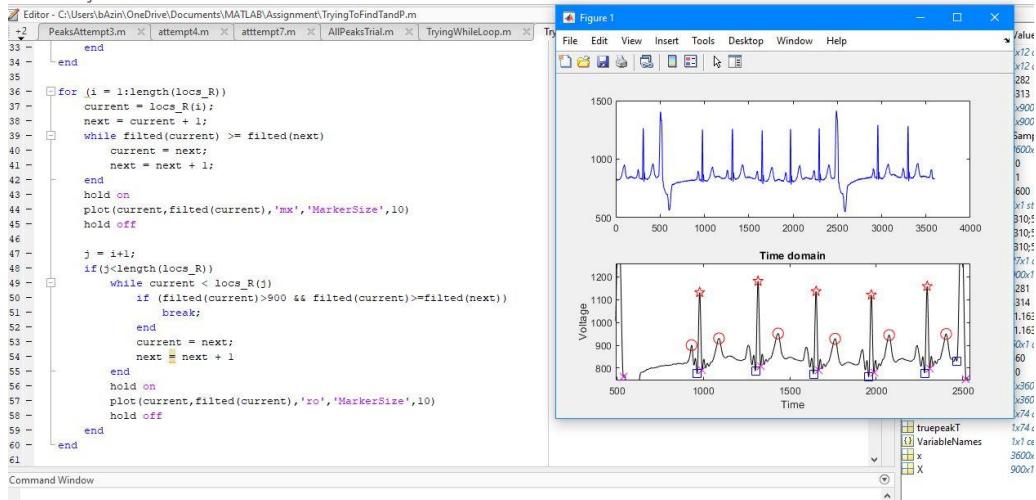
I decided to just try the butter filter. I started by calculating W_n and messing around with the nth degree filter. I originally graphed the filtered over the unfiltered to see the differences like in tutorial 9. Once I got a filter I thought was good as shown above. Then I graphed both the filtered and unfiltered in a figure shown below :



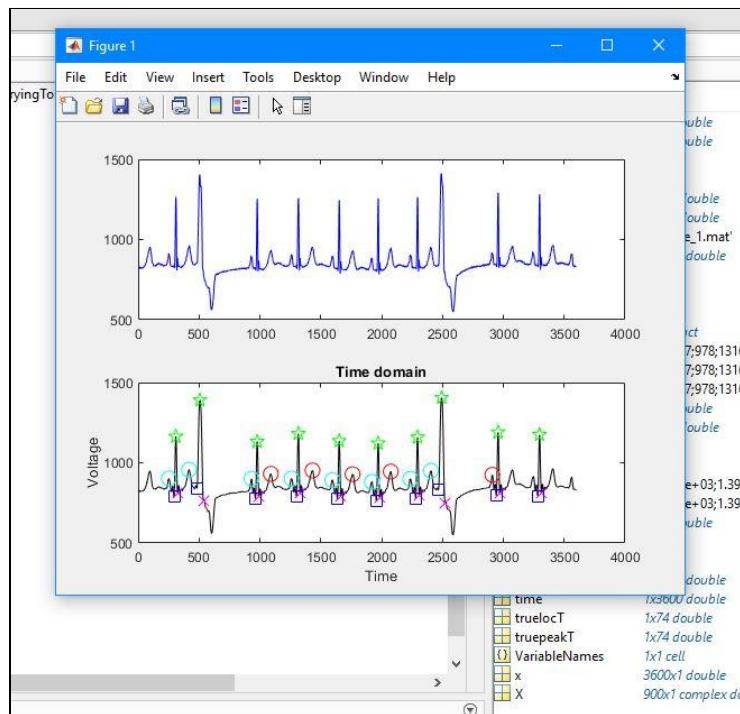
Once I had filtered my graphs I moved on to finding the Q,R and S points. To find the R peaks I used the same code I had in Part A and then just plotted them so they would appear on the graph as shown below.



To find the Q and S points I used the sample code from tutorial 9 and expanded on it. That code allowed users to find the next peak in the line for specific restrictions, so the challenge was to make it so it would find the next peak next to the R peak for all R peaks so that it would label all of the S peaks. To find the Q peaks involved the same loop however looking for the left hand side so it was a minus (going backward) . This is shown below. I would've liked to properly understand the code instead of just tinkering with it till it was right, however time became a real issue.



To find P and T I used code that I found in one of the week 12 labs. There were other ways to do this however I only had a small understanding of them such as creating an array using S and Q values and then looking for specific points in that segment however my coding ability was not good enough to do it in the time I had given myself. Forcing me to adapt the code from the week 12 lab slightly to create a loop for points on the LHS and one for the RHS. Again similar to the loop to find the Q and S points, still using a while loop to get all of the points, it creates a new variable 'j' and looks for the point next to the variable 'i'. Then it looks for the peaks next to Q or S and if they are above a certain value it will plot them. These are not the true P or T values and this affected the calculations for finding the intervals. Below is the finished graph with all of its markers plotted. The black line was chosen in order for the symbols to stand out.



This was by no means the best way to find the T and P points however due to a lack of time and better coding ability it was the best I could do. For the Q and S points I could've created an if statement that included two variables to find peaks between two y locations however the values would have to be changed for every single graph. Another option was to find the individual x components for each peak in the sample however this is incredibly unrealistic and would require a mountain of unnecessary code. Taking a segment and looking for specific peaks in that segment on certain conditions was the easiest method. To find the intervals was also very rushed, I knew it was possible to create an array/matrix and create new values for the locs and peaks to find more specific intervals between each point. However due to a lack of time I had to use a really simple equation involving using the mean and then dividing by the rate. The if statements to print depending on the situation were the only thing that made logical sense. I didn't get to try and do it the proper way due to time so nearly all of the values were wrong. Thus the only justification I have for this decision is a lack of time / poor time management.

Conclusion and recommendation

Finally to finish this report. Although it didn't go as planned, I was able to complete most of the requirements for both Part A and Part B. For Part A I was able to achieve all I set out to. Although struggling at first it was a much simpler code than I had imagined and was able to get it working to find the required peaks to calculate the BPM for all of the samples except sample 8. Moving on, Part B was much more of a struggle. It was difficult to visualize how to write the code and knowing which functions I needed to use and variables I needed to create in order to get it to do what I had envisioned for it to do. After going through many different tutorials through weeks 9-12 and learning as much as possible I was able to mark the correct Q,R and S points. And get somewhat close to finding the true T and P points. I could've done this by taking a segment between certain areas, for example Q and S and then using `findpeaks` to find the actual values. The coded solution definitely could've been a lot better for Part B. And I know of multiple other ways I could've coded it that would've worked much better. Including calculating a more accurate interval by creating new variables and using empty vectors. However I was not able to finish Part B due to miss-managed time and a lack of critical thinking and coding skills. This meant there was lots of room for improvement, and taught me learning coding was more difficult than I thought it would've been. Overall I learnt a lot however there is much room for improvement for Part B of the software solution.

References

Used MathsWorks throughout, however didn't want to reference them too many times :

MathWorks. 2021. *Prominence*. [online] Available at:
<<https://au.mathworks.com/help/signal/ug/prominence.html>> [Accessed 28 October 2021].

MathWorks. 2021. *findpeaks*. [online] Available at:
<<https://au.mathworks.com/help/signal/ref/findpeaks.html>> [Accessed 2 November 2021].

Design Brief Provided

Labs / Tutorial Worksheets and Recordings

Reflective Journal :

Week 7 :

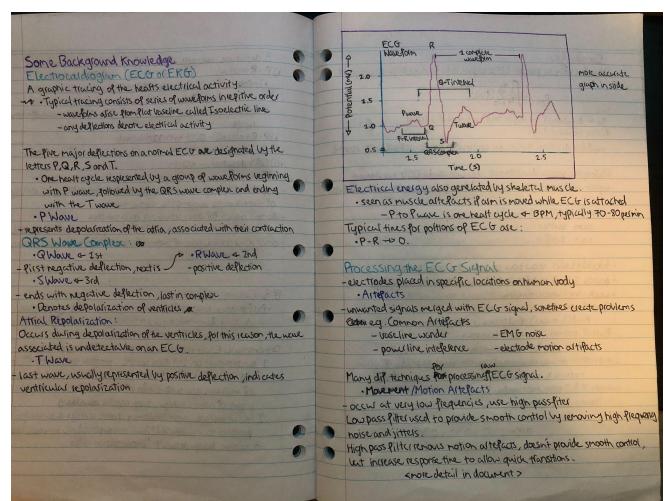
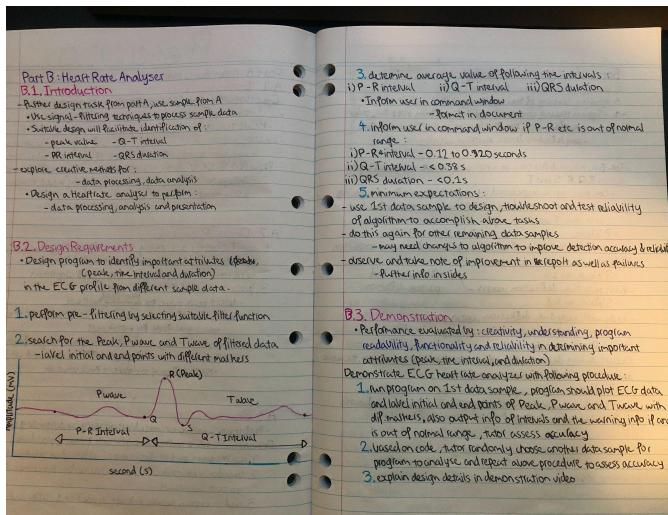
Starting this journey from Week 7 the main tasks were still to do with the portfolio assignment. I focused on completing this by Sunday and was able to do so. For Software Design it wasn't properly introduced yet however I did spend some time skimming over the different links in Canvas to look at the assignment and what was required. Also this week I spent time looking through the lecture notes for Week 7 which had some helpful information for the Software Design however nothing vital.

Week 8 :

In Week 8 I didn't attend the weekly Lab live, however I rewatched it in my own time and this seemed like a much more helpful one than Week 7. We learnt about the basics of filtering in MATLAB. The different types of filters and some super basic code which will probably be useful for the Software design. Next week the Presentation Slides are due on the 8th (Friday). I had a short look at the requirements for it but nothing else. I looked at the Lecture Slides for this week but prepare to summarize them later down the line.

Week 9 :

The busiest week so far week 9 involved finishing the presentation slides to be submitted by Friday. The first thing I did this week was attend the Lab. This Lab was the best lab and coming back to the journal after everything is done I can see it was easily the most important one I attended. I stayed for around an hour and followed along with the class. However I did not complete all the code in MATLAB, as I planned to come back to it later when I started to do my actual code. I looked at the lecture notes again this week however nothing of much value was there yet. Most importantly this week I finished my presentation slides, firstly I went through and summarized the entire project guideline to get a much clearer idea of what the slides and future tasks entailed. I made sure to look at all the criteria to get as good of a mark as I could on the slides. I created a summary for each topic in a separate document before creating the presentation and recording my voiceover. A page of the notes are shown below :



Week 10 :

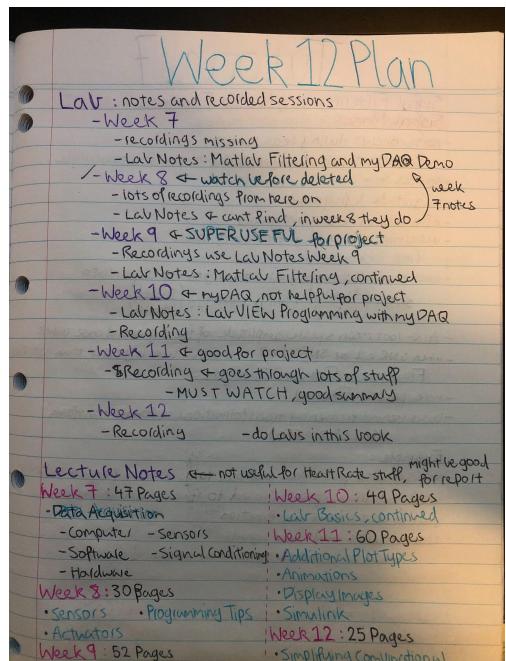
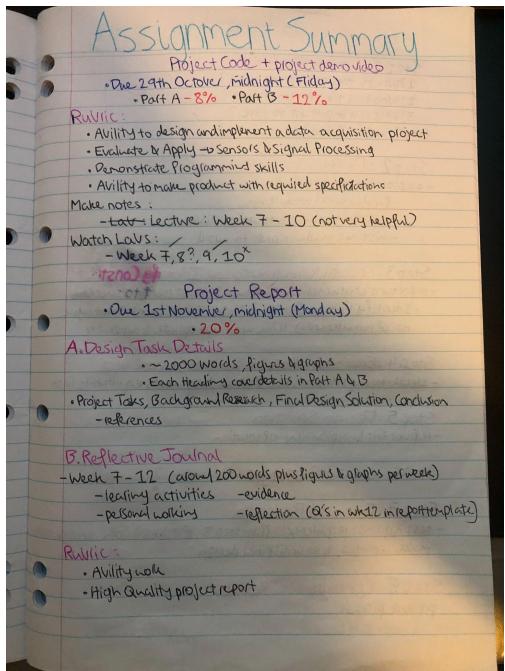
In Week 10 I did a lot less work than originally intended. I spent roughly around 28 hours this week preparing for my maths test so didn't have much time for Data and Digital. As well as this I spent a lot of time this week trying to get ahead in physics so that I would have more time in Week 11 and 12 to focus on my software project. Despite this I still attended the weekly Lab, however this week we learnt about myDaq and Labview. Although it was interesting it wasn't helpful to the software project. Again I looked over the lecture notes but it was nothing of great importance.

Week 11 :

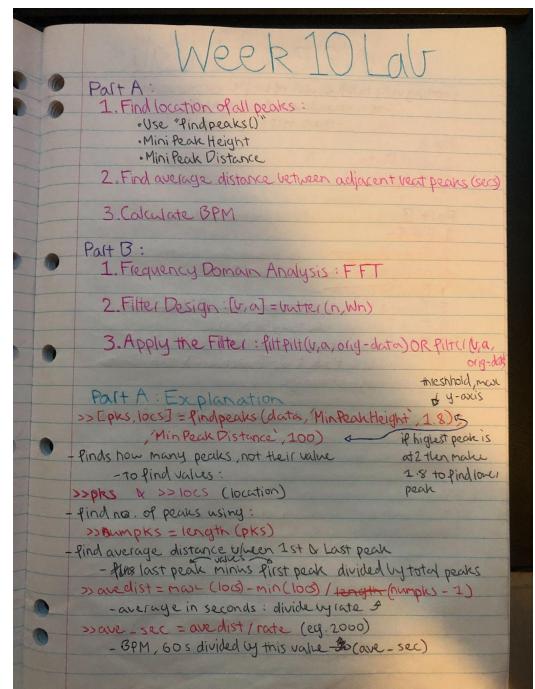
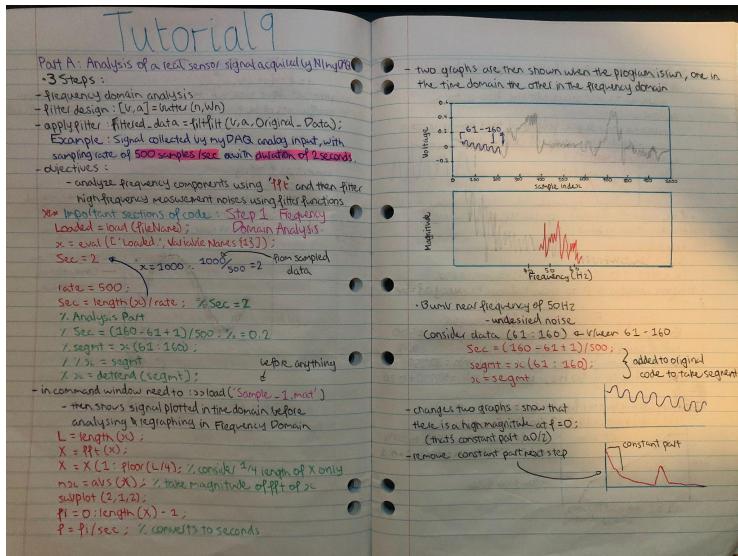
For Week 11 I once again attended the Lab, however this week it was used for asking questions about the project. I only attended this lab for just over an hour, as they wouldn't be helpful until I properly started my code in Week 12. And I had no questions to ask the teacher myself. I made sure to listen to what was being said however, as there were some important things discussed mainly for part A. I summarized what I thought was important in the lecture notes for all the previous weeks, however I found it mainly just stuff for LabView and although it was interesting it wasn't worth the time doing specifically weeks 9-12. I prepared myself for Week 12 as I knew I was going to have to really get a lot done, to make sure of this I completed all of my physics and maths work by the end of Week 11 so I could focus all of my attention on this assignment in week 12.

Week 12 :

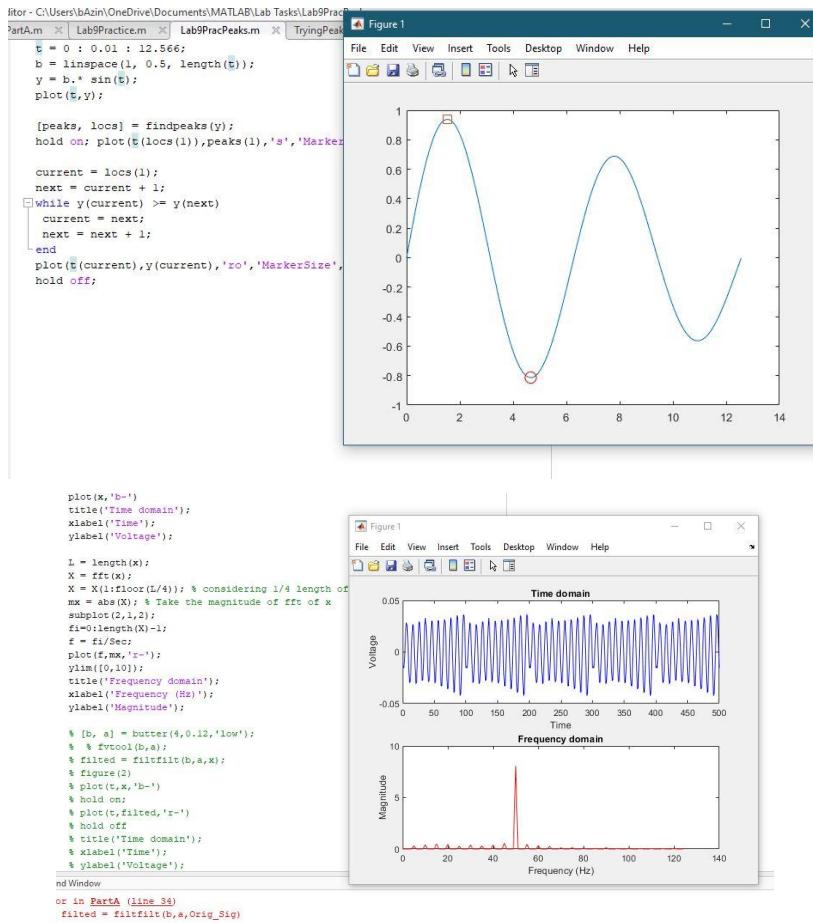
To begin this week I summarized my plans for week 12 by breaking down the tasks I wanted to do, where the first thing was to check over the lecture notes and make sure I didn't miss any vital information, than the biggest task was to go through weeks 7-12s labs for my class as well as summarizing every tutorial and all of the important information that was spoken about. Some of my pages of work are shown below.



This helped massively as the lecture notes did not help with this assignment, which was the opposite for the portfolio task. Firstly I went through and watched all of the previous recordings of my tutorial classes (2.30-5.30, tuesday) and took down any extra notes that I missed. Especially with weeks 11 and 12 as they were Q&A's, and if I ran into trouble myself most likely the information I needed would be in week 11 and 12 labs. A few pages of my notes can be seen below :



Once I watched and skimmed through the tutorials I did the Lab 9 code as this gave a great basis towards both part A and B of the project. By this time it was Thursday.



Week 9 Lab Code

It must be added that I had the vaccine on Tuesday and felt very nauseous only on that night as well as having extreme fatigue through to Thursday night. So I was unable to get as much as I should've done by this date. On Friday I started coding for Part A. I played around with the code from the tutorial and tried to get it to work on the samples given. Thankfully we were given an extension, as I was unconfident I'd be able to get any of Part B done if it was due on the Friday. I played around with the different filtering that was shown, and attempted to do the signal analysis and filter design that day. I was able to expand on the information I was given in the Week 10 lab at the end to find the bpm. So by Early Saturday morning I had played around with filtering which I used for Part B and not Part A in the end and figured out the filtering for the assignment. On Saturday I worked from around 2pm to 4am on the code. I had ideas of what my code needed to be for Part A however was really struggling to get something cohesive. Some of my attempts can be shown below.

The screenshot shows the MATLAB Editor and Command Window.

Editor:

```

1 - fileName = 'Sample_1.mat';
2 - Loaded = load(fileName);
3 - VariableNames = fieldnames(Loaded);
4 - x = eval(['Loaded.',VariableNames(1)]);
5 - Sec = 10;
6 - rate = 360;
7 -
8 - Sec = (160-61+1)/500; % = 0.2
9 - segmt = x(61:160);
10 - x = segmt;
11 - x = detrend(segmt);
12 -
13 - figure(1)
14 - subplot(2,1,1)
15 - plot(x,'b-')
16 - title('Time domain');
17 - xlabel('Time');
18 - ylabel('Voltage');
19 -
20 - L = length(x);
21 - X = fft(x);
22 - X = X(1:floor(L/4)); % considering 1/4 length of X only;
23 - mx = abs(X); % Take the magnitude of fft of x
24 - subplot(2,1,2);
25 - fi=0:length(X)-1;
26 - f = fi/Sec;
27 - plot(f,mx,'r-');
28 - ylim([0,300000]);
29 - title('Frequency domain');

```

Command Window:

```

>> tryingpeaks6
Unrecognized function or variable 'fprintf'.

Error in tryingpeaks6 (line 23)
fprintf(bpm);

>> bpm
bpm =
      55.3333

```

Part B : Failed Attempt At Finding BPM

I went back and started watching other labs/tutorials from other time slots to find information I might've not been informed on. In the end I probably watched around 25-40 different lab sessions as I was really struggling at this point. On Sunday morning I finally got my code for part A to work. This is shown below and it was a very nice feeling. The code turned out much simpler than I originally thought as I had learned how to properly use `findpeaks` through watching more labs as well as the use of `MinPeakProminence` which helped greatly. Originally I calculated the `bpm` incorrectly, however I fixed it.

```

Editor - C:\Users\bAzin\OneDrive\Documents\MATLAB\Assignment\tryngpeaks6.m
PartA.m TryingPeaks.m PeaksAttempt3.m attempt4.m PartB.m tryingstuff5.m tryingpeaks6.m + 

1 - fileName = 'Sample_1.mat'; %choose which sample
2 - Loaded = load(fileName);
3 - VariableNames = fieldnames(Loaded);
4 - x = eval(['Loaded.',VariableNames{1}]); %sample is loaded than the original signal is made into x
5 -
6 - Sec = 10;
7 - rate = 360;
8 -
9 - figure(1) %plots original data
10 - subplot(2,1,1)
11 - plot(x,'b-')
12 - title('Time domain');
13 - xlabel('Time');
14 - ylabel('Voltage');
15 - %can smooth in order to get rid of multiple peaks in similar location
16 -
17 - [peaks, locs] = findpeaks(x,'MinPeakHeight',1000,'MinPeakProminence',150);
18 - hold on; plot(t(locs),peaks,'s','MarkerSize',7)
19 -
20 - numpeaks = length(locs);
21 - avedist = (max(locs) - min(locs))/(numpeaks-1);
22 - bpm = (60/avedist)*360;
23 - fprintf('.f Beats Per Minute.\n', BPM);
24 - %. round the number
25 - %can increase minpeakheight to get more accurate for sample 8
26 - %however this is an outlier and destroys accuracy of the other 7 samples
27 

```

Command Window

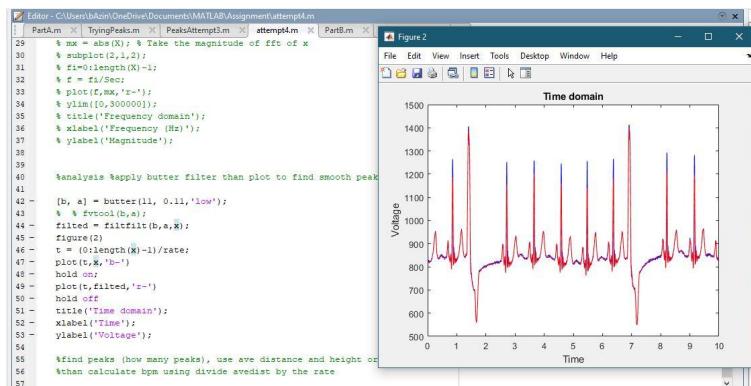
```

>> tryingpeaks6
>> tryingpeaks6
65 Beats Per Minute.

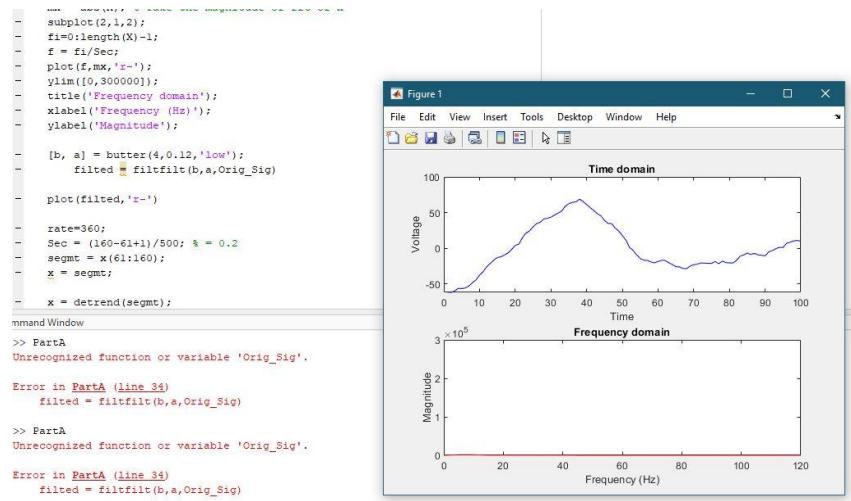
```

Part B Code : Finished

From here I moved onto part B. Although I didn't finish my Heart Rate Analyzer this was due to bad time management, as it was much easier to progress than Part A. First I started by finalizing my filtering using analysis and the design steps shown in Lab 9, this is shown below :

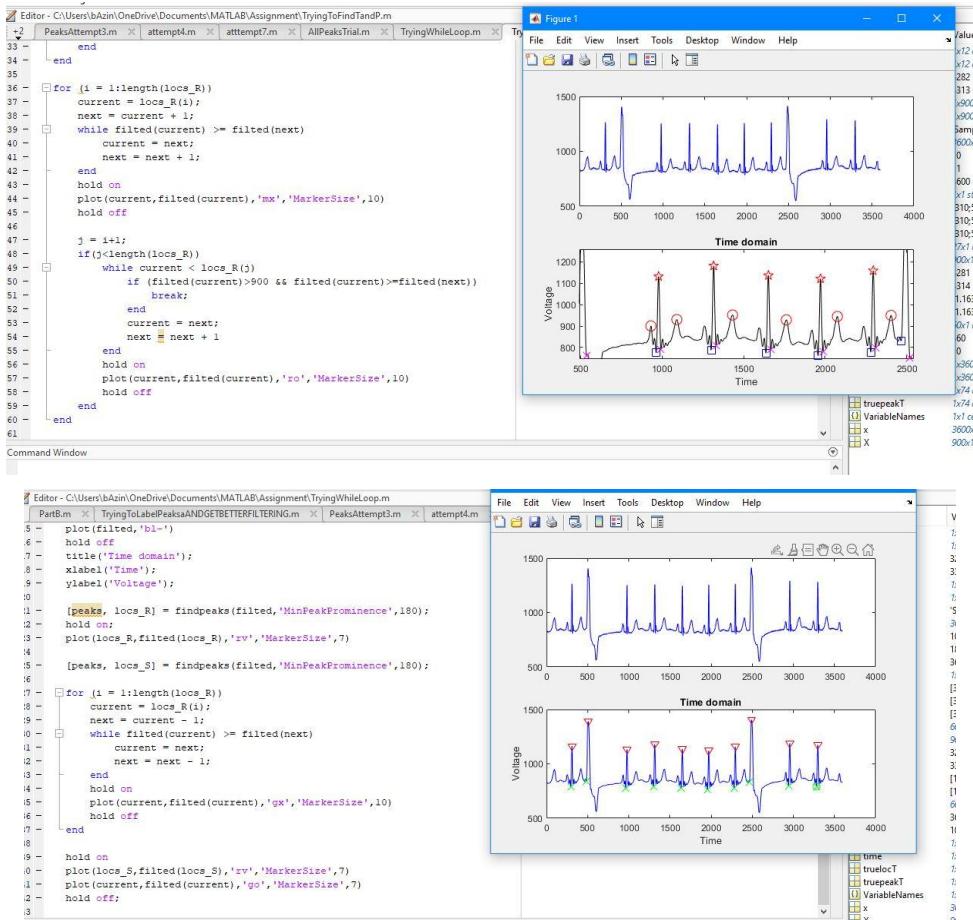


Filtered Graph : Made A lot of Adjustments



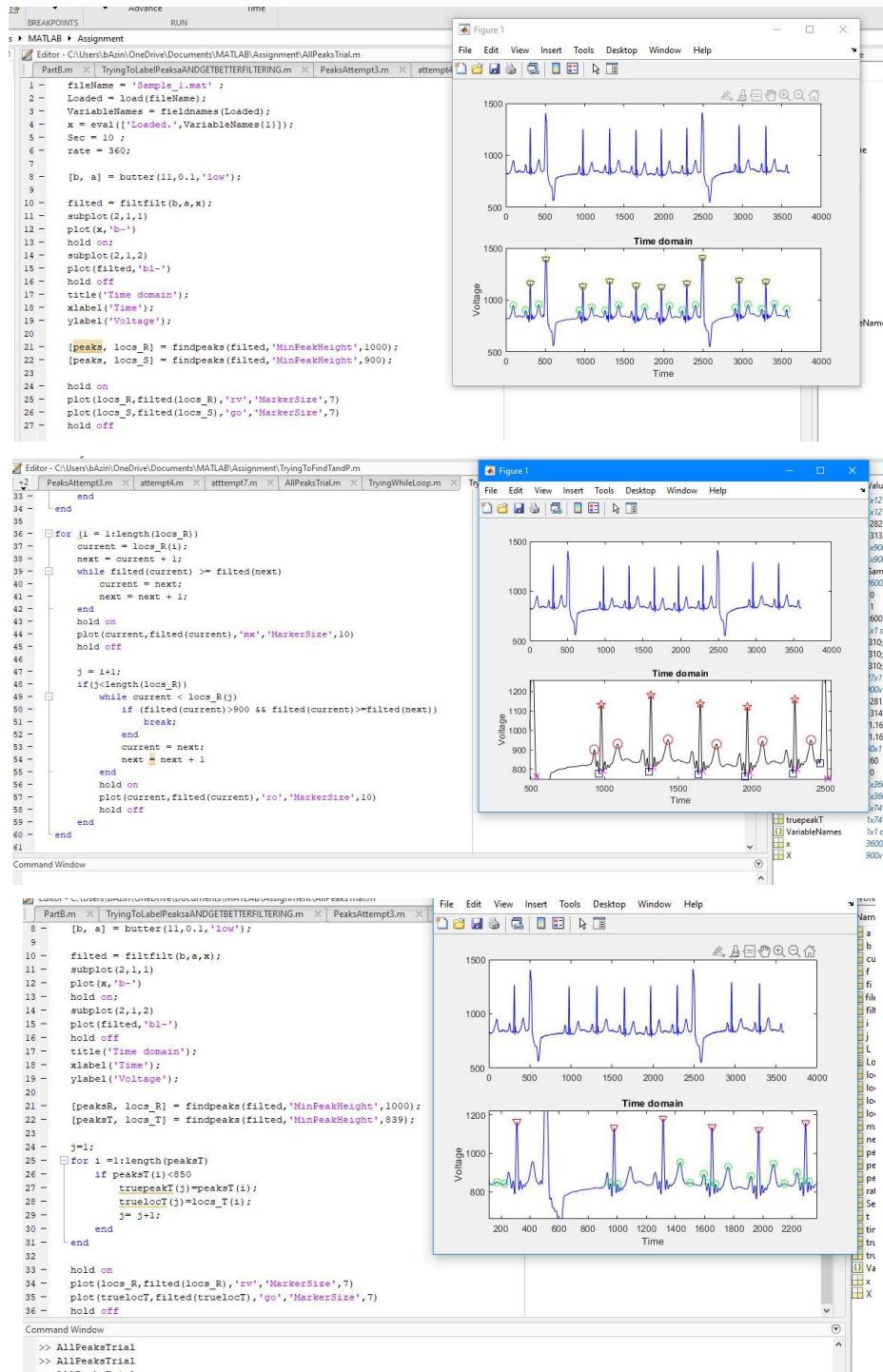
Filtering : Looking for Unwanted Noise

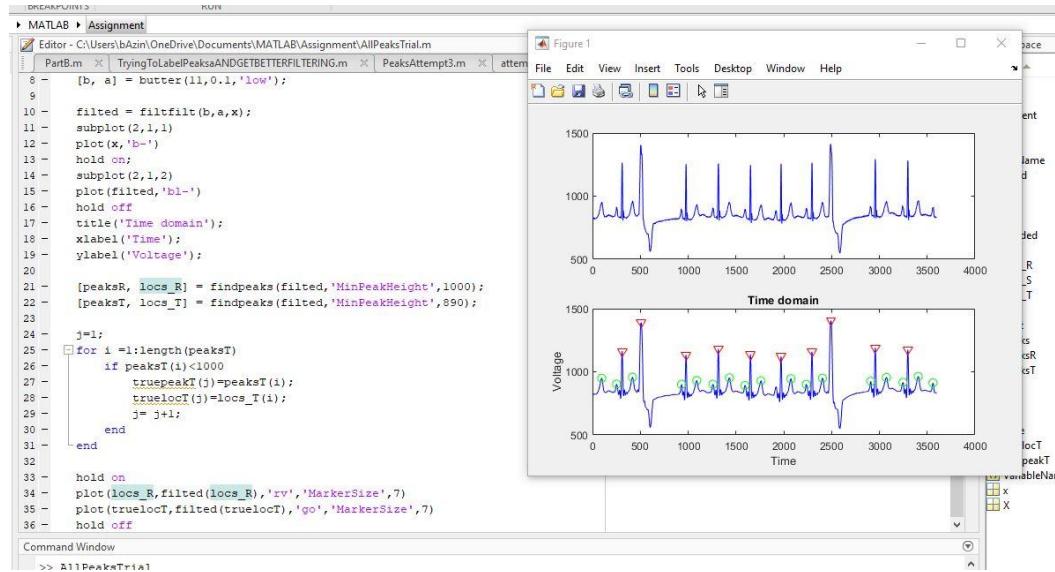
Once I had the correct filtering using the butter filter, I used my code to findpeaks from Part A and plotted the markings. Once I did that I moved on to trying to find the Q and S peaks. These didn't take me much time to do and only required slight changes to the provided code from Lab 9. This can be shown below :



R, Q and S Peaks Found and Marked

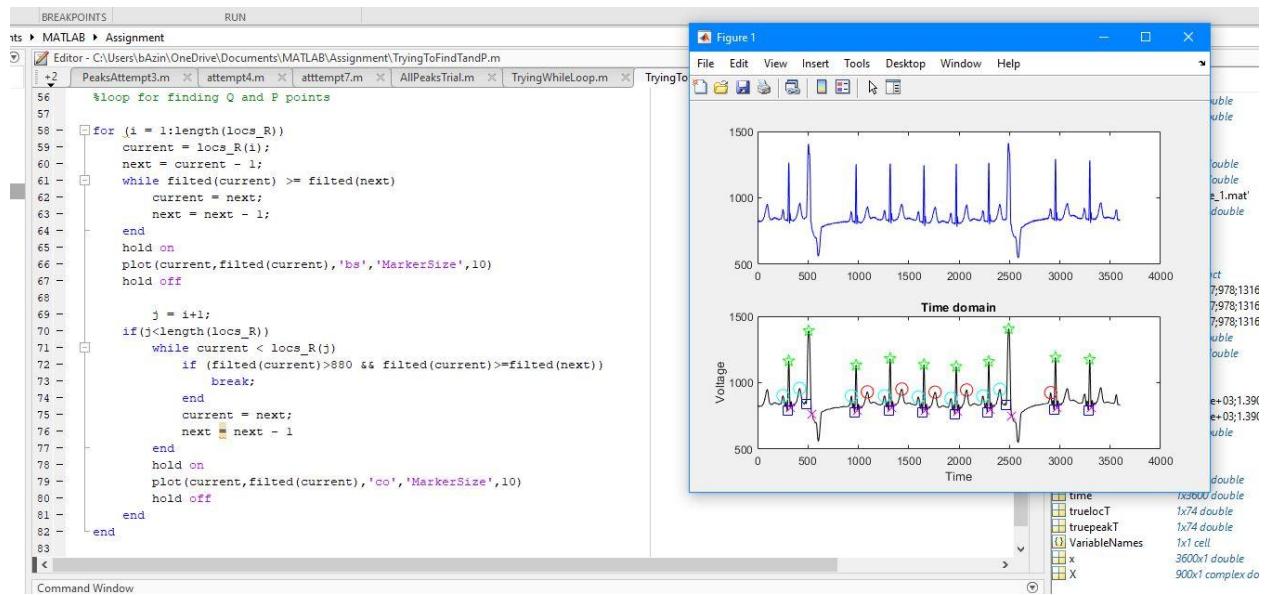
Next I moved on to finding the T and P peaks and this is by far the most difficult step, unfortunately I didn't have much time and thus didn't find their true locations. In a week 12 tutorial I watched they showed the basics of finding the T peak, however in it showed how to do it in multiple ways. I tried a lot of things and this can be seen in the images below :





Different ways of finding T and P peaks

However in the end I ran out of time and had to use the code he had shown with a couple of changes to find both the T and P peaks, then I had to do the intervals however they aren't near as accurate as I would like. Once this was done by Monday I spent the whole of Tuesday writing this report. My Part B code is shown below :



Part B : Graph Finished

Reflection on my learning experience

What are the most important things you learnt from this project?

The most important things I learnt in this project were more advanced coding from what had been done previously. That matlab has a lot more functionality than at first glance. I learned about signals and it showed me how important problem solving and breaking down bigger tasks can be. Another thing I learnt was I need to have better time management, as in the last week I had to learn so much information it was difficult to keep a clear mind and not get overwhelmed as I needed to be clear on how my code was going to work.

Did you meet your project goals?

I was able to meet nearly all of my project goals. I managed to finish part A to the level I wanted and most of part B however I underestimated the difficulty of the code and my own lack of problem solving ability that was essential for part B of this project. Leaving part B unfinished and not as accurate as I would've liked.

What parts of the project do you particularly like? Why?

The parts of the project that I enjoyed was that it felt challenging but fair, we were given set examples and a basis of code to go off of. So when you passed a step in the code it was a really great feeling of satisfaction. I also liked the structure of the unit and the amount of freedom. It meant there were many ways to do the code making completing an objective more satisfying, however I also wasn't capable of actually attempting a lot of the other ways due to my lack of coding ability.

What do you find particularly challenging?

It definitely tested my problem solving ability. My strength is my memory however this was not very helpful for this project so I struggled a lot. Because I did so much in the last week there was a lot of information to properly understand. Although I enjoy the idea of coding I was not very good at it. So it was extremely difficult to put my ideas for the code into actual matlab. My basic understanding of MATLAB language wasn't very good so trying to expand upon this made it difficult. I also found time management very difficult. I have a weekly planner however I had this subject as my third as I was more focused on physics and maths up until week 12. So I had to do a lot in the last week. Thankfully I prepared for this.

What are the things that helped you most in this project?

For this project the Labs and Mathworks were incredibly helpful. As stated I watched nearly every Lab / Tutorial I could and summarized the lab sheets as they had so much helpful information in them. When I was trying to learn what different functions and variables did Mathworks made it much easier.

One thing I would like to improve upon is?

I would like to improve my problem solving skills. I think my basic knowledge of MATLAB wasn't very good, and it made it really difficult to think of different ways to create my design solution. My abilities were very limited so it felt very tedious at times to get through. I did get better but nowhere near as good as I would like. There are many things I would like to improve upon with this unit however this is definitely the main one.