

Charlie Maclean

# Synthesis of Heart-Rate Detection Methods

Computer Science Tripos – Part II

King's College

January 19, 2020



# Proforma

Name: **Charlie Maclean**  
College: **King's College**  
Project Title: **Synthesis of Heart-Rate Detection Methods**  
Examination: **Computer Science Tripos – Part II, July 2020**  
Word Count: **TODO<sup>1</sup>**  
Project Originator: **Dr Robert Harle**  
Supervisor: **Dr Robert Harle**

## Original Aims of the Project

To research and implement the detection of heart rate from smartwatch sensors. To investigate the effectiveness of a selection of filters and peak finding algorithms. To use accelerometer data to find motion artifacts within the data, and compare methods of removing these artifacts.

## Work Completed

All that has been completed appears in this dissertation.

## Special Difficulties

Learning how to incorporate encapsulated postscript into a  $\text{\LaTeX}$  document on both Ubuntu Linux and OS X.

---

<sup>1</sup>This word count was computed by `detex diss.tex | tr -cd '0-9A-Za-z \n' | wc -w`

## Declaration

I, Charlie Maclean of King's College, being a candidate for Part II of the Computer Science Tripos, hereby declare that this dissertation and the work described in it are my own work, unaided except as may be specified below, and that the dissertation does not contain material that has already been used to any substantial extent for a comparable purpose.

Signed [signature]

Date [date]

# Contents

<b>1</b>	<b>Introduction</b>	<b>9</b>
<b>2</b>	<b>Preparation</b>	<b>11</b>
2.1	Gathering data . . . . .	11
2.2	Filtering . . . . .	11
2.2.1	Butterworth Filter . . . . .	12
2.2.2	Chebyshev Filter . . . . .	13
2.3	Heart Rate Calculation . . . . .	14
2.3.1	Local Maxima . . . . .	14
2.3.2	Phase Vocoder . . . . .	14
2.4	Motion Artifact Reduction . . . . .	14
2.4.1	Adaptive Noise Cancellation . . . . .	15
<b>3</b>	<b>Implementation</b>	<b>17</b>
3.1	Gathering Data . . . . .	17
3.1.1	Wear OS Development . . . . .	17
3.1.2	Synchronising Signals . . . . .	17
3.2	Verbatim text . . . . .	18
3.3	Tables . . . . .	19
3.4	Simple diagrams . . . . .	19
3.5	Adding more complicated graphics . . . . .	19
<b>4</b>	<b>Evaluation</b>	<b>23</b>
4.1	Printing and binding . . . . .	23
4.2	Further information . . . . .	23
<b>5</b>	<b>Conclusion</b>	<b>25</b>
	<b>Bibliography</b>	<b>25</b>
<b>A</b>	<b>Latex source</b>	<b>27</b>
A.1	diss.tex . . . . .	27
<b>B</b>	<b>Makefile</b>	<b>37</b>
B.1	makefile . . . . .	37
B.2	refs.bib . . . . .	38



# List of Figures

2.1	Diagram showing filter characteristics . . . . .	12
2.2	Comparison of Butterworth filter orders . . . . .	13
2.3	Comparison of Chebyshev filter rs values . . . . .	13
2.4	ANC overall flow . . . . .	15
3.1	A picture composed of boxes and vectors. . . . .	19
3.2	A diagram composed of circles, lines and boxes. . . . .	20
3.3	Example figure using encapsulated postscript . . . . .	20
3.4	Example figure where a picture can be pasted in. . . . .	21
3.5	Example diagram drawn using <b>xfig</b> . . . . .	21

## Acknowledgements

This document owes much to an earlier version written by Simon Moore [?]. His help, encouragement and advice was greatly appreciated.



# Chapter 1

## Introduction

Elite runners have historically used heart rate to provide an accurate measure of fitness, and allow them to train more effectively. [Expand on uses of heart rate]. Previously, Electrocardiography (ECG) chest straps have been used to measure heart rate, by detecting the electrical signals controlling the expansion and contraction of the heart. They are accurate devices however often prohibitively expensive, and hence inaccessible to casual runners.

In recent years, a new technology has emerged - Photoplethysmogram (PPG) - light is directed at the skin, and sensors measure how much blood vessels scatter it. PPG sensors are cheaper than ECG sensors, and hence are available in a variety of products, particularly smartwatches. This innovation has brought a new wave of advanced training and monitoring onto the wrists of any runner.

Switching from ECG to PPG is not without flaws though - ECG sensors return a clean signal, as opposed to PPG signals which are contaminated with noise. [More details about noise].

In particular when running, the motion can cause blood velocity to change, and the sensor can slip across the skin [?], [?]. These result in a distortion to the PPG signal, known as a motion artifact (MA). Fortunately, smartwatches contain other sensors, such as accelerometers and gyroscopes which can be used to predict the presence of MAs, and hence compensate for them.

My project is to research and develop a heart rate detection algorithm for smartwatches worn during running.



# Chapter 2

## Preparation

In this chapter we introduce the relevant concepts for this project. I begin with a description of the challenges that come with collecting data. Then I move on to describing the techniques I use to actually calculate heart rate.

### 2.1 Gathering data

### 2.2 Filtering

PPGs produce a large amount of noise from various sources:

- Light pollution. Both DC and AC lights in the environment can infiltrate the sensor, affecting the PPG signal [?].
- Temperature. As temperature increases, so does the volume of blood, which will be picked up by the PPG sensor [?].
- Breathing. The change in pressure associated with respiration causes variations in the flow of blood, and hence can be seen by the PPG sensor [?].

In order to remove much of this noise, we can use filters to remove frequencies we know are irrelevant. We know the heartbeat can vary from 30 to 220 beats per minute, and hence we would like to disregard any noise outside of this range. In this section, I will introduce the concept of filters, and describe two filters useful for PPG signals.

Figure 2.1 displays an example filter magnitude response diagram. This plots the amount of gain applied to each frequency. A gain of zero means the signal is removed, a gain of one means the signal is unchanged. The characteristics displayed are as follows.

The passband is the range of frequencies we would like to remain. In an ideal filter, there is no loss within the passband.

Passband ripple describes the variation in amplitude *within* the passband. An ideal filter will have no passband filter, such that all frequencies within the passband are permitted equally.

The transition width is the frequency range between the start and stop band. Ideally, this is zero, such that frequencies outside of the passband are instantly reduced.

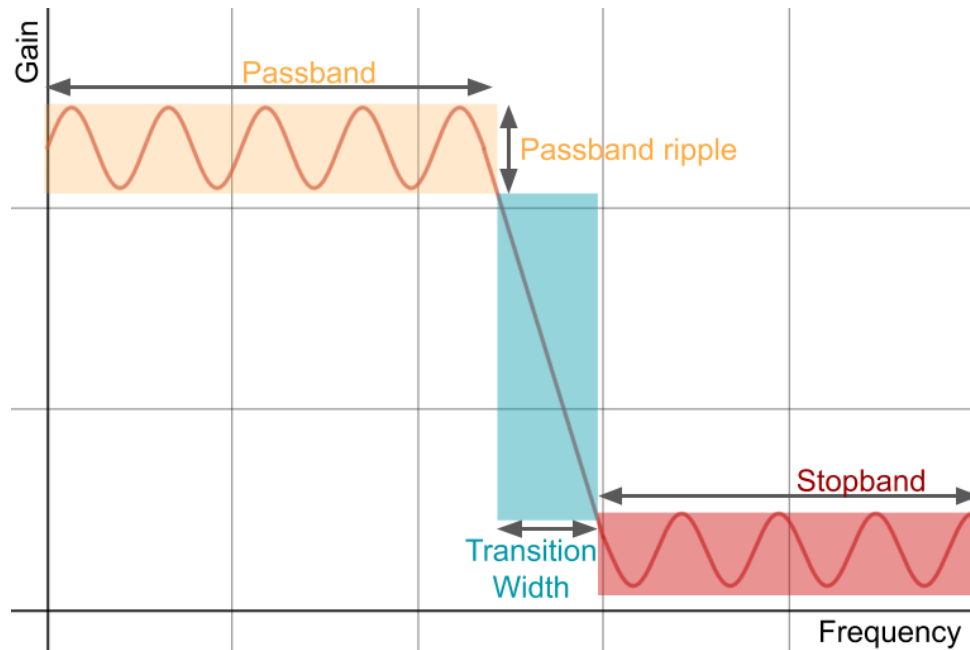


Figure 2.1: Diagram showing filter characteristics

The stopband is the range of frequencies we wish to remove. An ideal filter completely removes all frequencies within the stopband.

The ideal filter, as described above, is impossible - and hence we have a variety of filters which compromise between the desirable characteristics. Following this, I detail two of these compromises - the Butterworth filter, and the Chebyshev filter.

### 2.2.1 Butterworth Filter

The Butterworth filter aims to minimize passband ripple, at the expense of a larger transition width. To define a Butterworth filter, multiple parameters are used, which I will describe further here.

Order ( $N$ ) controls the transition width - the higher the order, the lower the transition width. This is desirable - we want to keep the passband as large as possible, and transition quickly to the stopband. However, it comes at the cost of computation complexity, as higher order filters take longer to apply. See figure 2.2 for a plot displaying the effect of different orders on a filter's response.

Critical frequency gives the frequency at which we want gain to drop below  $1/\sqrt{2}$  from the passband. Hence we use this parameter to define the point from which we wish to cut out noise. Figure 2.2 uses critical frequencies 0.4 and 4 hz, which correspond to 24 and 240 bpm, a reasonable heart rate range.

Type describes which frequencies we cut out. There are four different types of Butterworth filters, as follows:

- Lowpass - allow frequencies below the critical frequency.
- Highpass - allow frequencies above the critical frequency.

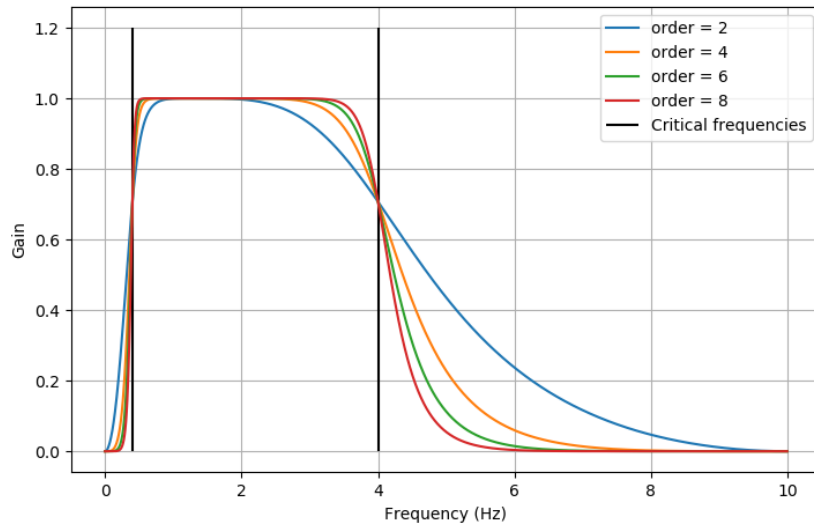


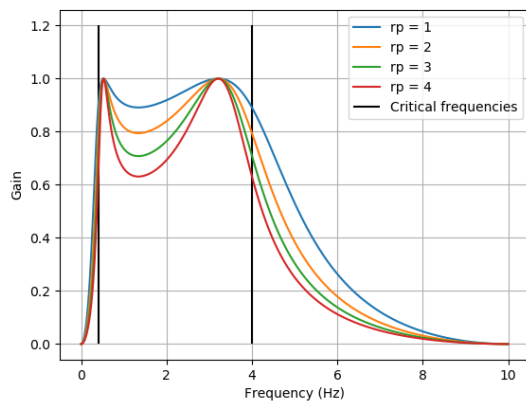
Figure 2.2: Comparison of Butterworth filter orders

- Bandpass - combination of lowpass and highpass - given two frequencies, we want the passband to be between those frequencies.
- Bandstop - given two frequencies, put the stopband between them, leaving the passband outside those frequencies.

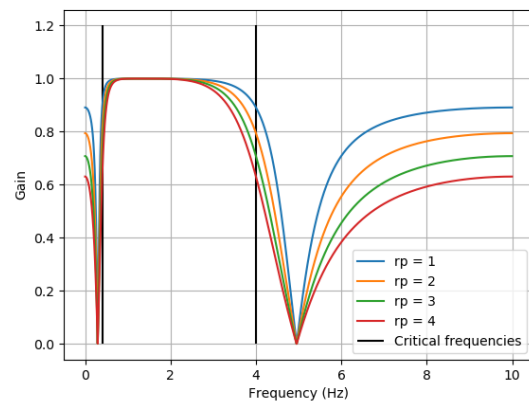
We will be using a bandpass filter, as there is low and high frequency noise we wish to remove.

### 2.2.2 Chebyshev Filter

The Chebyshev filter aims to reduce the transition width as much as possible, but to do this it introduces increased ripple.



(a) Chebyshev Type 1



(b) Chebyshev Type 2

Figure 2.3: Comparison of Chebyshev filter rs values

The filter is defined with similar parameters to a Butterworth filter - order, critical frequencies, and type all have very similar meanings. However, there are type 1 and 2 Chebyshev filters, with different aims. Additionally, there is a new parameter  $rs$ . I will explain these differences here.

Type 1 Chebyshev filters introduce passband ripple, whereas type 2 Chebyshev filters introduce stopband ripple. Figure 2.3 shows how these differences manifest in the response of a second order Chebyshev filter.

The parameter  $rs$  is defined as the maximum ripple permitted below unity gain. Higher  $rs$  results in more ripple for a type 1 filter, but less ripple for a type 2 filter, as is demonstrated in figure 2.3. Introducing more ripple is undesirable as it leads to more signal distortion, however can be advantageous as it reduces the transition width.

## 2.3 Heart Rate Calculation

Now we have removed noise from the signal, we should have a signal representing the volume of blood in veins within the arm. As the heart beats, it sends a wave of blood through the body, forcing more blood through these veins. Hence, peaks in the PPG signal correspond to heart beats.

There are multiple approaches to finding heart-rate from our filtered signals, and I will detail a few here.

### 2.3.1 Local Maxima

If we can find the peaks in the signal, we can count them up and divide by the time period we count those peaks within, giving us a heart rate. One way we can define peaks is as local maxima.

The local maxima in a signal are defined as samples which are larger than both neighbouring samples. Finding local maxima is an easy problem to solve - intuitively, we can create an algorithm which iterates through the samples, comparing each sample to its neighbours, which has complexity  $O(n)$ .

### 2.3.2 Phase Vocoder

An alternative approach to estimating the heart rate uses the signal's Fourier Transform. The Fourier Transformation takes a signal, and returns its constituent frequencies. We can look at this Fourier Transform, knowing one of the peaks is the frequency associated with the heart beating.

Fourier Transforms alone are not fantastic for finding heart rate, as they are too low resolution. However we can use a Phase Vocoder to increase the resolution.

## 2.4 Motion Artifact Reduction

In this section I explore methods that can be used to reduce the effect of Motion Artifacts (MAs) on the PPG Signal.

### 2.4.1 Adaptive Noise Cancellation

Adaptive Noise Cancellation (ANC), as described by Widrow et al. [?], is a technique which allows us to remove noise from a signal, given we have another signal correlated to the noise in some way. In our situation, we know MAs are correlated to motion, so we can use the accelerometer to remove MAs. This technique is extremely useful, as it can still produce good results when the MAs are at the same frequency as the heart beat, even if the MAs have a larger amplitude.

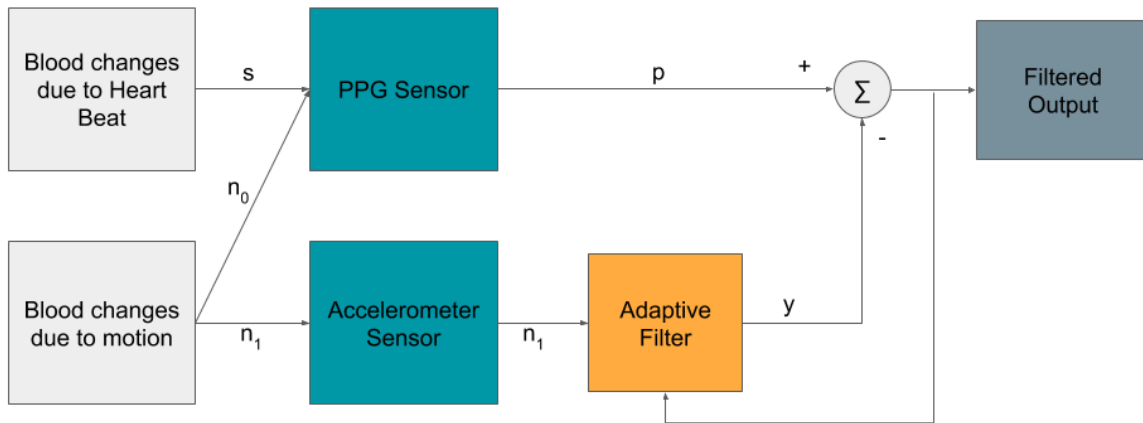


Figure 2.4: ANC overall flow

An overview of the algorithm follows. We have signal from heartbeat  $s$  that we want to figure out, but this is contaminated by noise from MAs  $n_0$ . We assume the PPG sensor reading  $p$  is such that  $p = s + n_0$ . Additionally, we have accelerometer sensor readings  $n_1$ . We filter  $n_1$ , producing signal  $y$  and subtract this from  $p$ , to produce  $z = p - y$ . Our aim is to adjust the filter, such that  $y$  is as close to  $n_0$  as possible.

The adaptive filter aims to choose a filter such that the power output  $E[z^2]$  is minimized. Given  $s$  is uncorrelated with  $n_0$  and  $n_1$ , and  $n_0$  is correlated with  $n_1$ , it can be proven that minimizing the output power is equivalent to





# Chapter 3

## Implementation

### 3.1 Gathering Data

The project required data from two sources: a PPG sensor mounted on a watch, and a chest-mounted ECG. In this section I detail the process of developing an application to record PPG data on a Wear OS watch.

#### 3.1.1 Wear OS Development

#### 3.1.2 Synchronising Signals

Once I have collected the PPG signal from the watch, and have recorded the ECG from the chest, I need a method to synchronise the signals, such that they start at the same time and we can compare them accurately.

I set out to produce a solution with the following properties:

- Able to synchronize signals with  $\pm 0.3s$  accuracy. With a heartbeat of 200 bpm this represents being within a heartbeat. This is an acceptable level of delay, as heart rate is averaged over several beats anyway.
- Can synchronize signals given the two recordings are started within two minutes of each other. This constraint is helpful as it prevents wasted time searching through the signal.
- Does not use the clocks built into the device. We must assume the clock within the ECG is unreliable. Additionally, the two devices may be synchronized to different time, and hence could be out by any amount of time.

Given I know both devices have an accelerometer, I realised I could ask the wearer to move in some motion which is picked up by both devices. Then, I could compare the two accelerometer signals in order to discover the movement. Then comparing the starting times of the two signals would enable calculation of the time difference between them.

The type of motion I chose was important, as it had to be easy to explain to the wearer, but also provide sufficient motion for it to be identifiable against normal motion.

I decided jumping was appropriate - I would ask the wearer to hold the watch close to their chest and jump three to five times.

The cross correlation of two signals  $f$  and  $g$  is a measure of similarity as a function of the displacement between them. It is simple to calculate, as a sum of the products between the samples of  $f$  and of  $g$  displaced by  $n$ .

To synchronise the accelerometer, I developed the following three step algorithm:

1. Normalize signals, by moving to zero mean,
2. Crop PPG signals to two minutes, crop ECG signals to four minutes, compute cross correlation with  $f$  = PPG acceleration,  $g$  = ECG acceleration.
3. Compute the other way - crop ECG to two minutes, PPG to four minutes, compute cross correlation with  $f$  = ECG acceleration,  $g$  = PPG acceleration. This is
4. Find the maximum cross correlation across

## 3.2 Verbatim text

Verbatim text can be included using `\begin{verbatim}` and `\end{verbatim}`. I normally use a slightly smaller font and often squeeze the lines a little closer together, as in:

```
GET "libhdr"

GLOBAL { count:200; all  }

LET try(ld, row, rd) BE TEST row=all
    THEN count := count + 1
    ELSE { LET poss = all & ~(ld | row | rd)
        UNTIL poss=0 DO
            { LET p = poss & -poss
                poss := poss - p
                try(ld+p << 1, row+p, rd+p >> 1)
            }
        }
    }

LET start() = VALOF
{ all := 1
  FOR i = 1 TO 12 DO
  { count := 0
    try(0, 0, 0)
    writef("Number of solutions to %i2-queens is %i5*n", i, count)
    all := 2*all + 1
  }
  RESULTIS 0
}
```

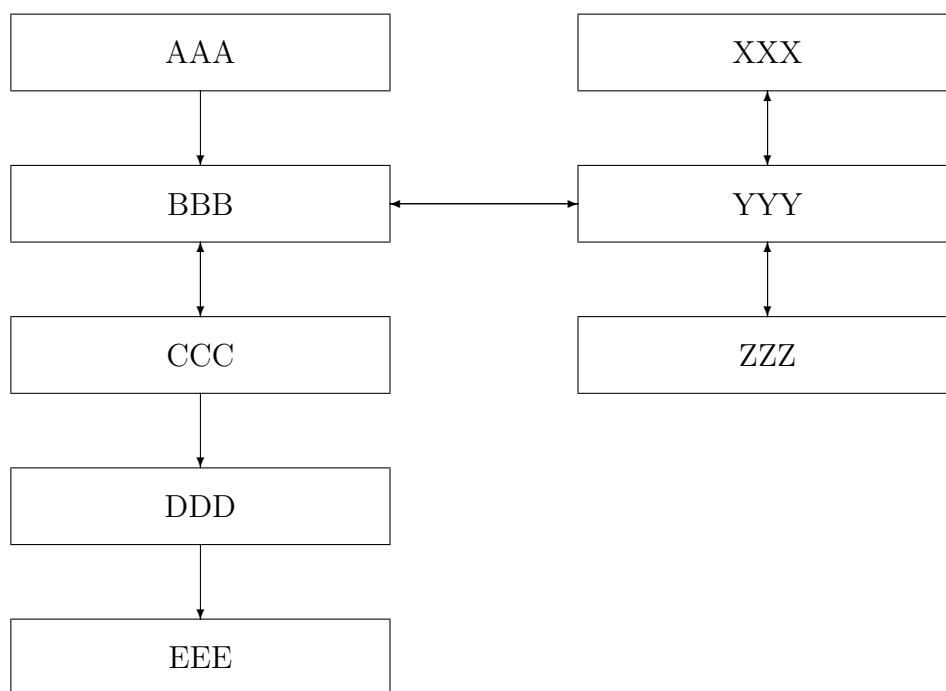


Figure 3.1: A picture composed of boxes and vectors.

### 3.3 Tables

Here is a simple example<sup>1</sup> of a table.

Left Justified	Centred	Right Justified
First	A	XXX
Second	AA	XX
Last	AAA	X

There is another example table in the proforma.

### 3.4 Simple diagrams

Simple diagrams can be written directly in  $\text{\LaTeX}$ . For example, see figure 3.1 on page 19 and see figure 3.2 on page 20.

### 3.5 Adding more complicated graphics

The use of  $\text{\LaTeX}$  format can be tedious and it is often better to use encapsulated postscript (EPS) or PDF to represent complicated graphics. Figure 3.3 and 3.5 on page 21 are

---

<sup>1</sup>A footnote

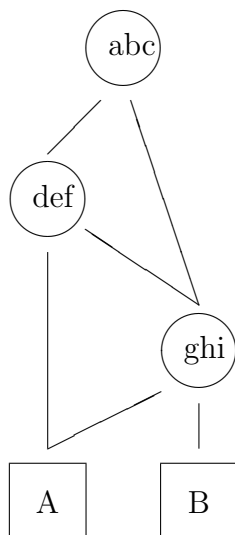


Figure 3.2: A diagram composed of circles, lines and boxes.

examples. The second figure was drawn using `xfig` and exported in `.eps` format. This is my recommended way of drawing all diagrams.

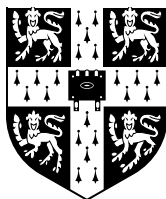


Figure 3.3: Example figure using encapsulated postscript

Figure 3.4: Example figure where a picture can be pasted in

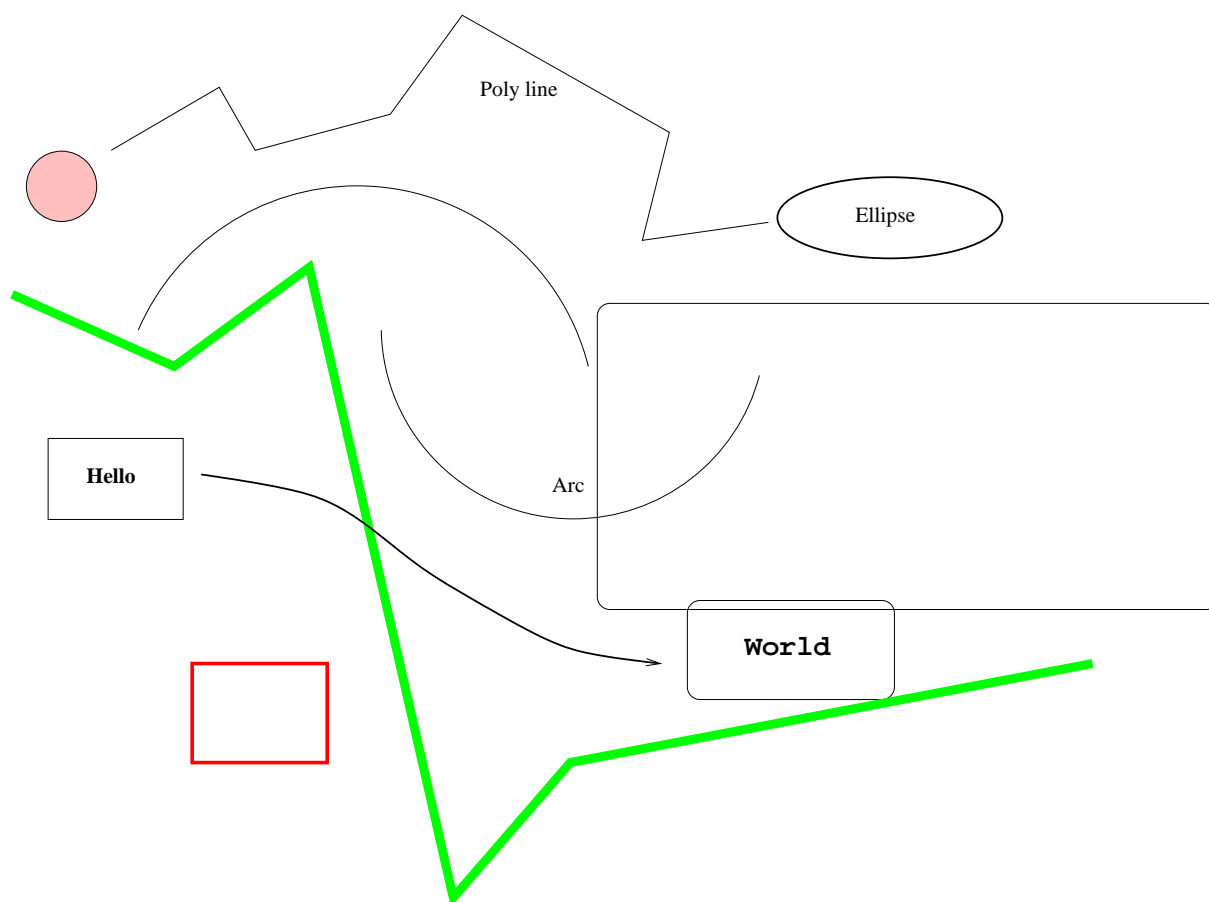


Figure 3.5: Example diagram drawn using `xfig`



# Chapter 4

## Evaluation

### 4.1 Printing and binding

Use a “duplex” laser printer that can print on both sides to print two copies of your dissertation. Then bind them, for example using the comb binder in the Computer Laboratory Library.

### 4.2 Further information

See the Unix Tools notes at

<http://www.cl.cam.ac.uk/teaching/current-1/UnixTools/materials.html>





# Chapter 5

## Conclusion

I hope that this rough guide to writing a dissertation in L<sup>A</sup>T<sub>E</sub>X has been helpful and saved you time.



# Bibliography

- [1] John Allen, John R Frame, and Alan Murray. Microvascular blood flow and skin temperature changes in the fingers following a deep inspiratory gasp. *Physiological measurement*, 23(2):365, 2002.
- [2] J. Kim, T. Lee, J. Kim, and H. Ko. Ambient light cancellation in photoplethysmogram application using alternating sampling and charge redistribution technique. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 6441–6444, Aug 2015.
- [3] Hangsik Shin. Ambient temperature effect on pulse rate variability as an alternative to heart rate variability in young adult. *Journal of clinical monitoring and computing*, 30(6):939–948, 2016.
- [4] B. Widrow, J. R. Glover, J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, J. Eugene Dong, and R. C. Goodlin. Adaptive noise cancelling: Principles and applications. *Proceedings of the IEEE*, 63(12):1692–1716, Dec 1975.
- [5] R. W. C. G. R. Wijshoff, M. Mischi, and R. M. Aarts. Reduction of periodic motion artifacts in photoplethysmography. *IEEE Transactions on Biomedical Engineering*, 64(1):196–207, Jan 2017.
- [6] L. B. Wood and H. H. Asada. Noise cancellation model validation for reduced motion artifact wearable ppg sensors using mems accelerometers. In *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 3525–3528, Aug 2006.



# Appendix A

## Latex source

### A.1 diss.tex

```
% Template for a Computer Science Tripos Part II project dissertation
\documentclass[12pt,a4paper,twoside,openright]{report}
\usepackage[pdftborder={0 0 0}]{hyperref} % turns references into hyperlinks
\usepackage[margin=25mm]{geometry} % adjusts page layout
\usepackage{graphicx} % allows inclusion of PDF, PNG and JPG images
\usepackage{verbatim}
\usepackage{pdfpages}
\usepackage{caption}
\usepackage{subcaption}

\raggedbottom % try to avoid widows and orphans
\sloppy
\clubpenalty1000%
\widowpenalty1000%

\renewcommand{\baselinestretch}{1.1} % adjust line spacing to make
% more readable

\begin{document}

\bibliographystyle{plain}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Title

\pagestyle{empty}

\rightline{\LARGE \textbf{Charlie Maclean}}

\vspace*{60mm}
\begin{center}
\Huge
\textbf{Synthesis of Heart-Rate Detection Methods} \\\[5mm]
Computer Science Tripos -- Part II \\\[5mm]
King's College \\\[5mm]
\today % today's date
\end{center}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Proforma, table of contents and list of figures

\pagestyle{plain}
```

```

\chapter*{Proforma}

{\large
\begin{tabular}{ll}
Name: & & \bf Charlie Maclean & \\
College: & & \bf King's College & \\
Project Title: & & \bf Synthesis of Heart-Rate Detection Methods & \\
Examination: & & \bf Computer Science Tripos -- Part II, July 2020 & \\
Word Count: & & \bf TODO\footnotemark[1] & \\
Project Originator: & & Dr Robert Harle & \\
Supervisor: & & Dr Robert Harle & \\
\end{tabular}
}
\footnotetext[1]{This word count was computed
by \texttt{detex diss.tex | tr -cd '0-9A-Za-z $\tt\backslash$' | wc -w}
}
\stepcounter{footnote}

\section*{Original Aims of the Project}

To research and implement the detection of heart rate from smartwatch
sensors. To investigate the effectiveness of a selection of filters and
peak finding algorithms. To use accelerometer data to find motion artifacts
within the data, and compare methods of removing these artifacts.

\section*{Work Completed}

All that has been completed appears in this dissertation.

\section*{Special Difficulties}

Learning how to incorporate encapsulated postscript into a \LaTeX\
document on both Ubuntu Linux and OS X.

\newpage
\section*{Declaration}

I, Charlie Maclean of King's College, being a candidate for Part II of the
Computer Science Tripos, hereby declare that this dissertation and the work
described in it are my own work, unaided except as may be specified below,
and that the dissertation does not contain material that has already been
used to any substantial extent for a comparable purpose.

\bigskip
\leftline{Signed [signature]}

\medskip
\leftline{Date [date]}

\tableofcontents

\listoffigures

\newpage
\section*{Acknowledgements}

This document owes much to an earlier version written by Simon Moore
\cite{Moore95}. His help, encouragement and advice was greatly
appreciated.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% now for the chapters

\pagestyle{headings}

```

`\chapter{Introduction}`

Elite runners have historically used heart rate to provide an accurate measure of fitness, and allow them to train more effectively. [Expand on uses of heart rate]. Previously, Electrocardiography (ECG) chest straps have been used to measure heart rate, by detecting the electrical signals controlling the expansion and contraction of the heart. They are accurate devices however often prohibitively expensive, and hence inaccessible to casual runners.

In recent years, a new technology has emerged - Photoplethysmogram (PPG) - light is directed at the skin, and sensors measure how much blood vessels scatter it. PPG sensors are cheaper than ECG sensors, and hence are available in a variety of products, particularly smartwatches. This innovation has brought a new wave of advanced training and monitoring onto the wrists of any runner.

Switching from ECG to PPG is not without flaws though - ECG sensors return a clean signal, as opposed to PPG signals which are contaminated with noise. [More details about noise].

In particular when running, the motion can cause blood velocity to change, and the sensor can slip across the skin \cite{Wijshoff17}, \cite{Wood06}. These result in a distortion to the PPG signal, known as a motion artifact (MA). Fortunately, smartwatches contain other sensors, such as accelerometers and gyroscopes which can be used to predict the presence of MAs, and hence compensate for them.

My project is to research and develop a heart rate detection algorithm for smartwatches worn during running.

`\chapter{Preparation}`

In this chapter we introduce the relevant concepts for this project. I begin with a description of the challenges that come with collecting data. Then I move on to describing the techniques I use to actually calculate heart rate.

`\section{Gathering data}`

`\section{Filtering}`

PPGs produce a large amount of noise from various sources:

`\begin{itemize}`

`\item` Light pollution. Both DC and AC lights in the environment can infiltrate the sensor, affecting the PPG signal \cite{kim15}.

`\item` Temperature. As temperature increases, so does the volume of blood, which will be picked up by the PPG sensor \cite{shin16}.

`\item` Breathing. The change in pressure associated with respiration causes variations in the flow of blood, and hence can be seen by the PPG sensor \cite{allen02}.

`\end{itemize}`

In order to remove much of this noise, we can use filters to remove frequencies we know are irrelevant. We know the heartbeat can vary from 30 to 220 beats per minute, and hence we would like to disregard any noise outside of this range. In this section, I will introduce the concept of filters, and describe two filters useful for PPG signals.

`\begin{figure}[h!]`

`\centerline{\includegraphics[width=0.8\textwidth]{figs/filter.png}}`

`\caption{Diagram showing filter characteristics}`

`\label{fig:filterdiag}`

`\end{figure}`

Figure \ref{fig:filterdiag} displays an example filter magnitude response

diagram. This plots the amount of gain applied to each frequency. A gain of zero means the signal is removed, a gain of one means the signal is unchanged. The characteristics displayed are as follows.

The passband is the range of frequencies we would like to remain. In an ideal filter, there is no loss within the passband.

Passband ripple describes the variation in amplitude `\emph{within}` the passband. An ideal filter will have no passband filter, such that all frequencies within the passband are permitted equally.

The transition width is the frequency range between the start and stop band. Ideally, this is zero, such that frequencies outside of the passband are instantly reduced.

The stopband is the range of frequencies we wish to remove. An ideal filter completely removes all frequencies within the stopband.

The ideal filter, as described above, is impossible - and hence we have a variety of filters which compromise between the desirable characteristics. Following this, I detail two of these compromises - the Butterworth filter, and the Chebyshev filter.

`\subsection{Butterworth Filter}`

The Butterworth filter aims to minimize passband ripple, at the expense of a larger transition width. To define a Butterworth filter, multiple parameters are used, which I will describe further here.

```
\begin{figure}[h]
\centerline{\includegraphics[width=0.8\textwidth]{figs/butter-order-comparison.png}}
\caption{Comparison of Butterworth filter orders}
\label{fig:butterworth-order}
\end{figure}
```

Order (`\(N\)`) controls the transition width - the higher the order, the lower the transition width. This is desirable - we want to keep the passband as large as possible, and transition quickly to the stopband. However, it comes at the cost of computation complexity, as higher order filters take longer to apply. See figure `\ref{fig:butterworth-order}` for a plot displaying the effect of different orders on a filter's response.

Critical frequency gives the frequency at which we want gain to drop below  $(1/\sqrt{2})$  from the passband. Hence we use this parameter to define the point from which we wish to cut out noise. Figure `\ref{fig:butterworth-order}` uses critical frequencies 0.4 and 4 hz, which correspond to 24 and 240 bpm, a reasonable heart rate range.

Type describes which frequencies we cut out. There are four different types of Butterworth filters, as follows:

```
\begin{itemize}
\item Lowpass - allow frequencies below the critical frequency.

\item Highpass - allow frequencies above the critical frequency.

\item Bandpass - combination of lowpass and highpass - given two
frequencies, we want the passband to be between those
frequencies.

\item Bandstop - given two frequencies, put the stopband between them,
leaving the passband outside those frequencies.
\end{itemize}
```

We will be using a bandpass filter, as there is low and high frequency noise we wish to remove.



### \subsection{Chebyshev Filter}

The Chebyshev filter aims to reduce the transition width as much as possible, but to do this it introduces increased ripple.

```
\begin{figure}[h]
\begin{subfigure}{.5\textwidth}
  \centering
  \includegraphics[width=\linewidth]{figs/cheby1-rp-comparison.png}
  \caption{Chebyshev Type 1}
  \label{fig:cheby1rs}
\end{subfigure}%
\begin{subfigure}{.5\textwidth}
  \centering
  \includegraphics[width=\linewidth]{figs/cheby2-rp-comparison.png}
  \caption{Chebyshev Type 2}
  \label{fig:cheby2rs}
\end{subfigure}
\caption{Comparison of Chebyshev filter rs values}
\label{fig:chebyrs}
\end{figure}
```

The filter is defined with similar parameters to a Butterworth filter - order, critical frequencies, and type all have very similar meanings. However, there are type 1 and 2 Chebyshev filters, with different aims. Additionally, there is a new parameter `\emph{rs}`. I will explain these differences here.

Type 1 Chebyshev filters introduce passband ripple, whereas type 2 Chebyshev filters introduce stopband ripple. Figure `\ref{fig:chebyrs}` shows how these differences manifest in the response of a second order Chebyshev filter.

The parameter `\emph{rs}` is defined as the maximum ripple permitted below unity gain. Higher `\(rs\)` results in more ripple for a type 1 filter, but less ripple for a part 2 filter, as is demonstrated in figure `\ref{fig:chebyrs}`. Introducing more ripple is undesirable as leads to more signal distortion, however can be advantageous as it reduces the transition width.

### \section{Heart Rate Calculation}

Now we have removed noise from the signal, we should have a signal representing the volume of blood in veins within the arm. As the heart beats, it sends a wave of blood through the body, forcing more blood through these veins. Hence, peaks in the PPG signal correspond to heart beats.

There are multiple approaches to finding heart-rate from our filtered signals, and I will detail a few here.

#### \subsection{Local Maxima}

If we can find the peaks in the signal, we can count them up and divide by the time period we count those peaks within, giving us a heart rate. One way we can define peaks is as local maxima.

The local maxima in a signal are defined as samples which are larger than both neighbouring samples. Finding local maxima is an easy problem to solve - intuitively, we can create an algorithm which iterates through the samples, comparing each sample to it's neighbours, which has complexity  $\mathcal{O}(n)$ .

#### \subsection{Phase Vocoder}

An alternative approach to estimating the heart rate uses the signal's Fourier Transform. The Fourier Transformation takes a signal, and returns it's constituent frequencies. We can look at this Fourier Transform, knowing one of the peaks is the frequency associated with the heart beating.

Fourier Transforms alone are not fantastic for finding heart rate, as they

are too low resolution. However we can use a Phase Vocoder to increase the resolution.

`\section{Motion Artifact Reduction}`

In this section I explore methods that can be used to reduce the effect of Motion Artifacts (MAs) on the PPG Signal.

`\subsection{Adaptive Noise Cancellation}`

Adaptive Noise Cancellation (ANC), as described by Widrow et al. `\cite{Widrow75}`, is a technique which allows us to remove noise from a signal, given we have another signal correlated to the noise in some way. In our situation, we know MAs are correlated to motion, so we can use the accelerometer to remove MAs. This technique is extremely useful, as it can still produce good results when the MAs are at the same frequency as the heart beat, even if the MAs have a larger amplitude.

`%TODO: https://ieeexplore.ieee.org/abstract/document/7867772 this may be helpful`

`\begin{figure}[tbh]`  
`\centerline{\includegraphics[width=\textwidth]{figs/ANC-concept.png}}`  
`\caption{ANC overall flow}`  
`\label{epsfig}`  
`\end{figure}`

An overview of the algorithm follows. We have signal from heartbeat  $(s)$  that we want to figure out, but this is contaminated by noise from MAs  $(n_0)$ . We assume the PPG sensor reading  $(p)$  is such that  $(p=s+n_0)$ . Additionally, we have accelerometer sensor readings  $(n_1)$ . We filter  $(n_1)$ , producing signal  $(y)$  and subtract this from  $(p)$ , to produce  $(z=p-y)$ . Our aim is to adjust the filter, such that  $(y)$  is as close to  $(n_0)$  as possible.

The adaptive filter aims to choose a filter such that the power output  $(E[z^2])$  is minimized. Given  $(s)$  is uncorrelated with  $(n_0)$  and  $(n_1)$ , and  $(n_0)$  is correlated with  $(n_1)$ , it can be proven that minimizing the output power is equivalent to  
`% TODO: do this maths better`

`\chapter{Implementation}`

`\section{Gathering Data}`

The project required data from two sources: a PPG sensor mounted on a watch, and a chest-mounted ECG. In this section I detail the process of developing an application to record PPG data on a Wear OS watch.

`%TODO: more detail on watch?`

`\subsection{Wear OS Development}`

`\subsection{Synchronising Signals}`

Once I have collected the PPG signal from the watch, and have recorded the ECG from the chest, I need a method to synchronise the signals, such that they start at the same time and we can compare them accurately.

I set out to produce a solution with the following properties:

`\begin{itemize}`  
`\item` Able to synchronize signals with  $(\pm 0.3)$ s accuracy. With a heartbeat of 200 bpm this represents being within a heartbeat.  
 This is an acceptable level of delay, as heart rate is

averaged over several beats anyway.

\item Can synchronize signals given the two recordings are started within two minutes of each other. This constraint is helpful as it prevents wasted time searching through the signal.

\item Does not use the clocks built into the device. We must assume the clock within the ECG is unreliable. Additionally, the two devices may be synchronized to different time, and hence could be out by any amount of time.

\end{itemize}

Given I know both devices have an accelerometer, I realised I could ask the wearer to move in some motion which is picked up by both devices. Then, I could compare the two accelerometer signals in order to discover the movement. Then comparing the starting times of the two signals would enable calculation of the time difference between them.

The type of motion I chose was important, as it had to be easy to explain to the wearer, but also provide sufficient motion for it to be identifiable against normal motion. I decided jumping was appropriate - I would ask the wearer to hold the watch close to their chest and jump three to five times.

The cross correlation of two signals  $\langle f \rangle$  and  $\langle g \rangle$  is a measure of similarity as a function of the displacement between them. It is simple to calculate, as a sum of the products between the samples of  $\langle f \rangle$  and of  $\langle g \rangle$  displaced by  $\langle n \rangle$ .

To synchronise the accelerometer, I developed the following three step algorithm:

\begin{enumerate}

\item Normalize signals, by moving to zero mean,

\item Crop PPG signals to two minutes, crop ECG signals to four minutes, compute cross correlation with  $\langle f \rangle$  PPG acceleration,  $\langle g \rangle$  ECG acceleration.

\item Compute the other way - crop ECG to two minutes, PPG to four minutes, compute cross correlation with  $\langle f \rangle$  ECG acceleration,  $\langle g \rangle$  PPG acceleration. This is

\item Find the maximum cross correlation across

\end{enumerate}

\section{Verbatim text}

Verbatim text can be included using `\verb|\begin{verbatim}|` and `\verb|\end{verbatim}|`. I normally use a slightly smaller font and often squeeze the lines a little closer together, as in:

```
{\renewcommand{\baselinestretch}{0.8}\small
\begin{verbatim}
GET "libhdr"
```

```
GLOBAL { count:200; all }
```

```
LET try(ld, row, rd) BE TEST row=all
    THEN count := count + 1
    ELSE { LET poss = all & ~(ld | row | rd)
        UNTIL poss=0 DO
            { LET p = poss & -poss
              poss := poss - p
              try(ld+p << 1, row+p, rd+p >> 1)
            }
        }
    }
```

```

LET start() = VALOF
{ all := 1
  FOR i = 1 TO 12 DO
  { count := 0
    try(0, 0, 0)
    writef("Number of solutions to %i2-queens is %i5*n", i, count)
    all := 2*all + 1
  }
  RESULTIS 0
}
\end{verbatim}
}

\section{Tables}

\begin{samepage}
Here is a simple example\footnote{A footnote} of a table.

\begin{center}
\begin{tabular}{l|c|r}
Left      & Centred & Right \\
Justified &         & Justified \\
\hline
First     & A       & XXX \\
Second    & AA      & XX  \\
Last      & AAA     & X   \\
\end{tabular}
\end{center}

\noindent
There is another example table in the proforma.
\end{samepage}

\section{Simple diagrams}

Simple diagrams can be written directly in \LaTeX. For example, see
figure~\ref{latexpic1} on page~\pageref{latexpic1} and see
figure~\ref{latexpic2} on page~\pageref{latexpic2}.

\begin{figure}
\setlength{\unitlength}{1mm}
\begin{center}
\begin{picture}(125,100)
\put(0,80){\framebox(50,10){AAA}}
\put(0,60){\framebox(50,10){BBB}}
\put(0,40){\framebox(50,10){CCC}}
\put(0,20){\framebox(50,10){DDD}}
\put(0,00){\framebox(50,10){EEE}}

\put(75,80){\framebox(50,10){XXX}}
\put(75,60){\framebox(50,10){YYY}}
\put(75,40){\framebox(50,10){ZZZ}}

\put(25,80){\vector(0,-1){10}}
\put(25,60){\vector(0,-1){10}}
\put(25,50){\vector(0,1){10}}
\put(25,40){\vector(0,-1){10}}
\put(25,20){\vector(0,-1){10}}

\put(100,80){\vector(0,-1){10}}
\put(100,70){\vector(0,1){10}}
\put(100,60){\vector(0,-1){10}}
\put(100,50){\vector(0,1){10}}

\put(50,65){\vector(1,0){25}}
\put(75,65){\vector(-1,0){25}}
\end{picture}
\end{center}
\end{figure}

```

```

\end{center}
\caption{A picture composed of boxes and vectors.}
\label{latexpic1}
\end{figure}

\begin{figure}
\setlength{\unitlength}{1mm}
\begin{center}

\begin{picture}(100,70)
\put(47,65){\circle{10}}
\put(45,64){abc}

\put(37,45){\circle{10}}
\put(37,51){\line(1,1){7}}
\put(35,44){def}

\put(57,25){\circle{10}}
\put(57,31){\line(-1,3){9}}
\put(57,31){\line(-3,2){15}}
\put(55,24){ghi}

\put(32,0){\framebox(10,10){A}}
\put(52,0){\framebox(10,10){B}}
\put(37,12){\line(0,1){26}}
\put(37,12){\line(2,1){15}}
\put(57,12){\line(0,2){6}}
\end{picture}

\end{center}
\caption{A diagram composed of circles, lines and boxes.}
\label{latexpic2}
\end{figure}

```

```
\section{Adding more complicated graphics}
```

The use of `\LaTeX` format can be tedious and it is often better to use encapsulated postscript (EPS) or PDF to represent complicated graphics. Figure~\ref{epsfig} and~\ref{xfig} on page \pageref{xfig} are examples. The second figure was drawn using `\texttt{xfig}` and exported in `{\tt.eps}` format. This is my recommended way of drawing all diagrams.

```

\begin{figure}[tbh]
\centerline{\includegraphics{figs/cuarms.pdf}}
\caption{Example figure using encapsulated postscript}
\label{epsfig}
\end{figure}

\begin{figure}[tbh]
\vspace{4in}
\caption{Example figure where a picture can be pasted in}
\label{pastedfig}
\end{figure}

\begin{figure}[tbh]
\centerline{\includegraphics{figs/diagram.pdf}}
\caption{Example diagram drawn using \texttt{xfig}}
\label{xfig}
\end{figure}

```

```
\chapter{Evaluation}
```

```

\section{Printing and binding}

Use a ‘‘duplex’’ laser printer that can print on both sides to print
two copies of your dissertation. Then bind them, for example using the
comb binder in the Computer Laboratory Library.

\section{Further information}

See the Unix Tools notes at

\url{http://www.cl.cam.ac.uk/teaching/current-1/UnixTools/materials.html}

\chapter{Conclusion}

I hope that this rough guide to writing a dissertation is \LaTeX\ has
been helpful and saved you time.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% the bibliography
\addcontentsline{toc}{chapter}{Bibliography}
\bibliography{refs}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% the appendices
\appendix

\chapter{Latex source}

\section{diss.tex}
{\scriptsize\verbatiminput{diss.tex}}

\chapter{Makefile}

\section{makefile}\label{makefile}
{\scriptsize\verbatiminput{makefile.txt}}

\section{refs.bib}
{\scriptsize\verbatiminput{refs.bib}}

\chapter{Project Proposal}

\includepdf[pages=-,pagecommand={},width=\textwidth]{proposal.pdf}

\end{document}

```

# Appendix B

## Makefile

### B.1 makefile

```
# This is the makefile for the Part II demonstration dissertation
#
# Note that continuation lines require '\' and
# that a TAB character precedes any shell command line

.DELETE_ON_ERROR:

# Rules for building LaTeX documents (see Unix Tools course)
%.pdf %.aux %.idx: %.tex
    pdflatex -halt-on-error $<
    while grep 'Rerun to get ' $*.log ; do pdflatex $< ; done
%.ind: %.idx
    makeindex $*
%.bbl: %.aux
    bibtex $*
# Rules for exporting xfig diagrams into PDF or EPS
%.pdf: %.eps
    epstopdf --outfile=$@ $<
%.eps: %.fig
    fig2dev -L eps $< $@
%.pdftex %.pdftex_t: %.fig
    fig2dev -L pdftex_t -p $*.pdftex $< $*.pdftex_t
    fig2dev -L pdftex $< $*.pdftex

help:
    @echo
    @echo "USAGE:"
    @echo
    @echo "make                display help information"
    @echo "make proposal.pdf   format the proposal as PDF"
    @echo "make diss.pdf        format the dissertation as PDF"
    @echo "make all             make proposal.pdf and diss.pfd"
    @echo "make view-proposal   format and view the proposal"
    @echo "make view-diss       format and view the dissertation"
    @echo "make count           display an estimated word count"
    @echo "make pub             put demodiss.pdf onto your homepage"
    @echo "make clean           delete all intermediate files"
    @echo

view-%: %.pdf
    ( okular --unique $< || evince $< ) &

diss.pdf: figs/cuarms.pdf figs/diagram.pdf makefile.txt diss.bbl

makefile.txt: makefile
```

```

expand makefile >makefile.txt

count:
    detex diss.tex | tr -cd '0-9A-Za-z \n' | wc -w

all: proposal.pdf diss.pdf

pub: diss.pdf
    rsync -t $+ $(HOME)/public_html/demodiss.pdf

clean:
    rm -f *.aux *.log *.err *.out
    rm -f *~ *.lof *.toc *.blg *.bbl
    rm -f makefile.txt

distclean: clean
    rm -f figs/*.pdf proposal.pdf diss.pdf

```

## B.2 refs.bib

```

@INPROCEEDINGS{Wood06,
author={L. B. {Wood} and H. H. {Asada}},
booktitle={2006 International Conference of the IEEE Engineering in Medicine and Biology Society},
title={Noise Cancellation Model Validation for Reduced Motion Artifact Wearable PPG Sensors Using MEMS Accelerometers},
year={2006},
pages={3525-3528},
doi={10.1109/IEMBS.2006.260359},
ISSN={1557-170X},
month={Aug},}

@ARTICLE{Wijshoff17,
author={R. W. C. G. R. {Wijshoff} and M. {Mischi} and R. M. {Aarts}},
journal={IEEE Transactions on Biomedical Engineering},
title={Reduction of Periodic Motion Artifacts in Photoplethysmography},
year={2017},
volume={64},
number={1},
pages={196-207},
doi={10.1109/TBME.2016.2553060},
ISSN={1558-2531},
month={Jan},}

@ARTICLE{Widrow75,
author={B. {Widrow} and J. R. {Glover} and J. M. {McCool} and J. {Kaunitz} and C. S. {Williams} and R. H. {Hearn} and J. R.},
journal={Proceedings of the IEEE},
title={Adaptive noise cancelling: Principles and applications},
year={1975},
volume={63},
number={12},
pages={1692-1716},
doi={10.1109/PROC.1975.10036},
ISSN={1558-2256},
month={Dec},}

@INPROCEEDINGS{Kim15,
author={J. {Kim} and T. {Lee} and J. {Kim} and H. {Ko}},
booktitle={2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)},
title={Ambient light cancellation in photoplethysmogram application using alternating sampling and charge redistribution tec},
year={2015},
volume={},
number={},
pages={6441-6444},
doi={10.1109/EMBC.2015.7319867},
ISSN={1558-4615},
month={Aug},}

```



```
@article{allen02,
  title={Microvascular blood flow and skin temperature changes in the fingers following a deep inspiratory gasp},
  author={Allen, John and Frame, John R and Murray, Alan},
  journal={Physiological measurement},
  volume={23},
  number={2},
  pages={365},
  year={2002},
  publisher={IOP Publishing}
}

@article{shin16,
  title={Ambient temperature effect on pulse rate variability as an alternative to heart rate variability in young adult},
  author={Shin, Hangsik},
  journal={Journal of clinical monitoring and computing},
  volume={30},
  number={6},
  pages={939--948},
  year={2016},
  publisher={Springer}
}
```



# Appendix C

## Project Proposal

*Charlie Maclean  
King's  
cm927*

Diploma in Computer Science Project Proposal

## **Synthesis of Heart-Rate Detection Methods**

*December 18, 2019*

**Project Originator:** Dr Robert Harle

**Project Supervisor:** *Dr Robert Harle*

**Signature:**

**Director of Studies:** *Dr Timothy Griffin*

**Signature:**

**Overseers:** *Dr Anil Madhavapeddy and Professor John Daugman*

**Signatures:** *<no need to obtain Overseers' signatures yourself>*

## Introduction and Description of the Work

Heart-rate signals from watches are unreliable while exercising. Watches make use of photoplethysmography (PPG) sensors - sensors which detect the volume of blood in the skin and use variances in this to reconstruct a heart-rate. PPG sensors are preferred to the more accurate electrocardiogram (ECG) due to user comfort. However, the signals they provide are harder to process - I want to compare strategies to process these signals to extract heart rate.

There are several sources of noise within a PPG signal. There is often high frequency contamination caused by electrical interference or light from external sources. Additionally, there is a constant low frequency variation in the DC background of the signal, as a result of capillary density, blood volume and temperature variations.

In the context of running, motion caused by the arms swinging forward and back causes the sensor to slide along the skin, creating motion artifacts (MAs). These are particularly challenging as they can have a much larger amplitude than the pulse we are looking for. Additionally, they can be at the same frequency as the heart rate signal, making them challenging to filter out. Therefore, there is research [1][2] that suggests using accelerometer data in order to predict MAs. I will implement algorithms which find the MAs based on data from the accelerometer. Following this, I will look into the implementation of filters to remove these MAs.

In order to get accurate heart-rate measurements with which to compare the PPG signals, I will make use of a chest-mounted portable ECG. I will need to synchronize the signals received from the ECG and PPG, due to clock skew between the different devices. Potentially, there could be drift between the two clocks as well.

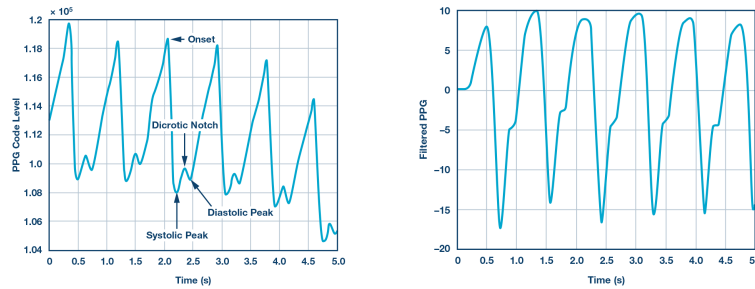


Figure 1: PPG signals before and after filtering [3]

## Starting Point

### Wearable Development

I will need to develop an application for the smart watch which will record PPG and motion signals without interruption. I will use Android Studio and Kotlin.

I have never used Kotlin before and have used Android Studio once before, but never to develop something for a wearable.

### Digital Signal Processing

Manipulation of the PPG signals I receive will require much digital processing, and there are two languages I am considering using: MATLAB and Python.

I have never used MATLAB before, but I am familiar with Python.

## Substance and Structure of the Project

### Core

1. Developing a wearable application to capture PPG signals and motion data. I would like to develop this using Android Studio and Kotlin. I will need to make use of wake locks in order to ensure that the application can continuously record data.
2. Collecting data using a PPG-enabled watch and a portable ECG. I will record my heartbeat over the course of several runs.
3. Synchronizing signals. Implementation of an algorithm to synchronize the data output from the watch records to the data output from the portable ECG. I will have to develop a program to find the lag between the two signals.
4. Removing noise. Due to the various sources of noise, I will investigate potential low and high pass filters, to remove both high and low frequency disturbance.
5. Peak finding - implementing algorithms to find the actual beat given a clean PPG signal. Some potential algorithms are:
  - (a) Adaptive threshold [4]
  - (b) Wavelet transformation [5]
6. Finding MAs - implement an algorithm which uses the accelerometer in order to detect segments of the PPG which are likely to have been affected by motion.

7. Removal of MAs - implementation of filters to remove the previously detected MAs.

## Possible Extensions

- Investigating PPG-enabled earbuds
  - Evaluation of the quality of heart-rate provided by earbuds
  - Exploring the potential to merge signals from a smartwatch and earbuds in order to provide a higher quality signal.
- There is research [6] to suggest gyroscope information is also helpful in filtering out MAs. I could include gyroscope data in my MA filtering technique.
- Comparing more heart-rate detection algorithms:
  - Digital filters
  - Adaptive filters
  - Singular value decomposition
  - Empirical mode decomposition
  - Spectrum analysis

## References

- [1] Z. Zhang. Photoplethysmography-based heart rate monitoring in physical activities via joint sparse spectrum reconstruction. *IEEE Transactions on Biomedical Engineering*, 62(8):1902–1910, Aug 2015.
- [2] Z. Zhang, Z. Pi, and B. Liu. Troika: A general framework for heart rate monitoring using wrist-type photoplethysmographic signals during intensive physical exercise. *IEEE Transactions on Biomedical Engineering*, 62(2):522–531, Feb 2015.
- [3] Foroohar Foroozan. Music-based algorithm for on-demand heart rate estimation using photoplethysmographic (ppg) signals on wrist, 2018.
- [4] Ivaylo I. Christov. Real time electrocardiogram qrs detection using combined adaptive threshold. *BioMedical Engineering OnLine*, 3(1):28, 2004.
- [5] Jake D. Campbell, Christopher G. Pretty, J. Geoffrey Chase, and Phillip J. Bones. Near-real-time detection of pulse oximeter ppg peaks using wavelet decomposition. *IFAC-PapersOnLine*, 51(27):146 – 151, 2018. 10th IFAC Symposium on Biological and Medical Systems BMS 2018.

- [6] Alexander J. Casson, Arturo Vazquez Galvez, and Delaram Jarchi. Gyroscope vs. accelerometer measurements of motion from wrist ppg during physical exercise. *ICT Express*, 2(4):175 – 179, 2016. Special Issue on Emerging Technologies for Medical Diagnostics.

## Success Criteria

The following should be achieved:

- Develop an application which records and stores PPG signals on a watch.
- Create program which synchronises ECG signals with PPG signals.
- Implement at least two filtering algorithms, demonstrate filtering works by displaying signals before and after filtering.
- Implement at least two peak finding algorithms, demonstrate they work by comparing peaks on the PPG signal to peaks on the ECG signal.
- Implement a MA detection algorithm.
- Be able to remove MAs, demonstrating that the peak finding algorithm is not affected by signals caused by motion.

## Resources Required

I will use my own laptop. I will regularly backup my project to GitHub and an external HDD, so that I can recover data in the event of hardware failure. I accept full responsibility for this machine and and I have made contingency plans to protect myself against hardware and/or software failure.

I will use a smart watch, and a chest-mounted ECG sensor in order to collect data.

## Timetable and Milestones

### Weeks 1-2 (28/10/19 - 10/11/19)

- Project set-up:
  - Installation and setup of Android Studio.



- Investigate and install an IDE for Python or MATLAB.
- Use GitHub to set up a backup system for the project.
- Learn to use Kotlin for the wearable app.
- Create a wearable app to record and store PPG and motion data, using Android Studio and Kotlin. Research into wake logs and interface design.
- Additionally, test application by recording data while running.

### **Weeks 3-4 (11/11/19 - 24/11/19)**

- Collect data from runs.
- Build program to synchronize data between ECG and PPG (based on either heart rate variation or motion information).
- Researching and then implementing filters to remove both high and low frequency noise from the PPG signal.

### **Weeks 5-6 (25/11/19 - 08/12/19)**

- Implement program to detect peaks in the signal.
- Begin writing dissertation document.

I will take the next week off.

### **Weeks 7-8 (16/12/19 - 29/12/19)**

- Implement MA detection algorithm.
- Continue writing dissertation.

### **Weeks 9-10 (30/12/19 - 12/01/20)**

- Implement MA removal algorithm.
- Continue writing dissertation.

**Weeks 11-12 (13/01/20 - 26/01/20)**

- Begin extension work, testing earbud HR detection against wrist-watches and portable ECGs.
- Continue writing dissertation.

**Friday 31st January - Progress Report due**

**Weeks 13-14 (27/01/20 - 09/02/20)**

- Extension work, testing whether earbud HR can be merged with wristwatch signal to create a more accurate output.
- Continue writing dissertation.

**Weeks 15-16 (10/02/2020 - 23/02/2020)**

- Extension work, including gyroscope feedback into the MA detection algorithm.
- Continue writing dissertation.

**Weeks 17-18 (24/02/2020 - 08/03/2020)**

- Extension work, including gyroscope feedback into the MA detection algorithm.
- Continue writing dissertation.

**Weeks 19-20 (09/03/2020 - 22/03/2020)**

- Writing dissertation.

**Weeks 20-21 (23/03/2020 - 05/04/2020)**

- Writing dissertation, aim to have first draft by second week of Easter holiday.

**Weeks 22-26 (06/04/2020 - 08/05/2020)**

- Continue improving dissertation.
- Leaving time for exam preparation.

**Final deadline for dissertation 08/05/2020**