



Combinational Photoplethysmography Based Model for Blood Pressure Measurement

A thesis submitted in fulfilment of the requirements for the degree of Master of Engineering

Ross Nye

Bachelor Computer Science (Embedded Systems), RMIT University
Diploma of Computer Systems, RMIT University
Associate Diploma in Engineering (Computer Systems), RMIT University

School of Engineering
College of Science, Engineering and Health
RMIT University

June 2018

Declaration

I certify that except where due acknowledgement has been made, the work is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program; any editorial work, paid or unpaid, carried out by a third party is acknowledged; and, ethics procedures and guidelines have been followed.

I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

Ross Nye

09th June 2018

Acknowledgements

I wish to sincerely thank Professor Qiang Fang, Chair and a professor in Biomedical Engineering at the Department of Biomedical Engineering, Shantou University, China. Professor Fang was initially my supervisor when I started working on this research project. He encouraged me to begin and made all the arrangements and affiliations listed below.

This research began as a research project paid for and instigated by Mr Jie Yu and his company is Beijing Sanjack Ltd, China. Without his interest in this subject this research would not have commenced. For this I offer my humble thanks to him and to his company.

I would also like to thank Doctor Hui Li who conducted the clinical experiments on behalf of Beijing Sanjack.

I also offer my eternal gratitude to Associate Professor Elena Pirogova who has been my guiding light especially since Professor Fang took up his new position in China. Without her support and guidance this thesis would never have been completed.

To my fellow research partner at the beginning of this research, Doctor Zhe Zhang, I thank you for everything. Now we can talk about something else whenever we see each other.

To my fellow HDR students Matt Dabin, Xiaoying Wang, Yinjun Tu, Dr. Ghazwan Haddad and Tilak Rajapaksha thanks for making the time we shared in our little office more than just about work.

Finally, to my family, Mom, Graeme, David, Ross, Kristy and Joan, thanks for your love and support, and for putting up with me, now and forever; especially to Mom for bringing me all those dinners.

Table of Contents

Declaration	i
Acknowledgements	ii
Table of Contents	iii
List of Figures	vii
List of Tables	ix
List of Abbreviations	x
Executive Summary	1
Chapter 1: Background	3
What is Blood Pressure	3
Blood Pressure Regulation	4
Blood Pressure as a Health Indicator	6
BP Measurement Methodologies	8
Invasive Blood Pressure	8
Non-Invasive Blood Pressure	9
Continuous Non-Invasive Blood Pressure	13
Photoplethysmography (PPG)	14
Chapter 2: Literature Review	17
Pulse Transit Time (PTT)	18
Pulse Wave Velocity (PWV)	19
Pulse Wave Analysis (PWA)	20

BP Monitoring Bodies and Standards	23
Association for Advancement of Medical Instrumentation Standards	23
British Hypertension Society Classifications	24
Effectiveness of Standards	24
Research Hypothesis	26
Research Objectives	26
Research Questions	28
Chapter 3: Materials and Methods	29
Materials.....	29
Biopac hardware and software	29
Materials of Dual Channel Device	31
Development Board.....	31
Open Hardware Pulse Sensor	34
Keil µVision IDE.....	35
3D Modelling and Printing	36
MATLAB	36
Methodology of Using Dual Channel Device	39
Sensor Positions Using Dual Channel Device.....	39
Dual Channel Device Usage Protocol	41
Chapter 4: Results	45
Biopac Experiments	45

Dual Channel Device	46
Device Software	46
3D Printed Case and Sensor Housing.....	47
Data Collection Tools.....	54
MATLAB Data Collection tool.....	54
MATLAB GUI data collection tool	56
Windows .Net GUI.....	57
Data Collection.....	59
Initial Tests	59
Circadian Experiments	60
Collaborators Data Collection	64
Data Consolidation – Common Data Format	66
Algorithm Development.....	70
PEAK – Systolic peak	71
MSUS – Mid-systolic up stroke	71
SSUS – Start of the systolic up stroke	71
PSUS – Pre-systolic up stroke	72
Detection of PPG Signal Points.....	72
Analysis Algorithm Output	73
Analysis Data.....	75
Heart Rate Validation	77

Problems Analysing Data.....	78
Fixing Device Read Data Error	78
Signal Abnormalities	82
Creating a Fit.....	84
Evaluation Against the AAMI and BHS Standards	88
Chapter 5: Discussion	95
Assessment of the Fitting Algorithm	95
Assessment of the Proposed PPG Methodologies	96
Assessment of the Dual Channel PPG Device	99
Effect of White Coat Syndrome on Results	100
Chapter 7: Conclusion.....	101
Chapter 6: Future Work	102
References.....	105
Publications.....	108
Appendix.....	109

List of Figures

Figure 1: Arterial branching of aortic arch and carotid arteries [2]	4
Figure 2: Hales invasively measuring a horse's BP [8]	8
Figure 3: Tonometry method [14].....	11
Figure 4: Reflective PPG sensor [26]	15
Figure 5: Example PPG signals	16
Figure 6: Graphical definition of PTT [33].....	18
Figure 7: Example of PWV sensor placement [40]	19
Figure 8: Project outline.....	27
Figure 9: Biopac MP100 specifications [55]	29
Figure 10: Biopac hardware setup	30
Figure 11: Open Hardware PPG sensor contents.....	34
Figure 12: Open Hardware PPG sensor design [57]	35
Figure 13: PPG locations	39
Figure 14: PWV Distance Approximation.....	39
Figure 15: PPG finger showing PPG housing and Velcro attachment	42
Figure 16: Example PPG output on LCD screen	43
Figure 17: Biopac TSD200 PPG sensor [59]	45
Figure 18: Model of Alientek Warship board.....	48
Figure 19: Base component of dual channel device case	49
Figure 20: Alientek board on case base	50
Figure 21: Model of Alientek board on case base	50
Figure 22: Rear view of model of case with cover attached	51
Figure 23: Top view model of case with cover.....	52
Figure 24: Internal view of model of case cover	52

Figure 25: Underside of model of PPG sensor housing.....	53
Figure 26: Side view of model of PPG sensor housing	53
Figure 27: Windows .Net GUI.....	58
Figure 28: Omron HEM-6221 [62].....	60
Figure 29: SBP, DBP and HR Omron data from day 4 of Circadian Experiments	63
Figure 30: SBP, DBP and HR Omron Data from all days of Circadian Experiments.....	63
Figure 31: Conversion to common data format	68
Figure 32: PPG detection points	70
Figure 33: Flow chart of method for PPG signal point detection	72
Figure 34: Segment of analysis algorithm output	73
Figure 35: Analysis algorithm output	74
Figure 36: Data acquisition from device to PC.....	79
Figure 37: Determining a swapped channel.....	81
Figure 38: Signal showing poor PPG sensor placement.....	82
Figure 39: Signal showing movement artefacts and poor ear PPG placement	83
Figure 40: Example output of fittest.m	86
Figure 41: Fit validation output graph	94

List of Tables

Table 1: WHO Hypertension Classification [6].....	7
Table 2: State of the art research.....	21
Table 3: British Hypertension Society Classifications [54].....	24
Table 4: STM32F103T6 Key Features [56].....	33
Table 5: Example SBP, DBP and HR data collected during Circadian Experiments.....	62
Table 6: Common Log Format	69
Table 7: PPG Meta data structure	75
Table 8: PTT/PWV calculations examples	76
Table 9: Fit validation with 10 calibration tests.....	90
Table 10: Fit validation with 25 calibration tests.....	90
Table 11: Fit Classifications of Full Circadian Experiments.....	91
Table 12: Fit Validation: Circadian, Fit: 25, Eval: 55, Weight: None.....	92
Table 13: Fit Validation: Circadian Full, Fit: 25, Eval: 142, Weight: None	93

List of Abbreviations

AAMI – Association for Advancement of Medical Instrumentation	23
ADC - Analogue to Digital Converter	31
APG - Acceleration PlethysmoGram.....	68
BHS – British Hypertension Society	24
BP - blood pressure.....	1
BPM – beats per minute.....	77
CNIBP – Continuous Non-Invasive Blood Pressure	13
CO - cardiac output.....	4
CSV – Comma Separated Values	58
CVD - cardiovascular diseases	6
DBP – Diastolic Blood Pressure	3
DVP - Digital Volume Pulse.....	14
FIR - Frequency Impuse Response	55
IABP – Intra-Arterial Blood Pressure.....	8
IBP Invasive Blood Pressure	8
IC – Integrated Circuit	35
IDE – Integrated Developlement Environment	35
LCD – Liquid Crystal Display.....	32
LED - Light Emitting Diode	14
MAP – Mean Arterial Pressure.....	3
PAT - Pulse Arrival Time	18
PD - Photo Diode.....	14

PPG - Photoplethysmography.....	1
PTG - Plethysmogram.....	14
PTT – Pulse Transit Time	17
PWA – Pulse Wave Analysis.....	17
PWTT - Pulse Wave Transit Time.....	18
PWV – Pulse Wave Velocity	17
SBP - Systolic Blood Pressure	3
TPR _ Total peripheral resistance.....	4
USB –Universal Serial Bus.....	32
WHO - World Health Organisation	6

Executive Summary

Blood pressure (BP) is an important vital sign commonly used to monitor a patient's health. Blood pressure measured at the extremes, or changes in measurements over time, can be an important indicator and determining factor in diagnosis of a large variety of health conditions. As such, monitoring of blood pressure for extended periods can be used to help monitor a patient's overall health as well as to track the progress or onset, of an illness.

Current methods for monitoring blood pressure are unsuitable for continuous, uninterrupted measurement outside of a hospital setting as these methods are invasive and could pose risks to the patient if complications occur. It would be desirable to be able to monitor blood pressure in a continuous yet non-invasive way. The development of such a technique would allow for continuous monitoring outside of a hospital setting and allow for a person to have their blood pressure continuously monitored whilst engaging in their everyday activities.

This research has investigated the possibility of using two photoplethysmography (PPG) sensors to measure blood oxygen saturation and through this monitor a patient's pulse at two sites; the ear and the finger. Using these two simultaneous and continuous measurements the ultimate aim is develop a method of monitoring blood pressure.

There have been several developments of the research. The first has been the creation of a device that is capable of simultaneously collecting data from two photoplethysmography sensors and sending this data to a connected computer.

The second outcome has been the writing of programs capable of capturing the data delivered from the device to a PC and storing this data for later analysis.

The third outcome is a software tool that processes and analyses the data collected from the dual channel PPG device. This tool calculates four variants of pulse transit time (PTT) and pulse wave velocity (PWV) in an attempt to map the collected PPG data to blood pressure measurements taken through conventional means using oscillometric sphygmomanometry at the time the PPG data was recorded.

Chapter 1: Background

What is Blood Pressure

Blood pressure is a commonly used vital sign of the human body used to monitor a patient's health and is used as indicator for the cardiovascular system. BP is a measurement of the force exerted on the blood vessel walls by blood as it is pumped around the body by the heart. As blood pressure increases the heart must work harder to pump blood.

The measurement of BP consists of two separate measurements each corresponding to two different phases of the cardiac cycle; systole and diastole. Systole is the stage at which the heart has contracted to force blood along the arteries and around the body. Diastole is when the heart is at rest between each contraction. [1]

BP is described as systolic blood pressure (SBP) *over* diastolic blood pressure (DBP). For historic reasons both SBP and DBP are measured in millimetres of mercury, (mm Hg). These two measurements give an indication of maximum and minimum BP respectively.

Mean arterial pressure (MAP) is another BP measurement, also given in mm Hg. It combines SBP and DBP into a single measurement, but in doing so loses some of the information the separate measurements provide.

BP is an ever changing and not constant and is determined by the following four factors

- Heart rate; how fast the heart is beating
- Stroke volume; the volume of blood that can be pumped by a single heart contraction
- Blood volume and viscosity; the volume of blood within the circulatory system and how thick that blood is

- Arterial elasticity and thickness; elasticity reduces with age and onset of arteriosclerosis (thickening of the arteries)

Heart rate and stroke volume can be combined into a single measurement named cardiac output (CO). Total peripheral resistance (TPR), can be categorised as the resistance the blood and the blood vessels apply to the movement of blood throughout the circulatory system; it is a combination of the other two factors mentioned above. As shown below in Equation 1, BP is can be defined as a combination of both CO and TPR [1].

Equation 1

$$BP = CO \cdot TPR$$

As cardiac output and/or total peripheral resistance increases as does blood pressure. Conversely as either decreases so too does BP.

Blood Pressure Regulation

The body regulates BP though a variety of mechanisms which alter CO or TPR as a means of maintaining BP at normotensive, or normal, levels. Much of this regulation is controlled from the medulla oblongata at the base of the brain.

Baroreceptors measure the pressure that blood is applying to the vessel walls within the aortic arch (near the heart) and the bifurcation point of the left common carotid artery (in the neck) where it splits into the left internal carotid and left external carotid arteries; see Figure 1.

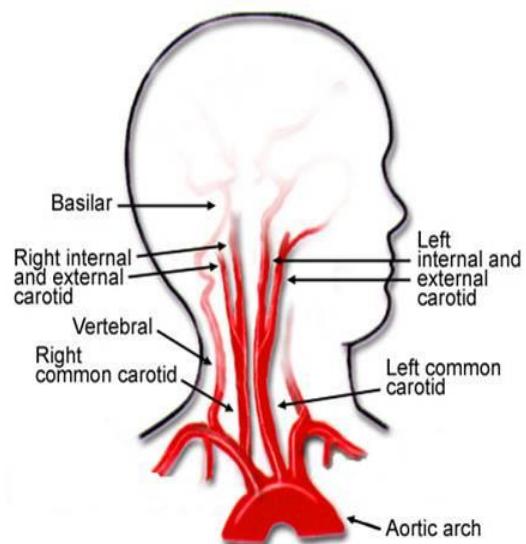


Figure 1: Arterial branching of aortic arch and carotid arteries [2]

These pressure signals from the baroreceptors are sent to the medulla oblongata which contains the following centres for controlling BP:

- Cardiac acceleratory centre – sympathetic nervous system
- Vasomotor centre – sympathetic nervous system
- Cardiac inhibitory centre – parasympathetic nervous system

The cardiac acceleratory centre triggers the release of chemicals in the heart so as to increase both the heart rate and the stroke volume and thus increasing cardiac output.

Conversely the cardiac inhibitory centre stimulates the heart, so cardiac output is decreased, but unlike the cardiac acceleratory centre is it does not stimulate the heart muscles directly.

When responding to stimulus from the baroreceptors the vasomotor centre controls the size of blood vessels as follows:

- vasodilation – increase the diameter of the blood vessels
- vasoconstriction – decrease the diameter of the blood vessels

The kidneys also play an important role in maintaining normotensive BP mainly though the secretion of renin. The secretion of this enzyme from the kidneys is triggered by the sympathetic nervous system. It acts in three ways to influence BP. When combined with angiotensin which are peptides that are produced in the liver and the lungs, it causes constriction of the blood vessels, aiding in vasoconstriction. Renin also triggers re-absorption water within the kidneys and an increase in thirst (and thus increased water intake) both of which aid in preserving or increasing the volume of water in the blood and thus effect the blood volume and its viscosity [1].

Blood Pressure as a Health Indicator

As mentioned BP is commonly used vital sign of the human body. It is an important indicator of overall health and in particular it reflects the state of a person's cardiovascular system.

According to the World Health Organisation (WHO) cardiovascular diseases (CVDs) are the number one cause of death across the world. Annually more people die from CVDs than any other cause. During 2015 an estimated 17.7 million people died from CVDs [3].

High blood pressure, or hypertension, is both a risk factor and early warning sign of many CVDs. Hypertension occurs when blood vessels endure a persistently raised pressure. This increased BP places the vascular system under stress and the heart must work harder to pump blood throughout the body. This not only risks damage to the heart, but also the brain, kidneys and other organs. The WHO's classification of hypertension can be seen in Table 1 on the following page.

Prolonged hypertension can lead to heart failure, aneurysms, stroke and/or cognitive impairment, kidney failure or blindness.

Hypotension, or low blood pressure, is less likely to be associated with long term health problems and so for some people low BP is normal. However, hypotension can also be associated with other underlying health problems and can cause dizziness, nausea or other symptoms.

The WHO recommends routine checks of BP in adults as changes in BP can be an early indicator of impending or current health concerns [4]. As early as the 1920s it has been common for physicians to use BP as routine measurement in assessing patient health [5].

Table 1: WHO Hypertension Classification [6]

Category blood pressure*	Systolic (upper) blood pressure mm Hg	Diastolic (bottom) blood pressure mm Hg
Norm		
Optimal**	120	80
Normal	130	85
High normal	130-139	85-89
Hypertension		
1 degree (soft)	140-159	90-99
grade 2 (moderate)	160-179	100-109
3rd degree (severe)	180	110
edge	140-149	90
Isolated systolic hypertension	140	90

* If systolic and Diastolic blood pressure in different categories, the higher category is selected.

** Optimal with respect to the risk of developing cardiovascular complications and mortality

BP Measurement Methodologies

Invasive Blood Pressure

Invasive blood pressure (IBP), or Intra-Arterial Blood Pressure (IABP), is a commonly used methodology for measuring BP. It is particularly useful in measuring BP in an ongoing clinical setting such as a hospital ward or intensive care unit as well as in an operating theatre during surgery when constant monitoring of BP is necessary [7].

This method of measuring blood pressure involves insertion of an arterial cannula into a suitable artery (often the radial or femoral arteries) of the subject. The cannula is connected to a sterile fluid filled tube. As the blood pressure in the subject's artery changes the fluid in the tube rises and falls with this change in pressure.

Reverend Stephen Hales is credited with being the first person to measure BP when he used IBP to measure the blood pressure of horses in 1733 [8, 9]. During these experiments he inserted a brass pipe attached to glass tube into the horse's artery as depicted in Figure 2.

The main advantage of IBP is that it allows for continuous beat-to-beat monitoring of BP.

This is essential during operations or any other time a patient is under anaesthesia, but it is also highly desirable for monitoring of patients suffering acute conditions.

Primarily due to dire complications that could result from the cannula dislodging



Figure 2: Hales invasively measuring a horse's BP [8]

from the patient's artery, IBP is not suitable outside of a hospital or clinical setting where a patient is able to be constantly monitored. If such dislodging occurs the patient could bleed out and lose vast amounts of blood.

Still, IBP is the most reliable method of accurately measuring blood pressure.

Non-Invasive Blood Pressure

Mercury Sphygmomanometers

Sphygmomanometers are a device for measuring blood pressure and were first introduced in the late 1800's. The name is derived from the Greek for pulse, *sphygmo* and *manometer* which is an instrument for measuring the pressure of a fluid.

Early sphygmomanometers were cumbersome, and their results were difficult to reproduce. The first non-invasive method for measuring BP was created by Samuel von Basch, an Austrian physician, in 1881. This water filled cuff instrument was the earliest sphygmomanometer [8, 10].

In 1896 however, Italian physician Scipione Riva-Rocci first published details of a mercury manometer and inflatable cuff based device capable of reliably measuring BP [5, 8]. His device occluded blood flow in the arm using a rubber squeeze bag to inflate the cuff which was placed over the upper arm. Palpation was used to determine the point where there was no longer a discernible pulse in the radial artery. The mercury manometer displayed the pressure to which the cuff was inflated at this point, in millimetres of mercury. The pressure at which the cuff was inflated is measured as the subject's systolic blood pressure.

It was not until 1905 and Russian Nikolai Korotkoff described his ‘Korotkoff Sounds’ that diastolic blood pressure could be measured [5, 8, 11]. These sounds, heard through auscultation using a stethoscope during the inflation and deflation of Riva-Rocci’s inflatable cuff. Korotkoff described five distinct sounds that occur after the cuff is inflated to beyond the subject’s systolic BP and slowly released. Initially no noise is heard whilst the cuff completely collapses the artery and no blood is flowing. The first Korotkoff sound occurs as blood starts to flow. The onset of this sound indicates the subject’s systolic pressure. The fifth Korotkoff sound indicates the diastolic pressure, which occurs when the pressure in the cuff no longer occludes blood during diastole when the heart is at rest.

Korotkoff’s method for auscultation using a stethoscope with a mercury sphygmomanometer has not changed and is still used over a century later. There are however aneroid sphygmomanometers where a mechanically driven dial replaces the mercury tube used to measure the pressure. These instruments tend to lose accuracy over time due to wear and tear. As such auscultation using mercury sphygmomanometers remain the ‘gold standard’ measurement instrument for non-invasive blood pressure monitoring [8].

Oscillometric Sphygmomanometers

Most modern automated sphygmomanometers developed for home use fall into this category. The earliest version of the oscillatory method was first described in 1897 by Hill and Barnard [8].

Like traditional mercury sphygmomanometers an inflatable cuff is used to compress the artery in the arm and occlude blood flow. However, this method does not use auscultation to listen for Korotkoff sounds, but instead a pressure gauge is used to monitor tiny oscillations

in amplitudes of the measured cuff pressure caused by the pulse passing under the cuff as part of the beat-to-beat heart rhythm.

The cuff is first inflated and then deflated over time. As this occurs the pressure in the cuff passes from being above systolic BP, and thereby totally occluding blood flow, to moving through to systolic pressure where maximum oscillations on the gauge occur. Pressure in the cuff is continually dropped until minimal oscillations occur on the gauge, which indicates diastolic blood pressure.

These devices are gaining increased popularity due to their ease of use without any need for specialist training. However, their accuracy is questionable for subjects with certain pre-existing conditions such as atrial fibrillation, obesity and atherosclerosis [12, 13].

NIBP – Tonometry and Vascular Unloading

Non-Invasive Blood Pressure (NIBP) devices have existed for commercially for decades. These non-invasive devices also allow for continuous measuring of BP [14] . These devices are typically based on two techniques. The first of which is known as applanation tonometry. This method acquires the arterial pressure wave by applying force onto an artery squeezing and flattening the vessel wall against a bone as shown in Figure 3.

Transmural forces are measured perpendicular to the arterial surface which is then used to estimate BP [14-17]. Although there are other commercially available

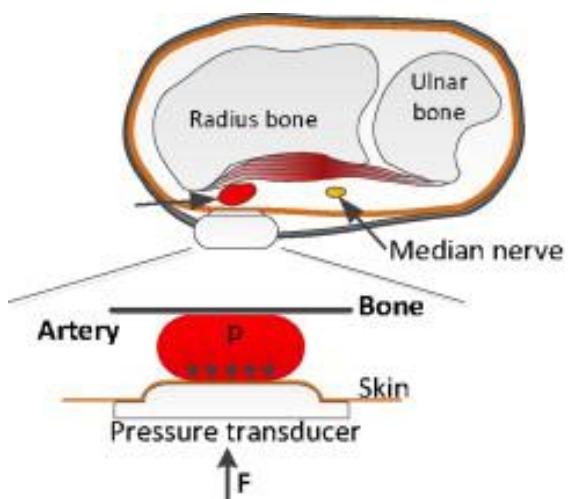


Figure 3: Tonometry method [14]

devices based on the tonometry principle, SphygmoCor (AtCor Medical, Australia) is the most established [18].

The force applied to the artery must be small enough so that blood flow is not completely shut. The correct positioning of the devices is also particularly important. Incorrect readings would be obtained if the position is even slightly out. Tonometry is also particularly susceptible to incorrect readings caused from movement of the subject [14].

The other NIBP method is known as vascular unloading [19] or alternatively as the volume clamp method [12, 14]. This method does use a PPG sensor, which as seen in the following sections forms the basis of this research; see the Photoplethysmography (PPG) section on page 14. Vascular unloading also employs an occlusive cuff to restrict blood flow in a controlled manner, but only on the finger rather than the whole arm or limb. Using oscillometry blood volume is estimated while zero transmural pressure is being applied. The device uses the PPG sensor to control how much the cuff restricts blood flow in the finger in order to keep the blood volume under the PPG sensor constant throughout the beat-to-beat cycle of the heart. The amount of restriction required to keep the blood flow constant is used to measure the subject's BP as the pressure applied by the cuff equates to the BP [12]. This method was first developed by Czech physiologist Ian Penaz in the early 1970 with his device called FINAPRES – FINger Arterial PRESSure [20].

NIBP devices are rarely used outside of clinical research. This is due to their discomfort of use, complex mechanical structure and high manufacturing cost [19]. A wide ranging study by Kim et al [21] which pooled results from many other studies found that the accuracy of devices based on the volume clamp method do not satisfy the AAMI standards; see Association for Advancement of Medical Instrumentation Standards section on page 23 for more information about the standard.

Continuous Non-Invasive Blood Pressure

A method of continuous and non-invasive blood pressure (CNIBP) is desirable for several reasons. The first and most obvious being the ability to continuously monitor BP without the risks associated with using an arterial catheter that are inherent in IBP. Removing these risks is of itself a good thing but doing so would also open the possibility of continuous BP readings being taken outside of a hospital setting; if there is no catheter involved there is no risk of it falling out at home, or elsewhere, and no risk of the patient bleeding out.

CNIBP monitoring is also desirable for the extra information that can be gathered from continuous monitoring of BP. Intermittently measuring BP has some inherent disadvantages. For example, subtle changes over time can easily be missed in between each instantaneous measurement of BP; CNIBP solves this problem. CNIBP also allows long term trends in BP to be observed in greater detail and without gaps in the information.

Another problem potentially solved by CNIBP is white coat syndrome. Most current forms of BP monitoring involve your BP being measured in a clinical setting by a doctor, nurse or other trained professional. Some patients find the environment stressful. This can in turn have an effect on the patients BP which is then reflected in the measurement [22]. A fully realised version of CNIBP would most likely be able to be used outside such a setting, for example within the home, where such effects would not exist and therefore could not have an impact on the BP measurement.

With the ability to solve these problems and perhaps others too, much research is being undertaken in order to develop a reliable and easy to use method of CNIBP.

Photoplethysmography (PPG)

A plethysmograph is a device that measures changes in the volume of an organ, limb or the whole body. Such a device produces a plethysmogram (PTG) [23] through the act of plethysmography. The word itself is derived from the works ‘*plethysmos*’ and ‘*graph*’, both ancient Greek words meaning increase and write respectively. A photoplethysmograph is a plethysmograph that uses photoelectric, or optical, means to perform the measurement [24].

In literature PPG can be used interchangeably to mean to either a photoplethysmograph device or sensor as well as photoplethysmography. The latter can in fact mean the signal derived from a photoplethysmography or the act of using the device to create the signal.

A pulse oximeter is a common PPG device that monitors the oxygen saturation of a patient’s blood and in doing so it also monitors changes in blood volume as well. Usually such a device is worn on the finger and specifically measures digital volume pulse (DVP) [24, 25]. Often when referring to PPGs a pulse oximeter is the device that is being used.

A PPG sensor is a simple electronic device consisting of three main components; one or more light emitting diodes (LED), a photo diode (PD) and a small amplifier/rectifier circuit. The light from the LEDs are shone into the patient. Depending on the type of PPG sensor this light may be either reflected or transmitted to the PD in varying amounts depending on the oxygen saturation of the blood. This distinction gives way to the main classification of PPG sensors; reflective or transmissive.

In a transmissive PPG sensor the LED and PD are located on either side of the device and the light is shone through the patient; the light from the LED passes through the subject’s finger and is detected by the PD on the other side.

As can be seen in Figure 4, in a reflective PPG sensor the LED and PD are located on the same *side* of the device. Light from the LED is shone into the subject and reflected to varying degrees by the blood in the vascular system beneath the sensor back to the PD.

PPG sensors measure the volume of blood in the vascular system directly beneath them. Blood is not instantly transported throughout the body, but rather it is forced around the vascular system by the heart.

During systole, when the heart contracts to pump blood around the body, the oxygen saturation of the blood is increased to the point called the systolic peak. This can be seen in Figure 5 on the following page marked as point ‘a’, with time between each of the ‘d’ points and ‘a’ points being systole.

Conversely when the heart relaxes during diastole, between points ‘a’ and ‘d’ in Figure 5, the oxygen saturation decreases. As such the oxygen saturation increases and decreases as part of

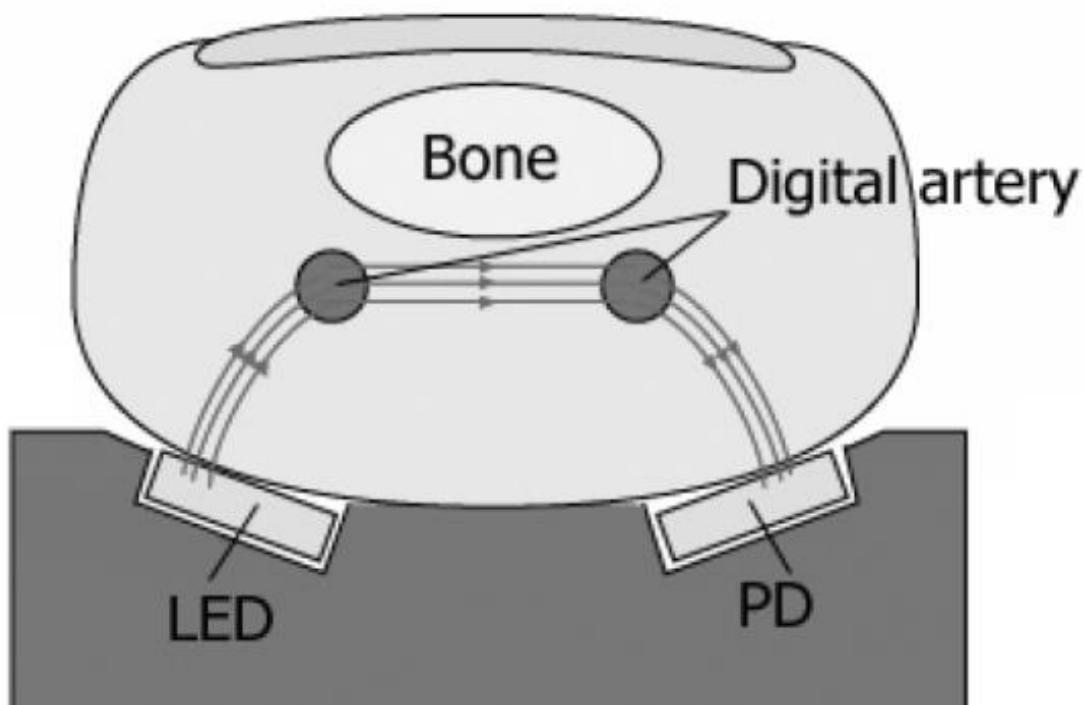


Figure 4: Reflective PPG sensor [26]

the body's natural heart rhythm. The peaks and troughs in the PPG readings correspond to each heart beat and as such can be used to determine a patient's heart rate.

Due to their relative simplicity PPG sensors are not only low-cost devices, they are also very durable as they have no moving parts. They can also easily be manufactured in a small form factor. These are key reasons all for their prevalence in research aiming to develop a methodology for CNIBP.

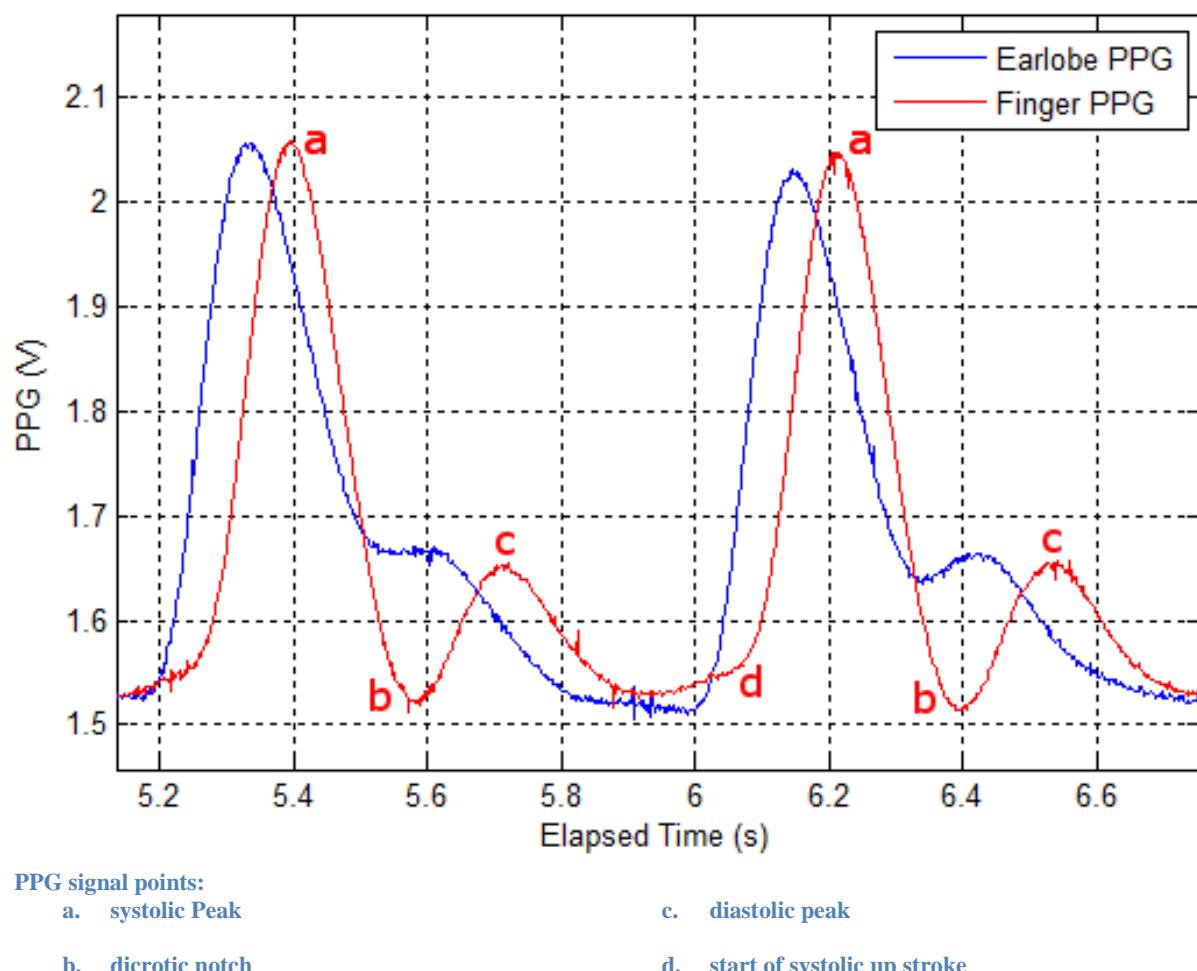


Figure 5: Example PPG signals

Chapter 2: Literature Review

A number of methods have been proposed for using PPGs in the estimation of blood pressure by many different researchers and research groups. Categorising them can be difficult due to the sometimes differing use of the same terms or differing terms used for the same methodology.

However, they can generally be grouped into the following three categories:

- Pulse Transit Time (PTT)
- Pulse Wave Velocity (PWV)
- Pulse Wave Analysis (PWA).

The following sections of this chapter detail each of the above methodologies.

It should be noted that, despite the inverse relationship between a pulse wave velocity and its transit time, there is a difference in the methodologies, and the names merely serve as an identifier for the methodology. For example Gesche et al [27] obtained their measurements using PTT methodology, but later converted the measured PTT into Pulse Wave Velocity to form the calculations in their model. In their paper Myint et al [28] described PTT as being the time between the two maxima of two PPG waveforms at different sites and that it was inversely proportional to the proportional to the propagation velocity of the pulse wave. This definition does not align with most other research.

This confusion regarding terminologies was specifically addressed by Mohamed Elgendi in his paper “Standard Terminologies for Photoplethysmogram Signals” [29] where he suggests standard terminologies to alleviate the confusion that is persistent throughout research in this area.

Pulse Transit Time (PTT)

Pulse Transit Time (PTT) [30-36] can also be referred to as Pulse Wave Transit Time (PWTT) (e.g. by Zhang et al [37]) or Pulse Transmit Time (e.g. Y. Chen et al[38]) or Pulse Arrival Time(PAT) (e.g. Tang et al [39]). This technique uses a single PPG signal along with an Electrocardiogram (ECG) in order to calculate the PTT.

The placement of the PPG and the ECG electrodes differ between research groups and there are pros and cons around where each device is placed. For example, in order to maximise the delay between and make the variations easier to detect the toe is a good site for a PPG sensor. However, this is at odds with a patient's comfort and everyday living if the monitoring is being performed in an at home-based setting.

The transit time in PTT is defined as the time between the R-Peak of the ECG and specific points of the PPG signal waveform, for example the systolic peak ('a' in Figure 5). In Figure 6 Parry et al [33] have shown their relevant PPG point as being the point of the maximum slope in the PPG signal.

Whichever characteristic PPG point is chosen both events, the R-Peak in the ECG and the selected PPG point, occur with each heart beat on a beat-to-beat basis and can be detected through signal analysis.

A downside of this approach is that two types of sensors are required, which makes the device's design and construction more complex.

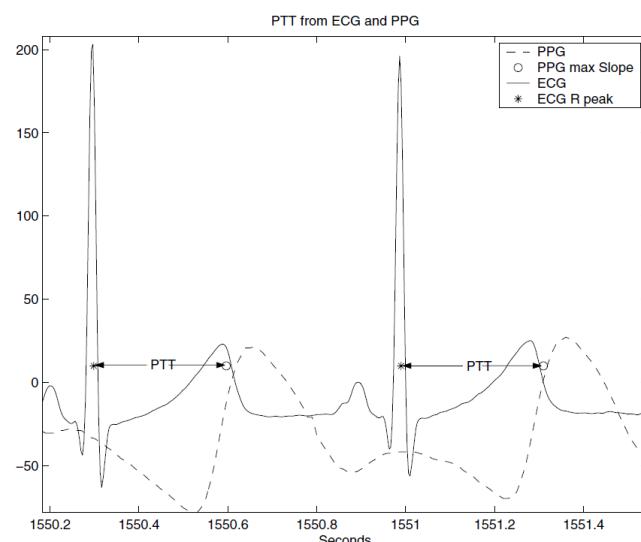


Figure 6: Graphical definition of PTT [33]

Pulse Wave Velocity (PWV)

The Pulse Wave Velocity (PWV) methodology uses signals from two PPG sensors, much like in Figure 5. Again, this approach has quite a broad definition but always involves two PPG sensors being placed at a known distance apart along the same arterial branch. The PPG is signals are analysed and the same characteristic point is located for each beat in the beat-to-beat heart cycle. For each heart beat the time difference between the times that the two points are detected on each of the two PPG sensors is calculated. This time difference is then used along with the distance between them to calculate the velocity of the pulse wave.

As already mentioned the description of this methodology is quite broad and there are many varying factors between different implementations. One distinguishing factor amongst the differing techniques is the placement of the PPG sensors. For example McCrombie et al [40] calculated PWV by placing one PPG sensor above the ulnar artery at the wrist joint and the second on the proximal phalange of the pinky finger (see Figure 7). Peter et al [14] described another form of PWV in which the common carotid artery in the neck is used along with femoral artery in the thigh. Both sensor sites were chosen due their proximity to the aorta, the body's main artery. Nabeel et al [41, 42] have constructed a device containing two PPG sensors designed to be worn on the neck to measure PWV from the common carotid artery.

A direct correlation between measured velocities derived using PWV and BP exists [14, 40]. However, in order to calculate BP accurately a precisely measured distance between the two sensors is required. Also, precise placement of sensors is required to achieve optimum results. Sensors may move during the measurement or over time which can make calculations inaccurate.

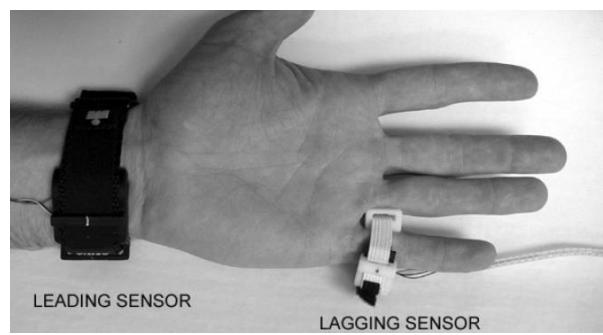


Figure 7: Example of PWV sensor placement [40]

Pulse Wave Analysis (PWA)

Pulse Wave Analysis (PWA), is another classification of PPG based BP estimation methodologies. This methodology differs from the other two in that only a single PPG sensor is used. BP is estimated by analysing different properties of PPG signal.

Among others this technique was employed by Teng and Zhang [43]. Their study extracted diastolic time from a single PPG sensor on the right index finger comparing their results with their previous PTT based study they found that they were able to better estimate DBP, but not SBP with this single PPG based method.

As stated by Samria et al their method was “not highly accurate” [44]. Their methodology was to extract features like systolic time and diastolic time form a single DVP PPG signal as well as calculating the time delay between systolic and diastolic peaks (‘a’ to ‘c’ in Figure 5). This information was all used in order to estimate BP. However they had to split their results of their small sample group between two age groups (above and below 26 years old).

Ruiz-Rodriguez et al [45] noted that, due to a high degree of variability and wide error margin, the signal analysis method is not currently viable.

A similar approach had been conducted by S. Suzuki and K. Oguri [46] and a better outcome was achieved. The test subjects were grouped into classes defined by the characteristics of their systolic upstroke and systolic peak, and then split into two age groups. They concluded that the separation of the classes had improved the accuracy of the method.

The primary obstacle in using PWA using a single PPG sensor to estimate or measure BP is that PPG signals are prone to dramatic non-systematic error, for example due to patient movement. However, this may be offset in the future by removing PPG signal artefacts

caused motion corruption as proposed by Joseph et al [47] using Discrete Wavelet Transform.

A. Suzuki and K. Rye also point out that factors such as gender and age have dramatic effects on how PPG signals should be interpreted in this area [48].

Table 2: State of the art research

Research Group	Methodology		Description
	<i>Type</i>	<i>Sensors Used</i>	
D. B. McCombie, A. T. Reisner, and H. H. Asada	Pulse Wave Velocity	PPG (wrist) + PPG (finger)	Method requires calibration procedure but it was has been validated, though not in real time [40].
J. M. Zhang, P. F. Wei, and Y. Li	Pulse Transit Time	ECG (wrist) + PPG (finger)	Proposed a linear relation between PWTT and BP. Careful calibration is required [37].
I. C. Jeong, J. I. Ko, S. O. Hwang, and H. R. Yoon	Pulse Transit Time	ECG (chest) + PPG (ear) w/ Pressure and Temperature	Used temperature and pressure sensors attached to the PPG to expand the model correlating PTT to BP [26].
H. Gesche, D. Grosskurth, G. Küchler, and A. Patzak	Pulse Transit Time	ECG (chest) + PPG (finger)	Used a one-time calibration using sphygmomanometric cuff. Sampled using PTT but converted to PWV (using $0.5 * \text{patient's height}$) to create a model for calculating BP [27].
Y. Chen, C. Wen, G. Tao, and M. Bi	Pulse Transit Time	PPG (ear) + PPG (toe)	Examined 6 different timing points on PPG signal waveform to best points to detect in order to calculate both SBP and DBP [38].
X. He, R. A. Goubran, and X. P. Liu	Pulse Transit Time	ECG + PPG (finger)	Used data from MIMIC database to confirm a correlation between PTT SBP, DBP and mean BP [31].
Y. Choi, Q. Zhang, and S. Ko	Pulse Transit Time	ECG + PPG (finger)	Also used MIMIC database. Applied Hilbert-Huang transform (HHT) to the ECG and PPG signals [32].
C. Myint, K. H. Lim, K. I. Wong, A. A. Gopalai, and M. Z. Oo	Pulse Wave Velocity	PPG (wrist) + PPG (finger)	Calculated BP in real time but have not published any results only stated that is showed promising potential [28].
Teng and Zhang	Pulse Wave Analysis	PPG	Analysed the signal from a single PPG and proposed two functions for estimating SBP and DBP [43].
R. Samria, R. Jain, A. Jha, S. Saini, and S. R. Chowdhury	Pulse Wave Analysis	PPG (finger)	Proposed multiple linear regression relationships for estimating BP dependant on age group of patient [44].

Research Group	Methodology		Description
	<i>Type</i>	<i>Sensors Used</i>	
J. C. Ruiz-Rodriguez, A. Ruiz-Sanmartin, V. Ribas, J. Caballero, A. Garcia-Roche, J. Riera, et al	Pulse Wave Analysis	PPG (finger)	Working with patients in hospital this group examined data simultaneously collected via PPG and intravenous BP measurement. They determined that clinical application of PSA is currently not possible [45].
S. Suzuki and K. Oguri	Pulse Wave Analysis	PPG (finger)	Analysed the second directive of the PPG signal specifically monitoring systolic upstroke and systolic peak [46].
P.M. Nabeel, S. Karthik, J. Jospeh, M. Sivaprakasam	Pulse Wave Velocity	2x PPG (neck)	Two PPG sensors mounted in a device worn on neck that measures PWV in common carotid artery [41, 42].
B. Ibrahim, V. Nathan and R. Jafari	Pulse Transit Time	PPG (finger) and Bio-Z (wrist)	Proposed a unique methodology for measuring PTT by using a Bio-Z impedance sensor on the wrist to replace ECG signal. This was compared to ECG based PTT measurements [49].
Soo-young Ye, Gi-Ryon Kim, Dong-Keun Jung, Seong-wan Baik, and Gye-rok Jeon	Pulse Transit Time	ECG + PPG	Derived a pulse pressure from ECG/PPG signals in enhance PTT-BP estimation [34]
SH Song, JS Cho, HS Oh, JS Lee and IY Kim	PPG replacing auscultation	PPG (finger) and PPG (wrist)	Used an occluding wrist cuff but rather than used auscultation to detect Korotkoff sounds used PPG to detect them [50].
Zunyi Tang, Toshiyo Tamura, Masaki Sekine, Ming Huang, Wenxi Chen, Masaki Yoshida, Kaoru Sakatani, Hiroshi Kobayashi, and Shigehiko Kanaya	Pulse Transit Time	ECG + PPG	Constructed a chair with embedded ECG and PPG sensors used to estimate BP whilst subject seated in chair [39].
Thomas Wibmer, Coy Denner, Christoph Fischer, Benedikt Schildge, Stefan Rüdiger, Cornelia Kropf-Sanchen, Wolfgang Rottbauer and Christian Schumann	Pulse Transit Time	ECG + PPG (finger)	Performed a study comparing PTT estimated BP with BP measurements obtained from automated sphygmomanometry and volume clamp methods [51].
Edmond Zahedi, Vahid Sohani, M. A. Mohd. Ali, Kalaivani Chellappan and Gan Kok Beng	Pulse Wave Analysis	PPG (wrist)	Investigated the possibility of substituting a PPG signal from radial artery into the generalised transfer functions used by devices that use applanation tonometry [18].

BP Monitoring Bodies and Standards

Association for Advancement of Medical Instrumentation Standards

Association for Advancement of Medical Instrumentation (AAMI) is an American body who aims to promote the use and improvement of medical instruments and their use. AAMI first published their SP10 standard covering sphygmomanometers in 1987 [52, 53]. The AAMI SP10 standard has undergone a number of revisions since.

The AAMI SP10 standard covers both manual and automatic sphygmomanometers, covering all aspects of their accuracy and use right down to the point of effective labelling and product information. The standard is a voluntary one but is designed to increase safety and performance as well as enabling comparison of different products through the use of uniform testing.

The most relevant part of the AAMI SP10 standard to this research is its classification of automatic blood pressure monitoring devices. It prescribes a testing regime for validating readings from an automatic blood pressure monitor against readings taken from auscultation using a standard mercury-based sphygmomanometer. It states that a device must not differ from the mercury standard by a mean difference of greater than 5 mm Hg or a have a standard deviation of greater than 8 mm Hg [52, 53].

The AAMI SP10 standard is referred to heavily in literature and has been adopted by International Organization for Standardization (ISO), ISO 81060-1 and ISO 81060-2. The first part refers to non-automated non-invasive sphygmomanometers and the second part relates to automated devices.

British Hypertension Society Classifications

Initially developed around the same time as the AAMI standard, the British Hypertension Society (BHS) also developed a standard for classifying automated blood pressure monitoring devices. The BHS's aims are similar to that of the AAMI standard. However, rather than a binary pass or fail of the American standard, the British standard classifies a device using four separate grades based on the comparison with reading given from the mercury sphygmomanometer. The classifications are shown in Table 3.

Table 3: British Hypertension Society Classifications [54]

Grade	Absolute difference between standard and test device (%)		
	≤ 5 mmHg	≤ 10 mmHg	≤ 15 mmHg
A	60	85	95
B	50	75	90
C	40	65	85
D	Worse than C		

Effectiveness of Standards

The stated main aim of AAMI is to promote the development of medical instrumentation through the use of standards. Their SP10 standard for sphygmomanometers was developed and refined to further this cause. The BHS standard was developed with similar intent. AAMI has been somewhat effective in its aims by having their standard be adopted as ISO 81060 by International Organization for Standardization.

However, both the AAMI and BHS standards are voluntary. A manufacturer of any blood pressure monitoring device is not obligated to conform to these standards. As such most devices on the market, particularly those that are not designed for use in a clinical setting, have never been validated against these standards. This results in devices on the market and in use by the community of which there is no way of knowing how accurate, or perhaps inaccurate, they are.

This poses a real issue for the future adoption of these technologies for mainstream use in ongoing prediction of patient health. Putting faith in a device that may be completely inaccurate could be a recipe for failure.

Further evidence of this; twice, years apart, the researcher enquired with the distributors of Omron in Australia asking if they had a list of Omron devices that had been validated against the AAMI standard. On both occasions the distributors were unaware as to what the researcher was referring to and were unable to produce such a list. In fact, they thought it was a reference to the Australian insurance company called AAMI.

Research Hypothesis

The focus of existing research into PPG based BP estimation has predominately investigated the use of PTT either PWV methodologies. To date a reliable PPG based method for estimating BP in a continuous and non-invasive manner has not been established. However, there is a prevailing presumption that such a methodology may be possible. This presumption is also made as a basis for this research. If this presumption is proven to be true it would allow for the development of continuously and non-invasively measure blood pressure (CNIBP).

The ultimate aim of this research would be to develop a model that could be used estimate BP by using data collected from one or two PPG sensors. This model could be used to monitor changes in BP over the short and long term.

To evaluate this model the estimated BP is to be compared to measurements obtained using a sphygmomanometer. This would be in a manner similar to what is stipulated in the AAMI SP10 / ISO 81060-2 and BHS standards, using the standards as a guide to evaluate the accuracy of the model where if the difference between the estimated and measured BP readings determines the model's effectiveness.

Research Objectives

This research project has two main aims. The first of which was to develop a device that is capable of simultaneously capturing two PPG signals and recording them. This device should be agnostic to the PPG methodology used in endeavouring to estimate BP from the PPG signal(s). It should also be capable of sampling the analogue PPG signals at a high enough rate and with enough sensitivity to detect subtle changes in the signal(s).

This device should be used to collect dual channel PPG data on a number of subjects which can be used in the second aim of this project described below.

The second main aim of this project was to use the data collected from a dual channel PPG device and endeavour to estimate the subject's blood pressure. This estimation should be validated against blood pressure measured using conventional means. This validation of the measurement would be akin to the validation of devices in AAMI SP10 / ISO 81060-2 standard.

As seen in Figure 8 the research project can be broken down into many components, each contributing to the overall aims of the research. As seen below and discussed in Chapter 4 the development of hardware to capture PPG signals as the data acquisition software that allows for the recording of this information was crucial to the latter stages of data analysis from which in turn leads to the development of the model.

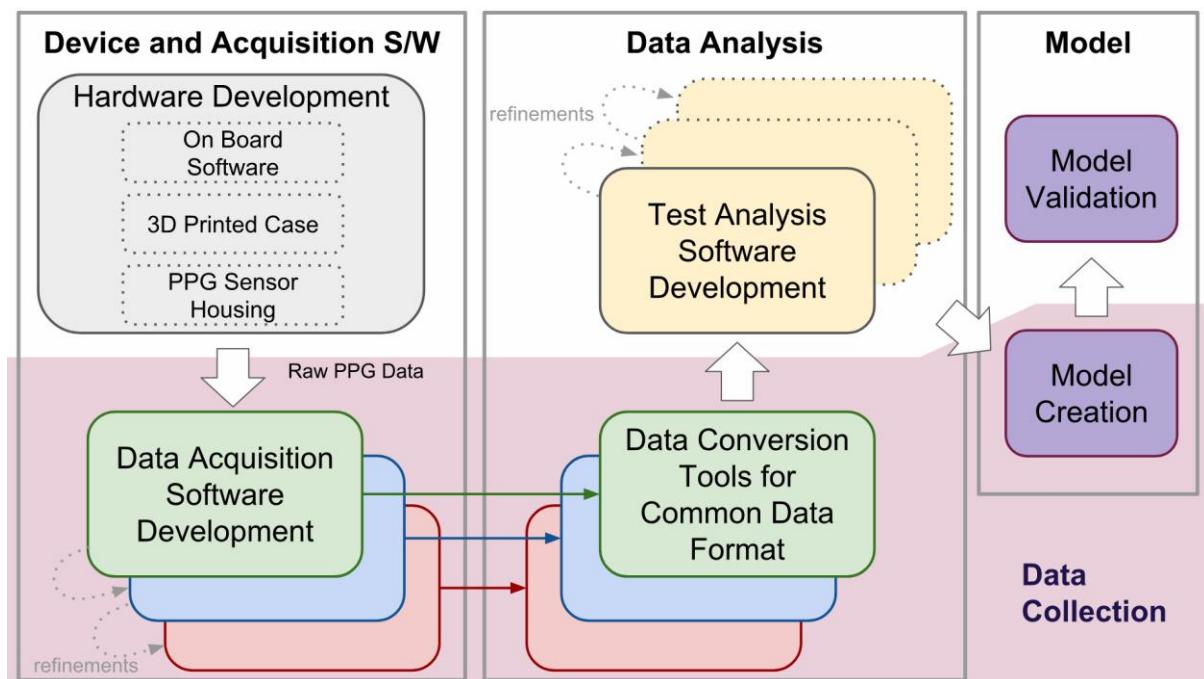


Figure 8: Project outline

Research Questions

1. Is estimation of blood pressure in a continuous and non-invasive way possible using two PPG sensors?

Can the estimations obtained be validated against BP measurement standards?

2. Can a PPG from a sensor on a patient's ear be used as a substitute for an ECG in calculating PTT?

3. Can differing versions of PTT or PWV be derived in order to improve BP estimation?

Do these differing versions offer different information from one and other?

Chapter 3: Materials and Methods

Materials

Biopac hardware and software

Biopac is a commercial manufacturer of physiological data acquisition systems targeted at the education and research markets. They produce a wide variety of hardware and software for acquiring a range of different biological signals.

The researcher used the Biopac MP100 data acquisition device along with its associated PPG amplifier the PPG100C and TSD200 PPG sensor. The specifications for the MP100 are listed in Figure 9 below.

The MP100 is capable of acquiring many different biometric signals from multiple sources including ECG, EMG, EEG, PPG and other sources. This is achieved through the use of

MP150 AND MP100 DATA ACQUISITION UNIT SPECIFICATIONS		
Analog Inputs		
Number of Channels:	16	Time Base
Input Voltage Range:	$\pm 10V$	Min Sample Rate: 2 samples/hour
A/D Resolution:	16 Bits	Trigger Options: Internal, External or Signal Level
Accuracy (% of FSR):	± 0.003	
Input Impedance:	1.0 M Ω	
Analog Outputs		
Number of Channels:	2	Power
Output Voltage Range:	$\pm 10V$	Amplifier Module Isolation: Provided by the MP data acquisition unit
D/A Resolution:	MP150: 16 bits, MP100: 12 Bits	CE Marking: EC Low Voltage and EMC Directives
Accuracy (% of FSR):	MP150: ± 0.003 , MP100: ± 0.02	Leakage current: <8 μA (Normal), <400 μA (Single Fault)
Output Drive Current:	$\pm 5mA$ (max)	Fuse: 2A (fast blow)
Output Impedance:	100 Ω	
Digital I/O		
Number of Channels:	16	
Voltage Levels:	TTL, CMOS	
Output Drive Current:	$\pm 20mA$ (max)	
External Trigger Input:	TTL, CMOS compatible	
Device specific specs	MP150	MP100
Max Sample Rate		
MP Internal Memory:	200K samples/sec (400K aggregate)	70K samples/sec (70 K aggregate)
PC Memory/Disk:	200K samples/sec (400K aggregate)	11K samples/sec (16K aggregate)
Internal Buffer Size:	6M samples	16K samples
Serial Interface Type/Rate:	Ethernet: DLC type II (10M bits/sec)	SERIAL: RS422 (800K bits/sec)
Transmission Type:	Ethernet (10/100 Base T)	USB (via USB1W or USB1M)
Maximum Cable Length:	100 meters (Ethernet cable)	7 meters (USB + SERIAL cable)
Power Requirements:	12VDC @ 2 amp (uses AC150A)	12 VDC @ 1amp (uses AC100A)
Dimensions:	10cm x 11cm x 19cm	7cm x 29cm x 25cm
Weight:	1.0 kg	1.8 kg

Figure 9: Biopac MP100 specifications [55]

modules that are attached to the MP100. Each module connects to one of more sensors or transducers; the number of sensors is dependent on the module type. The modules amplify the signal from the sensors and can be all connected together in a daisy chain fashion. The end module in the daisy chain is then connected to the MP100, which acts as an intermediary, or controller, between the modules and a computer attached to it. It also converts analogue signals into digital. This setup allows for the real time data collection from a variety of sources.

The MP100 connects to a PC (or Mac) serially. The connection can also be converted to USB using an adapter. Data collection using the Biopac hardware also involves using a proprietary piece of software called AcqKnowledge running on the attached PC. This software communicates with Biopac hardware such as the MP100. AcqKnowledge allows for data collection and rudimentary processing of the physiological signals collected using the hardware. Figure 10 below shows the components all connected as described. The experiments conducted with this hardware are described in the following Biopac Experiments section in Chapter 4.



Figure 10: Biopac hardware setup
Left to right: PC running AcqKnowledge, Biopac Serial to USB adapter, MP100, various Biopac amplifier modules connected to MP100 via both Digital and Analog DB cables.

Materials of Dual Channel Device

As described in the previous chapter, one of the main aims and outcomes of this research was the development of a dual channel PPG device that is capable of simultaneously collecting PPG sensor data from two PPG sensors. The following sections describe the hardware, software and materials used in the creation of this device, which the device itself is described in the following chapter.

Development Board

To create the prototype dual channel PPG device an early decision that was made was to use a development board that already contained the appropriate hardware necessary for the needs of the research rather than building something completely from scratch.

In selecting the development board to be used the following criteria needed to be satisfied.

- Low powered device
- Self-contained Analogue to Digital Converters (ADC)
 - Capable of sampling 2 discrete analogue (PPG) signals simultaneously
 - Capable of sampling at a reasonable resolution; 12-bits or above
- Easily connected to a PC to enable data to be transferred for analysis
- Easily programmable Flash storage to store the small program that does the PPG sampling

Many microcontrollers on the market today include an ADC controller on their IC. As such there was a wide variety of microcontrollers that were fit for purpose and many development boards that used them.

These stages of research were conducted as a group partnership. Both research partners were already familiar with ARM Cortex microcontrollers. As such, it was decided a development board based on one would be used. As there are many ARM Cortex variants there was a wide pool to select from.

In the end it was decided to use an Alientek Warship which uses an STM32F103T6 microcontroller. The key features of this microcontroller are listed in Table 4 on the following page.

Other considerations when choosing the Warship included

- a touchscreen LCD (Alientek TFTLCD Module V2.1) is included
- USB ports on the device transfer data as well as supply power to it
- the General Purpose Input / Output (GPIO) header on the board enabled easy access to the inputs for the ADC
- the board is capable of supplying enough power to drive the PPG sensors via power headers on the board
- With the combination of GPIO and power pins the PPG sensors could be connected directly to the board without the need for any additional hardware

Another advantage of the Warship it is supplied with Keil μ Vision IDE and device tool chain. This software is used for writing software that executes on the board and is also used for writing the binary executable (program) to the flash storage inside the STM32F103T6 microcontroller on the Warship. Both research partners were already familiar with this software. Refer to the specific section about this software on page 35.

Table 4: STM32F103T6 Key Features [56]

ARM 32-bit Cortex™-M3 CPU Core 72 MHz maximum frequency, 1.25 DMIPS/MHz (Dhrystone 2.1) performance at 0 wait state memory access Single-cycle multiplication and hardware division
Memories 16 or 32 Kbytes of Flash memory 6 or 10 Kbytes of SRAM
Clock, reset and supply management 2.0 to 3.6 V application supply and I/Os POR, PDR, and programmable voltage detector (PVD) 4-to-16 MHz crystal oscillator Internal 8 MHz factory-trimmed RC Internal 40 kHz RC PLL for CPU clock 32 kHz oscillator for RTC with calibration
Low power Sleep, Stop and Standby modes VBAT supply for RTC and backup registers
2 x 12-bit, 1 µs A/D converters (up to 16 channels) Conversion range: 0 to 3.6 V Dual-sample and hold capability Temperature sensor
DMA 7-channel DMA controller Peripherals supported: timers, ADC, SPIs, I ² Cs and USARTs
Up to 51 fast I/O ports 26/37/51 I/Os, all mappable on 16 external interrupt vectors and almost all 5 V-tolerant
Debug mode Serial wire debug (SWD) & JTAG interfaces
6 timers Two 16-bit timers, each with up to 4 IC/OC/PWM or pulse counter and quadrature (incremental) encoder input 16-bit, motor control PWM timer with dead-time generation and emergency stop 2 watchdog timers (Independent and Window) SysTick timer 24-bit downcounter
6 communication interfaces 1 x I ² C interface (SMBus/PMBus) 2 × USARTs (ISO 7816 interface, LIN, IrDA capability, modem control) 1 × SPI (18 Mbit/s) CAN interface (2.0B Active) USB 2.0 full-speed interface
CRC calculation unit, 96-bit unique ID
Packages are ECOPACK®

Open Hardware Pulse Sensor

Sourced from World Famous Electronics LLC (<http://pulsesensor.com>) these PPG sensors were used in the development of the PPG device developed as part of this research. This device is described below and in the following chapter.

At US\$25 each these Open Hardware PPG sensors were much cheaper than commercially available PPG sensors. They are designed to be used in small projects and because they are available under the Open Hardware licence there are few intellectual property restrictions.

The sensors are supplied with include Velcro strips and pads as well as a metal fastening clip for use in attaching the sensors to different human body periphery such as fingers and ears. The contents of the complete package can be seen in Figure 11 along with a pen for scale.

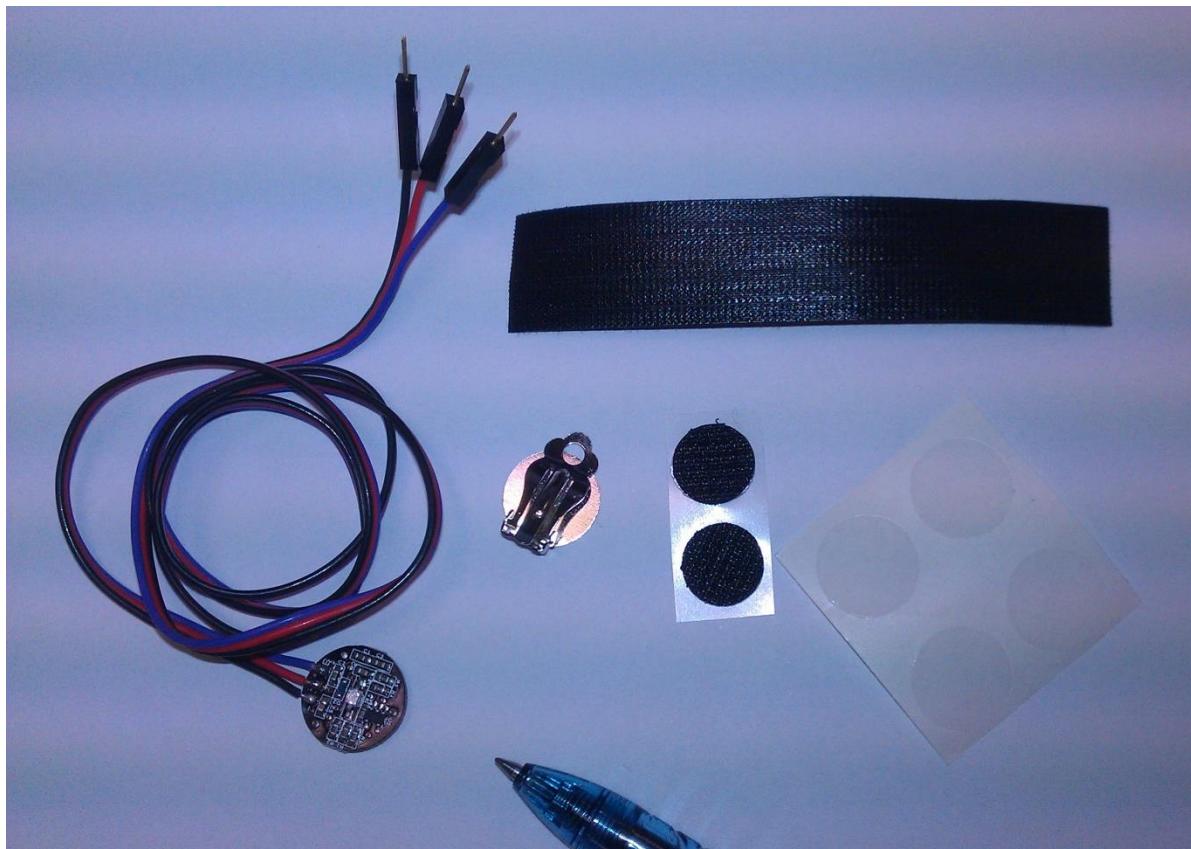


Figure 11: Open Hardware PPG sensor contents
PPG sensor with 3 wire ribbon cable on left; Velcro strip top right; metal clip, 2x Velcro pads and 4x insulation stickers in centre; pen (for scale) at bottom.

The sensors themselves are reflective, visible light PPG sensors and contain the LED, PD and a small rectifier/amplifier circuit IC all in single small form factor package. All they require is an approximately 5V DC power and ground and the analogue PPG signal is output to its output (OUT) pin. The circuit design of these sensors is shown in Figure 12 below.

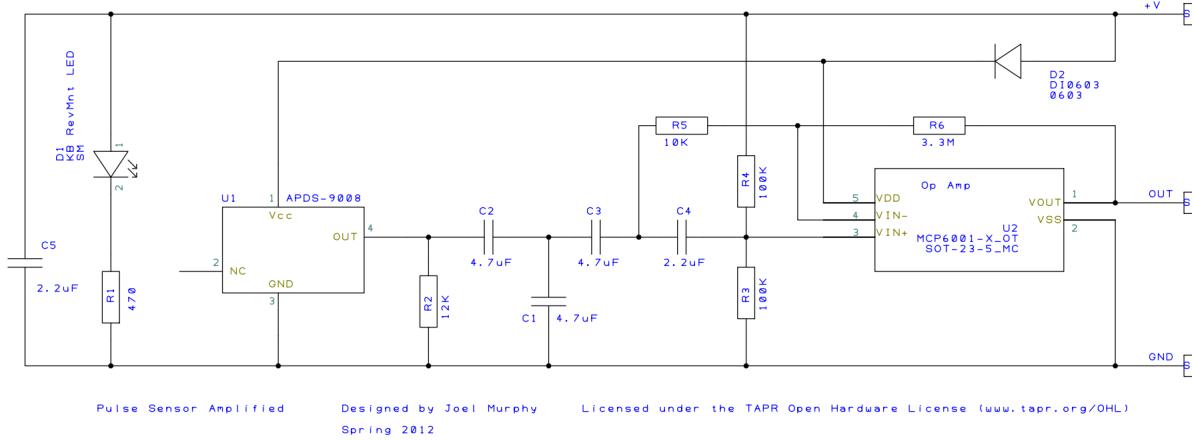


Figure 12: Open Hardware PPG sensor design [57]
See appendix for larger image

Keil µVision IDE

As mentioned in previously in Development Board section of this chapter the development of the program that ran on the development board was done in Keil µVision IDE. This software is produced by ARM and was supplied with the Alientek development board.

The Keil µVision IDE combines all the many of the tools needed for embedded software development, particularly on an ARM platform. It combines a source code editor, tool chain, run-time environment, real time debugger and more into a single program [58].

The IDE not only allows for the software to be written in assembly language relevant to the device being used, but it also allows for code to be written in high level languages C or C++; all in an integrated editor within the IDE. The software can be targeted to a specific

microcontroller (or many), compiled and written directly to an attached development board's Flash storage using the JTAG connector.

3D Modelling and Printing

As described in detail in Chapter 4 a 3D printed case was created for the dual channel PPG device and housing for its PPG sensors. The modelling done to create the design of the case and housing was performed using SketchUp 8. This is a free 3D modelling program designed for a variety of uses from architectural and interior design to mechanical and civil engineering.

Models of the case and housing were converted from the SketchUp SKP file format to STL files using an online converter. STL files are the most common file format for 3D printers and were what was required for the models to be printed on Stratasys uPrint Plus by Dimension, which was used to print the case and housing.

SketchUp, which was owned by Google at the time, is now available from its current owner Trimble as a free online version at <https://www.sketchup.com/products/sketchup-free>. This newer version operates in any web browser and can export files to STL natively without the need for an external tool.

MATLAB

MATLAB, by Mathworks, is a mathematical computing environment the name of which stands for matrix laboratory. This software is available to this research under a licence provided through the university. The MATLAB software environment is primarily built

around the MATLAB scripting language, but MATLAB has many other features available as part of the main installation or as separate toolboxes that can be added to it.

Much of the software developed for this research was written in the MATLAB scripting language, all of which is provided in the Appendices. Using MATLAB meant the tools available in it to analyse the data collected were easily accessible and did not need to be written from scratch.

By the end of the project all the MATLAB scripts that had been written were using MATLAB 2017a. This would be the minimum version required to be able to run many of the scripts as they use some features and syntax (particularly around how strings are handled) that is not available in earlier versions.

The MATLAB environment was used in three different phases of this research. It was used as a part of the data collection where software which ran on the PC collecting the PPG signals from the dual channel device was connected to (with the exception of the .Net GUI – see the section titled Windows .Net GUI in Chapter 4). This software was either a MATLAB script running inside the MATLAB environment or as a standalone MATLAB GUI that was written in MATLAB and packaged with the MATLAB runtimes. This packaging enables the program to be run as a standalone application without even the need for MATLAB to be installed on the PC; it does however create software licensing issues that need to be addressed. See Chapter 4 sections titled MATLAB Data Collection tool and MATLAB GUI data collection tool for more information about these programs.

MATLAB was also used in creating the analysis algorithm that was used to analyse the data from each test. This analysis program, which is actually a conglomerate of MATLAB scripts that are all reliant of each other, was written and refined over many months. This software was used to process the PPG signals recorded as a part of each test and to locate reoccurring

points that exist as part of the beat-to-beat cycle of the PPG signals. These points can then be used to calculate PTT/PWV for each beat as well as an average for the test.

The graphing functionality built into MATLAB was very helpful during this stage of research as plotting graphs with detected points highlighted on them was quite valuable in determining whether detection working as expected or conversely when it was not.

During the third stage of this research MATLAB was also used was in the analysis of the results produced in the previous stages. This very important step aimed to take the beat-to-beat PPG points derived in the previous stage and develop a model, or in MATLAB parlance, a fitting function, to correlate the calculated PTT/PWV values to the patient's BP readings taken using the sphygmomanometers and recorded as part of each test.

Work during the third stage was heavily reliant of the Curve Fitting Toolbox that can be added to MATLAB. Without this toolbox and the functionality built into it the work could not have been completed. Any of the other stages could have been done using almost any programming environment.

The toolbox was feed a variety of inputs from the previous stage and it automatically generated a fitting function to map the PPG data to the corresponding sphygmomanometer recordings. The toolbox also evaluates the accuracy of these functions through a series of metrics.

The aim of stage was indeed one of the main aims of this research; creating a model that uses PPG input and converts this input into a BP measurement. The recorded PPG and sphygmomanometer recordings were used to create the model and then the model was evaluated by comparing the output given PPG inputs to the corresponding BP readings for the test. See Creating a Fit section of the following chapter.

Methodology of Using Dual Channel Device

Sensor Positions Using Dual Channel Device

The dual channel PPG device's hardware on its own does not stipulate the location of its two PPG sensors. The hardware allows the flexibility for the sensors can be placed anywhere on the body that the PPG sensors are able to obtain a usable signal. The choice of PPG sensor locations relies purely on decisions made in relation to the PPG methodology being used.

For the purpose of this research the following two PPG sites were chosen with the rational explained below. Their location is also indicated in Figure 13 below.

- PPG ear – left ear lobe
- PPG finger – left index finger

The finger site can be used like regular pulse oximeter to produce familiar signals; existing / proposed PWA methods can be used with the signal.

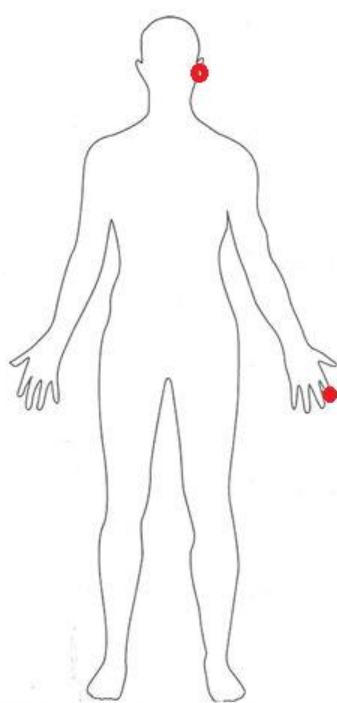


Figure 13: PPG locations

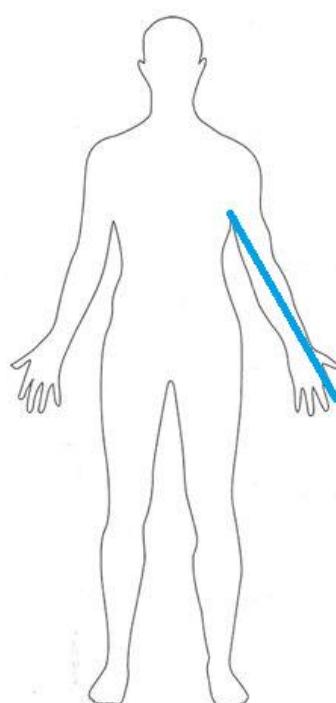


Figure 14: PWV Distance Approximation

Secure placement of sensor on the periphery of finger and the ear lobe is relatively simple using Velcro strip and 3D printed housing for the finger and clip for the ear.

Existing devices recommended using the left side of the body where possible due to layout of vascular system and placement of heart. The left carotid artery branches off from aortic arch first and in a different manner to right carotid artery and as discussed in Blood Pressure Regulation section of Chapter 1 plays a direct role in the body's regulations of BP.

In creating an alternative to PTT, the ear PPG site was chosen to provide a signal that could be analogous to the R-Peak of an ECG signal. Given the ear lobe receives blood supply directly from the carotid arteries and the carotid arteries are fed directly from the aortic arch, which is attached directly to the heart, it was surmised that the systolic peak taken from the PPG ear signal could be used to replace the R-Peak in PTT methodology.

Similarly, PWV was simplified with the chosen PPG placement. The PWV methodology requires the two PPG sensors be placed at a known distance apart on the same arterial branch. For this research this measurement was simplified to using the subject's arm length, measured from the arm pit to the tip of the index finger as shown in Figure 14.

This distance was used due to the ease of measurement and because it roughly equates to the distance between the two sensors if they were both placed on the same arterial branch; the distance to the heart from both the arm pit and the ear lobe is roughly the same. As such the pulse wave would arrive at both locations at a similar time.

These two choices simplified the setup of the equipment and, if proven successful, would allow for enhanced ease of use for any possible device developed using this principle. Also, if the arm measurement was performed incorrectly the PPG signals could be re-analysed using an updated measurement to amend the calculated BP values.

Dual Channel Device Usage Protocol

Almost all the data collected during this research was done so using the dual channel PPG device. The data that was collected came from multiple sources which are discussed in the Chapter 4.

In order to effectively collect data a protocol had to be established and followed to ensure correct and meaningful data was collected.

Collection of dual channel PPG data varied slightly depending on the version of the collection tool being run. Using some of the original tools require steps in the process that were later automated by software. However, for the most part the procedure was mostly the same as detailed below.

Step 1

Turn on the PC being used to run the test, start the MATLAB software and load the data collection tool. Alternatively, start the MATLAB stand-alone GUI tool or the Windows .Net GUI. The steps thereafter are almost identical.

Step 2

Plug the two sensors into the dual channel PPG device using the 3.5mm sockets on the rear of the case. Both sockets have raised dot(s) to indicate which channel the socket is for. The leftmost socket, with the single dot below it, is channel 1. The rightmost being channel 2 with two dots. The PPG sensor for the ear, which has a metal clip attached to it, should be plugged into channel 1. The finger PPG sensor should be plugged into channel 2. The finger sensor in its housing is shown in Figure 15.

Step 3

Attach a USB A to USB Mini-B cable between the device and the PC running the software. This will power the device. When initially powered, the device will display some text about the version of the software running on its LCD screen. This is immediately followed by the device displaying the signals being received from the two PPG sensors; which when not correctly positioned appropriately on the patient will appear as noise.

Step 4

As shown in Figure 13 fit the two PPG sensors to the patient in the appropriate positions; one on the left ear lobe and the other on the distal phalanx of the digitus secundus manus (index finger) of the left hand. Once positioned in place adjustment may be needed to obtain a useable signal; this is easily done with experience.

The key to the placement and attachment is to not completely occlude blood flow.

The clip holding the ear PPG in place cannot be adjusted so the sensor must be positioned in such a way that the way the clip is not too tight. The finger PPG should be placed as shown in Figure 15 with the sensor on the underside of the finger. A usable signal should easily be obtained when the Velcro is not strapped too tight. The sensor needs to be in full contact with the skin, but the Velcro cannot be so tight that it occludes blood from the finger.



Figure 15: PPG finger showing PPG housing and Velcro attachment

Example output from usable PPG signals can be seen in Figure 16. Here the blue signal is taken from the PPG positioned on the subject's ear lobe and the red signal is from the finger. Smaller amplitude and absence of PPG features indicate incorrect PPG placement.

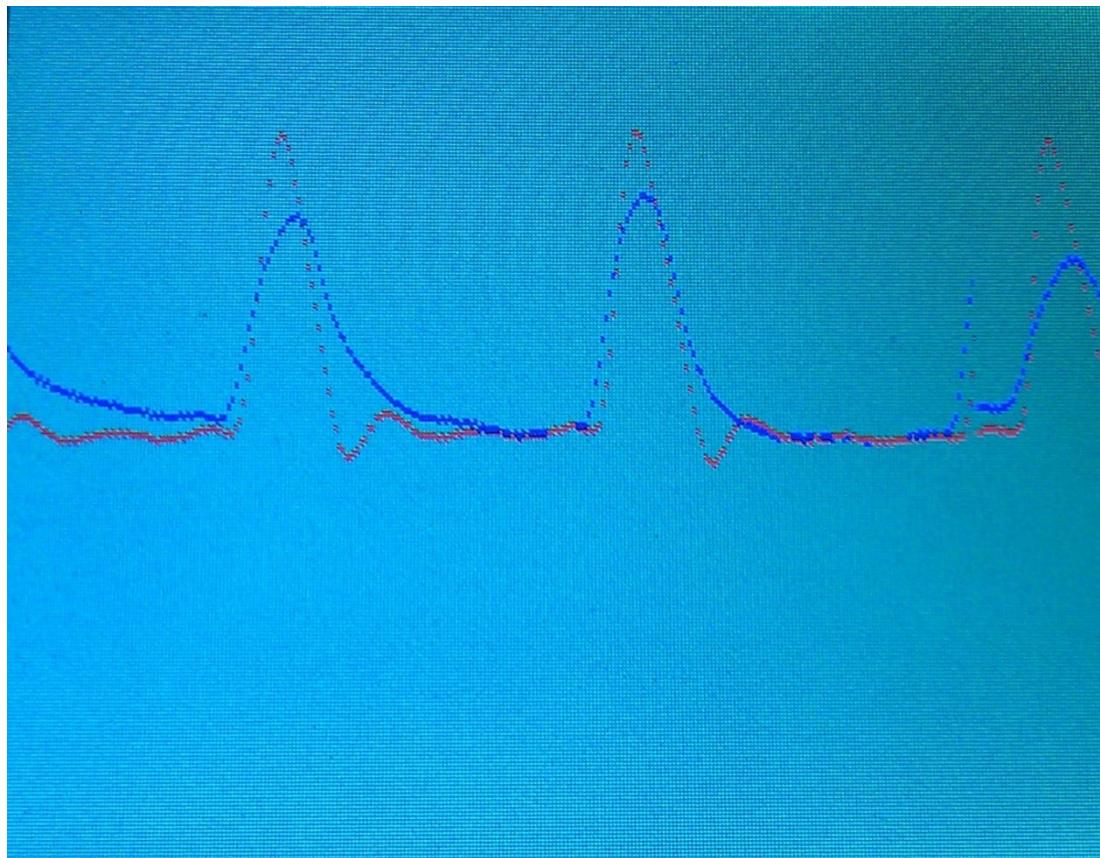


Figure 16: Example PPG output on LCD screen
The LCD display scans from left to right horizontally along the time axis.
The vertical axis is the amplitude of the PPG signals.
The blue signal is taken from channel 1 (PPG ear).
The red signal is taken from channel 2 (PPG finger).

Step 5

Once both sensors are correctly placed in position the subject should be seated and remain still for a period of around 30 seconds and begin breathing in a steady manner. This ensures any movement and irregular breathing patterns do not impact readings taken during the test.

Step 6

Start the data collection tool. Earlier versions of the MATLAB script required the duration of the test to be entered, by altering a variable in the code prior to pressing the run (play) button in the toolbar. In the MATLAB GUI this can be done using the duration can be selected using a pull-down list. In the Windows .Net GUI the test will keep running until the program is told to stop.

When the test begins the LCD on the dual channel device will flash and reset signifying the beginning of the test before once again continuing to display the signals received from the PPG sensors.

The test subject should remain still and breathing steadily throughout the duration of the test.

Step 7

When the collection of PPG has concluded the LCD screen will once again flash and reset. At this stage BP readings should be taken using a secondary device; a sphygmomanometer. The instructions governing use of this device should be followed to ensure accurate BP readings. These readings should be commenced immediately at the conclusion of the PPG readings and with as little movement as possible from the subject both prior and during the operation of the sphygmomanometer. The SBP, DBP and HR readings obtained from the sphygmomanometer should then be entered into the appropriate prompts within the data collection tool. These readings will then be saved along together with the PPG data.

Chapter 4: Results

Biopac Experiments

During the early stages of this research some initial investigation into PPG sensors and the waveforms produced they was conducted using commercially available hardware and software. These experiments were performed using a MP100 and associated PPG module (PPG100C) and PPG transducers (TSD200) (shown in Figure 17) as well as the AcqKnowledge software all from Biopac. These were described in Chapter 3.

The only real aim of these experiments was to develop a familiarity with collecting PPG sensor data and being able to visibly identify specific aspects of the signals. The experiments were performed on this researcher over a number of sessions.

Whilst data was collected as a part of these experiments, this data was never used. The outcomes however were not nil. Experience with the equipment and firsthand knowledge of how movement can affect the PPG signal was obtained. It was also determined that the Biopac PPG sensor's design was inflexible; the Velcro strap was only suitable for strapping the sensor to a subject's finger.

It also established as to how prohibitively expensive the Biopac hardware was. There was initial interest in trying to establish if the PWV methodology could be performed using two PPG sensors placed apart, perhaps on separate phalanxes of the same finger. Only having access to a single PPG Biopac sensor, a quotation for sought for another sensor from the Australian distributor, SDR Scientific Solutions. A single PPG sensor (TSD200) cost AU\$320 ex GST.



Figure 17: Biopac TSD200 PPG sensor [59]

Dual Channel Device

This device was originally created as a prototype for a pilot study for gathering dual channel PPG data as means of using these signals to continuously and non-invasively measure blood pressure.

For the purposes of evaluation and calibration the efficacy of the device was established during the course of the initial investigation and data collection. When comparing this device to what could be achieved with the Biopac hardware and software solution this device not only met the needs of the project but was also more flexible and the PPG signals obtained from the device were easier to analyse.

All this was achieved at a fraction of the cost of the commercial solution. The reduced cost and increased flexibility was at least in part due to the different PPG sensors used in the device. The Biopac PPG transducers were much more expensive and only designed to be worn on the finger.

Device Software

A small program was written in the C programming language and programmed on the Flash RAM on the Alientek board using Keil μ Vision as mentioned in the previous chapter.

This program set up two ADC channels for input by writing specific values to certain ports controlling the ARM microcontroller's ADC. Once set up the program then continually sampled those two ADC channels at a rate of 1 kHz. Therefore, every millisecond both PPG sensors were read, their analogue signal converted to a digital signal and stored temporarily.

This data was then sent to an attached PC for storing and/or processing using virtual serial communications over the USB port. The same USB port was also used for powering the device. Whenever it is powered the device also displays both PPG signals in real time on its LCD screen. In order to not have the data drawn too quickly to the screen only every 5th sample is drawn to the screen. The signals are drawn from left to right before returning to the left of the screen to repeat the process.

3D Printed Case and Sensor Housing

To ensure that the dual channel device was more durable as well as presentable a 3D printed case to enclose the development board and the attached LCD screen was created along with housings for the PPG sensor so one could easily be worn on the finger. As mentioned in the previous chapter both the case and sensor housing were purpose designed in SketchUp 8 and printed using Stratasys uPrint Plus by Dimension.

In order to create the case for the device firstly a model of the Alientek board and its attached LCD screen was created. This model was made by measuring dimensions on the board itself and the components on it as well as their precise position. This information was used to make a 3D model of the board so an appropriate case could be modelled around it. The model of the Alientek Warship board can be seen in Figure 18 on the following page.

The model shows the attached LCD screen in the middle of the board. Note the two mounting legs and screws which needed to be taken into account when designing the case in order to make a snug fit. To the right of the LCD is the orange GPIO header that contained the ADC input pins for the PPG sensors. The two orange headers in the middle right edge are the 5V and ground pins used to create the power circuit for the PPG sensors.

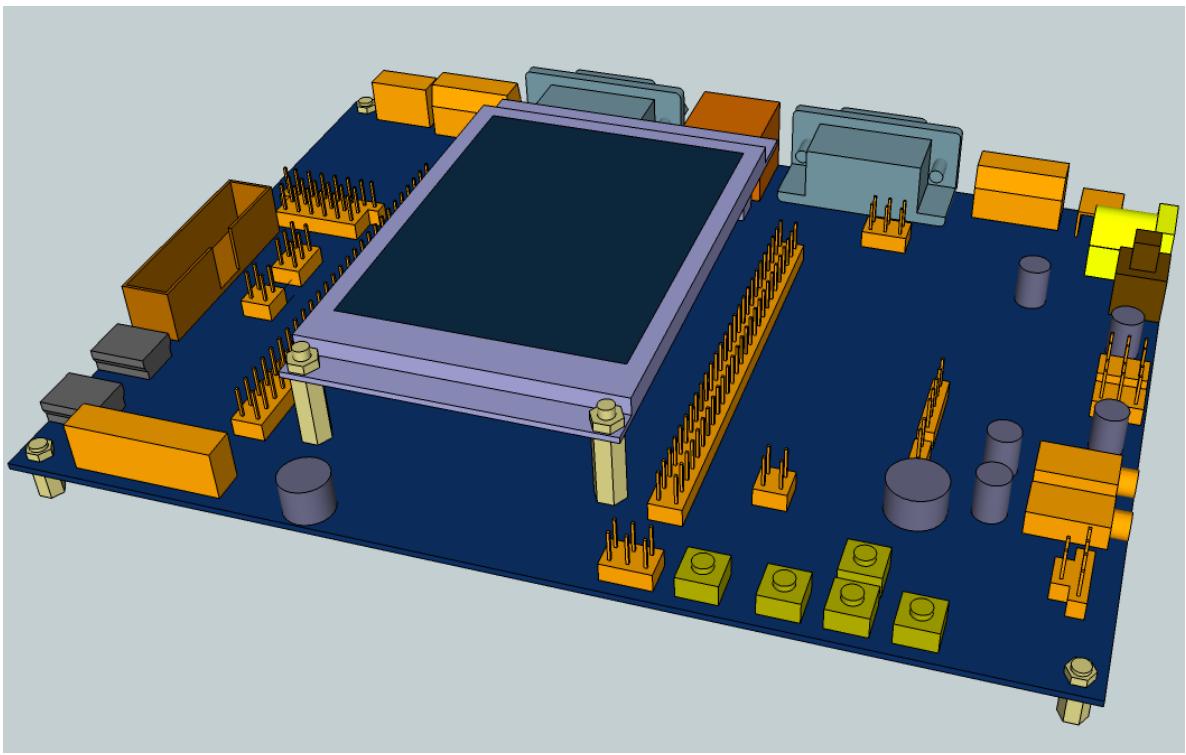


Figure 18: Model of Alientek Warship board

Another important factor in the design of the case was the position of the two USB ports on the board. They are shown as the two grey components on the left of the board. The USB port closest to the brown socket is the one that is used to communicate with the associated PC and to power the board.

The final pertinent factor in the design of the case was the position of the four mounting screws in each of the corners of the board. Two of these can be seen along the front edge of the board in Figure 18. These screws are used to secure the board inside the case.

The case was printed in two parts; a base (shown in Figure 19 on page 49) and the cover (shown in Figure 24 on page 52). In the design the Alientek board is attached to the base and the cover placed over the top including the base's back plate. The cover is secured to the base using 3 tongue and groove slot where the tongues on the base slot into the grooves on the inside of the cover. You can see three of the five tongues on the base in Figure 19.

In this Figure 19 you can also see the four holes that allow screws to be screwed through the base into the boards mounting screws to secure it in place. The model showing the Alientek Warship board on the case's base is shown in Figure 21. On Figure 20 shows what this looks like in reality. Both of these figures are on the following page.

Also visible in Figure 19 on the rear of the base are the two holes that accommodate the board's two USB ports can be seen on the left. A USB micro connector passes through these orifices to plug into the receptacles on the board. On the right there is a slot that lines up with the boards SD cards reader, allowing cards to be inserted and removed from the board whilst still in the case (this was never used). Above the SD card slot are the mounting points for two 3.5mm headphone sockets.

To aid ease of use and portability two 3.5mm headphone jacks and sockets were added for each of the PPG sensors. This allowed the sensors to be easily to be unplugged and plugged into the device. In hindsight these jacks were a poor design choice as the fiddly solder connection inside the jack was prone to failure and had to be fixed a number of times. The sockets are secured inside the device by pushing their shaft though the holes designed to

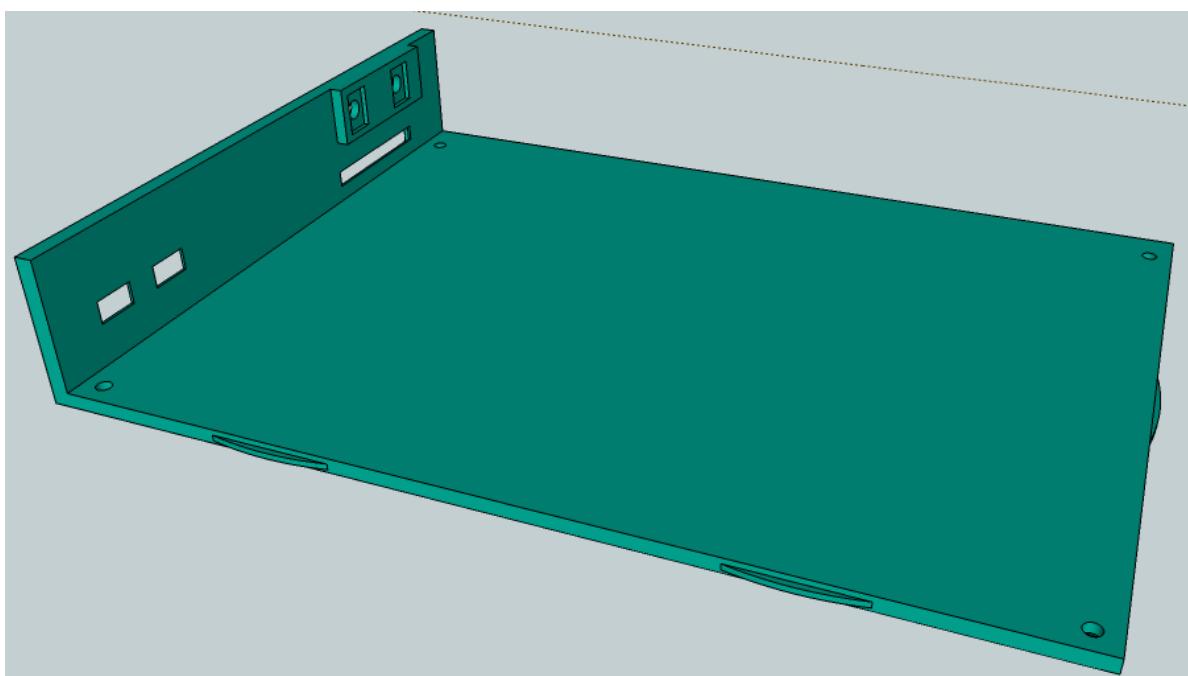


Figure 19: Base component of dual channel device case

accommodate them and screwing sheath on the outer side of the base. In order to create the socket mount point on the base the 3.5mm sockets themselves were also measured and modelled in SketchUp.

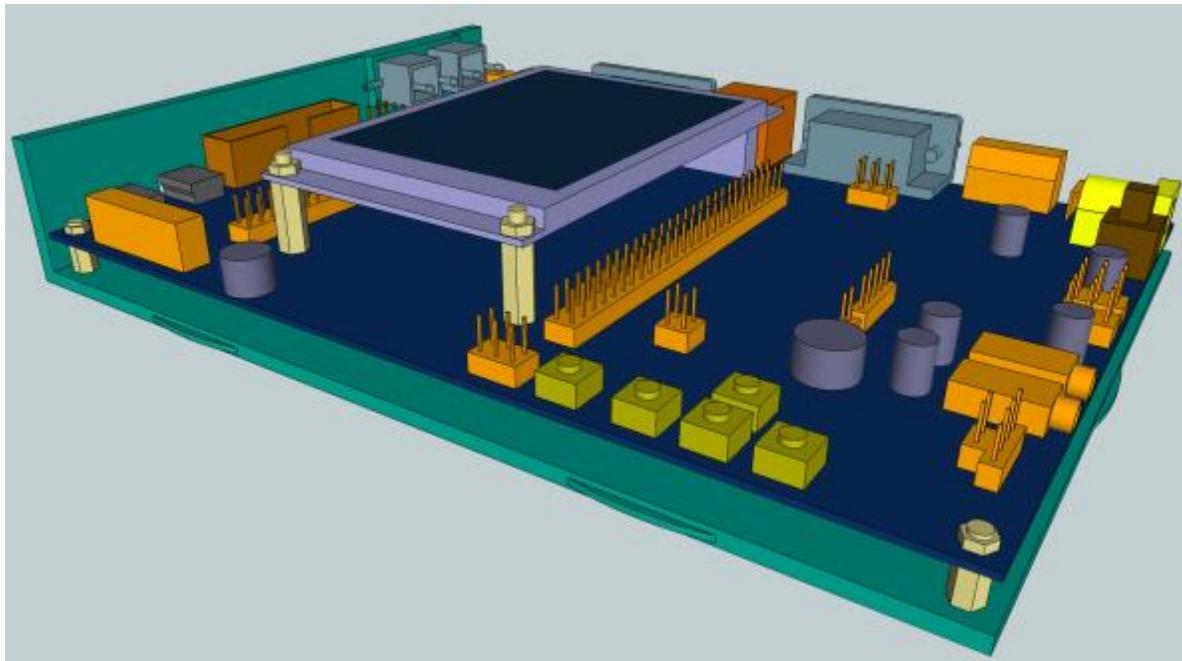


Figure 21: Model of Alientek board on case base

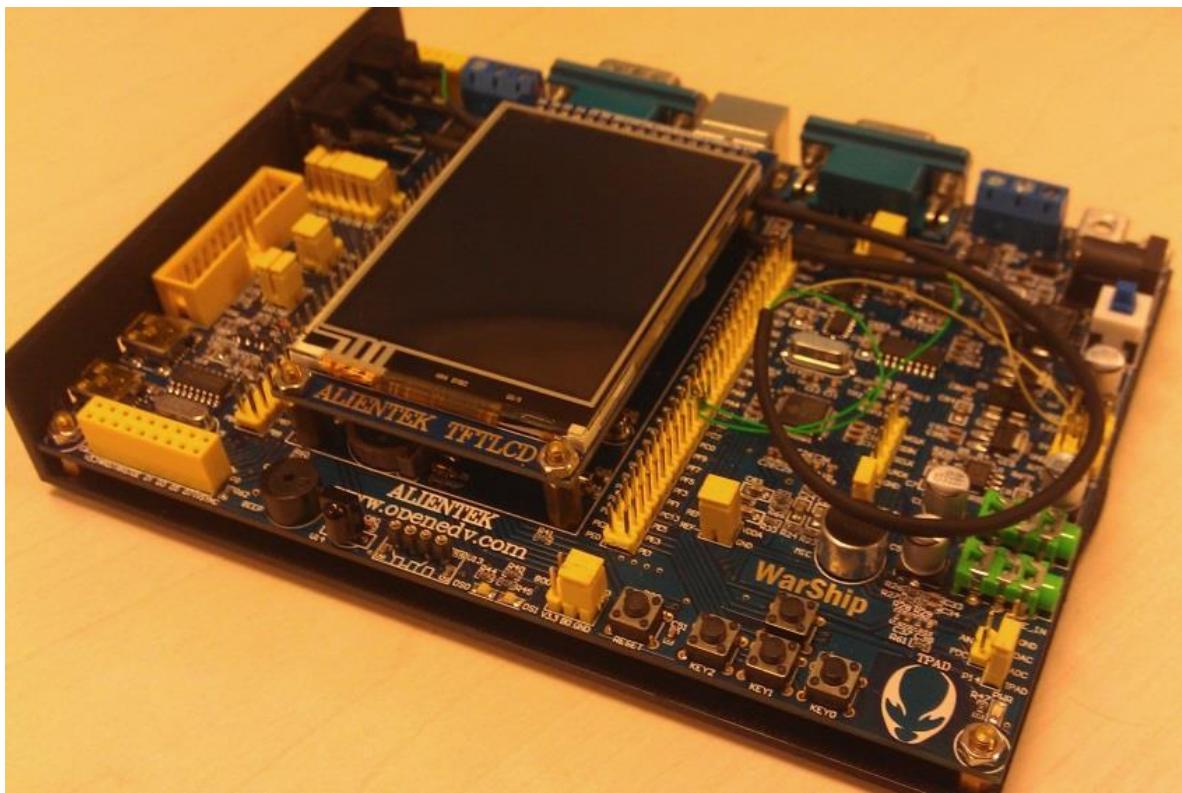


Figure 20: Alientek board on case base

When assembled and the case secured to the base the sockets each had the 3 connecting wires soldered to them and further secured using heat shrink insulation tube, which also prevented electrical shorts from occurring. These three wires were wire-wrapped to the appropriate ADC pin on the GPIO header as well as the power and ground pins. These wires can be seen in Figure 20 passing under the LCD screen inside the same insulation tube which was used in this instance for the purpose of cable management.

The model of the case's cover is shown in Figure 22 - Figure 24. In Figure 22 the rear of the case is shown where the USB ports, SD card slot and 3.5mm socket PPG sensor connectors are shown. The LCD screen can also be seen at the top of the case.

In Figure 24 the internal space where the case's base and board reside is shown. The cut outs for the rear connectors can be seen as well as the opening for the LCD which is surrounded by some indentation to accommodate the screen and some cut to size crafting foam used to create a seal between the cover and the LCD.

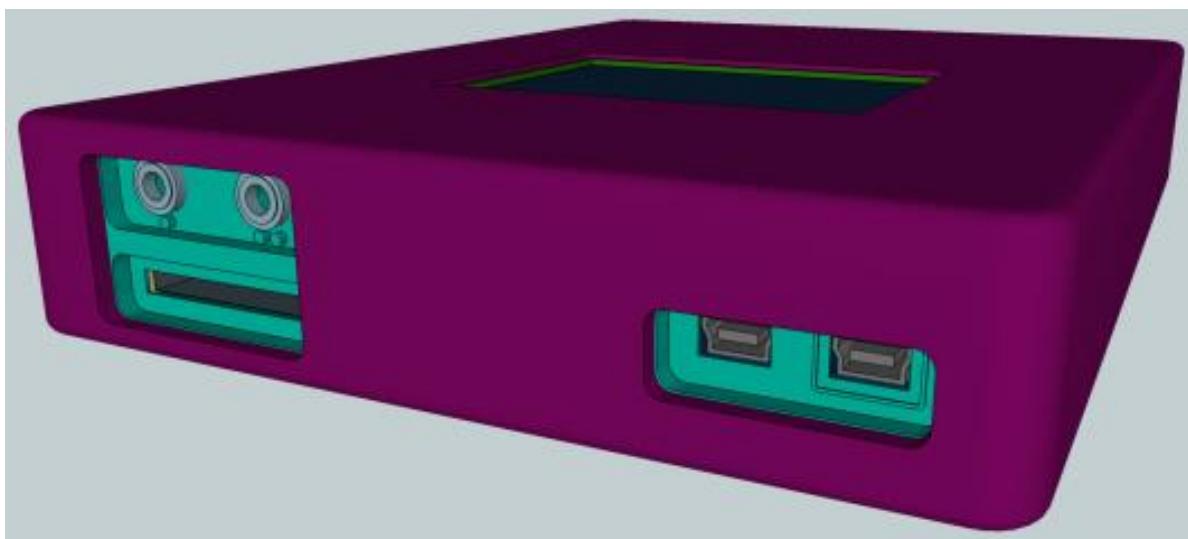


Figure 22: Rear view of model of case with cover attached

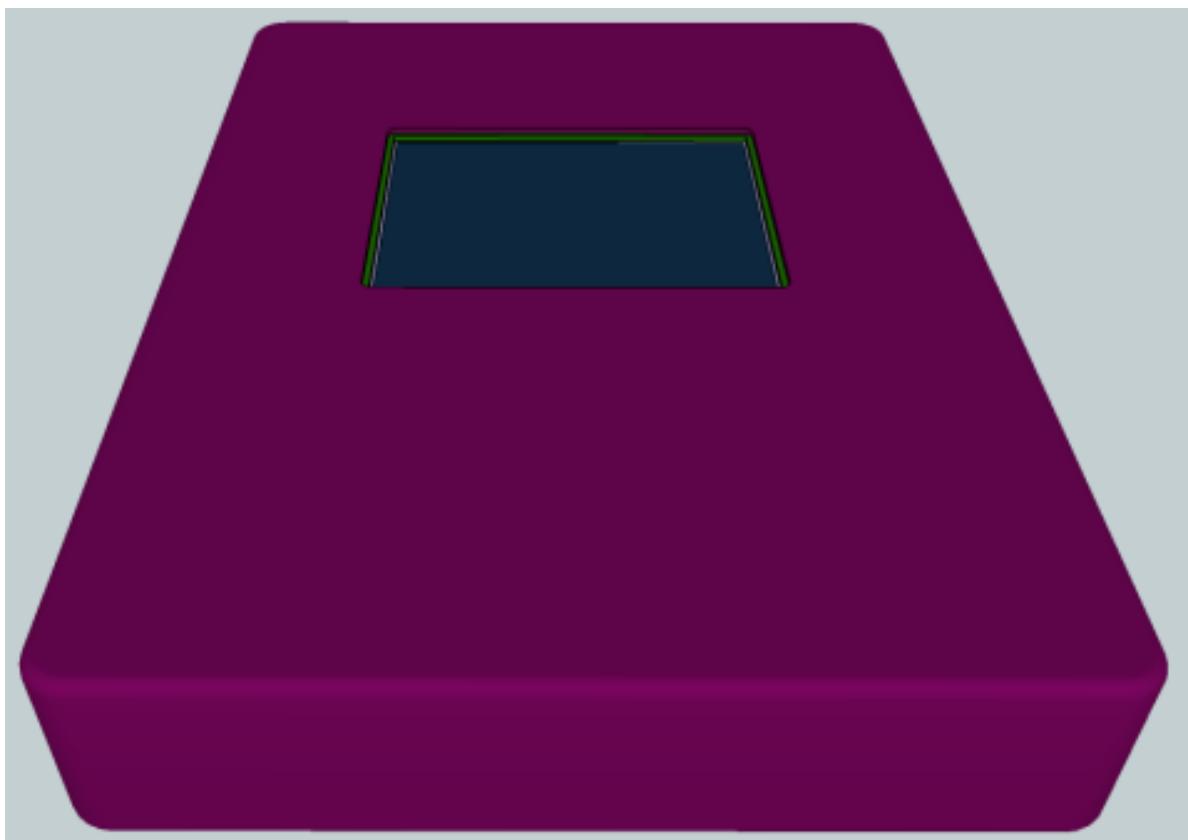


Figure 23: Top view model of case with cover

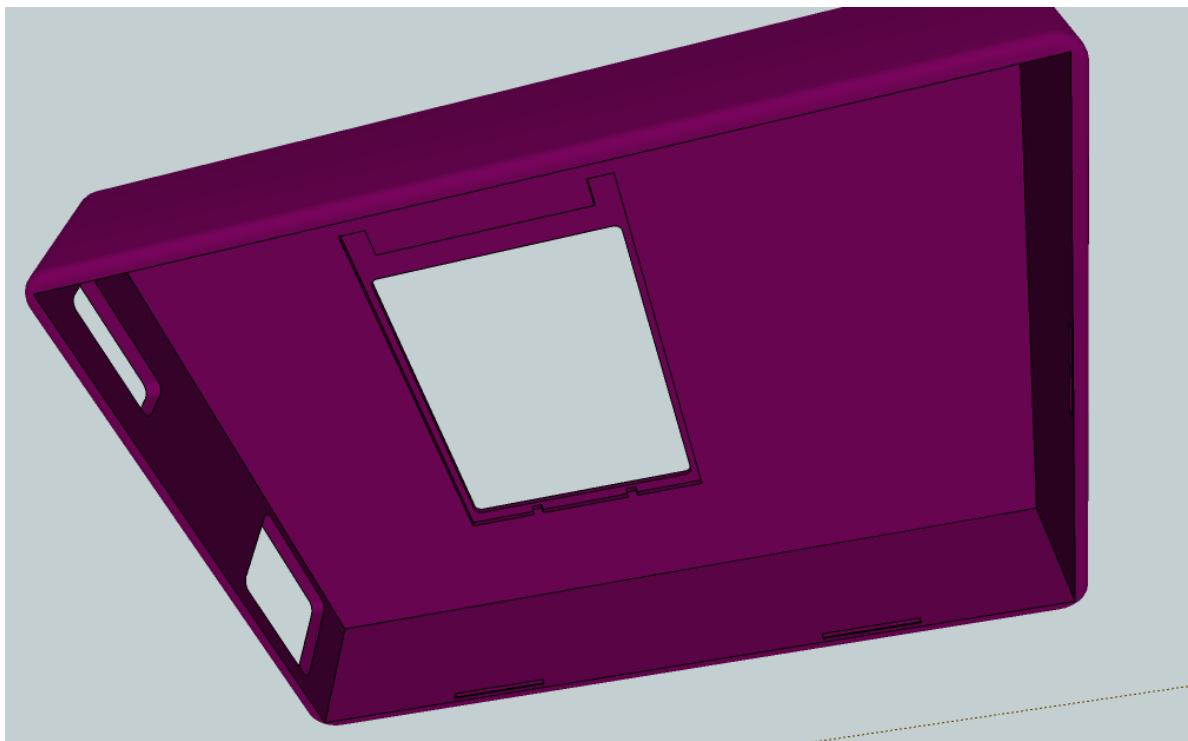


Figure 24: Internal view of model of case cover

A housing for a PPG sensor to be used on the finger was also modelled and 3D printed. This housing was designed to protect the electronics on the back of the sensor and to allow a Velcro strap to be attached to the sensor to allow for easy application and removal of the sensor.

The sensors snuggly fit into the indentation shown in Figure 25 with the 3 wire ribbon cable connected to the sensor sits in the gap created for it. In Figure 26 the hole where a Velcro strap passes directly through the housing can be seen. The strap can then be wrapped around the finger or other periphery holding the sensor against the skin.

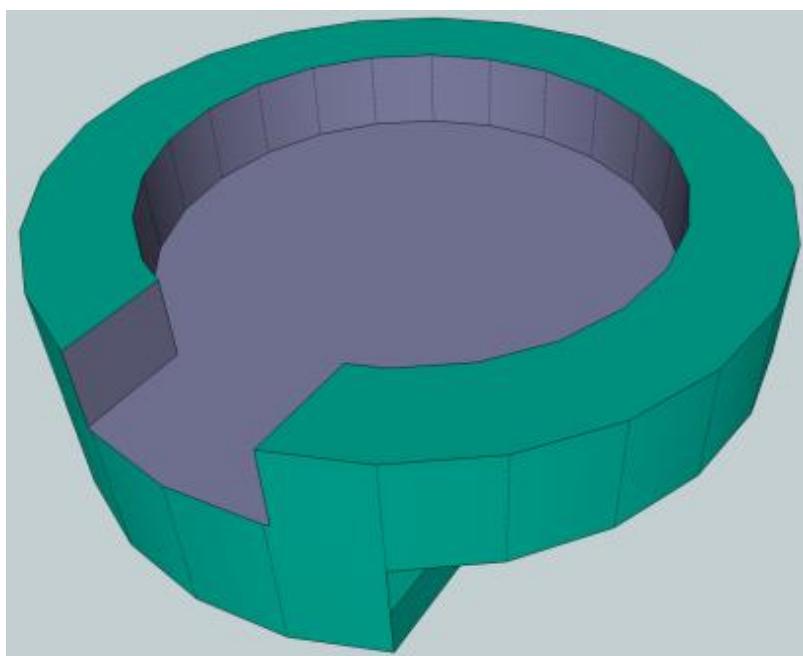


Figure 25: Underside of model of PPG sensor housing

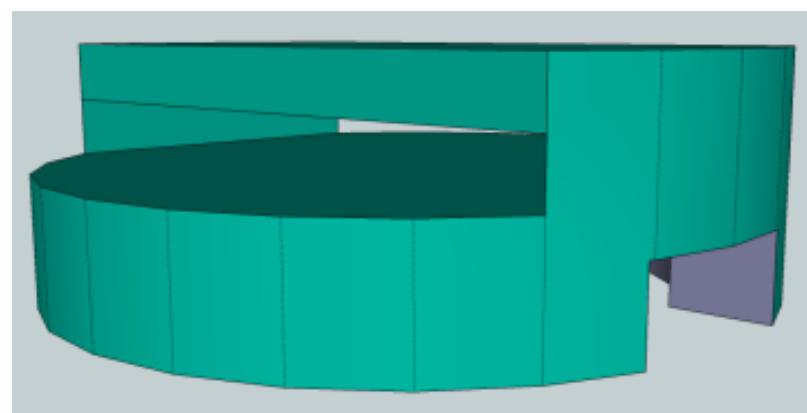


Figure 26: Side view of model of PPG sensor housing

Data Collection Tools

During the design phase of this research three methods of data collection were developed.

- MATLAB (*rudimentary*) data collection tool
- MATLAB GUI data collection tool
- Windows .Net GUI

The main function of all three tools was to read the PPG signals from the dual channel PPG device and record the subjects SBP, DBP and HR taken at the end of the test using a sphygmomanometer. The PPG and recorded sphygmomanometer readings were then stored together for later retrieval and processing.

MATLAB Data Collection tool

The purpose of this tool is to read in data from the dual channel PPG device and store all the PPG information. At the conclusion of each test the user was prompted to enter in the SBP, DBP and HR readings taking using a sphygmomanometer. This data could then be used to develop a model form measuring BP, continuously and non-invasively using two PPG signals.

The tool itself involves a number of scripts that were initially written in an older version of MATLAB but have since been converted to be used with, as well as take advantage of features in MATLAB 2017a.

The tool first established a serial connection with the device over USB connecting at a baud rate of 256000 bits per second. Once connected the dual channel PPG device collected PPG signals and converted the analogue signals to 12-bit digital values, ranging from 0 to 4095.

As both channels were sampled simultaneously this resulted in a total of 4 bytes of data being read at a rate of 1 kHz. The sampling and sending of data occurred continuously for a specified time of 15, 30, 60 or 120 seconds. Usually these tests were run for 30 seconds.

Once the sampling and sending of the PPG data had concluded the user was then prompted to take blood pressure and heart rate readings using the sphygmomanometer. The user was then asked to enter these readings, SBP, DBP and HR, in that order.

The script then processed some of the PPG data. Using a filter created in MATLAB's filter designer the PPG signals were filtered to remove noise which was present on the signals.. The filter was a low pass frequency impulse response (FIR) filter with a 9600 Hz pass band and a 12 kHz stop band. This smoothed the signal to enable further processing by filtering out power line interference on the signal [24] . Next the heart rate was calculated using peak detection to detect the systolic peaks on both signals. The number of systolic peaks per minute corresponds to the heart rate.

There were several different iterations of the MATLAB data collection tool. Initially a MATLAB script was written that read in data from the 2 channel PPG device and stored the raw PPG data for processing at a later time without any modification pre-processing of the data. Associated SBP, DBP and HR reading were not recorded or stored with the PPG data.

Later versions of the tool prompted for the entry of the subject's SBP, DBP and HR taken after the PPG readings using a sphygmomanometer. The PPG data and the BP readings were collated by the script and saved together in a file for later use. Some rudimentary processing of the data was done also done at this stage and the PPG signals were displayed on the computer screen. This was done to ensure the test had executed correctly and if not the test could be immediately discarded.

All modifications done to the data collection tool were made carefully so as to preserve the use of data that had already been collected. The data collected was not changed it was just dealt with and stored in a more appropriate manner making it easier to deal with in future.

Other modifications made to the collection tool ensured that it functioned correctly and was more user-friendly. Early versions crashed under certain circumstances for various reasons. The causes of these crashes were isolated and removed over the course of the development of the tool.

MATLAB GUI data collection tool

This was developed as an enhancement to the earlier developed MATLAB collection tool. Rather than requiring the MATLAB environment to run in this tool was written in MATLAB but packaged to run a stand-alone program.

This tool also uploaded the data to a specified web server rather than just storing the data locally on the PC it was running on.

As it was based on earlier version of the processing tool there were however problems with it. If the tool encountered issues processing the data it crashed without keeping any of the results.

This tool was what was used by collaborators at the Second Jiaxing Hospital of Jiaxing Municipal in China.

Windows .Net GUI

This tool was written in Microsoft Visual Studio and is a .Net Windows based GUI alternative to the MATLAB data collection tool. It was written mainly to facilitate data collection by our counterparts at the Jiaxing Rehabilitation Hospital in China, but it never ended up being used by them.

This software not only collected data, but it also performed some of the functions found in the analysis tool which was still concurrently under development itself at the time. It performed some of the rudimentary analysis of the PPG signals was able to display the PPG signals in real time within the application.

The application also had added functionality of being able to detect problems with PPG placement or movement of the sensors and alert the operator drawing their attention to this problem.

As this program was paid for, owned by and produced for our Chinese affiliates the user interface for the application is available in both English and Chinese. Part of the interface can be seen in Figure 27.

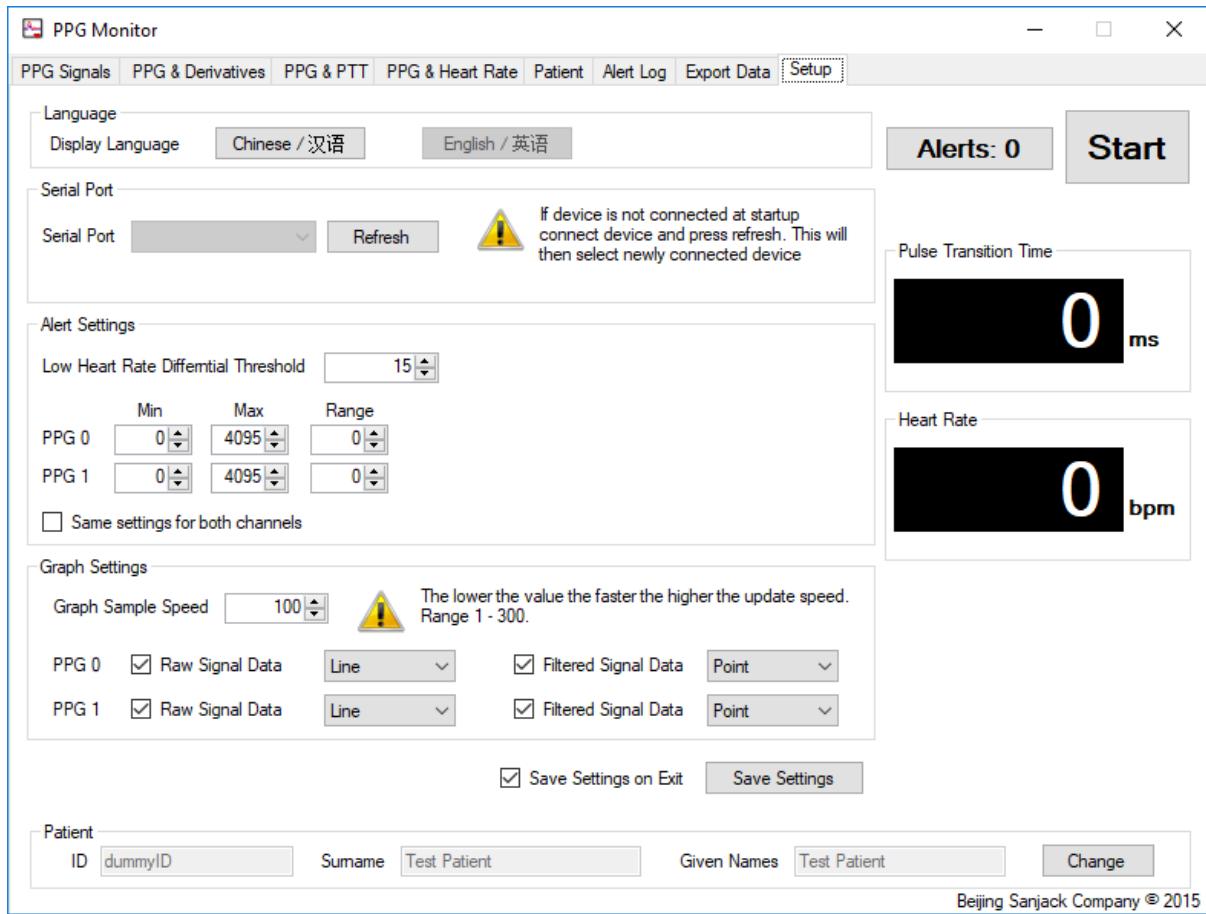


Figure 27: Windows .Net GUI

One of the main aims of this program was to remove the reliance on using MATLAB to collect the data from the dual channel PPG device. To that end this program allowed the PPG data to be exported as plain text CSV files or in a Microsoft Excel spread sheet. This allowed for the data to be analysed within Excel or any other program that could open either of the two file formats, which is a major departure from the other MATLAB based programs.

Data Collection

Initial Tests

The main aim of these tests was to develop and evaluate the effectiveness of the dual channel device hardware and the PC software needed to operate it and to collect the data from the device. At this stage a simple data collection tool was created in MATLAB. This tool established serial communications with the dual channel device, collated and stored the dual channel PPG data and saved this data into a two .m files.

Reading from the Omron were also taken during these tests but were stored in a separate spread sheet. It was not until later revisions of the MATLAB tools that both channels of PPG data as well as the SBP, DBP and HR readings for each test were all stored together in a single file.

These initial tests were conducted on the two research partners conducting this research; initially this research was part of a group project and this researcher had a research partner who was working on the project for the first year. Any of the data that was collected using the research partner as a test subject has not been used in the research presented in this thesis. However, since the data collected on the author of this paper included all the necessary information – the dual channel PPG data and SBP, DBP and HR readings – but just in a different format, the author's initial test data has been able to be used.

It was during this phase of initial testing that the protocol that governed the use of the dual channel device was first established. This protocol is discussed in Dual Channel Device Usage Protocol section of the previous chapter.

Circadian Experiments

The circadian experiments were initially designed to test the efficacy of the dual channel PPG device and to see if the relationship between BP and the time of day could be confirmed and if this circadian rhythm is evident on the PPG readings. These experiments were performed based on the findings of previous research that in most individuals over the course of a day their SBP and DBP varies displaying a circadian rhythm [60, 61].

These experiments were run using one human subject, the author of this thesis, as a test subject over the course of a week. Over this week long experiment a total of 245 tests were run resulting in over 2 hours of dual channel PPG data being recorded. This was the main long-term outcome of this experiment.

These tests were conducted using the dual channel PPG device, the MATLAB data collection tool and an Omron HEM-6221 was used to obtain SBP, DBP and HR. The HEM-6221 is an automatic, wrist worn oscillometric sphygmomanometer. This device is shown in Figure 28 [62].

The PPG and BP data were collected every two hours between 6am and 12am over the course of a week. During each 2 hour time slot the PPG data and BP readings were taken multiple times and recorded using the data collection tool. The number of times a test was run per slot varied, but there was always a minimum of three. The first of which were run at the start of the hour and subsequent tests after a delay of around one minute from the preceding test.



Figure 28: Omron HEM-6221 [62]

Each test involved the subject (the author) in the seated position and connected to the dual channel device, with the two PPG sensors placed on the left ear lobe and left index finger as discussed in Sensor Positions Using Dual Channel Device section of Chapter 3.

PPG data was collected for 30 seconds per test and was immediately followed by BP readings taken using the Omron sphygmomanometer which were consequently recorded and stored using the data collection tool. This was all done according to the same experimental protocol discussed in the previous chapter.

The data was collected using an older version of the data collection tool. Later this data was converted to match a newly devised format that consolidated the data into a single structure and did some pre-processing to make analysis of the data faster.

In terms of the original aim for this circadian experiment, which was to investigate a correlation between BP and the time of day these experiments were a failure. As shown by the small selection of data in Table 5 on the following page and by the error bars used to indicate the range of values recorded in each experiment time slot in Figure 29 (on page 63) the recorded SBP, DBP and HR values varied wildly within a matter of minutes.

Over the course of the week-long experiment, even if the wide error bars are discounted, there is no evident correlation between any of the BP readings and the time of day for the subject (this researcher). This can be seen in the data shown in Figure 30 where each day's worth of data is shown in a different colour. It also shows that for this researcher BP did not follow the circadian rhythm found in most subjects of previous research undertaken by others.

Table 5: Example SBP, DBP and HR data collected during Circadian Experiments.

Slot	Time	SBP	DBP	HR
6am	6.02	145	92	71
	6.05	159	106	62
	6.11	165	108	60
	6.14	145	105	58
8am	8.04	158	107	59
	8.08	149	99	57
	8.12	161	104	57
10am	10.05	143	105	65
	10.09	143	99	58
	10.17	148	106	57
12pm	12.03	144	99	62
	12.06	138	95	60
	12.10	132	92	64
2pm	14.10	147	98	52
	14.13	147	105	60
	14.21	151	106	57

Slot	Time	SBP	DBP	HR
4pm	16.06	152	103	65
	16.10	148	104	61
	16.13	145	102	60
	16.16	143	98	58
6pm	18.13	159	109	59
	18.16	151	103	57
	18.19	148	103	58
	18.21	145	108	59
8pm	20.02	152	102	62
	20.04	147	100	63
	20.07	141	99	63
10pm	22.01	165	110	61
	22.05	161	103	62
	22.08	154	105	63
	22.12	154	104	67
12am	00.01	149	95	51
	00.04	149	97	53
	00.07	147	99	54
	00.12	164	106	53

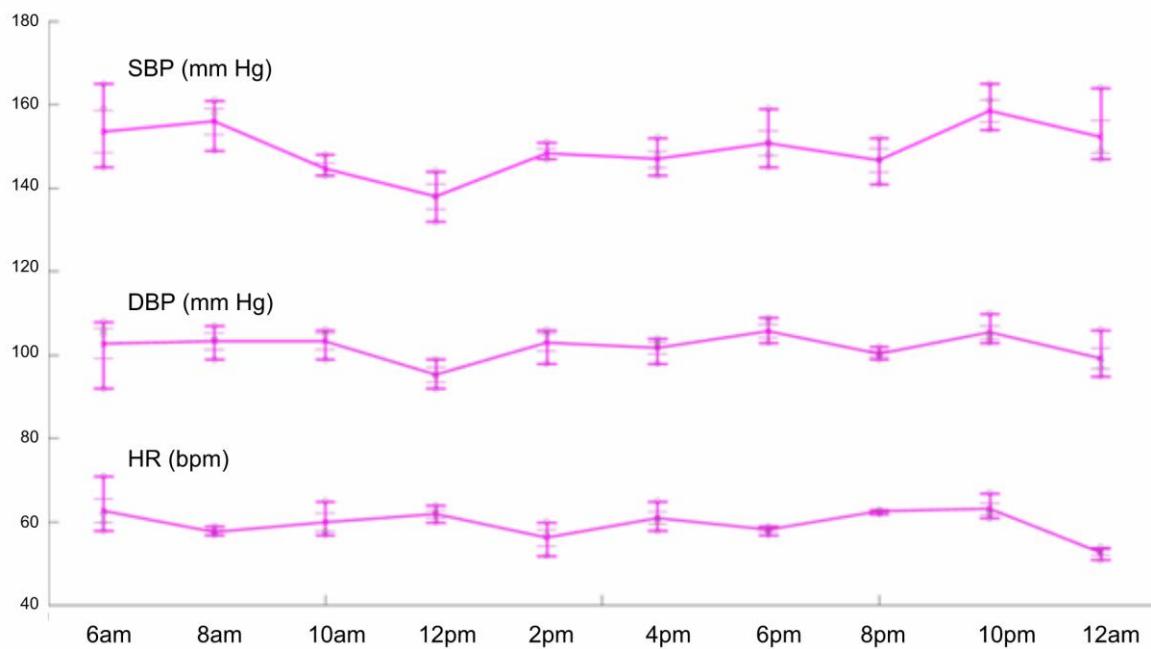


Figure 29: SBP, DBP and HR Omron data from day 4 of Circadian Experiments

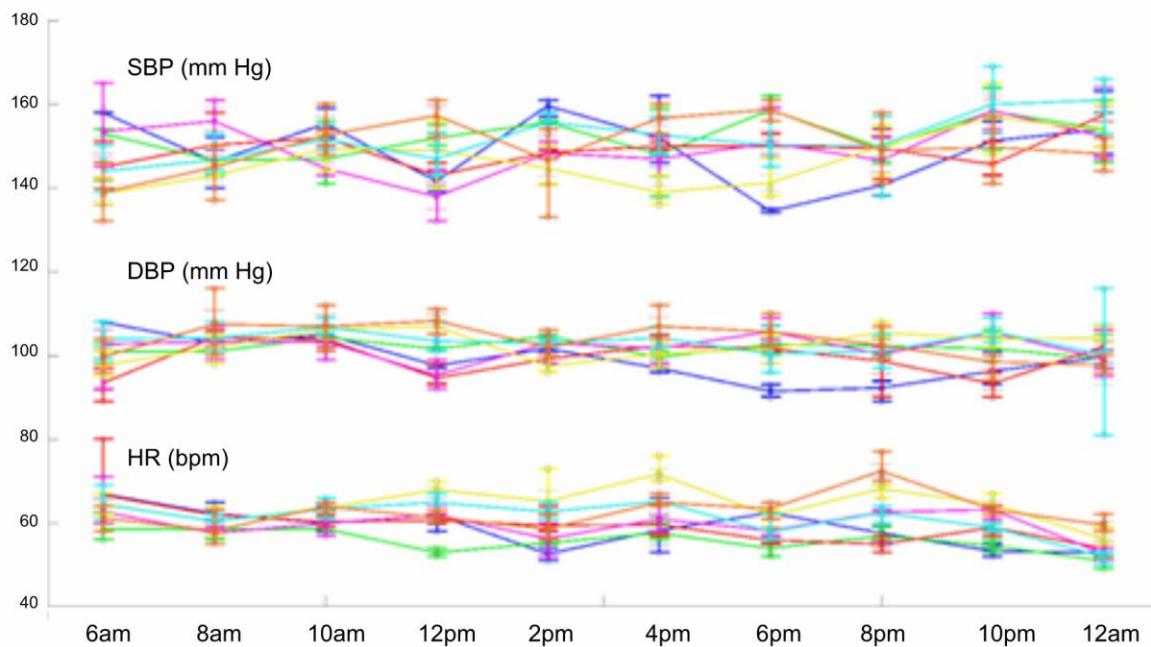


Figure 30: SBP, DBP and HR Omron Data from all days of Circadian Experiments

n.b. Each day on the above graph is represented by a different colour.

Collaborators Data Collection

This research began as collaboration with the Beijing Sanjack Ltd, China. The dual channel PPG device and the associated software is the result of this collaboration and was created to be used as a part of research conducted for them by researchers at the Second Hospital of Jiaxing Municipal in China.

The researchers in China conducted their own experiments, without any guidance or input from this researcher. At a later date access was granted to the data that was collected at the Hospital. This came in the form of two different experiments which were run over two stages.

All the data collected at the Second Hospital of Jiaxing Municipal was done after approval from the hospitals ethics committee. All the patients whom were subjects of the hospitals research also completed a certified consent for allowing for the data collected to be used in any future research within or outside the hospital. A letter of authorisation from the hospital was obtained to enable the use of the data the collected there in this research.

Stage 1

During this stage of their research the Chinese researchers conducted tests on 4 patients of the rehabilitation hospital. In these experiments each patient was placed in the supine position on an initially flat tilt bed. Over the course of each experiment the tilt bed was slowly tilted to varying degrees up to a maximum inclination of about 80 degrees.

The idea behind this tilting action was that it was designed to induce a change in BP which was to be detected by the monitoring of the PPG signals.

Unfortunately, this data was mostly unusable for this research as the SBP and DBP sphygmomanometer readings were only recorded once per patient as opposed to once per

test. To investigate the correlation between BP and PPG, SBP and DBP readings are required per test so they can be the PPG data can be better modelled to these readings.

It was possible to use the collected dual channel PPG data as a means of improving the analysis algorithm's ability to detect the appropriate events occurring on the PPG signals. This provided limited value to the research overall however.

The data collected in this stage of their research was collected using a very early version of the data collection tool. This tool stored the PPG data in two separate files and did not include the sphygmomanometer data together with this information. Due to this data being separate it was not until after much time was spent working with this data that it was realised all the necessary information had not been collected.

Stage 2

The second stage of experiments conducted by the researchers/doctors the Second Hospital of Jiaxing Municipal hospital differed in several ways from the experiments originally conducted by them.

The second stage of experiments conducted at the Second Hospital of Jiaxing Municipal involved many more patients than in their first stage. During this stage there were a total of 21 patients used as the subjects in the experiment. These patients all signed consent for their medical data to be collected and used for research purposes. This was all approved through the ethics processes of the hospital, who also granted access to use the data in this research.

Many of the patients were used as subjects of the tests multiple times, some of them only once. It varied from patient to patient. Some patients participated in the tests over the course of a few days, others in rapid succession one test immediately following another.

During each test, unlike in the hospitals earlier experiment, the patients all remained in a seated and still position throughout the duration of the test.

As before the tests were conducted using a dual channel PPG device. However, the software that ran on the PC it was connected to had been updated. This updated software now had an easy to use GUI interface making it much more user-friendly. However, the main improvement in functionality from the point of view of this research was that the data collected was now automatically uploaded to a web server. The uploaded data included both the dual channel PPG raw data as well as SBP, DBP and HR measurements. These readings were taken by a doctor or nurse at the hospital using an oscillometric sphygmomanometer at the conclusion of each test. As recording this information was required in order to proceed to the conclusion and uploading of the data all tests had this information included. This resulted in BP information existing for each set of PPG data which was necessary for this research.

There were a few notable problems with the collection of this data. The main being a problem of data entry of the SBP, DBP and HR reading taken using the sphygmomanometer by doctor or nurse. The entry of this data was not always accurate. For example, the one test had a recorded SBP of 1212 mmHg, which could not possibly be correct.

Data Consolidation – Common Data Format

The data collected during the initial and circadian experiments as well as both stages of the Chinese experiments were all collected using different versions of the data collection tool. While these tools all performed the same task of connecting to the dual channel device and recording the PPG data sent from it, they all worked differently and more importantly saved the data in differing formats.

Initially data collected was manually saved by the operator and in doing so the operator chose the filename. All the variables in the MATLAB workspace were saved along with the raw PPG data. Much of the information stored in this file was relevant to being able to analyse the data, but some of it was not.

By the time the circadian tests had been completed the data collection tool had progressed to automatically saving the relevant data into file. The file created contained all the information relevant to the test. This included the SBP, DBP and HR readings taken using the sphygmomanometer at the conclusion of the test and entered by the operator, as well as a patient identifier used to group tests covering the same patient together, information about the test itself like the length that it ran for, and the PPG data itself. Having all the data in the one file was a vast improvement in terms of being able to analyse the data.

The results for Stage 1 and Stage 2 of the Jinxing Rehabilitation Hospital experiments were also in a different format. Stage one had files for each PPG signal for each test and a spread sheet, in Mandarin, containing information about the test. As mentioned in the Collaborators Data Collection section of Chapter 4 these results turned out to be unusable for this research as each test was not accompanied by their own SBP, DBP and HR information.

During the Stage 2 experiment at the Second Hospital of Jiaxing Municipal the data collected was automatically uploaded to a server by the updated data collection tool. The uploaded information included the PPG data, each stored in a separate file, but also the additional information about the test, including the sphygmomanometer measurements, were recorded.

Processing and analysing all the data from the disparate experiments proved to be difficult without having a uniform data set to work with. As such a common data format was created where all the relevant information experiments could be stored in a common file that could be easily process by the analysis tool in MATLAB.

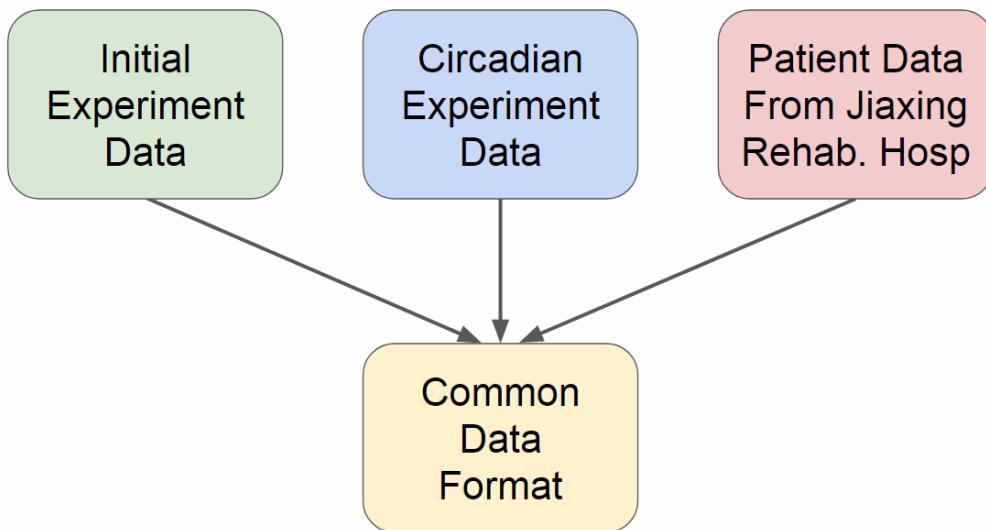


Figure 31: Conversion to common data format

In creating the common log format, it was decided to allow all the tests from a particular experiment group should be stored in the same file, which was a table in MATLAB. Each row of this table would reflect the results for a particular test. In creating the common log file it was decided to alleviate some of the processing that would always occur as part of the analysis of the test, such as filtering using the low pass FIR filter to filter out any power line interference [24] present in the PPG signals. Also the first and second derivative (often referred to as acceleration plethysmogram (APG) in literature [23, 24, 48, 63]) of each of the filtered PPG signals was calculated. All this signal information, including the raw, unprocessed PPG signals, was included as part of the log along with timestamp, patient id and other information about the test.

To accommodate the creation of the common log format conversion tools were written in MATLAB script that converted each of the disparate datasets from three experiments into the common log format.

Refer to Table 6 on the following page for the fields that make up the common log format.

Table 6: Common Log Format

Field	Usage
test Id	An ID that can be used to refer to the test within a larger data pool
valid	Set during analysis this indicated if the test data valid and should it be included in the overall results; true or false
comment	Comment about the data set during analysis
timestamp	The date and time the test was conducted.
userID	An identifier to uniquely identify patients so tests using the same subject could be associated with each other.
armLength	The measured length of the arm of the test's subject
recSBP	SBP reading taken using sphygmomanometer at the conclusion of the test and recorded by operator.
recDBP	DBP reading taken using sphygmomanometer at the conclusion of the test and recorded by operator.
recHR	HR reading taken using sphygmomanometer at the conclusion of the test and recorded by operator.
sampleTime	The duration of the test; i.e. how long PPG data was collected (in seconds)
PpgData	This is a structure array, one structure for each PPG channel containing the raw, filtered and 1 st and 2 nd derivative signal data.
PpgMeta	This is a second structure array, one element per channel. Each structure contains meta data about the PPG signals and it created during analysis.

Algorithm Development

In traditional PTT calculations the time is measured between detection of the R-Peak of the ECG until the corresponding systolic wave is observed on the PPG signal. Different approaches have used differing methods of detecting this wave, but often the systolic peak itself or its mid-point are used.

Four different ways to detect and define the start of each heart beat using a single PPG signal were used in this research. This was done using recurring and detectable *events* that occurred as part of a regular PPG signal. Each *event* occurred on or around the systolic up stroke so as to maintain a relationship between the detection point and the R-Peak of an ECG used in traditional PTT calculations.

The four events, or points, in the PPG signal as seen in Figure 32 were as follows (in order)

- Pre-systolic up stroke (PSUS)
- Start of the systolic up stroke (SSUS)
- Mid systolic up stroke (MSUS)
- Systolic peak (Peak)

These four points where derived from either the first or second derivative of the PPG signals and involved determining the zero crossing points within these signals.

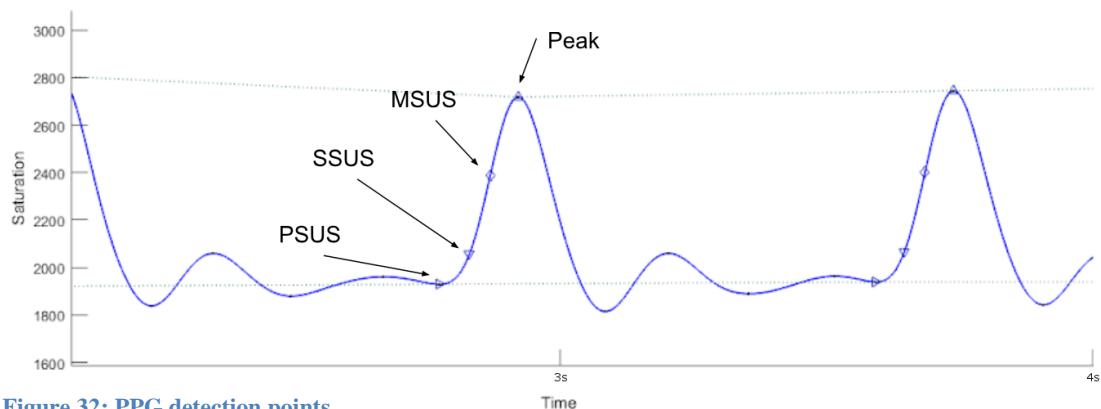


Figure 32: PPG detection points

PEAK – Systolic peak

The systolic peak detection is the easiest to visualise on the PPG signal as shown in Figure 32. It occurs as maximum blood flow beneath the sensor is detected for the corresponding heartbeat. The peak can be detected using two methods; finding the local maximum and using the zero crossing of the first derivative.

In the end the peak was determined by using the zero-crossing point in the first derivatives and the known locations of the MSUS point described below; each peak occurs at the zero-crossing point of the first derivative that occurs after a the MSUS of each heartbeat.

MSUS – Mid-systolic up stroke

The MSUS, or mid-systolic up stroke, occurs during at the midpoint in the rising edge of the systolic upstroke. This is detected by the algorithm developed for this research using the local maximum of the first derivative. In detecting the MSUS the local maximum is detected by finding the peaks in the first derivative that at least 50% of the value of the maximum of the first derivative.

SSUS – Start of the systolic up stroke

The SSUS, or start of the systolic upstroke, occurs prior to the MSUS of the same heart beat cycle. The MSUS point is used in the detection of the SUSS, so MSUS must be calculated first. SSUS occurs at the peak in the second derivative of the PPG signal that occurs just prior to the MSUS.

PSUS – Pre-systolic up stroke

PSUS, or pre-systolic up stoke is the first detectable point in the heart beat cycle. It is probable, though unproven, that it occurs at pre-ejection point of the heartbeat. Like the other 3 detection points the detection of PSUS relies on MSUS for detection. It is detected by determining the zero crossing points in the second derivative of which there are many. PSUS has been defined and detected as the first zero crossing point of the second derivative that occurs just prior to a MSUS point.

Detection of PPG Signal Points

As part of the analysis algorithm all the PPG signal points for both PPG signals (ear and finger) are detected. The process of detecting these points is outlined above and in Figure 33 below. Refer to the source of **findPpgPoints.m** in the Appendix for the complete source code for this algorithm. Note that like all of the analysis this does not happen in real time and analyses a complete test from start to finish. However, there is no reason that this process could not be applied to small windowed segments of PPG signal at a time and thus allow for real time processing, albeit with a small delay determined by the window size.

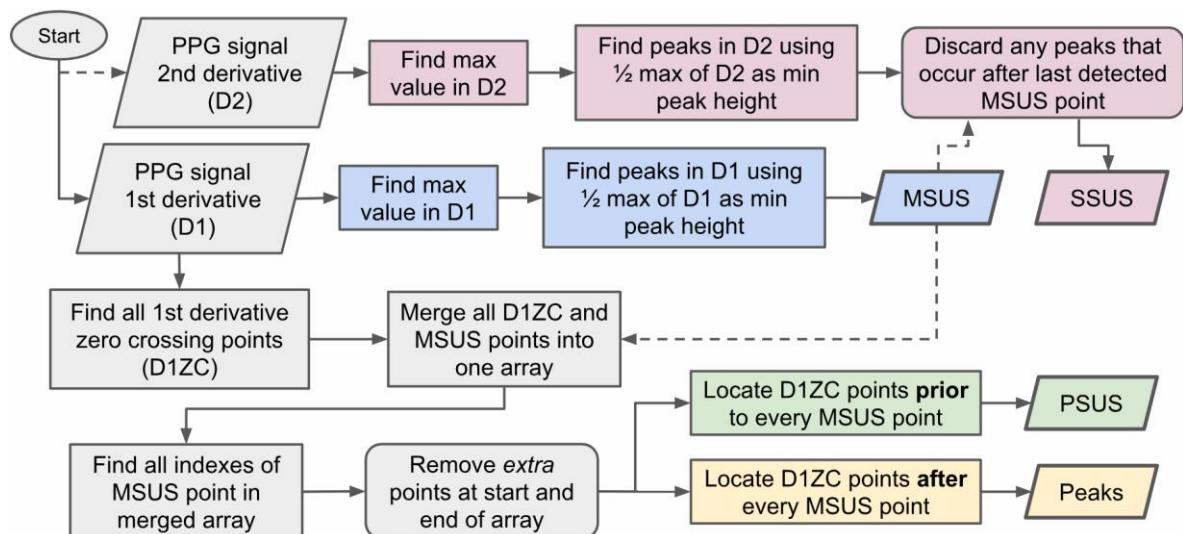


Figure 33: Flow chart of method for PPG signal point detection

Analysis Algorithm Output

Overall the most amount of time spent on this research was spent developing the analysis algorithm, and in particular locating the desired points in the PPG signals as described in the previous section. As previously mentioned this algorithm was developed in MATLAB and written in MATLAB script. All of the source code can be found in the Appendices.

The role of the analysis algorithm was to parse the PPG data from each test, analyse it and locate the 4 points of interest mentioned in the previous section for each complete heart beat cycle present on both PPG signals from the ear and finger.

This process was both manual and automatic. The MATLAB scripts produced output such as in Figure 35 on the following page, a segment of which is below in Figure 34 . This output was visually inspected and verified. This inspection, or inspection of multiple outputs, may have resulted in refinements being made to the algorithm to improve its detection of beat-to-beat points. This process was also used to determine the tests where there were problems with the data which made them unusable or there were too many irregularities for the test to be processed with the analysis program.

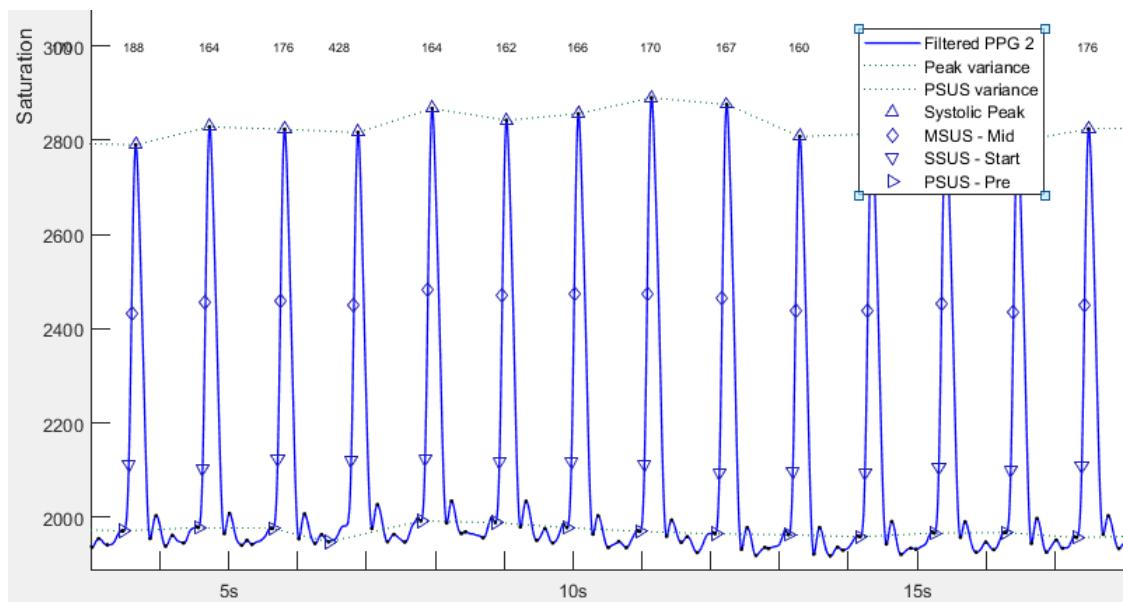


Figure 34: Segment of analysis algorithm output

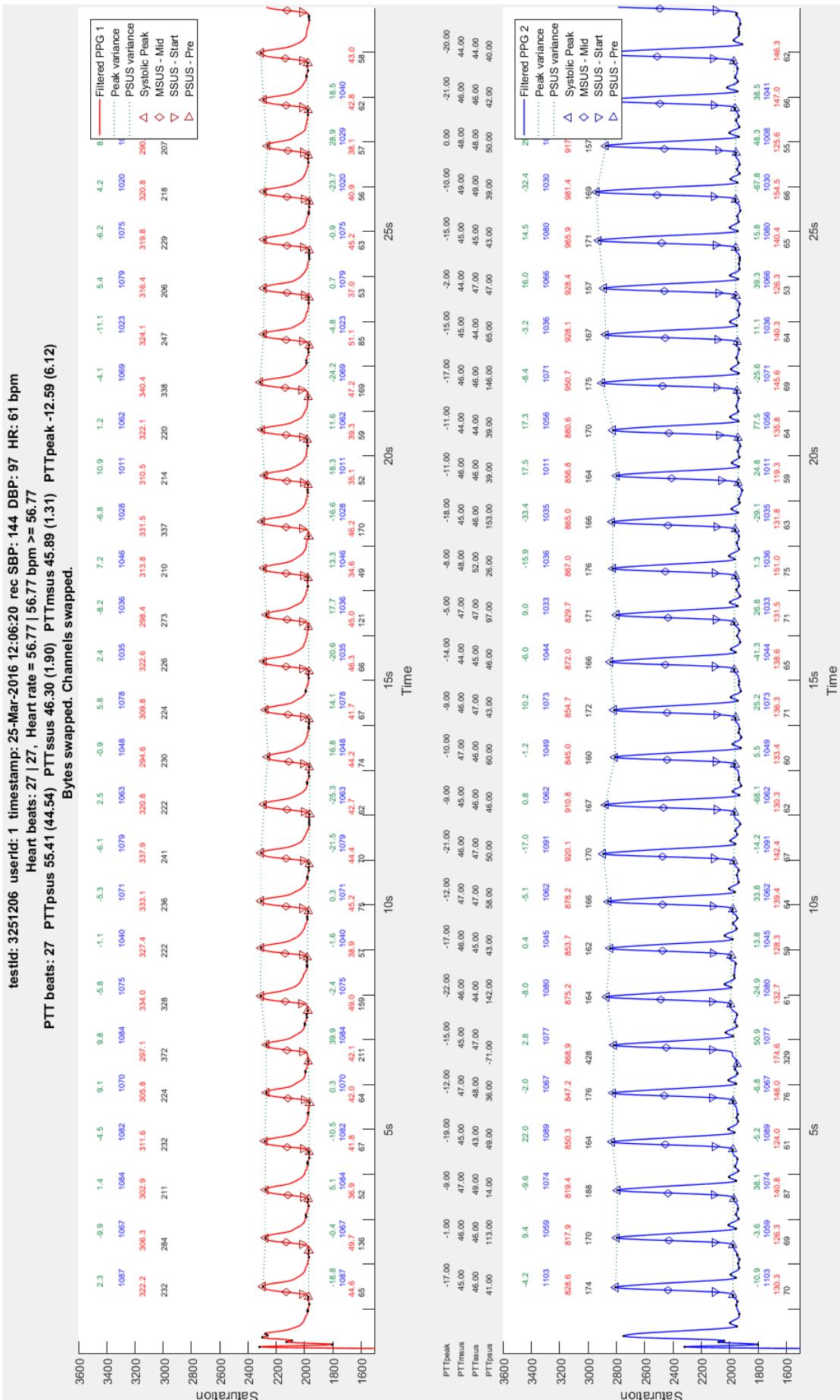


Figure 35: Analysis algorithm output

Analysis Data

The analysis algorithm extracted a large amount of data from the PPG signals in each test. All this data was added to the test's data stored in the common log format. Adding to the existing information was done in such a way that the same log file could be again used as input to the analysis algorithm. This was done in order to be able to make refinements to the analysis algorithm and still be able to reuse the same data where necessary.

There were two categories of information generated from the analysis. There was data which related to each specific PPG signal from the test and the data that related to both. The data pertaining to each PPG signal was stored in the *PPG Meta* structure outlined in Table 6.

The *PPG Meta* data stored per PPG channel in for each test's entry in the common log format .mat files is shown below in Table 7. Many of the fields in this structure were inspired by

Table 7: PPG Meta data structure

Field	Meaning / use
psus	The beat-to-beat PSUS points – stored as sample index of occurrence (time)
suss	The beat-to-beat SSUS points
msus	The beat-to-beat MSUS points
peaks	The beat-to-beat peak points
d1zc	Zero crossing points of the PPG signal's first derivative
beats	Number of detected full heart beat cycles
bpm	Heart rate (beats per minute) for test using beats (above) & test's sample time
systolicTime	The beat-to-beat systolic time in ms. (time from PSUS to Peak)
p2pTime	The time in ms (samples) between each peak on a beat-to-beat basis
peakVariance	Difference in PPG saturation level between one peak to the next (beat-to-beat)
sysInitTime	Time between PSUS and SSUS points (beat-to-beat)
sysInitAmplitude	Difference in PPG saturation level between PSUS and SSUS (beat-to-beat)
ssus2ssusTime	Time in ms (samples) between each SSUS point (beat-to-beat)
ssusVariance	Difference in PPG saturation level between successive SSUS points (b-to-b)

similar items measured by Birrenkott et al [64] in their paper evaluating measurements used in analysing ECG and PPG signals.

The PPG Meta data can also be seen displayed along with the signals in the analysis algorithm output in Figure 35 on the page 74.

Along with the metadata derived from each PPG signal more information was obtained from combining this metadata together. In terms of this research was the PTT and PWV calculations based on the four different methodologies (PSUS, SSUS, MSUS and Peak). PTT here does not refer to the traditional definition of PTT, but rather a revised definition where the ECG R-peak is replaced with timings from the PPG ear signal. PWV is also not the standard definition from other research but is also a modified version based on the procedure outlined in Sensor Positions Using Dual Channel Device section in Chapter 3. This PWV was calculated using the measured PTT and the subject's arm length as an analogous measurement to that of the distance between the two PPG sensors.

Not only were the modified versions of PTT and PWV were calculated using each of the four methodologies on a beat-to-beat basis a mean was calculated for each as well as the standard deviation. For example, see Table 8 below for all the data calculated and stored for PSUS methodology. All these were replicated using the other three methodologies; SSUS, MSUS and Peak.

Table 8: PTT/PWV calculations examples

Field	Meaning / use
pttPsus	Beat-to-beat PTT calculations using PSUS
pttPsusMean	Mean of all of the PTT_{PSUS} calculations
pttPsusSD	Standard deviation of all the PTT_{PSUS} calculations
pwvPsus	Beat-to-beat PWV calculations using PTT_{SUS} and arm length
pwvPsusMean	Mean of all the PWV_{PSUS} calculations
pwvPsusSD	Standard deviation of all the PWV_{PSUS} calculations

Finally, the means of the following metadata was also taken from the PPG finger signal were also calculated and added to the log; systolicTime, p2pTime, peakVariance, sysInitTime, sysInitAmplitude, ssus2ssusTime and ssusVariance.

Heart Rate Validation

There is a good reason for calculating and including different variants of obtaining the heart rate form the test. Heart rate is a relatively simple thing to obtain from a PPG signal. Visually this can be done just by counting the peaks in the PPG signal and dividing by the time period being measure before converting to beats per minute (BPM).

As the number of beats can easily be calculated and visually verified it was used as a means of verification of the analysis algorithm's functionality. This was particularly handy as it can be used with each of the four PPG PTT/PWV methodologies.

Differences in the number beats calculated per channel were used as an indication of potential problem with the analysis of the PPG signals which was particularly helpful when refining the analysis algorithm in order to improve its accuracy and performance.

Comparison of the heart rate derived from the PPG signals to the heart rate measured from using the sphygmomanometer was also as a means of validation of data. When there was a discrepancy of more the 10 BMP the test was marked invalid and not used.

Heart rate was also used in the weighting of results when creating a fit between PPG PTT/PWV and BP readings. See the Creating a Fit section on page 82.

Problems Analysing Data

A number of issues made the task of analysing the PPG data from the dual channel device difficult. These issues can be classified into two categories

- Data error reading from the device
- Abnormalities with the PPG signal(s)

Fixing Device Read Data Error

Much of this category of problem can be put down to a design flaw with the device and how it communicates with the PC. This flaw was the lack of handshaking between the device and the PC. It was not until after all the data was collected and the analysis algorithm was being refined in MATLAB that this problem was first noticed.

The problem can be described as follows. Once a serial connection between the PC and the device was established the device constantly sends a stream of data to the PC until the connection is closed. There is no hand shaking, there is no data format, just a stream of consecutive bytes sent over the serial communications.

Each channel of PPG data is sampled by a 12-bit ADC, but over the serial communications this data is sent as 8-bit bytes. As such a single sample from one of the PPG sensors requires two bytes to be transmitted; the upper nibble (4-bits) of the high byte is padded with four zeros. Since the dual channel device sends both PPG samples from each channel together a single dual PPG channel sample is 4 bytes long.

However, when the PC starts reading in the data it can start reading at any point in the data stream. As such it could start reading at any one of the 4 bytes used to make up a dual PPG

sample. As a result, when reading in the data there are four ways the data could be interpreted and possibly read in correctly. This is illustrated below in Figure 36.

It was not until late in this research that the data acquisition problem was rectified. The steps outlined in the next two sections were both performed before it was fixed and had to be repeated once the problem was rectified.

The rectification of this problem was only made possible by the fact the upper nibble of each of the high bytes from each signal was always set to zeros. This meant the high byte from each PPG signal sample was always of the format 0000xxxx in binary, where xxxx is the higher 4-bits of the 12-bit sample. A MATLAB script, **swapChannelData.m** (see Appendix), was written to manipulate the data stored in the common log format so the test, which was previously marked as invalid, could be used.

Swapping the data around only half solved the problem however. As can be seen in Figure 36 one of the four places the PC could start reading is reading the second channels data, in the correct byte order, first. The result of which was the PC interpreted the PPG channels

Byte	1	2	3	4	5	6	7	8	9	10
	Channel 1 Sample #1		Channel 2 Sample #1		Channel 1 Sample #2		Channel 2 Sample #2		Channel 1 Sample #3	
	High Byte	Low Byte	High Byte	Low Byte						
	Channel 1 Sample #1		Channel 2 Sample #1		Channel 1 Sample #2		Channel 2 Sample #2			
	Low Byte	High Byte	Low Byte							
	Channel 1 Sample #1		Channel 2 Sample #1		Channel 1 Sample #2		Channel 2 Sample #2			
	High Byte	Low Byte	High Byte							
	Channel 1 Sample #1		Channel 2 Sample #1		Channel 1 Sample #2		Channel 2 Sample #2			
	Low Byte	High Byte	Low Byte							
	Channel 1 Sample #1		Channel 2 Sample #1		Channel 1 Sample #2		Channel 2 Sample #2			
	High Byte	Low Byte	High Byte							

Starts reading from 1st byte in data stream. No problem.

Starts reading from 2nd byte in data stream. Data is read as [Low > High].

Starts reading from 3rd byte in data stream. Channels are swapped.

Starts reading from 4th byte in data stream. Data is read as [Low > High].

Figure 36: Data acquisition from device to PC

incorrectly and the channels were swapped. This creates a problem for the calculations of PTT/PWV values needed in this research as the one of the basic assumptions is that the ear channel, which is being used as an analogy of ECG signal for the calculation of PTT values, will be seen on channel 1.

Visual inspection of the two PPG signals could be used to determine which channel was which and if necessary the channels were swapped. Initially the assumption was that the ear signal generally had a lower systolic amplitude than observable on a PPG finger signal. This was not always the case however. A secondary visual inspection method was employed using the assumption that dicrotic notches do not generally appear on PPG ear signals.

MSUS points were also used to programmatically determine if the channels should be swapped or not as MSUS on the ear PPG signal should usually lead the MSUS on PPG finger, but this was still visually confirmed manually and channels swapped as appropriate.

All of this can be seen in on the next page in Figure 37.

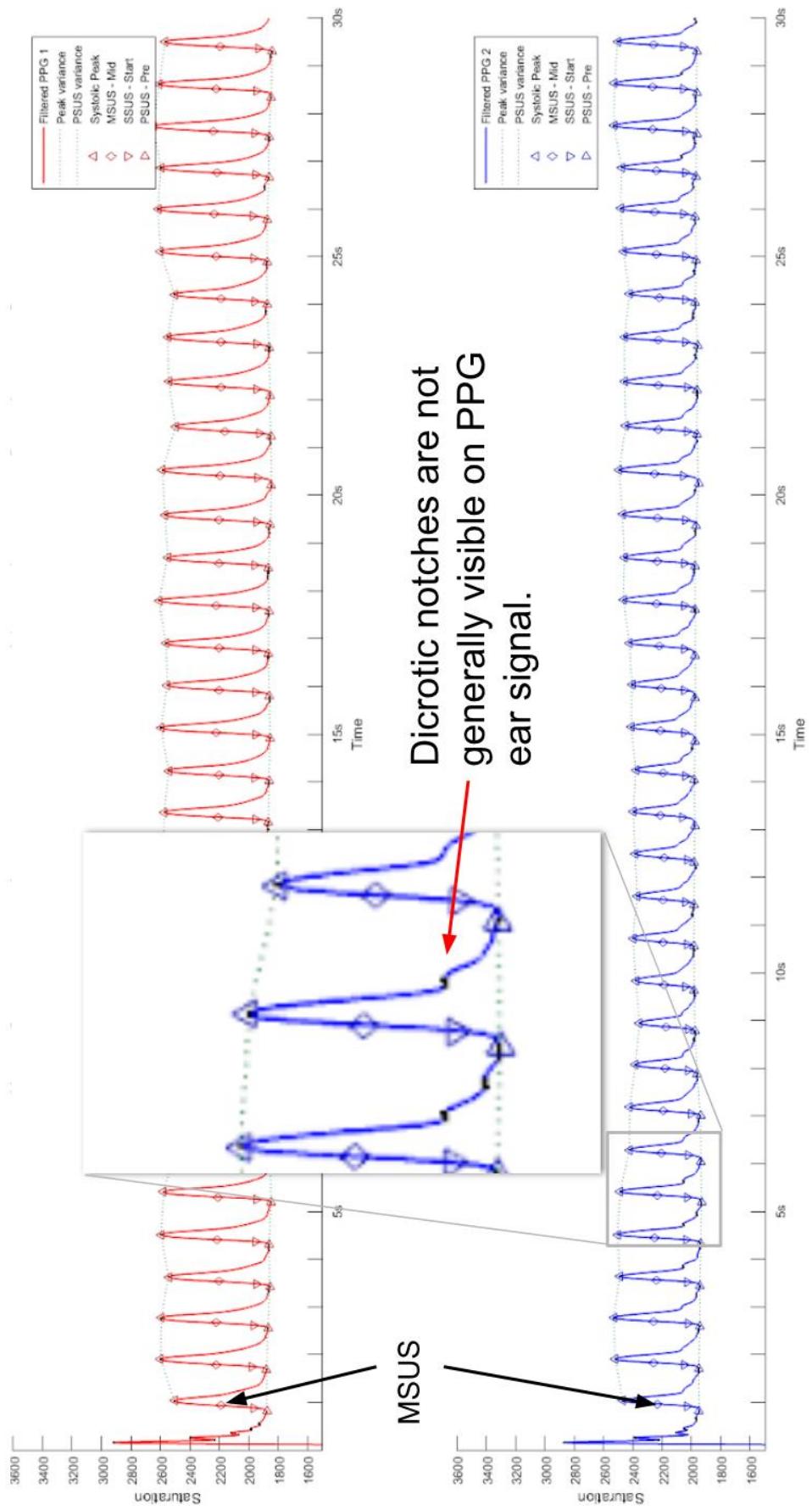


Figure 37: Determining a swapped channel

Signal Abnormalities

Abnormal PPG signals was another problem that affected the analysis algorithm's ability to analyse the data from the dual channel PPG device. As the analysis was refined the better equipped it was to be able to handle irregularities in the data, but some could not be recovered from. Most problematic PPG signals from the tests can be classified as two types:

- Poorly placed PPG sensors
- Movement artefacts

Shown is Figure 38 is data from a test where the PPG sensor had been poorly placed on the subjects ear. As can be seen in the figure the amplitude of the signal is quite small as the sensor was not able to properly detect the subject's oxygen saturation of the blood correctly. This weak signal could not be properly processed by the analysis algorithm; in particular failure to properly detect SSUS points was often an issue with this type of signal.

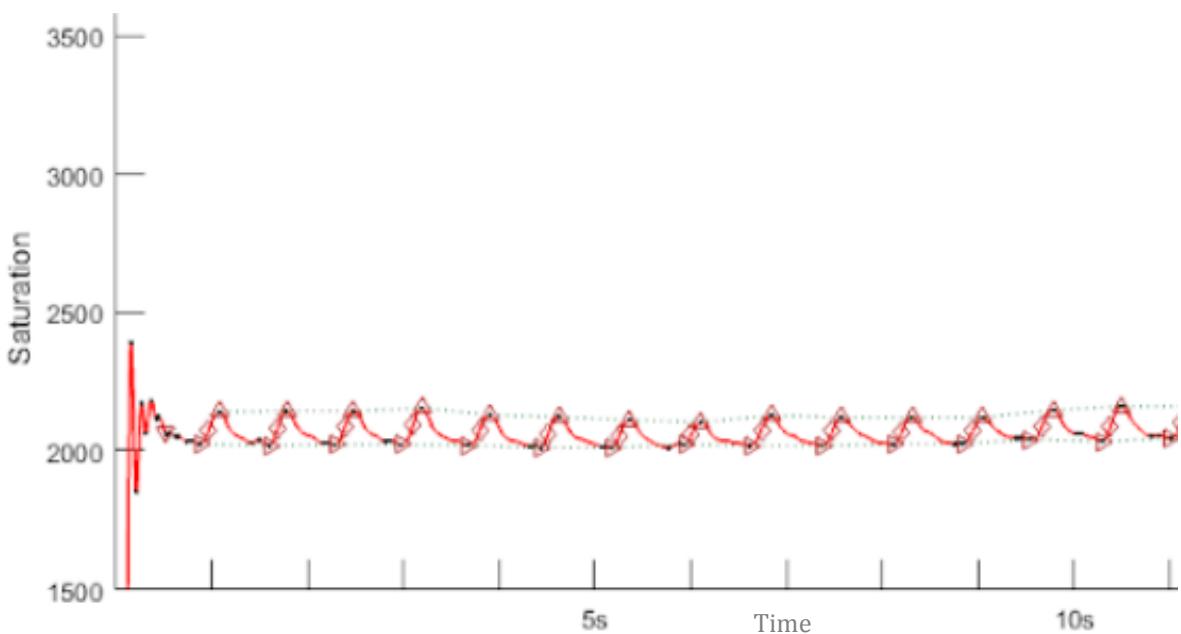


Figure 38: Signal showing poor PPG sensor placement

In Figure 39 two issues have occurred with the PPG data. As highlighted by the arrows in the figure a movement artefact has occurred. This is likely due to sudden movement of the subject which has caused a spike in both the PPG ear and finger signals.

It can also be seen that the red PPG ear signal is not producing a stable cyclical signal. This is also the result of poor PPG sensor placement.

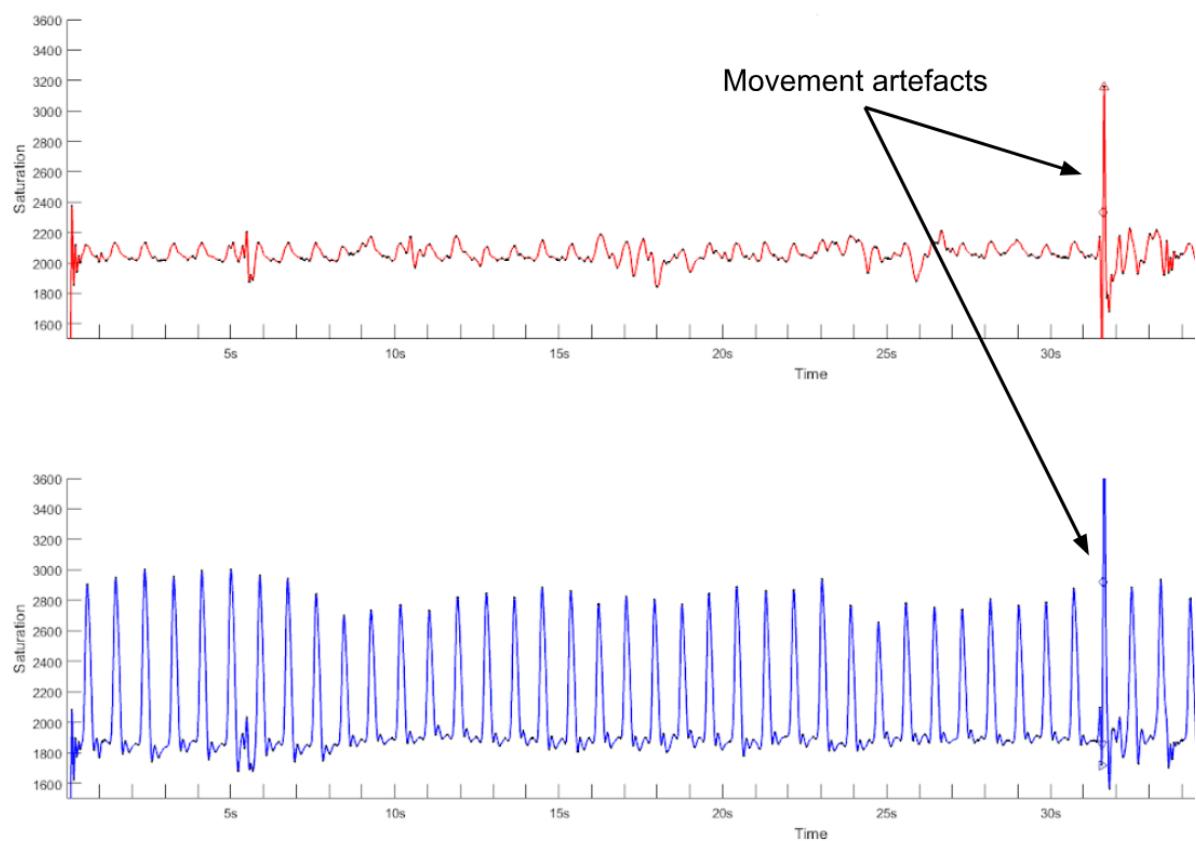


Figure 39: Signal showing movement artefacts and poor ear PPG placement

Creating a Fit

In order to estimate BP using the PPG signals a correlation between the PPG input and the BP readings obtained from the Omron must be found. As this research attempts to compare the four differing methods of calculating PTT/PWV a number of different sets of correlations needed to be calculated. This number is multiplied as SBP, DBP and MAP were all being compared as well. Consequently, for a given set of data 12 different correlations needed to be calculated.

This process was repeated using different sets of calibration data and comparison data which were all ultimately derived from subsets of all the data available to this research. The three sets of data that were used were as follows

- Data collected during Circadian Experiments
- Data collected during Circadian Experiments and Initial Experiments
- Data collected during Circadian Experiments, Initial Experiments and data obtained from stage 2 of the Second Hospital of Jiaxing Municipal Experiments.

To estimate BP from the PPG based methodologies a set of calibration data needed to be used to create a correlation between each of the PPG based PTT/PWV methodologies. This training data took a set of PPG based readings obtained after processing the data with the analysis algorithm. This provided PTT/PWV averages using the each of the 4 different methodologies for all the tests. Each test also has associated with it the sphygmomanometer readings taken as a part of the test.

For each of the 4 methodologies (PSUS, SSUS, MSUS and Peak) a correlating fitting algorithm was calculated between the average PWV of each test and the SBP, DBP and MAP readings. This process relied on the Curve Fitting Toolbox extension to MATLAB.

The first step in this process was to use the Curve Fitting Tool, **cftool**, in MATLAB to analyse all the data produced by the analysis algorithm. Output from different combinations of three sources was fed into the Curve Fitting Tool one by one until later the process was automated with a script, **fittest.m** (which can be found in the Appendix)

The MATLAB script takes a set of analysis data, extracts the tests that were marked as valid during analysis (though this can be overridden to select different tests as being valid or invalid) and attempts to establish a correlating fit using the different built in fitting functions in MATLAB. This script produced output is shown in Figure 40 on the following page. The graphs themselves are not as important as the numbers quoted above them; the sum of squares due to error (SSE) and the adjusted R-squared (coefficient of determination). These were used to determine the best inbuilt function to use for creating a correlation between PPG based PWV and BP. In the end the exponential function shown in Equation 2 was chosen as the base fitting function, where values for a and b are calculated for each given set of data.

Equation 2: Fitting function

$$y = a e^{bx}$$

Two different sized fit series were used in establishing a fit between the PPG PWV readings and the BP readings, both of which came from the data obtained in the Circadian Experiments.

- using first 10 valid circadian results
- using first 25 valid circadian results

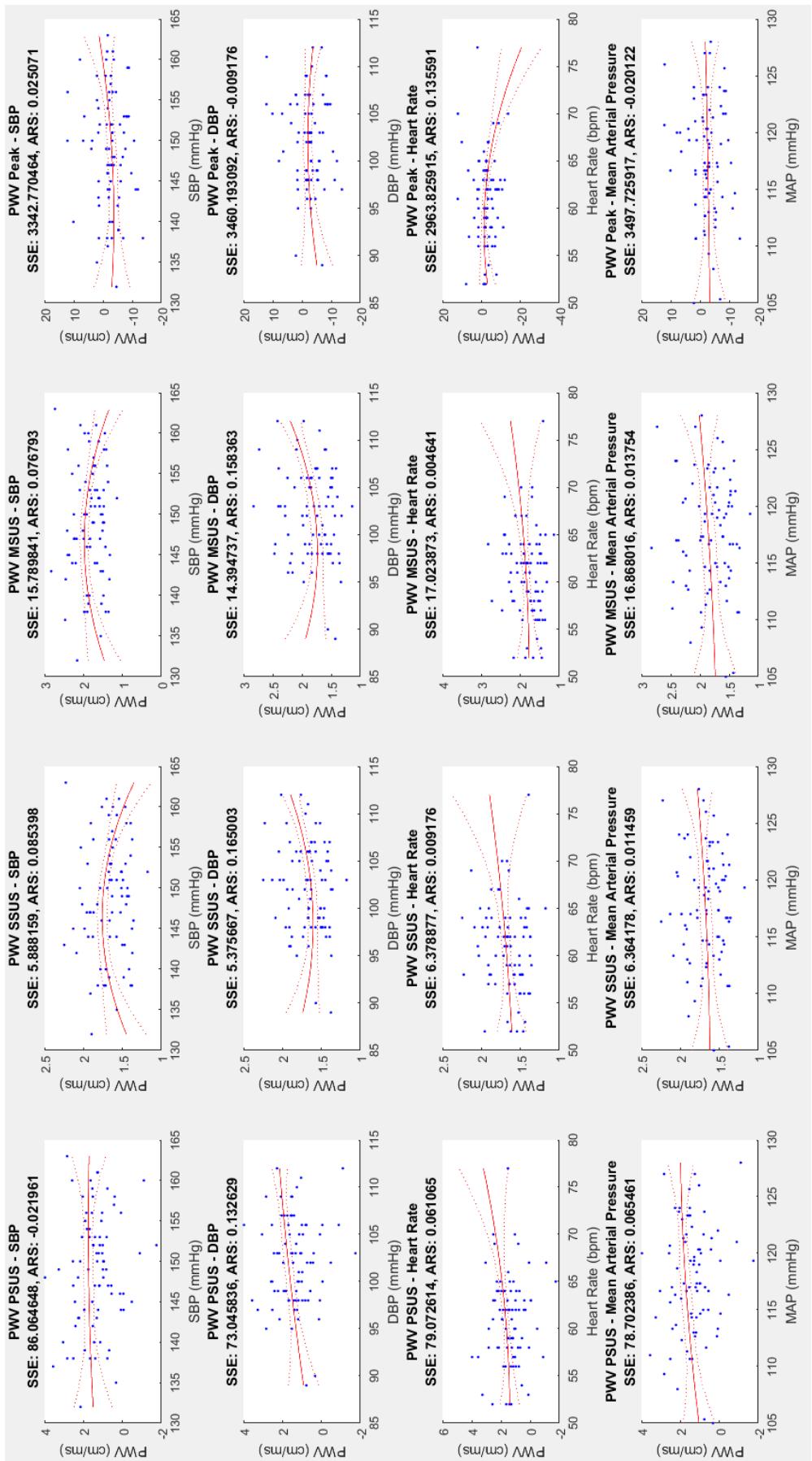


Figure 40: Example output of fittest.m

Each fit series was also weighted in three different ways to produce a total of 6 fit series. These 3 weightings involved comparing the heart rate measured by the sphygmomanometer to that which was detected using the PPG. The rational for this was that heart rate was something that is easily calculated using the PPG and was something that was easily verifiably during the processing of each test. Also, as there were two PPG there were two heart rate readings obtained from each test; one from each of the PPG signals.

The three weighting types were classified as follows

- no weighting – here the data was not weighted at all
- simple weighting (see Equation 3)
- reciprocal weighting (see Equation 3)

Equation 3: simple weighting

$$\omega_{simple} = 1 - |HR_{omron} - mean(HR_{ear}, HR_{finger})|$$

Equation 4: reciprocal weighting

$$\omega_{reciprocal} = \frac{1}{|HR_{omron} - mean(HR_{ear}, HR_{finger})|}$$

This all resulted in a total of 24 exponential curve fitting function that mapped the 4 variants of PWV averages obtained from each test to the BP (SBP, DBP and MAP) readings taken. These curve fitting functions were then used to estimate the BP readings for the remaining set of tests which were then validated against the AAMI and BHS standards which is discussed in the following section.

Evaluation Against the AAMI and BHS Standards

In this research the BHS and AAMI SP10 / ISO 81060 standards were used to guide the evaluations of the different PTT/PWV calculation methods. It should be noted that in evaluating the proposed methodologies against these two standards the complete evaluation procedures detailed within the standards were not followed. Both standards are quite specific in how devices should be evaluated.

Both AAMI and BHS consider mercury-based sphygmomanometers as the gold standard that should be used as the baseline for evaluation of new devices and techniques. In doing so they both stipulate that for the purposes of such evaluation a sufficiently trained person should use the auscultation method to obtain blood pressure readings from a subject for each measurement taken from the new device or using the proposed new technique. The BP readings obtained through auscultation and the mercury sphygmomanometer can then be compared to that readings obtained using the new device or technique.

Whilst all the tests that form the data that was used in this research include *standard* BP and heart rate readings taken at the time of the test, these readings were not obtained via the method outlined in the AAMI and BHS standard. Rather than using auscultation and a mercury sphygmomanometer these readings were taken using an oscillometric device, an Omron HEM-6221 in the case of the data collected by this researcher on them self. This is contrary to AAMI and BHS recommendations for the purposes of evaluation of the proposed methodology.

Although not strictly adhering to either standard, both AAMI and BHS were used in making a comparison between four variations of calculating PTT/PWV.

After analysis of each test was complete the four variants of PTT/PWV were compared to the BP readings from take from the oscillometric sphygmomanometer. These comparisons result in a grade using the BHS standard and a pass or fail using AAMI.

For the purposes of making these comparisons a MATLAB script was written that took the results of the PPG-BP analysis tool and compared the calculated SBP, DBP and MAP against the results obtained and recorded from the Omron. Examples of the validation results for one of the fitting algorithms can be seen in Table 12 and Table 13 on pages 92 and 93.

Full results can be seen in the Appendix, but as shown in Table 9 and Table 10 (on the next page) are a summary of the results obtained from the fitting algorithm validations against the BHS and AAMI standards. The results show that the higher number of tests used in calibrating the fitting algorithm the higher the accuracy of the BP readings extrapolated from the PPG derived PWV readings.

The results also show that the fitting algorithm performed better without any weighting of the input calibration data.

Also shown in the tables is that estimation of DBP using the PPG PTT/PWV produced more accurate results than using the same sets of calibration and validation data to estimate SBP from PPG PTT/PWV.

Table 9: Fit validation with 10 calibration tests
 Grade refers to BHS grade (A, B, C or D), where A and B grade is suitable to use.

		Circadian				Circadian + Initial				Circadian + Initial + Chinese				
		70 validated				78 validated				93 validated				
Reciprocal	Simple	PSUS	SSUS	MSUS	Peak	PSUS	SSUS	MSUS	Peak	PSUS	SSUS	MSUS	Peak	
		DBP	B	C	B	B	B	C	C	C	D	D	C	C
		SBP	D	D	D	D	D	D	D	D	D	D	D	C
Grade		D				D				D				
Reciprocal	None	DBP	B	C	C	D	B	C	C	D	D	D	D	D
		SBP	D	D	D	D	D	D	D	D	D	D	D	D
		Grade	D				D				D			
Reciprocal	None	DBP	C	D	D	D	C	D	C	D	D	D	D	D
		SBP	D	D	D	D	D	D	D	D	D	D	D	D
		Grade	D				D				D			

Table 10: Fit validation with 25 calibration tests
 Highlighted cells refer to passing the AAMI standard for this category, though not passed overall as SBP and DBP both need to be passed

		Circadian				Circadian + Initial				Circadian + Initial + Chinese				
		55 validated				63 validated				78 validated				
Reciprocal	Simple	PSUS	SSUS	MSUS	Peak	PSUS	SSUS	MSUS	Peak	PSUS	SSUS	MSUS	Peak	
		DBP	B*	A*	A*	A*	B*	B*	B*	B*	D	C	C	C
		SBP	D	C	C	C	D	C	C	D	D	D	D	D
Grade		C				C				D				
Reciprocal	None	DBP	B*	B*	B*	A*	B	B*	B	B*	D	C	C	C
		SBP	D	C	C	C	D	D	D	D	D	D	D	D
		Grade	C				D				D			
Reciprocal	None	DBP	C	B*	C	C	C	B	C	C	D	C	D	D
		SBP	D	D	D	C	D	D	D	D	D	D	D	D
		Grade	C				D				D			

Table 11 below shows the results using the full range of tests from the circadian experiments that were available after the data recovery of *byte swapped* and *channel swapped* tests as described in the Fixing Device Read Data Error section in Problems Analysing Data in Chapter 4.

As this table only shows results from the circadian tests they are only related to one subject, this researcher. They were all taken over the course of 7 days. Shown here is a higher number of overall test data with increased validation set. It should also be noted that whilst the first column of tests marked “Circadian” (only) in Table 9 and Table 10 do not use the same set of fit data as some of the tests recovered from the data errors now for a part of the calibration data. See Table 13 on page 93 and Figure 41 on page 94 for example output from validation of this full data set.

Table 11: Fit Classifications of Full Circadian Experiments

		Full Circadian 10 fit				Full Circadian 25 fit				Full Circadian 50 fit			
		157 validated				142 validated				117 validated			
None	DBP	A*	A*	A*	A*	A*	A*	A*	A*	A*	A*	A*	A*
	SBP	B	D	D	B	B	C	C	B	C	B	B	B
	Grade	B				B				B			
Simple	DBP	A*	A*	A*	A*	A*	A*	A*	A*	A*	A*	A*	A*
	SBP	B	D	D	B	B	B	C	B	C	B	B	C
	Grade	B				B				B			
Reciprocal	DBP	A*	B*	B*	B*	B	C	B	B	A*	C	C	C
	SBP	B	D	D	B	B	B	C	B	B	B	B	B
	Grade	B				B				B			

Table 12: Fit Validation: Circadian, Fit: 25, Eval: 55, Weight: None

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	BHS Grade	Difference in calculated and actual BP value		
						≤ 5mmHg	≤ 10mmHg	≤ 15mmHg
PWV PSUS	DBP	Y	4.44	3.24	B	32 (58.18%)	53 (96.36%)	54 (98.18%)
	SBP	N	9.72	7.48	D	19 (34.55%)	31 (56.36%)	43 (78.18%)
	MAP	Y	5.77	3.98	B	28 (50.91%)	47 (85.45%)	54 (98.18%)
PWV SSUS	DBP	Y	3.99	2.79	A	35 (63.64%)	53 (96.36%)	55 (100.00%)
	SBP	N	5.84	3.95	C	27 (49.09%)	45 (81.82%)	54 (98.18%)
	MAP	Y	4.37	2.73	A	33 (60.00%)	54 (98.18%)	55 (100.00%)
PWV MSUS	DBP	Y	4.09	2.73	A	36 (65.45%)	53 (96.36%)	55 (100.00%)
	SBP	N	6.15	4.07	C	26 (47.27%)	43 (78.18%)	53 (96.36%)
	MAP	Y	4.51	2.74	B	31 (56.36%)	53 (96.36%)	55 (100.00%)
PWV Peak	DBP	Y	4.2	3.00	A	35 (63.64%)	53 (96.36%)	55 (100.00%)
	SBP	N	6.15	4.36	C	25 (45.45%)	46 (83.64%)	54 (98.18%)
	MAP	N	4.34	3.10	A	34 (61.82%)	51 (92.73%)	55 (100.00%)

Table 13: Fit Validation: Circadian Full, Fit: 25, Eval: 142, Weight: None

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	BHS Grade	Difference in calculated and actual BP value		
						≤ 5mmHg	≤ 10mmHg	≤ 15mmHg
PWV PSUS	DBP	Y	4.11	3.14	A	97 (68.31%)	135 (95.07%)	141 (99.30%)
	SBP	N	5.87	4.50	B	74 (52.11%)	110 (77.46%)	137 (96.48%)
	MAP	Y	4.27	3.01	A	93 (65.49%)	134 (94.37%)	141 (99.30%)
PWV SSUS	DBP	Y	3.94	2.97	A	93 (65.49%)	137 (96.48%)	142 (100.00%)
	SBP	N	6.26	4.32	C	70 (49.30%)	115 (80.99%)	136 (95.77%)
	MAP	Y	4.25	2.83	A	94 (66.20%)	137 (96.48%)	142 (100.00%)
PWV MSUS	DBP	Y	4.16	3.15	A	93 (65.49%)	135 (95.07%)	142 (100.00%)
	SBP	N	7.01	5.23	C	58 (40.85%)	102 (71.83%)	130 (91.55%)
	MAP	Y	4.66	3.27	A	86 (60.56%)	132 (92.96%)	142 (100.00%)
PWV Peak	DBP	Y	4.43	3.23	A	86 (60.56%)	135 (95.07%)	141 (99.30%)
	SBP	N	5.41	4.12	B	80 (56.34%)	119 (83.80%)	138 (97.18%)
	MAP	Y	4.24	2.90	A	88 (61.97%)	135 (95.07%)	142 (100.00%)

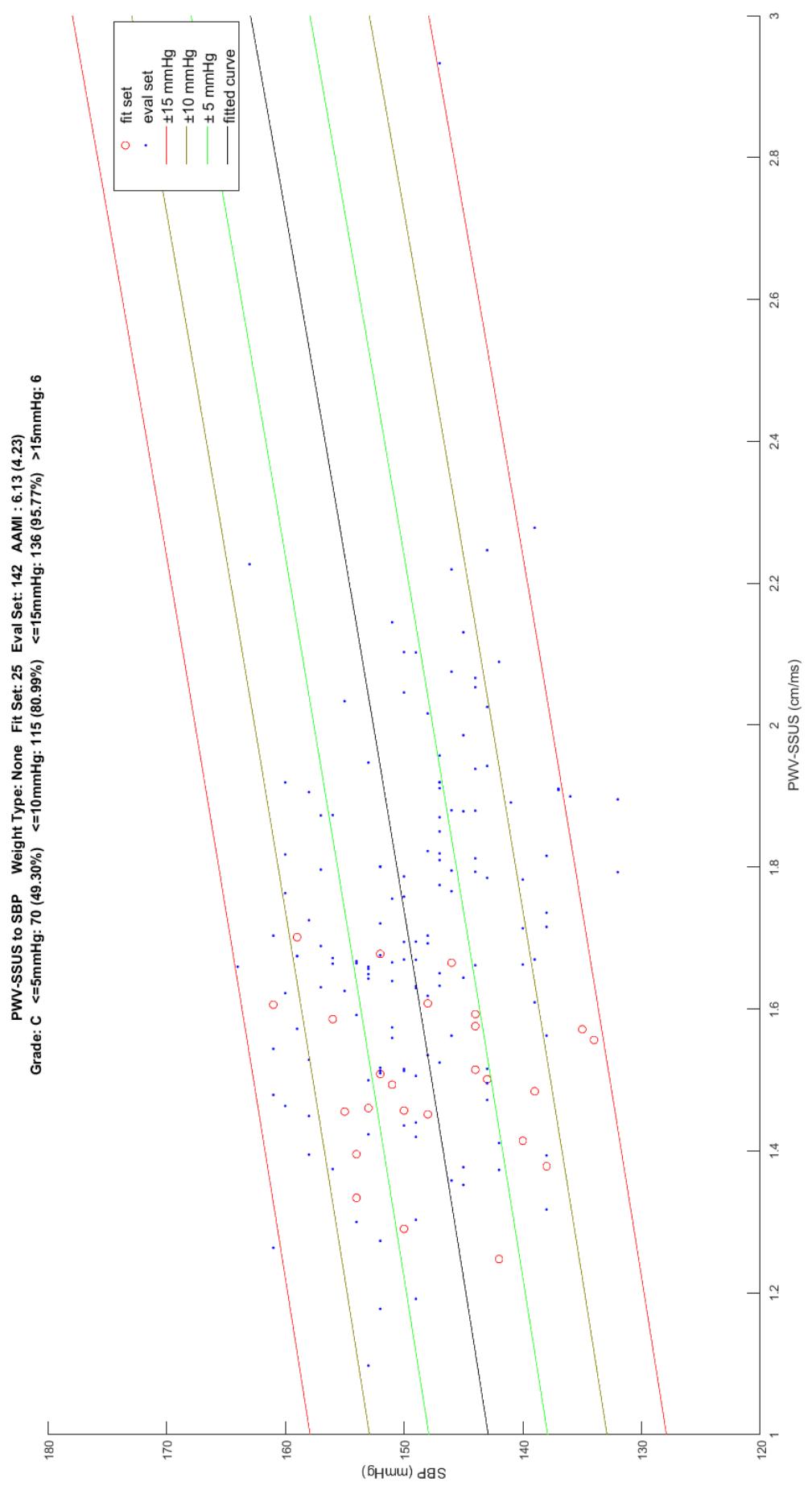


Figure 41: Fit validation output graph

Chapter 5: Discussion

Assessment of the Fitting Algorithm

As mentioned in the section Creating a Fit in the previous chapter the final outcomes of this research was to create a fitting algorithm for each of the 4 different methods of calculating PPG PTT/PWV. In effect however there actually 12 fitting algorithms created; each for the 4 methodologies had 3 different weighting systems using to weight the calibration data according to how close the PPG calculated heart rate was to the heart rate obtained from the sphygmomanometer. An unexpected result of this research was the fit performed better without any weighting of the input calibration data. The purpose of creating the two weighting models was to give a higher priority to PTT/PWV values that were derived from tests where the HR measured from the PPG were closer to the heart rate measured using the sphygmomanometer.

Also shown in Table 9 and Table 10 is that estimation of DBP using the PPG PTT/PWV methodologies has been assessed to be more accurate than estimation of SBP. Consistently DBP received a better BHS grade and satisfied the AAMI requirements more often. SBP never passed the AAMI specifications.

Noting the following with respect to the Circadian tests; all used the author as the subject and all were all performed in the same week. Given that these tests were used to create the fitting algorithm it us understandable that the algorithm performed best in estimating BP for the remainder of the set of results available in the group. It is also within reason that the validation of the fitting algorithm using either 10 or 25 of the Circadian tests as calibration data performed best in estimating BP from the remaining 70 or 55 circadian tests.

Accuracy in being able to estimate BP is reduced when data taken from the Initial tests is used. This test data again used the researcher as a subject, but the tests had been conducted many months prior – changes in the author’s BP had occurred in that time. Accuracy is reduced even further when the same fitting data is used to try and estimate BP from PPG readings collected at the Second Hospital of Jiaxing Municipal – i.e. completely different people.

Overall it seems clear that calibration data is necessary in order for the methodologies proposed here to function. This calibration data must be recently acquired as well as specific to the subject / patient.

Assessment of the Proposed PPG Methodologies

This research used 4 different methods for calculating PTT/PWV. These four methods can be used separately, or in combination, to perform PTT and/or PWV calculations. Of the four methods two of them are already being extensively used; namely Peak and MSUS.

The use of PPG peak as the detection point was, for example, used by Myint et al [28] in their research. Chen et al [38] also used systolic peak detection, as well as other detection mechanisms, such as detecting the diastolic peak.

MSUS is likely the most common approach used by most researchers in this area. It’s both stable and relatively easy to detect using a simple signal analysis. It’s not only used in this research as a basis for detection of all the other PPG detection points it’s also been used by others as their main diagnostic point. Gesche et al [27] used the mid-point of the systolic up stroke for the PPG component of their PTT calculations. As did Ibrahim et al [49] with their novel approach to calculating PTT by using bio-impedance measurement to replace an ECG.

Of the four methods detection methods evaluated in this research two are at least in some part novel; SSUS and to a greater extent PSUS. Methodologies akin to SSUS have been used in other research, but this wasn't evident to this researcher until after the experimental phase of this research was complete. In their research creating a neck worn PPG detection device Nabeel et al [41, 42] identified the maximum peak in the PPG signal's second derivative as the detection point. However, they did not detail how they selected the appropriate peak in the APG signal, as there are many.

SSUS may well offer a small improvement over using Peak or MSUS detections for use with PTT/PWV calculations. This can be seen in Table 12 (on page 92) by the marginal improvement in SBP estimations using SSUS.

To the knowledge of this researcher there seems to be no other research that has used PSUS as a detection point. Of the four methods used in this research this method was the most unstable and difficult to ensure correct detection however. It was especially susceptible to movement artefacts, so its use may be somewhat limited.

Many issues were encountered in the processing of the PPG signals by the analysis algorithm. A large amount of time was spent in trying to adapt this algorithm and to keep it completely automatic. It would have been much easier to manually examine the signals and mark points of interest or remove incorrectly automatically detected ones. This was in fact done on occasion.

The biggest issue in dealing and processing the PPG signals was detection of the first heartbeat. The heart is asynchronous with the beginning of the test. As such, when the test begins the PPG signal read in can start anywhere within the natural heartbeat cycle. It would be great if the heart could be paused and begin its cycle in line with the beginning of each test, but as the subjects are humans and not robots this is not possible.

This is further complicated by the use of two PPG signals; the test could start right at the point where a pulse wave has just transited the PPG ear sensor but has not yet propagated to the PPG sensor on the finger. This is one advantage of using an ECG – the R-peak is easily distinguished from the rest of the signal.

Another issue in using PPG signals as a means of estimating BP is that the ear beats detected on the ear and finger PPG sensors could on occasion go out of phase with each other. In theory the ear PPG, which as discussed in the Sensor Positions Using Dual Channel Device section is located close to the common carotid artery should be leading the PPG finger signal which is located right at the periphery in terms of arterial supply. This could possibly be explained by the vacillation of the carotid artery, which is one of the human bodies means of maintaining BP. This would be interesting, if not somewhat difficult, to investigate; at least without the use of IBP and the use of drugs or other means to induce a change in BP, both of which are outside the scope of a Masters thesis.

The final and most common problem in the use of PPG signals in this research was the movement artefacts that occurred when the subject moved. Turning of the neck or movement of the arm caused noticeable changes in the PPG signal when compared to the subject sitting completely still. These artefacts can be seen as a small change in the PPG signal's properties such as a change in signal amplitude or a change in the signal's waveform (i.e. the absence of a dicrotic notch). Movement artefacts can not only mess up the detection of a single heartbeat and associated point of interest but can also cause the de-synchronisation of the heartbeats on the two PPG channels, which completely destroys the chance of accurate PTT/PWV calculations.

This means that it is highly unlikely for a purely PPG based methodology to be used in a portable device unless an accurate way of detecting and removing movement artefacts can be established. This may make purely continuous CNIBP unobtainable.

Assessment of the Dual Channel PPG Device

Broadly speaking the efficacy of the dual channel PPG device developed in this research has been established. It is capable of collecting two channels of PPG data simultaneously and having that data stored and analysed externally to the device.

While the device was created using the Alientek Warship development board the functionality of this device could easily be replicated by a much smaller and easily portable device than development board that was used in this prototype. The dual channel device only ever relied on the basic functionality of the ARM Cortex M3 Microcontroller; the flash storage within the chip to store the small program that runs on the device and the ADC which is also a part of the chip. As such a device could easily be created that based on not much more than the STM32F103T6 or something similar (of which there are plenty of capable ARM or alternative microcontrollers) and a small power circuit to power the chip. The LCD that was used in this dual channel device, while handy for the purposes of this research may not be required in a fully commercialised device if one was ever built. The LCD screen was the only component of the development board used in the dual channel device that could not be provided by the microcontroller chip alone.

There were however problems with the design of the dual channel device and its software. Many issues occurred from an early design decision that could have, and should have, been easily rectified from the beginning. This decision was that the device did not perform any

handshaking with the attached PC or any marking of the data. The device just blindly sends the two samples from the PPG signals as 4 bytes over and over again. When the data is being read by the PC there is was no marking of the data or any way to distinguish which channel or even which byte was being read.

This can be rectified relatively easily but it also results in an assumption about which channel is which when analysing the data. As a PPG finger signal usually will generally have higher peak amplitudes and contain a dicrotic notch where a PPG ear signal will not visually this difference in the two signals is easily spotted. However, this is not so easily done with 100% accuracy using software.

A far better solution would be to implement handshaking communications between the dual channel device and the PC and marking the data using some sort of data format and not sending a continuous stream of bytes of raw PPG sensor data.

Effect of White Coat Syndrome on Results

As mentioned in previous chapter there were many issues with the data collected from the researchers in China. Some were overcome. Some were not. One problem that could never be overcome and cannot be accurately quantified is the effect of white coat syndrome on the results that the researchers at the Second Hospital of Jiaxing Municipal in China collected.

Given that all the subjects of all the tests conducted at the hospital were patients of the hospital it is perhaps inevitable that the subjects could be physiologically affected by the surroundings the tests were conducted in and by whom the tests were carried out by; namely doctors from the hospital. The opposite would be true of the data collected by this researcher with them self as the subject.

Chapter 7: Conclusion

The aim of this research was to investigate the possibility of estimating BP using two PPG signals. In this study, the author compared 4 different detection methodologies that could be used in when calculating PTT/PWV using PPG signals. Two of these methods, PSUS and SSUS, were incorporated and developed as part of the process of refining the analysis tool.

This research was unable to find any strong correlations or concrete methodology for dual PPG to BP estimation. There were however some positive outcomes from this research. The biggest of which was the construction of the prototype dual channel PPG device. Its efficacy has been validated though its use throughout this research. As discussed in the next chapter this prototype could easily refined and improved, but this research has provided evidence that the approach used in creating the device could indeed be a valid one.

Many tests were conducted using this dual channel PPG device and there is a great deal of data that has been collected by both the author and the researchers at the Second Hospital of Jiaxing Municipal. All of this data could be further analysed.

Ultimately however comparing PPG signals to sphygmomanometer readings taken after them is less than ideal, even if the AAMI and BHS standard both advocate something similar. To get completely clear indications of how PPG signals can be used to estimate BP and to monitor and predict changes within them, it would seem to be an advantage in having PPG signals being collected simultaneously with IBP using a cannula. Given that both IBP and PPG signals are continuous marrying changes in IBP signal to that of a PPG signal or signals may prove to be beneficial in creating an algorithm that estimates BP from PPG signals.

Chapter 6: Future Work

Future work in this research would be to incorporate extra features that have already been extracted by the analysis algorithm into the fitting algorithm that attempts to estimate BP using the dual channel PPG signals. These features are discussed in Analysis Data section of Chapter 4 and are listed in Table 7 on page 75.

These extra features could be used in further research to try another approach in analysis of the data to try and develop a better model between PPG and BP measurements thus improving BP estimation. Further data could also be collected, perhaps using more reliable equipment or more closely conforming to the AAMI SP10 / ISO 81060 standards both of which are discussed further below.

A second prototype of the dual channel device would be the top priority for any future work from this research. This new device could use a much smaller development board with fewer peripheral components or be a device built specifically for the purpose using an appropriate microcontroller and minimal external circuitry for powering the device. This device would inevitably be much smaller and therefore much more portable than the current device. Such a device could also be battery powered and store the PPG signal data itself for later retrieval rather than being reliant on an attached PC for data storage.

Refinement and improvement of the dual channel device could also allow for better communications between the device and PC used for analysis. A fully developed communications protocol could be developed to avoid the problems discussed in Fixing Device Read Data Error section of Chapter 4. Alternatively, the data could be sent from the device using Bluetooth or WiFi. Both of these protocols require packets of data, rather than a data stream as with serial communications (over USB as used with the current device).

Packetisation of the PPG sensor data for transmission over WiFi or Bluetooth would inherently resolve the situation.

Further improvements could see self-contained PPG sensor, wireless communications and battery all within a single unit where two separate sensor devices could be developed that work in tandem with each other to gather the dual channel PPG data. These could be used to continuously and non-invasively monitor BP without even the need for wires to connect the PPG sensors. The technology to produce such devices would seem to be not far off the horizon with battery life being the main sticking point. A somewhat similar device, the Apple AirPods, which are wireless all in one ear bud head phones, can currently operate for around 5 hours off a single charge [65]. Replace the speakers with PPG sensors and the hardware would match the requirements for self-contained, wireless PPG sensors.

An alternative approach could be to use thin film sensors [66] that could be attached or perhaps even 3D printed directly onto the subject. This approach could alleviate the issue of sensors moving out of position and causing incorrect readings to occur. Thin film sensors placed in situ directly on the epidermis could also allow from the placement of PPG sensors in sites which are currently difficult of uncomfortable to secure. This could allow ease of use of sensors at sites like the brachial or radial arteries without the risk of movement of the sensor from above the artery. It could also allow the development direct PPG monitoring of the common carotid artery. Nabeel et al [41, 42] developed a 3D printed dual PPG device which was positioned on the neck of the patient to measure PWV over a small section of the common carotid. This device was rather bulky and appeared that it would be quite uncomfortable for the subject. Using 3D printed, directly attached thin film PPG sensors such a device could be make much more comfortable and not suffer the risks of the sensors moving from the desired location above the artery.

If this researcher was to undertake further research in this area a specific change would be made around the collection of data used to validate any proposed method or device. This change would be to clearly adhere to the AAMI SP10 / ISO 81060 standards for the evaluation of any such procedure or device. In doing so the procedure of data collection would have to include validation BP readings taken using a mercury sphygmomanometer by a trained professional rather than reliance of oscillometric sphygmomanometers for such a purpose.

This would not only help increase the accuracy of the data collection, but it would also support another worthy aim of promoting the relevant standards which has been severely lacking even as more and more home-based devices appear on the market.

References

- [1] H. Robert and M. Elizabeth, *A Dictionary of Biology*. 2016.
- [2] S. Shrestha. (2018, 25/10/2018). *External Carotid Artery and Its Branches*. Available: <http://tube.medchrome.com/2011/06/how-to-remember-branches-of-external.html>
- [3] World Health Organisation. (2017, December). *Cardiovascular diseases (CVDs)*. Available: [http://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](http://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds))
- [4] World Health Organisation. (2015, December). *Q&As on hypertension*. Available: <http://www.who.int/features/qa/82/en/>
- [5] R. Porter, *The Cambridge illustrated history of medicine*. Cambridge University Press, 2001.
- [6] Qsota Medical. (2017, December). *Upper and lower blood pressure*. Available: <https://qsota.com/upper-and-lower-blood-pressure/>
- [7] L.-w. H. Lehman, M. Saeed, D. Talmor, R. Mark, and A. Malhotra, "Methods of blood pressure measurement in the ICU," *Critical care medicine*, vol. 41, no. 1, pp. 34-40, 2013.
- [8] A. Roguin, "Scipione Riva-Rocci and the men behind the mercury sphygmomanometer," *International Journal of Clinical Practice*, vol. 60, no. 1, pp. 73-79, 2006.
- [9] B. Saugel, R. Dueck, and J. Y. Wagner, "Measurement of blood pressure," *Best Practice and Research: Clinical Anaesthesiology*, vol. 28, no. 4, pp. 309-322, 2014.
- [10] "Samuel Siegfried Karl von Basch (1837–1905)," *Nature*, vol. 140, p. 393, 09/04/online 1937.
- [11] Y. L. Shevchenko and J. E. Tsitlik, "90th Anniversary of the development by Nikolai S. Korotkoff of the auscultatory method of measuring blood pressure," *Circulation*, vol. 94, no. 2, pp. 116-118, 1996.
- [12] R. Mukkamala *et al.*, "Toward Ubiquitous Blood Pressure Monitoring via Pulse Transit Time: Theory and Practice," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 8, pp. 1879-1901, 2015.
- [13] Z. M. Anastas, E. Jimerson, and S. Garolis, "Comparison of Noninvasive Blood Pressure Measurements in Patients With Atrial Fibrillation," *Journal of Cardiovascular Nursing*, vol. 23, no. 6, pp. 519-524, 2008.
- [14] L. Peter, N. Noury, and M. Cerny, "A review of methods for non-invasive and continuous blood pressure monitoring: Pulse transit time method is promising?," *IRBM*, vol. 35, no. 5, pp. 271-282, 2014/10/01/ 2014.
- [15] H. M. Cheng, D. Lang, C. Tufanaru, and A. Pearson, "Measurement accuracy of non-invasively obtained central blood pressure by applanation tonometry: A systematic review and meta-analysis," *International Journal of Cardiology*, vol. 167, no. 5, pp. 1867-1876, 2013.
- [16] C.-H. Chen *et al.*, "Estimation of Central Aortic Pressure Waveform by Mathematical Transformation of Radial Tonometry Pressure," *Validation of Generalized Transfer Function*, vol. 95, no. 7, pp. 1827-1836, 1997.
- [17] N. Langwieser *et al.*, "Radial artery applanation tonometry for continuous noninvasive arterial blood pressure monitoring in the cardiac intensive care unit," *Clinical Research in Cardiology*, vol. 104, no. 6, pp. 518-524, 2015.
- [18] E. Zahedi, V. Sohani, M. A. Mohd. Ali, K. Chellappan, and G. K. Beng, "Experimental feasibility study of estimation of the normalized central blood pressure waveform from radial photoplethysmogram," *Journal of Healthcare Engineering*, vol. 6, no. 1, pp. 121-144, 2015.
- [19] G. Fortino and V. Giampà, "PPG-based methods for non invasive and continuous blood pressure measurement: An overview and development issues in body sensor networks," in *2010 IEEE International Workshop on Medical Measurements and Applications, MeMeA 2010 - Proceedings*, 2010, pp. 10-13.
- [20] G. Parati, R. Casadei, A. Groppelli, M. Di Renzo, and G. Mancia, "Comparison of finger and intra-arterial blood pressure monitoring at rest and during laboratory testing," *Hypertension*, vol. 13, no. 6 Pt 1, pp. 647-655, 1989.

- [21] S. H. Kim *et al.*, "Accuracy and precision of continuous noninvasive arterial pressure monitoring compared with invasive arterial pressure: A systematic review and meta-analysis," *Anesthesiology*, vol. 120, no. 5, pp. 1080-1097, 2014.
- [22] P. Verdecchia, G. Schillaci, F. Boldrini, I. Zampi, and C. Porcellati, "Variability between current definitions of normal ambulatory blood pressure. Implications in the assessment of white coat hypertension," *Hypertension*, 10.1161/01.HYP.20.4.555 vol. 20, no. 4, p. 555, 1992.
- [23] K. Takazawa *et al.*, "Clinical usefulness of the second derivative of a plethysmogram (acceleration plethysmogram)," *J Cardiol*, vol. 23, no. suppl 37, pp. 207-217, 1993.
- [24] M. Elgendi, "On the analysis of fingertip photoplethysmogram signals," *Curr Cardiol Rev*, vol. 8, no. 1, pp. 14-25, Feb 2012.
- [25] N. Gayapersad, S. Rocke, Z. Ramsaroop, A. Singh, and C. Ramlal, "Beyond Blood Pressure and Heart Rate Monitoring: Towards a Device for Continuous Sensing and Automatic Feature Extraction of Cardiovascular Data," in *2016 8th International Conference on Computational Intelligence and Communication Networks (CICN)*, 2016, pp. 261-265.
- [26] I. C. Jeong, J. I. Ko, S. O. Hwang, and H. R. Yoon, "A new method to estimate arterial blood pressure using photoplethysmographic signal," in *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, 2006, pp. 4667-4670.
- [27] H. Gesche, D. Grosskurth, G. Küchler, and A. Patzak, "Continuous blood pressure measurement by using the pulse transit time: Comparison to a cuff-based method," *European Journal of Applied Physiology*, vol. 112, no. 1, pp. 309-315, 2012.
- [28] C. Myint, K. H. Lim, K. I. Wong, A. A. Gopalai, and M. Z. Oo, "Blood Pressure measurement from Photo-Plethysmography to Pulse Transit Time," in *IECBES 2014, Conference Proceedings - 2014 IEEE Conference on Biomedical Engineering and Sciences: "Miri, Where Engineering in Medicine and Biology and Humanity Meet"*, 2015, pp. 496-501.
- [29] M. Elgendi, "Standard terminologies for photoplethysmogram signals," *Current cardiology reviews*, vol. 8, no. 3, p. 215, 2012.
- [30] C. C. Y. Poon and Y. T. Zhang, "Cuff-less and noninvasive measurements of arterial blood pressure by pulse transit time," in *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, 2005, vol. 7 VOLS, pp. 5877-5880.
- [31] X. He, R. A. Goubran, and X. P. Liu, "Evaluation of the correlation between blood pressure and pulse transit time," in *MeMeA 2013 - IEEE International Symposium on Medical Measurements and Applications, Proceedings*, 2013, pp. 17-20.
- [32] Y. Choi, Q. Zhang, and S. Ko, "Noninvasive cuffless blood pressure estimation using pulse transit time and Hilbert–Huang transform," *Computers & Electrical Engineering*, vol. 39, no. 1, pp. 103-111, 2013.
- [33] F. Parry, G. Dumont, C. Ries, C. Mott, and M. Ansermino, "Continuous noninvasive blood pressure measurement by pulse transit time," in *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2004, vol. 1, pp. 738-741.
- [34] S.-y. Ye, G.-R. Kim, D.-K. Jung, S. Baik, and G. Jeon, "Estimation of systolic and diastolic pressure using the pulse transit time," *World Academy of Science, Engineering and Technology*, vol. 67, pp. 726-731, 2010.
- [35] M. Y. Wong, E. Pickwell-MacPherson, and Y. T. Zhang, "The acute effects of running on blood pressure estimation using pulse transit time in normotensive subjects," *Eur J Appl Physiol*, vol. 107, no. 2, pp. 169-75, Sep 2009.
- [36] Y. Yan, "Model-based Motion-resistant Method for Noninvasive and Continuous Measurement of Arterial Blood Pressure," *ProQuest Dissertations and Theses*, pp. 146-146, 2006.
- [37] J. M. Zhang, P. F. Wei, and Y. Li, "A LabVIEW based measure system for pulse wave transit time," in *5th Int. Conference on Information Technology and Applications in Biomedicine*,

ITAB 2008 in conjunction with 2nd Int. Symposium and Summer School on Biomedical and Health Engineering, IS3BHE 2008, 2008, pp. 477-480.

- [38] Y. Chen, C. Wen, G. Tao, and M. Bi, "Continuous and noninvasive measurement of systolic and diastolic blood pressure by one mathematical model with the same model parameters and two separate pulse wave velocities," *Ann Biomed Eng*, vol. 40, no. 4, pp. 871-82, Apr 2012.
- [39] Z. Tang *et al.*, "A Chair-Based Unobtrusive Cuffless Blood Pressure Monitoring System Based on Pulse Arrival Time," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 5, pp. 1194-1205, 2017.
- [40] D. B. McCombie, A. T. Reisner, and H. H. Asada, "Adaptive blood pressure estimation from wearable PPG sensors using peripheral artery pulse wave velocity measurements and multi-channel blind identification of local arterial dynamics," in *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, 2006, pp. 3521-3524.
- [41] P. M. Nabeel, S. Karthik, J. Joseph, and M. Sivaprakasam, "Experimental validation of dual PPG local pulse wave velocity probe," in *2017 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, 2017, pp. 408-413.
- [42] P. M. Nabeel, S. Karthik, J. Joseph, and M. Sivaprakasam, "Measurement of carotid blood pressure and local pulse wave velocity changes during cuff induced hyperemia," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 2017, pp. 1700-1703.
- [43] X. F. Teng and Y. T. Zhang, "Continuous and noninvasive estimation of arterial blood pressure using a photoplethysmographic approach," in *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No.03CH37439)*, 2003, vol. 4, pp. 3153-3156 Vol.4.
- [44] R. Samria, R. Jain, A. Jha, S. Saini, and S. R. Chowdhury, "Noninvasive cuffless estimation of blood pressure using photoplethysmography without electrocardiograph measurement," in *IEEE TENSYMP 2014 - 2014 IEEE Region 10 Symposium*, 2014, pp. 254-257.
- [45] J. C. Ruiz-Rodriguez *et al.*, "Innovative continuous non-invasive cuffless blood pressure monitoring based on photoplethysmography technology," *Intensive Care Med*, vol. 39, no. 9, pp. 1618-25, Sep 2013.
- [46] S. Suzuki and K. Oguri, "Cuffless and non-invasive Systolic Blood Pressure estimation for aged class by using a Photoplethysmograph," in *Proceedings of the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS'08 - "Personalized Healthcare through Technology"*, 2008, pp. 1327-1330.
- [47] G. Joseph, A. Joseph, G. Titus, R. M. Thomas, and D. Jose, "Photoplethysmogram (PPG) signal analysis and wavelet de-noising," in *2014 Annual International Conference on Emerging Research Areas: Magnetics, Machines and Drives, AICERA/iCMM 2014 - Proceedings*, 2014.
- [48] A. Suzuki and K. Ryu, "Feature selection method for estimating systolic blood pressure using the taguchi method," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1077-1085, 2014, Art. no. 6654315.
- [49] B. Ibrahim, V. Nathan, and R. Jafari, "Exploration and validation of alternate sensing methods for wearable continuous pulse transit time measurement using optical and bioimpedance modalities," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 2017, pp. 2051-2055.
- [50] S. H. Song, J. S. Cho, H. S. Oh, J. S. Lee, and I. Y. Kim, "Estimation of blood pressure using photoplethysmography on the wrist," in *Computers in Cardiology*, 2009, vol. 36, pp. 741-744.
- [51] T. Wibmer *et al.*, "Blood pressure monitoring during exercise: Comparison of pulse transit time and volume clamp methods," *Blood pressure*, vol. 24, no. 6, pp. 353-360, 2015.
- [52] B. Alpert, B. Friedman, and D. Osborn, "AAMI Blood Pressure Device Standard Targets Home Use Issues," (in English), *Biomedical Instrumentation & Technology*, pp. 69-72, 2010, 2014-04-02 2010.

- [53] E. O'Brien *et al.*, "European Society of Hypertension International Protocol revision 2010 for the validation of blood pressure measuring devices in adults," *Blood pressure monitoring*, vol. 15, no. 1, pp. 23-38, 2010.
- [54] E. O'Brien, B. Waeber, G. Parati, J. Staessen, and M. G. Myers, "Blood pressure measuring devices: recommendations of the European Society of Hypertension," *BMJ : British Medical Journal*, vol. 322, no. 7285, pp. 531-536, 11/17/accepted 2001.
- [55] Biopac Systems Inc. (2016, May). *MP System Comparison: MP150 vs. MP100*. Available: <https://www.biopac.com/knowledge-base/mp-system-comparison-mp150-vs-mp100/>
- [56] STMicroelectronics. (May). *STM32F103T6*. Available: http://www.st.com/content/st_com/en/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32-mainstream-mcus/stm32f1-series/stm32f103/stm32f103t6.html
- [57] World Famous Electronics LLC. (2016, July). *Pulse Sensor - Open Hardware*. Available: <https://pulsesensor.com/pages/open-hardware>
- [58] ARM Limited. (2018, May). *μVision® IDE*. Available: <https://www2.keil.com/mdk5/uvision/>
- [59] Biopac Systems Inc. (2018, May). *Photoplethysmogram Transducers*. Available: <https://www.biopac.com/product/photoplethysmogram-transducers/>
- [60] T. D. Giles, "Circadian rhythm of blood pressure and the relation to cardiovascular events," *Journal of Hypertension*, vol. 24, pp. S11-S16, 2006.
- [61] M. Millar-Craig, C. Bishop, and E. B. Raftery, "CIRCADIAN VARIATION OF BLOOD-PRESSURE," *The Lancet*, vol. 311, no. 8068, pp. 795-797, 1978/04/15/ 1978.
- [62] O. Australia. (2015, June). *Omron HEM6221 Premium Wrist Blood Pressure Monitor*. Available: <https://omronhealthcare.com.au/product/new-omron-hem6221-premium-wrist-blood-pressure-monitor/>
- [63] M. Elgendi *et al.*, "On Time Domain Analysis of Photoplethysmogram Signals for Monitoring Heat Stress," *Sensors*, vol. 15, no. 10, pp. 24716-24734, 2015.
- [64] D. A. Birrenkott, M. A. F. Pimentel, P. J. Watkinson, and D. A. Clifton, "Robust estimation of respiratory rate via ECG- and PPG-derived respiratory quality indices," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2016, pp. 676-679.
- [65] Apple Inc. (2018, May 28). *AirPods - Technical Specifications*. Available: https://support.apple.com/kb/SP750?locale=en_US
- [66] P. Gutruf *et al.*, "Transparent functional oxide stretchable electronics: micro-tectonics enabled high strain electrodes," *Npg Asia Materials*, Original Article vol. 5, p. e62, 09/13/online 2013.

Publications

R. Nye, Z. Zhang, and Q. Fang, "Continuous non-invasive blood pressure monitoring using photoplethysmography: A review," in Bioelectronics and Bioinformatics (ISBB), 2015 International Symposium on, 2015, pp. 176-179: IEEE.

Appendix

Fit Validation: Circadian, Fit Set: 10 Eval Set: 70 Weights: None	1
Fit Validation: Circadian, Fit Set: 10 Eval Set: 70 Weights: Simple.....	2
Fit Validation: Circadian, Fit: 10, Eval: 70, Weight: Reciprocal	3
Fit Validation: Circadian + Initial, Fit: 10, Eval: 78, Weight: None	4
Fit Validation: Circadian + Initial, Fit: 10, Eval: 78, Weight: Simple	5
Fit Validation: Circadian + Initial, Fit: 10, Eval: 78, Weight: Reciprocal.....	6
Fit Validation: Circadian + Initial + Chinese, Fit: 10, Eval: 93, Weight: None	7
Fit Validation: Circadian + Initial + Chinese, Fit: 10, Eval: 93, Weight: Simple	8
Fit Validation: Circadian + Initial + Chinese, Fit: 10, Eval: 93, Weight: Reciprocal	9
Fit Validation: Circadian, Fit: 25, Eval: 55, Weight: None	10
Fit Validation: Circadian, Fit: 25, Eval: 55, Weight: Simple	11
Fit Validation: Circadian, Fit: 25, Eval: 55, Weight: Reciprocal	12
Fit Validation: Circadian + Initial, Fit: 25, Eval: 63, Weight: None	13
Fit Validation: Circadian + Initial, Fit: 25, Eval: 63, Weight: Simple	14
Fit Validation: Circadian + Initial, Fit: 25, Eval: 63, Weight: Reciprocal.....	15
Fit Validation: Circadian + Initial + Chinese, Fit: 25, Eval: 78, Weight: None	16
Fit Validation: Circadian + Initial + Chinese, Fit: 25, Eval: 78, Weight: Simple	17
Fit Validation: Circadian + Initial +Chinese, Fit: 25, Eval: 78, Weight: Reciprocal	18
Fit Validation: Circadian Full, Fit: 10, Eval: 157, Weight: None.....	19
Fit Validation: Circadian Full, Fit: 10, Eval: 157, Weight: Simple.....	20

Fit Validation: Circadian Full, Fit: 10, Eval: 157, Weight: Reciprocal	21
Fit Validation: Circadian Full, Fit: 25, Eval: 142, Weight: None.....	22
Fit Validation: Circadian Full, Fit: 25, Eval: 142, Weight: Simple.....	23
Fit Validation: Circadian Full, Fit: 25, Eval: 142, Weight: Reciprocal	24
Fit Validation: Circadian Full, Fit: 50, Eval: 117, Weight: None.....	25
Fit Validation: Circadian Full, Fit: 50, Eval: 117, Weight: Simple	26
Fit Validation: Circadian Full, Fit: 50, Eval: 117, Weight: Reciprocal	27
MATLAB code: birrenkott.m	28
MATLAB code: compare.m	28
MATLAB code: createEmptyPpgAxis.m	31
MATLAB code: createEmptyPpgFigure.m	33
MATLAB code: createLogStructs.m.....	34
MATLAB code: extractPpgData.m	34
MATLAB code: findPpgPoints.m	35
MATLAB code: fittest.m.....	36
MATLAB code: fppp.m.....	38
MATLAB code: locatePpgChannelPeaks.m	44
MATLAB code: parseCircTests.m	46
MATLAB code: parseInitialTest.m.....	46
MATLAB code: parseInitialTests.m	49
MATLAB code: parsePpgData.m.....	49

MATLAB code: parseTestLog.m	52
MATLAB code: plotPpgSignals.m.....	54
MATLAB code: ppppg.m.....	57
MATLAB code: swapChannelData.m.....	58
MATLAB code: validatePpgPoints.m	58
Open Hardware PPG sensor design [57]	60

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	BHS Grade	Difference in calculated and actual BP value		
						≤ 5mmHg	≤ 10mmHg	≤ 15mmHg
PWV PSUS	DBP	N	8.07	8.99	B	38 (54.29%)	58 (82.86%)	64 (91.43%)
	SBP	N	10.95	9.73	D	26 (37.14%)	45 (64.29%)	56 (80.00%)
	MAP	N	8.82	9.12	C	33 (47.14%)	52 (74.29%)	63 (90.00%)
PWV SSUS	DBP	N	8.66	8.73	C	30 (42.86%)	52 (74.29%)	66 (94.29%)
	SBP	N	9.85	8.69	D	25 (35.71%)	49 (70.00%)	61 (87.14%)
	MAP	N	8.82	8.64	D	23 (32.86%)	55 (78.57%)	66 (94.29%)
PWV MSUS	DBP	N	7.60	8.69	B	35 (50.00%)	61 (87.14%)	69 (98.57%)
	SBP	N	9.64	8.70	D	25 (35.71%)	50 (71.43%)	63 (90.00%)
	MAP	N	8.11	8.56	C	32 (45.71%)	57 (81.43%)	67 (95.71%)
PWV Peak	DBP	N	7.40	8.72	B	35 (50.00%)	60 (85.71%)	69 (98.57%)
	SBP	N	8.78	8.71	D	27 (38.57%)	51 (72.86%)	64 (91.43%)
	MAP	N	7.66	8.56	C	34 (48.57%)	60 (85.71%)	70 (100.00%)

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	BHS Grade	Difference in calculated and actual BP value		
						≤ 5mmHg	≤ 10mmHg	≤ 15mmHg
PWV PSUS	DBP	N	8.05	8.91	B	37 (52.86%)	59 (84.29%)	66 (94.29%)
	SBP	N	10.91	9.69	D	27 (38.57%)	46 (65.71%)	57 (81.43%)
	MAP	N	8.80	8.99	C	32 (45.71%)	53 (75.71%)	65 (92.86%)
PWV SSUS	DBP	N	8.59	8.56	C	30 (42.86%)	55 (78.57%)	67 (95.71%)
	SBP	N	9.96	8.71	D	24 (34.29%)	46 (65.71%)	61 (87.14%)
	MAP	N	8.85	8.50	D	25 (35.71%)	53 (75.71%)	67 (95.71%)
PWV MSUS	DBP	N	7.79	8.61	C	32 (45.71%)	61 (87.14%)	70 (100.00%)
	SBP	N	9.30	8.74	D	25 (35.71%)	50 (71.43%)	63 (90.00%)
	MAP	N	8.11	8.51	C	34 (48.57%)	56 (80.00%)	69 (98.57%)
PWV Peak	DBP	N	8.38	8.64	D	26 (37.14%)	52 (74.29%)	68 (97.14%)
	SBP	N	9.18	8.76	D	24 (34.29%)	50 (71.43%)	64 (91.43%)
	MAP	N	8.39	8.57	C	32 (45.71%)	56 (80.00%)	68 (97.14%)

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	BHS Grade	Difference in calculated and actual BP value		
						≤ 5mmHg	≤ 10mmHg	≤ 15mmHg
PWV PSUS	DBP	N	8.38	8.84	C	32 (45.71%)	56 (80.00%)	66 (94.29%)
	SBP	N	10.30	8.69	D	16 (22.86%)	48 (68.57%)	59 (84.29%)
	MAP	N	8.76	8.71	D	26 (37.14%)	54 (77.14%)	67 (95.71%)
PWV SSUS	DBP	N	9.70	8.30	D	18 (25.71%)	45 (64.29%)	65 (92.86%)
	SBP	N	10.30	8.81	D	24 (34.29%)	43 (61.43%)	58 (82.86%)
	MAP	N	9.64	8.40	D	22 (31.43%)	44 (62.86%)	65 (92.86%)
PWV MSUS	DBP	N	8.85	8.46	D	27 (38.57%)	50 (71.43%)	68 (97.14%)
	SBP	N	10.37	8.67	D	20 (28.57%)	45 (64.29%)	60 (85.71%)
	MAP	N	9.05	8.38	D	27 (38.57%)	51 (72.86%)	67 (95.71%)
PWV Peak	DBP	N	10.46	8.34	D	21 (30.00%)	46 (65.71%)	62 (88.57%)
	SBP	N	10.93	8.97	D	22 (31.43%)	40 (57.14%)	57 (81.43%)
	MAP	N	10.44	8.41	D	20 (28.57%)	40 (57.14%)	64 (91.43%)

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	BHS Grade	Difference in calculated and actual BP value		
						≤ 5mmHg	≤ 10mmHg	≤ 15mmHg
PWV PSUS	DBP	N	6.26	5.11	B	41 (52.56%)	63 (80.77%)	71 (91.03%)
	SBP	N	9.55	7.27	D	28 (35.90%)	49 (62.82%)	60 (76.92%)
	MAP	N	7.13	5.68	C	35 (44.87%)	56 (71.79%)	70 (89.74%)
PWV SSUS	DBP	N	6.46	4.61	C	36 (46.15%)	60 (76.92%)	74 (94.87%)
	SBP	N	7.81	4.99	D	27 (34.62%)	57 (73.08%)	69 (88.46%)
	MAP	N	6.61	4.53	D	29 (37.18%)	63 (80.77%)	74 (94.87%)
PWV MSUS	DBP	N	5.49	3.98	C	38 (48.72%)	68 (87.18%)	77 (98.72%)
	SBP	N	7.61	4.92	D	27 (34.62%)	58 (74.36%)	71 (91.03%)
	MAP	N	6.01	3.97	C	36 (46.15%)	65 (83.33%)	75 (96.15%)
PWV Peak	DBP	N	5.36	3.96	C	38 (48.72%)	67 (85.90%)	77 (98.72%)
	SBP	N	6.98	4.59	D	29 (37.18%)	58 (74.36%)	72 (92.31%)
	MAP	N	5.68	3.73	C	37 (47.44%)	67 (85.90%)	78 (100.00%)

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	BHS Grade	Difference in calculated and actual BP value		
						≤ 5mmHg	≤ 10mmHg	≤ 15mmHg
PWV PSUS	DBP	N	8.43	8.65	B	40 (51.28%)	65 (83.33%)	73 (93.59%)
	SBP	N	11.21	9.33	D	29 (37.18%)	50 (64.10%)	61 (78.21%)
	MAP	N	9.17	8.72	C	34 (43.59%)	57 (73.08%)	72 (92.31%)
PWV SSUS	DBP	N	8.62	8.30	C	35 (44.87%)	63 (80.77%)	75 (96.15%)
	SBP	N	9.88	8.37	D	26 (33.33%)	54 (69.23%)	69 (88.46%)
	MAP	N	8.85	8.25	D	30 (38.46%)	61 (78.21%)	75 (96.15%)
PWV MSUS	DBP	N	8.04	8.33	C	35 (44.87%)	68 (87.18%)	78 (100.00%)
	SBP	N	9.43	8.37	D	27 (34.62%)	57 (73.08%)	71 (91.03%)
	MAP	N	8.33	8.23	C	37 (47.44%)	63 (80.77%)	77 (98.72%)
PWV Peak	DBP	N	8.48	8.36	D	31 (39.74%)	59 (75.64%)	76 (97.44%)
	SBP	N	9.37	8.38	D	26 (33.33%)	57 (73.08%)	72 (92.31%)
	MAP	N	8.54	8.29	C	35 (44.87%)	63 (80.77%)	76 (97.44%)

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	BHS Grade	Difference in calculated and actual BP value		
						≤ 5mmHg	≤ 10mmHg	≤ 15mmHg
PWV PSUS	DBP	N	8.55	8.54	C	37 (47.44%)	63 (80.77%)	74 (94.87%)
	SBP	N	10.44	8.36	D	18 (23.08%)	52 (66.67%)	66 (84.62%)
	MAP	N	8.94	8.41	D	30 (38.46%)	61 (78.21%)	75 (96.15%)
PWV SSUS	DBP	N	9.52	8.12	D	23 (29.49%)	52 (66.67%)	73 (93.59%)
	SBP	N	10.19	8.50	D	27 (34.62%)	50 (64.10%)	66 (84.62%)
	MAP	N	9.53	8.16	D	28 (35.90%)	51 (65.38%)	73 (93.59%)
PWV MSUS	DBP	N	9.00	8.17	C	32 (41.03%)	57 (73.08%)	76 (97.44%)
	SBP	N	10.61	8.37	D	22 (28.21%)	49 (62.82%)	66 (84.62%)
	MAP	N	9.27	8.09	D	30 (38.46%)	57 (73.08%)	74 (94.87%)
PWV Peak	DBP	N	10.10	8.19	D	28 (35.90%)	54 (69.23%)	70 (89.74%)
	SBP	N	10.63	8.66	D	25 (32.05%)	48 (61.54%)	65 (83.33%)
	MAP	N	10.09	8.25	D	27 (34.62%)	48 (61.54%)	72 (92.31%)

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	Difference in calculated and actual BP value		
					BHS Grade	≤ 5mmHg	≤ 10mmHg
PWV PSUS	DBP	N	8.98	9.28	D	42 (45.16%)	66 (70.97%)
	SBP	N	12.52	12.25	D	29 (31.18%)	54 (58.06%)
	MAP	N	9.78	9.98	D	37 (39.78%)	60 (64.52%)
PWV SSUS	DBP	N	9.93	10.16	D	36 (38.71%)	61 (65.59%)
	SBP	N	11.41	12.15	D	28 (30.11%)	59 (63.44%)
	MAP	N	9.97	10.58	D	30 (32.26%)	67 (72.04%)
PWV MSUS	DBP	N	8.66	9.42	C	39 (41.94%)	70 (75.27%)
	SBP	N	11.10	11.75	D	28 (30.11%)	60 (64.52%)
	MAP	N	9.07	9.93	D	38 (40.86%)	69 (74.19%)
PWV Peak	DBP	N	8.19	8.89	C	40 (43.01%)	70 (75.27%)
	SBP	N	10.43	11.46	D	29 (31.18%)	61 (65.59%)
	MAP	N	8.52	9.44	D	41 (44.09%)	71 (76.34%)

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	Difference in calculated and actual BP value		
					BHS Grade	≤ 5mmHg	≤ 10mmHg
PWV PSUS	DBP	N	8.80	9.01	D	42 (45.16%)	68 (73.12%)
	SBP	N	12.48	12.21	D	30 (32.26%)	55 (59.14%)
	MAP	N	9.66	9.74	D	36 (38.71%)	61 (65.59%)
PWV SSUS	DBP	N	9.32	9.04	D	36 (38.71%)	65 (69.89%)
	SBP	N	11.36	11.80	D	27 (29.03%)	57 (61.29%)
	MAP	N	9.56	9.75	D	32 (34.41%)	65 (69.89%)
PWV MSUS	DBP	N	8.44	8.70	D	37 (39.78%)	71 (76.34%)
	SBP	N	10.72	11.33	D	27 (29.03%)	60 (64.52%)
	MAP	N	8.79	9.30	C	41 (44.09%)	67 (72.04%)
PWV Peak	DBP	N	8.68	8.41	D	33 (35.48%)	63 (67.74%)
	SBP	N	10.59	11.23	D	26 (27.96%)	61 (65.59%)
	MAP	N	8.87	9.07	C	39 (41.94%)	67 (72.04%)

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	BHS Grade	Difference in calculated and actual BP value		
						≤ 5mmHg	≤ 10mmHg	≤ 15mmHg
PWV PSUS	DBP	N	8.61	8.39	D	39 (41.94%)	67 (72.04%)	79 (84.95%)
	SBP	N	11.54	11.07	D	19 (20.43%)	57 (61.29%)	71 (76.34%)
	MAP	N	9.19	8.95	D	33 (35.48%)	65 (69.89%)	81 (87.10%)
PWV SSUS	DBP	N	9.26	7.37	D	26 (27.96%)	57 (61.29%)	78 (83.87%)
	SBP	N	10.97	10.48	D	30 (32.26%)	54 (58.06%)	72 (77.42%)
	MAP	N	9.44	8.06	D	31 (33.33%)	56 (60.22%)	78 (83.87%)
PWV MSUS	DBP	N	8.60	7.50	D	36 (38.71%)	62 (66.67%)	82 (88.17%)
	SBP	N	11.40	10.84	D	25 (26.88%)	54 (58.06%)	71 (76.34%)
	MAP	N	9.11	8.15	C	33 (35.48%)	61 (65.59%)	82 (88.17%)
PWV Peak	DBP	N	9.96	7.57	D	30 (32.26%)	58 (62.37%)	75 (80.65%)
	SBP	N	11.48	10.61	D	27 (29.03%)	52 (55.91%)	70 (75.27%)
	MAP	N	10.09	8.25	D	30 (32.26%)	53 (56.99%)	77 (82.80%)

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	BHS Grade	Difference in calculated and actual BP value		
						≤ 5mmHg	≤ 10mmHg	≤ 15mmHg
PWV PSUS	DBP	Y	4.44	3.24	B	32 (58.18%)	53 (96.36%)	54 (98.18%)
	SBP	N	9.72	7.48	D	19 (34.55%)	31 (56.36%)	43 (78.18%)
	MAP	Y	5.77	3.98	B	28 (50.91%)	47 (85.45%)	54 (98.18%)
PWV SSUS	DBP	Y	3.99	2.79	A	35 (63.64%)	53 (96.36%)	55 (100.00%)
	SBP	N	5.84	3.95	C	27 (49.09%)	45 (81.82%)	54 (98.18%)
	MAP	Y	4.37	2.73	A	33 (60.00%)	54 (98.18%)	55 (100.00%)
PWV MSUS	DBP	Y	4.09	2.73	A	36 (65.45%)	53 (96.36%)	55 (100.00%)
	SBP	N	6.15	4.07	C	26 (47.27%)	43 (78.18%)	53 (96.36%)
	MAP	Y	4.51	2.74	B	31 (56.36%)	53 (96.36%)	55 (100.00%)
PWV Peak	DBP	Y	4.2	3.00	A	35 (63.64%)	53 (96.36%)	55 (100.00%)
	SBP	N	6.15	4.36	C	25 (45.45%)	46 (83.64%)	54 (98.18%)
MAP	N	4.34	3.10	A	34 (61.82%)	51 (92.73%)	55 (100.00%)	0

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	BHS Grade	Difference in calculated and actual BP value		
						≤ 5mmHg	≤ 10mmHg	≤ 15mmHg
PWV PSUS	DBP	Y	4.76	3.54	B	32 (58.18%)	52 (94.55%)	54 (98.18%)
	SBP	N	10.18	8.04	D	19 (34.55%)	31 (56.36%)	43 (78.18%)
	MAP	N	4.29	4.29	B	28 (50.91%)	44 (80.00%)	53 (96.36%)
PWV SSUS	DBP	Y	4.17	2.87	B	32 (58.18%)	53 (96.36%)	55 (100.00%)
	SBP	N	6.05	4.00	C	24 (43.64%)	45 (81.82%)	54 (98.18%)
	MAP	Y	4.54	2.78	A	34 (61.82%)	53 (96.36%)	55 (100.00%)
PWV MSUS	DBP	Y	4.34	2.96	B	31 (56.36%)	53 (96.36%)	55 (100.00%)
	SBP	N	6.65	4.64	C	24 (43.64%)	43 (78.18%)	52 (94.55%)
	MAP	Y	4.76	3.16	B	31 (56.36%)	53 (96.36%)	55 (100.00%)
PWV Peak	DBP	Y	4.45	3.23	A	35 (63.64%)	51 (92.73%)	55 (100.00%)
	SBP	N	6.25	4.40	C	25 (45.45%)	45 (81.82%)	54 (98.18%)
MAP	Y	4.48	3.23	A	33 (60.00%)	51 (92.73%)	55 (100.00%)	0

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	BHS Grade	Difference in calculated and actual BP value		
						≤ 5mmHg	≤ 10mmHg	≤ 15mmHg
PWV PSUS	DBP	N	6.22	4.81	C	26 (47.27%)	45 (81.82%)	52 (94.55%)
	SBP	N	11.61	9.27	D	17 (30.91%)	27 (49.09%)	37 (67.27%)
	MAP	N	7.59	5.89	C	23 (41.82%)	38 (69.09%)	49 (89.09%)
PWV SSUS	DBP	Y	4.93	3.44	B	30 (54.55%)	48 (87.27%)	55 (100.00%)
	SBP	N	7.56	5.28	D	21 (38.18%)	37 (67.27%)	51 (92.73%)
	MAP	N	5.36	3.52	B	29 (52.73%)	49 (89.09%)	55 (100.00%)
PWV MSUS	DBP	N	5.72	3.68	C	27 (49.09%)	45 (81.82%)	55 (100.00%)
	SBP	N	10.17	7.59	D	15 (27.27%)	31 (56.36%)	40 (72.73%)
	MAP	N	6.68	4.63	C	24 (43.64%)	42 (76.36%)	53 (96.36%)
PWV Peak	DBP	N	5.46	3.94	C	26 (47.27%)	47 (85.45%)	55 (100.00%)
	SBP	N	6.12	4.34	C	25 (45.45%)	43 (78.18%)	54 (98.18%)
	MAP	N	5.18	3.74	B	30 (54.55%)	48 (87.27%)	55 (100.00%)

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	BHS Grade	Difference in calculated and actual BP value		
						≤ 5mmHg	≤ 10mmHg	≤ 15mmHg
PWV PSUS	DBP	Y	4.88	3.57	B	34 (53.97%)	59 (93.65%)	61 (96.83%)
	SBP	N	10.14	7.51	D	20 (31.75%)	34 (53.97%)	47 (74.60%)
	MAP	N	6.18	4.32	C	30 (47.62%)	51 (80.95%)	61 (96.83%)
PWV SSUS	DBP	Y	4.61	3.41	B	36 (57.14%)	59 (93.65%)	62 (98.41%)
	SBP	N	6.64	4.41	C	27 (42.86%)	47 (74.60%)	61 (96.83%)
	MAP	N	5.08	3.41	B	34 (53.97%)	58 (92.06%)	62 (98.41%)
PWV MSUS	DBP	Y	4.77	3.45	B	37 (58.73%)	59 (93.65%)	62 (98.41%)
	SBP	N	7.04	4.66	C	26 (41.27%)	45 (71.43%)	57 (90.48%)
	MAP	N	5.29	3.56	B	32 (50.79%)	55 (87.30%)	62 (98.41%)
PWV Peak	DBP	Y	4.68	3.34	B	37 (58.73%)	59 (93.65%)	63 (100.00%)
	SBP	N	6.81	4.57	D	25 (39.68%)	48 (76.19%)	61 (96.83%)
	MAP	Y	4.95	3.49	B	35 (55.56%)	56 (88.89%)	63 (100.00%)

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	BHS Grade	Difference in calculated and actual BP value		
						≤ 5mmHg	≤ 10mmHg	≤ 15mmHg
PWV PSUS	DBP	N	5.07	3.77	B	34 (53.97%)	58 (92.06%)	61 (96.83%)
	SBP	N	10.58	8.04	D	20 (31.75%)	34 (53.97%)	47 (74.60%)
	MAP	N	6.47	4.59	C	31 (49.21%)	48 (76.19%)	60 (95.24%)
PWV SSUS	DBP	Y	4.78	3.50	B	33 (52.38%)	59 (93.65%)	62 (98.41%)
	SBP	N	6.90	4.55	D	24 (38.10%)	47 (74.60%)	58 (92.06%)
	MAP	N	5.26	3.51	B	35 (55.56%)	57 (90.48%)	62 (98.41%)
PWV MSUS	DBP	N	5.01	3.67	B	32 (50.79%)	58 (92.06%)	62 (98.41%)
	SBP	N	7.64	5.30	D	24 (38.10%)	45 (71.43%)	56 (88.89%)
	MAP	N	5.58	3.97	B	32 (50.79%)	56 (88.89%)	62 (98.41%)
PWV Peak	DBP	Y	4.82	3.43	B	37 (58.73%)	58 (92.06%)	63 (100.00%)
	SBP	N	6.90	4.59	D	25 (39.68%)	47 (74.60%)	61 (96.83%)
	MAP	N	5.02	3.52	B	34 (53.97%)	57 (90.48%)	63 (100.00%)

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	BHS Grade	Difference in calculated and actual BP value		
						≤ 5mmHg	≤ 10mmHg	≤ 15mmHg
PWV PSUS	DBP	N	6.31	4.85	C	29 (46.03%)	51 (80.95%)	59 (93.65%)
	SBP	N	11.92	9.18	D	19 (30.16%)	29 (46.03%)	41 (65.08%)
	MAP	N	7.76	5.97	D	25 (39.68%)	42 (66.67%)	56 (88.89%)
PWV SSUS	DBP	N	5.37	3.76	B	32 (50.79%)	54 (85.71%)	62 (98.41%)
	SBP	N	8.40	5.84	D	21 (33.33%)	40 (63.49%)	55 (87.30%)
	MAP	N	5.95	4.10	C	30 (47.62%)	53 (84.13%)	61 (96.83%)
PWV MSUS	DBP	N	6.25	4.12	C	28 (44.44%)	50 (79.37%)	62 (98.41%)
	SBP	N	11.12	8.04	D	15 (23.81%)	33 (52.38%)	44 (69.84%)
	MAP	N	7.38	5.20	D	25 (39.68%)	45 (71.43%)	59 (93.65%)
PWV Peak	DBP	N	5.50	3.85	C	29 (46.03%)	54 (85.71%)	63 (100.00%)
	SBP	N	6.63	4.41	D	25 (39.68%)	45 (71.43%)	62 (98.41%)
	MAP	N	5.42	3.73	C	31 (49.21%)	55 (87.30%)	63 (100.00%)

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	BHS Grade	Difference in calculated and actual BP value		
						≤ 5mmHg	≤ 10mmHg	≤ 15mmHg
PWV PSUS	DBP	N	8.86	10.28	D	34 (43.59%)	61 (78.21%)	65 (83.33%)
	SBP	N	13.81	13.29	D	22 (28.21%)	36 (46.15%)	53 (67.95%)
	MAP	N	9.82	10.84	D	32 (41.03%)	55 (70.51%)	65 (83.33%)
PWV SSUS	DBP	N	8.40	10.07	C	38 (48.72%)	61 (78.21%)	67 (85.90%)
	SBP	N	11.16	13.11	D	28 (35.90%)	48 (61.54%)	67 (85.90%)
	MAP	N	8.92	10.82	C	36 (46.15%)	62 (79.49%)	67 (85.90%)
PWV MSUS	DBP	N	8.52	10.07	C	39 (50.00%)	61 (78.21%)	67 (85.90%)
	SBP	N	11.39	12.94	D	26 (33.33%)	47 (60.26%)	63 (80.77%)
	MAP	N	9.03	10.77	C	35 (44.87%)	59 (75.64%)	67 (85.90%)
PWV Peak	DBP	N	8.87	10.56	C	37 (47.44%)	61 (78.21%)	67 (85.90%)
	SBP	N	11.55	13.68	D	26 (33.33%)	50 (64.10%)	66 (84.62%)
	MAP	N	9.19	11.43	C	37 (47.44%)	60 (76.92%)	67 (85.90%)

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	BHS Grade	Difference in calculated and actual BP value		
						≤ 5mmHg	≤ 10mmHg	≤ 15mmHg
PWV PSUS	DBP	N	8.86	10.03	D	35 (44.87%)	60 (76.92%)	65 (83.33%)
	SBP	N	14.16	13.46	D	22 (28.21%)	36 (46.15%)	53 (67.95%)
	MAP	N	9.97	10.70	D	33 (42.31%)	52 (66.67%)	65 (83.33%)
PWV SSUS	DBP	N	8.18	9.62	C	35 (44.87%)	62 (79.49%)	67 (85.90%)
	SBP	N	11.25	12.83	D	24 (30.77%)	49 (62.82%)	64 (82.05%)
	MAP	N	8.81	10.41	C	38 (48.72%)	61 (78.21%)	67 (85.90%)
PWV MSUS	DBP	N	8.40	9.75	C	34 (43.59%)	61 (78.21%)	67 (85.90%)
	SBP	N	11.72	12.87	D	24 (30.77%)	49 (62.82%)	61 (78.21%)
	MAP	N	9.01	10.55	C	34 (43.59%)	61 (78.21%)	67 (85.90%)
PWV Peak	DBP	N	8.87	10.35	C	37 (47.44%)	60 (76.92%)	67 (85.90%)
	SBP	N	11.64	13.72	D	26 (33.33%)	49 (62.82%)	64 (82.05%)
	MAP	N	9.19	11.32	C	36 (46.15%)	61 (78.21%)	67 (85.90%)

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	BHS Grade	Difference in calculated and actual BP value		
						≤ 5mmHg	≤ 10mmHg	≤ 15mmHg
PWV PSUS	DBP	N	9.41	9.44	D	31 (39.74%)	54 (69.23%)	64 (82.05%)
	SBP	N	15.13	13.70	D	20 (25.64%)	31 (39.74%)	46 (58.97%)
	MAP	N	10.77	10.46	D	27 (34.62%)	46 (58.97%)	62 (79.49%)
PWV SSUS	DBP	N	8.04	8.70	C	35 (44.87%)	59 (75.64%)	67 (85.90%)
	SBP	N	11.81	12.07	D	24 (30.77%)	44 (56.41%)	61 (78.21%)
	MAP	N	8.70	9.51	D	34 (43.59%)	57 (73.08%)	66 (84.62%)
PWV MSUS	DBP	N	8.71	9.01	D	31 (39.74%)	55 (70.51%)	67 (85.90%)
	SBP	N	14.18	13.73	D	17 (21.79%)	38 (48.72%)	49 (62.82%)
	MAP	N	9.87	10.26	D	29 (37.18%)	49 (62.82%)	67 (85.90%)
PWV Peak	DBP	N	9.09	9.70	D	31 (39.74%)	56 (71.79%)	67 (85.90%)
	SBP	N	11.18	13.14	D	26 (33.33%)	46 (58.97%)	68 (87.18%)
MAP	N	9.17	10.69	C	33 (42.31%)	59 (75.64%)	67 (85.90%)	11

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	BHS Grade	Difference in calculated and actual BP value		
						≤ 5mmHg	≤ 10mmHg	≤ 15mmHg
PWV PSUS	DBP	Y	3.93	2.80	A	106 (67.52%)	150 (95.54%)	157 (100.00%)
	SBP	N	5.50	4.31	B	83 (52.87%)	134 (85.35%)	152 (96.82%)
	MAP	Y	3.94	2.80	A	107 (68.15%)	151 (96.18%)	157 (100.00%)
PWV SSUS	DBP	Y	3.95	2.89	A	108 (68.79%)	151 (96.18%)	157 (100.00%)
	SBP	N	10.53	8.96	D	55 (35.03%)	86 (54.78%)	117 (74.52%)
	MAP	N	5.22	3.96	B	85 (54.14%)	137 (87.26%)	154 (98.09%)
PWV MSUS	DBP	Y	4.06	2.96	A	105 (66.88%)	149 (94.90%)	157 (100.00%)
	SBP	N	8.85	7.06	D	63 (40.13%)	94 (59.87%)	129 (82.17%)
	MAP	N	5.02	3.60	B	93 (59.24%)	141 (89.81%)	156 (99.36%)
PWV Peak	DBP	Y	3.96	2.95	A	110 (70.06%)	149 (94.90%)	156 (99.36%)
	SBP	N	5.37	4.13	B	84 (53.50%)	132 (84.08%)	153 (97.45%)
	MAP	Y	3.84	2.92	A	112 (71.34%)	150 (95.54%)	157 (100.00%)

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	BHS Grade	Difference in calculated and actual BP value		
						≤ 5mmHg	≤ 10mmHg	≤ 15mmHg
PWV PSUS	DBP	Y	3.96	2.81	A	108 (68.79%)	149 (94.90%)	157 (100.00%)
	SBP	N	5.40	4.12	B	88 (56.05%)	133 (84.71%)	152 (96.82%)
	MAP	Y	3.91	2.79	A	106 (67.52%)	152 (96.82%)	157 (100.00%)
PWV SSUS	DBP	Y	4.05	2.97	A	104 (66.24%)	151 (96.18%)	157 (100.00%)
	SBP	N	10.81	9.29	D	54 (34.39%)	83 (52.87%)	117 (74.52%)
	MAP	N	5.45	4.18	C	78 (49.68%)	134 (85.35%)	154 (98.09%)
PWV MSUS	DBP	Y	4.21	3.10	A	101 (64.33%)	150 (95.54%)	157 (100.00%)
	SBP	N	8.68	6.84	D	62 (39.49%)	95 (60.51%)	129 (82.17%)
	MAP	N	5.14	3.67	B	90 (57.32%)	139 (88.54%)	156 (99.36%)
PWV Peak	DBP	Y	4.04	2.94	A	109 (69.43%)	150 (95.54%)	156 (99.36%)
	SBP	N	5.47	4.19	B	82 (52.23%)	133 (84.71%)	153 (97.45%)
	MAP	Y	3.85	2.92	A	111 (70.70%)	149 (94.90%)	157 (100.00%)

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	BHS Grade	Difference in calculated and actual BP value		
						≤ 5mmHg	≤ 10mmHg	≤ 15mmHg
PWV PSUS	DBP	Y	4.23	3.34	A	109 (69.43%)	143 (91.08%)	156 (99.36%)
	SBP	N	5.53	4.11	B	87 (55.41%)	127 (80.89%)	155 (98.73%)
	MAP	Y	4.24	3.03	A	101 (64.33%)	149 (94.90%)	156 (99.36%)
PWV SSUS	DBP	Y	4.94	3.48	B	82 (52.23%)	143 (91.08%)	156 (99.36%)
	SBP	N	7.23	5.03	D	61 (38.85%)	115 (73.25%)	146 (92.99%)
	MAP	Y	4.36	3.05	A	98 (62.42%)	148 (94.27%)	157 (100.00%)
PWV MSUS	DBP	Y	4.64	3.24	B	91 (57.96%)	148 (94.27%)	156 (99.36%)
	SBP	N	7.17	4.93	D	62 (39.49%)	117 (74.52%)	144 (91.72%)
	MAP	Y	4.55	3.13	A	95 (60.51%)	146 (92.99%)	157 (100.00%)
PWV Peak	DBP	Y	4.84	3.39	B	87 (55.41%)	146 (92.99%)	156 (99.36%)
	SBP	N	5.43	4.18	B	86 (54.78%)	131 (83.44%)	153 (97.45%)
	MAP	Y	4.48	3.04	A	98 (62.42%)	148 (94.27%)	157 (100.00%)

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	BHS Grade	Difference in calculated and actual BP value		
						≤ 5mmHg	≤ 10mmHg	≤ 15mmHg
PWV PSUS	DBP	Y	4.11	3.14	A	97 (68.31%)	135 (95.07%)	141 (99.30%)
	SBP	N	5.87	4.50	B	74 (52.11%)	110 (77.46%)	137 (96.48%)
	MAP	Y	4.27	3.01	A	93 (65.49%)	134 (94.37%)	141 (99.30%)
PWV SSUS	DBP	Y	3.94	2.97	A	93 (65.49%)	137 (96.48%)	142 (100.00%)
	SBP	N	6.26	4.32	C	70 (49.30%)	115 (80.99%)	136 (95.77%)
	MAP	Y	4.25	2.83	A	94 (66.20%)	137 (96.48%)	142 (100.00%)
PWV MSUS	DBP	Y	4.16	3.15	A	93 (65.49%)	135 (95.07%)	142 (100.00%)
	SBP	N	7.01	5.23	C	58 (40.85%)	102 (71.83%)	130 (91.55%)
	MAP	Y	4.66	3.27	A	86 (60.56%)	132 (92.96%)	142 (100.00%)
PWV Peak	DBP	Y	4.43	3.23	A	86 (60.56%)	135 (95.07%)	141 (99.30%)
	SBP	N	5.41	4.12	B	80 (56.34%)	119 (83.80%)	138 (97.18%)
	MAP	Y	4.24	2.90	A	88 (61.97%)	135 (95.07%)	142 (100.00%)

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	BHS Grade	Difference in calculated and actual BP value		
						≤ 5mmHg	≤ 10mmHg	≤ 15mmHg
PWV PSUS	DBP	Y	4.24	3.23	A	92 (64.79%)	132 (92.96%)	141 (99.30%)
	SBP	N	5.93	4.55	B	76 (53.52%)	111 (78.17%)	136 (95.77%)
	MAP	Y	4.35	3.08	A	93 (65.49%)	132 (92.96%)	141 (99.30%)
PWV SSUS	DBP	Y	4.08	3.13	A	91 (64.08%)	135 (95.07%)	142 (100.00%)
	SBP	N	5.73	4.04	B	73 (51.41%)	116 (81.69%)	139 (97.89%)
	MAP	Y	4.27	2.81	A	89 (62.68%)	135 (95.07%)	142 (100.00%)
PWV MSUS	DBP	Y	4.34	3.31	A	89 (62.68%)	132 (92.96%)	141 (99.30%)
	SBP	N	6.70	4.83	C	65 (45.77%)	107 (75.35%)	133 (93.66%)
	MAP	Y	4.69	3.30	B	82 (57.75%)	130 (91.55%)	142 (100.00%)
PWV Peak	DBP	Y	4.69	3.40	B	82 (57.75%)	133 (93.66%)	141 (99.30%)
	SBP	N	5.42	4.13	B	80 (56.34%)	118 (83.10%)	138 (97.18%)
MAP	Y	4.40	3.03	A	86 (60.56%)	133 (93.66%)	142 (100.00%)	0

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	BHS Grade	Difference in calculated and actual BP value		
						≤ 5mmHg	≤ 10mmHg	≤ 15mmHg
PWV PSUS	DBP	Y	4.94	3.96	B	81 (57.04%)	127 (89.44%)	139 (97.89%)
	SBP	N	6.13	4.71	B	72 (50.70%)	111 (78.17%)	135 (95.07%)
	MAP	Y	4.89	3.61	B	83 (58.45%)	127 (89.44%)	141 (99.30%)
PWV SSUS	DBP	N	6.32	3.91	C	59 (41.55%)	120 (84.51%)	139 (97.89%)
	SBP	N	5.93	4.22	B	72 (50.70%)	116 (81.69%)	138 (97.18%)
	MAP	N	5.29	3.62	B	71 (50.00%)	126 (88.73%)	142 (100.00%)
PWV MSUS	DBP	N	5.80	3.82	C	69 (48.59%)	121 (85.21%)	141 (99.30%)
	SBP	N	7.00	4.83	C	60 (42.25%)	108 (76.06%)	130 (91.55%)
	MAP	N	5.12	3.62	B	73 (51.41%)	127 (89.44%)	141 (99.30%)
PWV Peak	DBP	N	6.89	4.32	C	57 (40.14%)	106 (74.65%)	136 (95.77%)
	SBP	N	6.02	4.81	B	71 (50.00%)	110 (77.46%)	133 (93.66%)
	MAP	N	6.00	4.09	C	66 (46.48%)	121 (85.21%)	140 (98.59%)

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	BHS Grade	Difference in calculated and actual BP value		
						≤ 5mmHg	≤ 10mmHg	≤ 15mmHg
PWV PSUS	DBP	Y	4.20	2.96	A	79 (67.52%)	112 (95.73%)	116 (99.15%)
	SBP	N	6.15	4.45	C	57 (48.72%)	93 (79.49%)	112 (95.73%)
	MAP	Y	4.13	2.89	A	78 (66.67%)	112 (95.73%)	116 (99.15%)
PWV SSUS	DBP	Y	4.55	3.26	A	73 (62.39%)	110 (94.02%)	116 (99.15%)
	SBP	N	5.41	4.16	B	69 (58.97%)	96 (82.05%)	113 (96.58%)
	MAP	Y	4.33	2.93	A	72 (61.54%)	109 (93.16%)	117 (100.00%)
PWV MSUS	DBP	Y	4.45	3.19	A	74 (63.25%)	112 (95.73%)	116 (99.15%)
	SBP	N	5.48	4.07	B	66 (56.41%)	96 (82.05%)	113 (96.58%)
	MAP	Y	4.20	2.87	A	74 (63.25%)	110 (94.02%)	117 (100.00%)
PWV Peak	DBP	Y	4.38	3.13	A	75 (64.10%)	112 (95.73%)	116 (99.15%)
	SBP	N	5.66	4.14	B	59 (50.43%)	97 (82.91%)	115 (98.29%)
	MAP	Y	4.09	2.91	A	80 (68.38%)	111 (94.87%)	117 (100.00%)

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	BHS Grade	Difference in calculated and actual BP value		
						≤ 5mmHg	≤ 10mmHg	≤ 15mmHg
PWV PSUS	DBP	Y	4.18	3.02	A	77 (65.81%)	111 (94.87%)	116 (99.15%)
	SBP	N	6.31	4.61	C	52 (44.44%)	94 (80.34%)	110 (94.02%)
	MAP	Y	4.22	2.99	A	75 (64.10%)	110 (94.02%)	116 (99.15%)
PWV SSUS	DBP	Y	4.60	3.30	A	72 (61.54%)	110 (94.02%)	116 (99.15%)
	SBP	N	5.39	4.21	B	68 (58.12%)	98 (83.76%)	114 (97.44%)
	MAP	Y	4.39	2.97	A	73 (62.39%)	110 (94.02%)	117 (100.00%)
PWV MSUS	DBP	Y	4.51	3.22	A	74 (63.25%)	112 (95.73%)	116 (99.15%)
	SBP	N	5.43	4.09	B	68 (58.12%)	95 (81.20%)	113 (96.58%)
	MAP	Y	4.23	2.89	A	74 (63.25%)	110 (94.02%)	117 (100.00%)
PWV Peak	DBP	Y	4.46	3.17	A	77 (65.81%)	110 (94.02%)	116 (99.15%)
	SBP	N	5.68	4.15	C	58 (49.57%)	97 (82.91%)	115 (98.29%)
	MAP	Y	4.12	2.94	A	79 (67.52%)	111 (94.87%)	117 (100.00%)

PWV Type	BP	AAMI Valid	Diff. Mean	Diff. SD	BHS Grade	Difference in calculated and actual BP value		
						≤ 5mmHg	≤ 10mmHg	≤ 15mmHg
PWV PSUS	DBP	Y	4.61	3.61	A	71 (60.68%)	109 (93.16%)	116 (99.15%)
	SBP	N	6.75	5.16	C	51 (43.59%)	86 (73.50%)	107 (91.45%)
	MAP	Y	4.73	3.53	A	72 (61.54%)	106 (90.60%)	116 (99.15%)
PWV SSUS	DBP	N	6.35	4.16	C	47 (40.17%)	96 (82.05%)	114 (97.44%)
	SBP	N	5.51	4.20	B	66 (56.41%)	99 (84.62%)	113 (96.58%)
	MAP	N	5.36	3.56	B	61 (52.14%)	105 (89.74%)	116 (99.15%)
PWV MSUS	DBP	N	5.78	3.84	C	50 (42.74%)	101 (86.32%)	115 (98.29%)
	SBP	N	5.75	4.02	B	60 (51.28%)	95 (81.20%)	114 (97.44%)
	MAP	Y	4.81	3.33	B	64 (54.70%)	108 (92.31%)	117 (100.00%)
PWV Peak	DBP	N	5.60	3.73	C	50 (42.74%)	103 (88.03%)	115 (98.29%)
	SBP	N	5.57	4.11	B	64 (54.70%)	97 (82.91%)	114 (97.44%)
	MAP	Y	4.81	3.33	B	69 (58.97%)	105 (89.74%)	117 (100.00%)

MATLAB code: birrenkott.m

```
1 function [systolicTime, amplitude, peakToPeakTime, peakVariance] = birrenkott(signal, peaks, troughs)
2 %birrenkott passes supplied signal and analyses it
3 %Uses vectors peaks and troughs to calculate outputs
4 %
5 % see also fppp.m
6
7
8 systolicTime = peaks - troughs;
9
10 amplitude = signal(peaks) - signal(troughs);
11
12 peakToPeakTime = peaks(2:end) - peaks(1:end-1);
13
14 peakSignal = signal(peaks);
15
16 peakVariance = peakSignal(2:end) - peakSignal(1:end-1);
17
18 end
```

MATLAB code: compare.m

```
1 yt = yt + 1;
2 if yt == 4
3     xt = xt + 1;
4     yt = 1;
5     fprintf('\n');
6 end
7 if xt == 5
8     xt = 1;
9     yt = 0;
10    disp('done');
11    return;
12 end
13
14 yTypes = {'DBP', 'SBP', 'MAP'};
15 xTypes = {'PSUS', 'SSUS', 'MSUS', 'Peak'};
16
17 yType = yTypes{yt};
18 xType = xTypes{xt};
19 wType = 'None';
20
21 fitType = fittype('a*exp(b*x)');
22
23 % Fit Set: 25   Eval Set: 55 / 63 / 78
24 %fitSet = [CLResults(1:92)];
25 %compSet = [CLResults(93:245)];
26 %compSet = [CLResults(93:245), RIL2];
27 %compSet = [CLResults(93:245), RIL2, ChinaResults];
28
29 %Fit Set: 10   Eval Set: 70 / 78 / 93
30 %fitSet = [CLResults(1:32)];
31 %compSet = [CLResults(33:245)]; %70
32 %compSet = [CLResults(33:245), RIL2]; % 78
33 %compSet = [CLResults(33:245), RIL2, ChinaResults]; % 93
34
35
36 % New - with byte swapped tests
37 %fitSet = [CLResults(1:16)]; % 20
38 %compSet = [CLResults(17:245)]; %157
39
40
41 % New - with byte swapped tests
42 fitSet = [CLResults(1:42)]; % 25
```

```

43 compSet = [CLResults(43:245)]; %142
44
45 % New - with byte swapped tests
46 %fitSet = [CLResults(1:80)]; % 50
47 %compSet = [CLResults(81:245)]; %117
48
49
50
51
52
53
54 % fitSet = [ZIL2(41:50)];
55 % compSet = [ZIL2(51:end)];
56
57 fitSetValid = find([fitSet.valid]);
58 compSetValid = find([compSet.valid]);
59 chinaValid = find([ChinaResults.valid]);
60 initValid = find([RIL2.valid]);
61
62 switch wType
63   case 'None'
64     fitSetWeights = ones(length(fitSetValid), 1);
65   case 'Simple'
66     fitSetWeights = 11 - ([fitSet(fitSetValid).diffHR]');
67   case 'Recip'
68     fitSetWeights = 1./([fitSet(fitSetValid).diffHR]');
69   otherwise
70     error('PPG:bjmCompare:yTypeInvalid','Invalid yType');
71 end
72
73
74
75 switch xType
76   case 'MSUS'
77     fitSetX = [fitSet(fitSetValid).pwwMsusMean]';
78     compSetX = [compSet(compSetValid).pwwMsusMean]';
79     chinaX = [ChinaResults(chinaValid).pwwMsusMean]';
80     initX = [RIL2(initValid).pwwMsusMean]';
81   case 'SSUS'
82     fitSetX = [fitSet(fitSetValid).pwwSsusMean]';
83     compSetX = [compSet(compSetValid).pwwSsusMean]';
84     chinaX = [ChinaResults(chinaValid).pwwSsusMean]';
85     initX = [RIL2(initValid).pwwSsusMean]';
86   case 'PSUS'
87     fitSetX = [fitSet(fitSetValid).pwwPsusMean]';
88     compSetX = [compSet(compSetValid).pwwPsusMean]';
89     chinaX = [ChinaResults(chinaValid).pwwPsusMean]';
90     initX = [RIL2(initValid).pwwPsusMean]';
91   case 'Peak'
92     fitSetX = [fitSet(fitSetValid).pwwPeakMean]';
93     compSetX = [compSet(compSetValid).pwwPeakMean]';
94     chinaX = [ChinaResults(chinaValid).pwwPeakMean]';
95     initX = [RIL2(initValid).pwwPeakMean]';
96   otherwise
97     error('PPG:bjmCompare:xTypeInvalid','Invalid yType');
98 end
99
100 switch yType
101   case 'DBP'
102     fitSetY = [fitSet(fitSetValid).recDBP]';
103     compSetY = [compSet(compSetValid).recDBP]';
104     chinaY = [ChinaResults(chinaValid).recDBP]';
105     initY = [RIL2(initValid).recDBP]';
106   case 'SBP'
107     fitSetY = [fitSet(fitSetValid).recSBP]';
108     compSetY = [compSet(compSetValid).recSBP]';
109     chinaY = [ChinaResults(chinaValid).recSBP]';
110     initY = [RIL2(initValid).recSBP]';
111   case 'MAP'
112     fitSetY = [fitSet(fitSetValid).MAP]';
113     compSetY = [compSet(compSetValid).MAP]';
114     chinaY = [ChinaResults(chinaValid).MAP]';
115     initY = [RIL2(initValid).MAP]';
116   otherwise

```

```

117      error('PPG:bjmCompare:yTypeInvalid','Invalid yType');
118  end
119
120 [f, gof] = fit(fitSetX, fitSetY, fitType, 'Weight', fitSetWeights, 'StartPoint', [67.638 0.191]);
121 %[f, gof] = fit(fitSetX, fitSetY, fitType, 'StartPoint', [67.638 0.191]);
122
123
124 figure('Name', sprintf('PWV-%s to %s', xType, yType), 'NumberTitle','off');
125
126 hold on;
127 legend('hide');
128
129 %plot(chinaX, chinaY, 'mo');
130 %plot(initX, initY, 'co');
131
132 plot(fitSetX, fitSetY, 'ro');
133 plot(compSetX, compSetY, 'b.');
134
135 fp15 = plot(f);
136 fp10 = plot(f);
137 fp05 = plot(f);
138 fm15 = plot(f);
139 fm10 = plot(f);
140 fm05 = plot(f);
141
142 set(fp15, 'YData', get(fp15, 'YData') + 15);
143 set(fm15, 'YData', get(fm15, 'YData') - 15);
144 set(fp15, 'Color', [1 0.1 0.1]);
145 set(fm15, 'Color', [1 0.1 0.1]);
146 %set(get(get(fm15,'Annotation'),'LegendInformation'), 'IconDisplayStyle','off');
147 %set(fm15, 'IconDisplayStyle','off');
148
149
150 set(fp10, 'YData', get(fp10, 'YData') + 10);
151 set(fm10, 'YData', get(fm10, 'YData') - 10);
152 set(fp10, 'Color', [0.5 0.5 0]);
153 set(fm10, 'Color', [0.5 0.5 0]);
154 %set(get(get(fm10,'Annotation'),'LegendInformation'), 'IconDisplayStyle','off');
155 %set(fm10, 'DisplayName', 'TEST');
156 %set(fm10, 'IconDisplayStyle','off');
157
158 set(fp05, 'YData', get(fp05, 'YData') + 5);
159 set(fm05, 'YData', get(fm05, 'YData') - 5);
160 set(fp05, 'Color', [0.1 1 0.1]);
161 set(fm05, 'Color', [0.1 1 0.1]);
162 %set(get(get(fm5,'Annotation'),'LegendInformation'), 'IconDisplayStyle','off');
163 %set(fm5, 'IconDisplayStyle','off');
164 %plot(f, 'k');

165
166
167 plot(f, 'k');
168 xlabel(sprintf('PWV-%s (cm/ms)', xType));
169 ylabel(sprintf('%s (mmHg)', yType));
170 legend('hide');
171
172
173 lt05 = 0;
174 lt10 = 0;
175 lt15 = 0;
176 gt15 = 0;
177
178 len = length(compSetX);
179 diffSet(len) = -1;
180 for ii = 1 : len
181     diff = abs(compSetY(ii) - f(compSetX(ii)));
182     diffSet(ii) = diff;
183     if diff <= 5
184         lt05 = lt05 + 1;
185     end
186     if diff <= 10
187         lt10 = lt10 + 1;
188     end
189     if diff <= 15
190         lt15 = lt15 + 1;

```

```

191     end
192     if diff > 15
193         gt15 = gt15 + 1;
194     end
195 end
196
197 diffSetMean = mean(diffSet);
198 diffSetSD = std(diffSet);
199 aamiPass = diffSetMean <= 5 && diffSetSD <= 8;
200
201 pcLt05 = (lt05/len) *100;
202 pcLt10 = (lt10/len) *100;
203 pcLt15 = (lt15/len) *100;
204
205
206
207 if pcLt05 >= 60 && pcLt10 >= 85 && pcLt15 >= 95
208     grade = 'A';
209 elseif pcLt05 >= 50 && pcLt10 >= 75 && pcLt15 >= 90
210     grade = 'B';
211 elseif pcLt05 >= 40 && pcLt10 >= 65 && pcLt15 >= 85
212     grade = 'C';
213 else
214     grade = 'D';
215 end
216
217 if aamiPass
218     aamiText = 'AAMI PASS';
219 else
220     aamiText = '';
221 end
222 desc = sprintf(['PWV-%s to %s      Weight Type: %s    Fit Set: %d    Eval Set: %d    AAMI : %.2f (%.2f)
223 %s\n' ...
224 'Grade: %s    <=5mmHg: %d (%.2f%)    <=10mmHg: %d (%.2f%)    <=15mmHg: %d (%.2f%)    >15mmHg:
225 %d'], ...
226     xType, yType, wType, length(fitSetValid), len, diffSetMean, diffSetSD, aamiText, ...
227     grade, lt05, pcLt05, lt10, pcLt10, lt15, pcLt15, gt15);
228
229 title(desc);
230 disp(desc);
231 %disp(coeffvalues(f));
232 %disp(gof);
233
234 clipboard('copy', sprintf('.2f',diffSetMean));

```

MATLAB code: createEmptyPpgAxis.m

```

1 function gHandle = createEmptyPpgAxis(Data, gHandle)
2 % createEmptyPpgFigure create the PPG figure setting axis and title etc and
3 % returns a handle to the figure.
4 % Data param is either a LogEntry or PpgData struct needed to determine the
5 % number of samples (ie the x-axis).
6 % figHandle = createEmptyPpgFigure(LogEntry).
7 %
8 % See also createLogStructs
9
10 [PpgData, dataParamIsLogEntry] = extractPpgData(Data);
11
12
13 % validate / set gHandle. If not param then use current figure
14 % can be figure handle or subplot handle
15 if ~exist('gHandle', 'var')
16     % don't have to do anything here.. just uses gcf
17 elseif ~(isgraphics(gHandle, 'figure') || isgraphics(gHandle, 'axes'))
18     error('PPG:createEmptyPpgAxis:invalidFigureHandle', 'Invalid graphics handle. Cannot create
19 axis');
20 elseif isgraphics(gHandle, 'figure')

```

```

21      figure(gHandle);
22  elseif isgraphics(gHandle, 'axes')
23      subplot(gHandle);
24  end
25
26
27 % set nSamples, ensuring data can be correctly read or throw and error
28 try
29     nSamples = length(PpgData(1).raw);
30 catch
31     error('PPG:createEmptyPpgAxis:cannotReadPpgRaw', 'Unable to read raw data and/or it'''
32 length');
33 end
34
35
36 % create figure and set axis based on number of samples
37 axis([0 nSamples 1500 3600]);
38 xlabel("Time");
39 ylabel("Saturation");
40
41
42 % create x-tick markers at 1000ms == 1s intervals
43 xTicksPositions = 1000 : 1000 : nSamples;
44 xticks(xTicksPositions);
45 xTicksPositions = xTicksPositions./1000;
46
47
48 % label every 10s x-ticks if > 60s of data, otherwise label every 5s
49 if nSamples > 60000
50     labelSpacing = 10;
51 else
52     labelSpacing = 5;
53 end
54
55 xTickLabels = cell(length(xTicksPositions));
56 for i = 1:length(xTicksPositions)
57     if mod(i,labelSpacing) == 0
58         xTickLabels{i} = strcat(num2str(xTicksPositions(i)), 's');
59     else
60         xTickLabels{i} = '';
61     end
62 end
63
64 xticklabels(xTickLabels);
65 end

```

MATLAB code: createEmptyPpgFigure.m

```
1 function figHandle = createEmptyPpgFigure(LogEntry)
2 % createEmptyPpgFigure create the PPG figure setting axis and title etc and
3 % returns a handle to the figure
4 % figHandle = createEmptyPpgFigure(LogEntry)
5 %
6 % See also createLogStructs
7
8
9 % validate parameter
10 if ~exist("LogEntry", "var")
11     error('PPG:createEmptyPpgFigure:noLogEntry', 'No LogEntry parameter supplied');
12 end
13
14 % set nSamples, ensuring data can be correctly read or throw and error
15 try
16     nSamples = length(LogEntry.PpgData(1).raw);
17 catch
18     error('PPG:createEmptyPpgFigure:cannotReadPppRaw', 'Unable to read raw data and/or it''s
length');
19 end
20
21
22
23
24 % create figure and set axis based on number of samples
25 figHandle = figure;
26 hold on;
27 axis([0 nSamples 1500 3500]);
28 xlabel("Time");
29 ylabel("Saturation");
30
31 title(sprintf("ID: %d:%d - Timestamp: %s", ...
32     LogEntry.userId, LogEntry.testId, LogEntry.timestamp));
33
34
35 % create x-tick markers at 1000ms == 1s intervals
36 xTicksPositions = 1000 : 1000 : nSamples;
37 xticks(xTicksPositions);
38 xTicksPositions = xTicksPositions./1000;
39
40 % label every 10s x-ticks if > 60s of data, otherwise label every 5s
41 if nSamples > 60000
42     labelSpacing = 10;
43 else
44     labelSpacing = 5;
45 end
46
47 xTickLabels = cell(length(xTicksPositions));
48 for i = 1:length(xTicksPositions)
49     if mod(i,labelSpacing) == 0
50         xTickLabels{i} = strcat(num2str(xTicksPositions(i)), 's');
51     else
52         xTickLabels{i} = '';
53     end
54 end
55
56 xticklabels(xTickLabels);
57 end
```

MATLAB code: createLogStructs.m

```
1 function [LogEntry, PpgDataStruct] = createLogStructs()
2 % createLogStructs creates the structures used in new logging paradigm.
3 % [LogEntry, PpgChannel] = createLogStructs()
4 %
5 % See also parseTestLog
6
7 PpgDataStruct = struct('raw',[], 'filtered', [], 'd1', [], 'd2', [], ...
8     'sysPeakVals', [], 'sysPeakSamp', [], ...
9     'sysStartVals', [], 'sysStartSamp', []);
10
11 LogEntry = struct('testId',[], 'valid', [], 'comment', [], ...
12     'timestamp', [], 'userId', [], 'userName', [], ...
13     'armLength', [], 'recSBP', [], 'recDBP', [], 'recHR', [], ...
14     'sampleTime', [], 'PpgData', PpgDataStruct);
15 end
```

MATLAB code: extractPpgData.m

```
1 function [PpgData, dataParamIsLogEntry] = extractPpgData(Data)
2 % extractPpgData extracts the PpgData struct from supplied param
3 % Data param can be "full" test log (row) or
4 % Data could also be just the PogData struct.
5 % Returns PpgData struct and dataParamIsLogEntry depending on what was
6 % supplied as the Data param.
7
8 [LogEntryStruct, PpgDataStruct] = createLogStructs();
9
10 dataFieldNames = fieldnames(Data);
11
12
13 if isstruct(Data)
14     if isequal(dataFieldNames, fieldnames(PpgDataStruct))
15         PpgData = Data;
16         dataParamIsLogEntry = false;
17
18     elseif isequal(dataFieldNames, fieldnames(LogEntryStruct))
19         PpgData = Data.PpgData;
20         dataParamIsLogEntry = true;
21     elseif all(contains(fieldnames(LogEntryStruct), dataFieldNames))
22         PpgData = Data.PpgData;
23         dataParamIsLogEntry = true;
24         warning('PPG:extractPpgData:reprocessingLog', 'Reprocessing existing results log');
25     end
26
27 end
28
29
30 % Data was not LogEntry or PpgData so throw and error
31 if ~exist('PpgData','var')
32     error('PPG:extractPpgData:invalidDataParam', 'Invalild PPG Channel data.');
33 end
34 end
35
36
```

MATLAB code: findPpgPoints.m

```
1 function PpgMeta = findPpgPoints(PpgData)
2
3 PpgData = extractPpgData(PpgData);
4
5 if length(PpgData) ~= 1
6     error('PPG:findPpgPoints:invalidDataParam', 'Invalid PPG Channel data. Single channel only!');
7 end
8
9 d1 = PpgData.d1;
10 d2 = PpgData.d2;
11
12 % d1Peaks = [];
13 % for iWindow = 500 : 500: length(d1) - 1000
14 %     d1Window = d1(iWindow : iWindow + 1000);
15 %     [~,d1PeakWindow] = findpeaks(d1Window, 'MinPeakHeight', max(d1Window) * 0.5);
16 %     d1Peaks = vertcat(d1Peaks, d1PeakWindow + iWindow);
17 % end
18
19 % find peaks in the 1st derivative, ignoring the first 500 samples
20 % all? of these are the midpoint of the systolic up stroke (MSUS).
21 [~,d1Peaks] = findpeaks(d1(500:end), 'MinPeakHeight', max(d1(500:end)) * 0.5);
22 %~,d1Peaks = findpeaks(d1(500:end), 'MinPeakProminence', max(d1(500:end)) *0.333);
23 d1Peaks = d1Peaks + 500;
24 msus = d1Peaks;
25
26
27
28 % find peaks in the 2nd derivative, ignoring the first 500 samples
29 % many of these are the start of the systolic up stroke (SSUS).
30 [~,d2Peaks] = findpeaks(d2(500:end), 'MinPeakHeight', max(d2(500:end)) * 0.5);
31 %~,d2Peaks = findpeaks(d2(500:end), 'MinPeakProminence', max(d2(500:end)) *0.333);
32 d2Peaks = d2Peaks + 500;
33 ssus = d2Peaks;
34
35
36 % discard any 2nd derivative peaks that occur after last d1 peak
37 while ssus(end) > msus(end)
38     ssus(end) = [];
39 end
40
41
42 % one by one discard any d2 peaks not just before a d1 peak
43 % the SSUS should always occur just prior to MSUS.
44 while length(ssus) > length(msus)
45     for n = 1 : length(ssus)
46         if ssus(n) < msus(n) - 100
47             ssus(n) = [];
48             break;
49         end
50     end
51 end
52
53
54 % find the zero crossing in 1st derivative
55 d1zc = zeroCrossing(d1);
56
57 % join d1 zero crossing and msus points into one sorted vector
58 d1zcPlusMsus = sort(vertcat(d1zc,msus));
59
60 % find the indexes of the msus points
61 % the peaks and the psus points are either side of each msus.
62 iMuss = find(ismember(d1zcPlusMsus, msus));
63
64 % protect for out of bounds
65 if iMuss(end) + 1 > length(d1zcPlusMsus)
66     iMuss(end) = [];
67 end
```

```

69     if iMuss(1) - 1 < 1
70         iMuss(1) = [];
71     end
72
73
74     peaks = d1zcPlusMsus(iMuss + 1);
75
76     psus = d1zcPlusMsus(iMuss - 1);
77
78
79     PpgMeta.peaks = peaks;
80     PpgMeta.psus = psus;
81     PpgMeta.ssus = ssus;
82     PpgMeta.msus = msus;
83     PpgMeta.d1zc = d1zc;
84
85
86
87
88 % the following inspired by Steffen Brueckner's crossing.m
89 % https://au.mathworks.com/matlabcentral/fileexchange/2432-crossing
90 function ind = zeroCrossing(signal)
91     ind0 = find(signal == 0);
92     signal1 = signal(1 : end-1) .* signal(2 : end);
93     ind1 = find(signal1 < 0);
94     ind = sort([ind0 ind1]);
95 end
96
97 end

```

MATLAB code: fittest.m

```

1 function fittest(ResultsLog, validRows)
2
3     if ~exist('ResultsLog', 'var')
4         error('PPG:fittest:noResultsLog', 'Must supply a results log')
5     end
6
7     if ~exist('validRows', 'var')
8         v = [ResultsLog(:,valid)];
9         validRows = find(v);
10    end
11
12    emptyValidRows = find(arrayfun(@(ResultsLog) isempty(ResultsLog.valid), ResultsLog));
13
14    validRowsIsEmpty = isempty(validRows);
15
16    % there is a problem if there are valid rows, but also empty valid fields
17    if ~validRowsIsEmpty && ~isempty(emptyValidRows)
18        error('PPG:fittest:validRowsPlusEmptyValidFields', 'There are valid rows, but also empty valid
19 fields');
20    end
21
22    % if no validRows then check for and use any empty valid fields
23    % instead.
24    if validRowsIsEmpty
25        validRows = emptyValidRows;
26    end
27
28
29
30    %validRows = validRows(validRows ~= 204); % manually removing outlier.
31    disp(validRows);
32
33
34
35    figure();
36    doPttPlot(ResultsLog, validRows, [ResultsLog(validRows).pwvPsusMean]', "PSUS", 1);
37    doPttPlot(ResultsLog, validRows, [ResultsLog(validRows).pwvSsusMean]', "SSUS", 2);

```

```

38 doPttPlot(ResultsLog, validRows, [ResultsLog(validRows).pwvMsusMean]', "MSUS", 3);
39 doPttPlot(ResultsLog, validRows, [ResultsLog(validRows).pwvPeakMean]', "Peak", 4);
40
41 %     xy = [[ResultsLog(validRows).pwvSsusMean]' [ResultsLog(validRows).pwvMsusMean]';
42 %     doPttPlot(ResultsLog, validRows, xy, "PSUS", 1);
43 %     doPttPlot(ResultsLog, validRows, [ResultsLog(validRows).pwvSsusMean]', "SSUS", 2);
44 %     doPttPlot(ResultsLog, validRows, [ResultsLog(validRows).pwvMsusMean]', "MSUS", 3);
45 %     doPttPlot(ResultsLog, validRows, [ResultsLog(validRows).pwvPeakMean]', "Peak", 4);
46
47 %
48 %     ax = [];
49 %     for ai = 1 : 16
50 %         ax = [ax subplot(4,4,ai)];                                %#ok<AGROW>
51 %     end
52 %
53 %     linkaxes(ax, 'y');
54
55
56 end
57
58 function doPttPlot(CLResults, validRows, pttData, pttName, plotCol, fitType)
59
60     if ~exist('fitType', 'var')
61         fitType = 'poly2';
62     end
63
64     w = 1./([CLResults(validRows).diffHR]');
65
66
67     figure();
68     %%%%%%subplot(4,4,plotCol);      hold on;
69
70     [f, gof] = fit([CLResults(validRows).recSBP]', pttData, fitType, 'Weight', w);
71     plot(f,[CLResults(validRows).recSBP]', pttData, 'predfunc');
72     %plot([CLResults(validRows).recSBP]', pttData, 'm.');
73
74
75     desc = sprintf("PWV %s - SBP\nSSE: %f, ARS: %f", pttName, gof.sse, gof.adjsquare);
76
77     %axis([100 170 -100 100]);
78     title(desc);
79     xlabel("SBP (mmHg)");
80     ylabel("PWV (cm/ms)");
81     legend(gca, 'off');
82
83     disp(desc);                                         %#ok<DSPS>
84     disp(gof);
85
86
87
88
89     figure();
90     %%%%%%subplot(4,4,plotCol + 4);      hold on;
91
92     [f, gof] = fit([CLResults(validRows).recDBP]', pttData, fitType, 'Weight', w);
93     plot(f,[CLResults(validRows).recDBP]', pttData, 'predfunc');
94     %plot([CLResults(validRows).recDBP]', pttData, 'm.');
95
96     desc = sprintf("PWV %s - DBP\nSSE: %f, ARS: %f", pttName, gof.sse, gof.adjsquare);
97
98     %axis([80 120 -100 100]);
99     title(desc);
100    xlabel("DBP (mmHg)");
101    ylabel("PWV (cm/ms)");
102    legend(gca, 'off');
103
104   disp(desc);                                         %#ok<DSPS>
105   disp(gof);
106
107
108
109    figure();
110    %%%%%%subplot(4,4,plotCol + 8);      hold on;
111
112    [f, gof] = fit([CLResults(validRows).recHR]', pttData, fitType, 'Weight', w);

```

```

112 plot(f,[CLResults(validRows).recHR]', pttData, 'predfunc');
113 %plot([CLResults(validRows).recHR]', pttData, 'm.');
114
115 desc = sprintf("PWV %s - Heart Rate\nSSE: %f, ARS: %f", pttName, gof.sse, gof.adjrsquare);
116
117 %axis([50 80 -100 100]);
118 title(desc);
119 xlabel('Heart Rate (bpm)');
120 ylabel("PWV (cm/ms)");
121 legend(gca, 'off');
122
123 disp(desc);                                     %#ok<DSPS>
124 disp(gof);
125
126
127 figure();
128 %%%%%%subplot(4,4,plotCol + 12);    hold on;
129
130 [f, gof] = fit([CLResults(validRows).MAP]', pttData, fitType, 'Weight', w);
131 plot(f,[CLResults(validRows).MAP]', pttData, 'predfunc');
132 %plot([CLResults(validRows).MAP]', pttData, 'm.');
133
134 desc = sprintf("PWV %s - Mean Arterial Pressure\nSSE: %f, ARS: %f", pttName, gof.sse,
135 gof.adjrsquare);
136
137 %axis([50 80 -100 100]);
138 title(desc);
139 xlabel('MAP (mmHg)');
140 ylabel("PWV (cm/ms)");
141 legend(gca, 'off');
142
143 disp(desc);                                     %#ok<DSPS>
144 disp(gof);
145
146
147 end

```

MATLAB code: fppp.m

```

1 function Data = fppp(Data, displayPlot, displayText)
2
3
4 % Data = parsePpgData(Data);
5 [PpgData, dataParamIsLogEntry] = extractPpgData(Data);
6
7 if ~exist('displayPlot', 'var')
8     displayPlot = false;
9 end
10
11 if ~exist('displayText', 'var')
12     displayText = false;
13 end
14
15
16 if dataParamIsLogEntry
17     testDescBasic = sprintf('testId: %d userId: %d timestamp: %s rec SBP: %d DBP: %d HR: %d
18 bpm', Data.testId, Data.userId, Data.timestamp, Data.recSBP, Data.recDBP, Data.recHR );
19     testId = Data.testId;
20     if displayPlot
21         figure('Name', sprintf('PPG testId #%d', testId), 'NumberTitle', 'off');
22     end
23 else
24     testDescBasic = 'Unknown test';
25     testId = NaN;
26     if displayPlot
27         figure('Name', 'PPG test', 'NumberTitle', 'off');
28     end
29 end
30

```

```

31
32
33     testDescError = '';
34     valid = [false false];
35
36
37     if displayPlot
38         createEmptyPpgAxis(Data, subplot(2,1,1));
39         createEmptyPpgAxis(Data, subplot(2,1,2));
40         set(gcf, 'Position', get(0, 'Screensize'));
41
42         channelColors = [1 0 0; 0 0 1];
43         markerColors = [0.7 0.1 0.1; 0.1 0.1 0.7];
44         varianceColor = [0 0.5 0.2];
45     end
46
47     for iChannel = 1 : 2
48         meta = findPpgPoints(PpgData(iChannel));
49
50         meta = manualRemoval(testId, iChannel, meta);
51
52
53         plotSignals(PpgData, meta, iChannel);
54
55
56         [meta, valid(iChannel)] = validatePpgPoints(meta);
57
58         % if ~valid(iChannel)
59         %     testDescError = [testDescError sprintf('PPG%d BAD! ', iChannel)];
60         % end
61
62
63         [meta.systolicTime, meta.systolicAmplitude, meta.p2pTime, meta.peakVariance] =
64         birrenkott(PpgData(iChannel).filtered, meta.peaks, meta.psus);
65         [meta.sysInitTime, meta.sysInitAmplitude, meta.ssus2ssusTime, meta.ssusVariance] =
66         birrenkott(PpgData(iChannel).filtered, meta.ssus, meta.psus);
67
68         displayMetaText(meta);
69
70         PpgMeta(iChannel) = meta;                                %%ok<AGROW>
71
72     end
73
74     meanBpm = mean([PpgMeta(1).bpm, PpgMeta(2).bpm]);
75
76     testDesCHR = sprintf('\nHeart beats: %d | %d, Heart rate = %.2f | %.2f bpm >= %.2f',
77     PpgMeta(1).beats, PpgMeta(2).beats, PpgMeta(1).bpm, PpgMeta(2).bpm, meanBpm);
78
79     if dataParamIsLogEntry
80         hrTolerance = 10;
81         hrDiff = Data.recHR - meanBpm;
82         if any(abs([hrDiff (PpgMeta(1).bpm - PpgMeta(2).bpm)]) > hrTolerance)
83             testDescError = [testDescError sprintf('Heart Rate ERROR! ±%d (%.2f)', hrTolerance,
84             hrDiff)];
85         end
86     end
87
88
89
90
91     if PpgMeta(1).beats == PpgMeta(2).beats
92         % we cool. beats match.
93         pttBeats = PpgMeta(1).beats;
94
95     elseif PpgMeta(1).beats == PpgMeta(2).beats + 1
96         % one beat more in channel 1
97         pttBeats = PpgMeta(2).beats;
98
99     elseif PpgMeta(1).beats + 1 == PpgMeta(2).beats
100        pttBeats = PpgMeta(1).beats;
101
102        % if ch2 1st peak before ch1 1st psus remove ch2 1st detections
103        ch2FistPeak = PpgMeta(2).peaks(1);
104        if ch2FistPeak < PpgMeta(1).psus(1) || ch2FistPeak < PpgMeta(1).ssus(1)

```

```

105     PpgMeta(2).peaks(1) = [];
106     if PpgMeta(2).msus(1) < ch2FistPeak
107         PpgMeta(2).msus(1) = [];
108     end
109     if PpgMeta(2).ssus(1) < ch2FistPeak
110         PpgMeta(2).ssus(1) = [];
111     end
112     if PpgMeta(2).psus(1) < ch2FistPeak
113         PpgMeta(2).psus(1) = [];
114     end
115
116
117     PpgMeta(2).beats = pttBeats;
118     warning('PPG:fppp:channel2FirstBeatRemoved', 'Removed extra first beat on channel 2 in
119     testId %d', testId);
120
121     %if ~valid(2)
122         % revalidate ch2 an remove PPG2 Bad from text
123         [meta, valid(2)] = validatePpgPoints(PpgMeta(2));
124     %
125     %
126     %
127     %end
128
129     [meta.systolicTime, meta.systolicAmplitude, meta.p2pTime, meta.peakVariance] =
130     birrenkott(PpgData(iChannel).filtered, meta.peaks, meta.psus);
131     [meta.sysInitTime, meta.sysInitAmplitude, meta.ssus2ssusTime, meta.ssusVariance] =
132     birrenkott(PpgData(iChannel).filtered, meta.ssus, meta.psus);
133
134     if displayPlot
135         subplot(2,1,2);
136         cla;
137         plotSignals(PpgData, meta, 2);
138         displayMetaText(meta);
139     end
140
141     PpgMeta(2) = meta;
142
143     else
144         warning('PPG:fppp:channel2Beats1Greater', 'One more beat detected and kept in channel 2 in
145         testId %d', testId);
146     end
147
148
149
150     else
151         pttBeats = min([PpgMeta(1).beats PpgMeta(2).beats]);
152         warning('PPG:fppp:channelBeatsUnmatched', 'Beats detected per channel don''t match in testId
153         %d', testId);
154     end
155
156
157     pttPeak = PpgMeta(2).peaks(1:pttBeats) - PpgMeta(1).peaks(1:pttBeats);
158     pttPsus = PpgMeta(2).psus(1:pttBeats) - PpgMeta(1).psus(1:pttBeats);
159     pttSsus = PpgMeta(2).ssus(1:pttBeats) - PpgMeta(1).ssus(1:pttBeats);
160     pttMsus = PpgMeta(2).msus(1:pttBeats) - PpgMeta(1).msus(1:pttBeats);
161
162     displayPttText(meta)
163
164
165     testDescPTT = sprintf('\nPTT beats: %d    PTTpsus %.2f (%.2f)    PTTssus %.2f (%.2f)    PTTmsus %.2f
166     (%.2f)    PTTpeak %.2f (%.2f) ', ...
167     pttBeats, mean(pttPsus), std(pttPsus), ...
168     mean(pttSsus), std(pttSsus), ...
169     mean(pttMsus), std(pttMsus), ...
170     mean(pttPeak), std(pttPeak));
171
172     ch1Invalid = '';
173     ch2Invalid = '';
174     if ~valid(1)
175         ch1Invalid = sprintf('PPG%d Invalid detections! ', 1);
176     end
177     if ~valid(2)
178         ch2Invalid = sprintf('PPG%d Invalid detections! ', 2);

```

```

179     end
180
181 if ~isempty(testDescError) || ~isempty(ch1Invalid) || ~isempty(ch2Invalid)
182     testDescError = sprintf('\n%s %s %s', ch1Invalid, ch2Invalid, testDescError);
183 end
184
185 if dataParamIsLogEntry && ~isempty(Data.comment)
186     comment = sprintf('\n%s', Data.comment);
187 else
188     comment = '';
189 end
190
191
192 testDesc = [testDescBasic testDescHR testDescPTT testDescError comment];
193
194 if dataParamIsLogEntry
195     Data.valid = valid(1) && valid(2);
196     if ~isempty(testDescError)
197         Data.comment = regexprep([Data.comment ' ' testDescError], '\n', '');
198     end
199 end
200
201 if displayPlot
202     subplot(2,1,1);
203     title(testDesc);
204 end
205
206 %%clipboard('copy', testDesc);
207
208 Data.PpgMeta(1) = PpgMeta(1);
209 Data.PpgMeta(2) = PpgMeta(2);
210 Data.meanBpm = meanBpm;
211
212 Data.diffHR = abs(Data.recHR - Data.meanBpm);
213 Data.MAP = (Data.recSBP + (2 * Data.recDBP)) / 3;
214
215 Data.pttPsusMean = mean(pttPsus);
216 Data.pttSsusMean = mean(pttSsus);
217 Data.pttMsusMean = mean(pttMsus);
218 Data.pttPeakMean = mean(pttPeak);
219
220 Data.pttPsusSD = std(pttPsus);
221 Data.pttSsusSD = std(pttSsus);
222 Data.pttMsusSD = std(pttMsus);
223 Data.pttPeakSD = std(pttPeak);
224
225 Data.pttPsus = pttPsus;
226 Data.pttSsus = pttSsus;
227 Data.pttMsus = pttMsus;
228 Data.pttPeak = pttPeak;
229
230 Data.pwvPsus = Data.armLength ./ pttPsus;
231 Data.pwvPsus(pttPsus == 0) = [];
232
233 Data.pwvSsus = Data.armLength ./ pttSsus;
234 Data.pwvSsus(pttSsus == 0) = [];
235
236 Data.pwvMsus = Data.armLength ./ pttMsus;
237 Data.pwvMsus(pttMsus == 0) = [];
238
239 Data.pwvPeak = Data.armLength ./ pttPeak;
240 Data.pwvPeak(pttPeak == 0) = [];
241
242 Data.pwvPsusMean = mean(Data.pwvPsus);
243 Data.pwvSsusMean = mean(Data.pwvSsus);
244 Data.pwvMsusMean = mean(Data.pwvMsus);
245 Data.pwvPeakMean = mean(Data.pwvPeak);
246
247 Data.pwvPsusSD = std(Data.pwvPsus);
248 Data.pwvSsusSD = std(Data.pwvSsus);
249 Data.pwvMsusSD = std(Data.pwvMsus);
250 Data.pwvPeakSD = std(Data.pwvPeak);
251
252 Data.systolicTime2Mean = mean(PpgMeta(2).systolicTime);

```

```

253 Data.systolicAmplitude2Mean = mean(PpgMeta(2).systolicAmplitude);
254 Data.p2pTime2Mean = mean(PpgMeta(2).p2pTime);
255 Data.peakVariance2Mean = mean(PpgMeta(2).peakVariance);
256
257 Data.sysInitTime2Mean = mean(PpgMeta(2).sysInitTime);
258 Data.sysInitAmplitude2Mean = mean(PpgMeta(2).sysInitAmplitude);
259 Data.ssus2ssusTime2Mean = mean(PpgMeta(2).ssus2ssusTime);
260 Data.ssusVariance2Mean = mean(PpgMeta(2).ssusVariance);
261
262 Data.sysSlope2Mean = mean(PpgMeta(2).systolicAmplitude ./ PpgMeta(2).systolicTime);
263 Data.siSlope2Mean = mean(PpgMeta(2).sysInitAmplitude ./ PpgMeta(2).sysInitTime);
264
265
266 Data.testDescHR = testDescHR;
267 Data.testDescPTT = testDescPTT;
268 Data.testDescError = testDescError;
269
270
271
272
273
274
275
276
277 function plotSignals(PpgData, meta, iChannel)
278     if displayPlot
279         subplot(2,1,iChannel);
280         cla;
281         hold on;
282
283
284     hSignal = plot(PpgData(iChannel).filtered);
285
286     hPeaksVar = plot(meta.peaks, PpgData(iChannel).filtered(meta.peaks), 'k:');
287     hPsusVar = plot(meta.psus, PpgData(iChannel).filtered(meta.psus), 'k:');
288
289     hPeaks = plot(meta.peaks, PpgData(iChannel).filtered(meta.peaks), 'k^');
290
291     hD1zc = plot(meta.d1zc, PpgData(iChannel).filtered(meta.d1zc), 'k.');
292     hMsus = plot(meta.msus, PpgData(iChannel).filtered(meta.msus), 'kd');
293
294     hSsus = plot(meta.ssus, PpgData(iChannel).filtered(meta.ssus), 'kv');
295     hPsus = plot(meta.psus, PpgData(iChannel).filtered(meta.psus), 'k>');
296
297
298     %hSsus = plot(meta.ssus, PpgData(iChannel).d2(meta.ssus)*20 + 1700, 'k.');
299     %hMsus = plot(meta.msus, PpgData(iChannel).d1(meta.msus)*10 + 1700, 'k.');
300
301     set(hSignal, 'Color', channelColors(iChannel,:));
302     set(hSignal, 'LineWidth', 1);
303
304     set([hPeaks hMsus hSsus hPsus], 'Color', markerColors(iChannel,:));
305
306     set(hSignal, 'DisplayName', sprintf('Filtered PPG %d', iChannel));
307     set(hPeaks, 'DisplayName', 'Systolic Peak');
308     set(hMsus, 'DisplayName', 'MSUS - Mid');
309     set(hSsus, 'DisplayName', 'SSUS - Start');
310     set(hPsus, 'DisplayName', 'PSUS - Pre');
311     set(get(get(hD1zc,'Annotation'),'LegendInformation'), 'IconDisplayStyle','off');
312
313
314     set([hPeaksVar hPsusVar], 'Color', varianceColor);
315     set(hPeaksVar, 'DisplayName', 'Peak variance');
316     set(hPsusVar, 'DisplayName', 'PSUS variance');
317
318     legend('show');
319 end
320 end
321
322
323
324 function displayMetaText(meta)
325     if displayPlot && displayText
326         textY = zeros(1,length(meta.peaks));

```

```

327
328     stText = strtrim(cellstr(num2str(meta.systolicTime(:))));
329     saText = strtrim(cellstr(num2str(meta.systolicAmplitude(:), '%.1f'))));
330     p2pText = strtrim(cellstr(num2str(meta.p2pTime(:))));
331     pvText = strtrim(cellstr(num2str(meta.ssusVariance(:), '%.1f'))));
332
333     sitText = strtrim(cellstr(num2str(meta.sysInitTime(:))));
334     siaText = strtrim(cellstr(num2str(meta.sysInitAmplitude(:), '%.1f'))));
335     s2sText = strtrim(cellstr(num2str(meta.p2pTime(:))));
336     svText = strtrim(cellstr(num2str(meta.peakVariance(:), '%.1f'))));
337
338
339     text(meta.psus, textY + 3000, stText, 'FontSize', 7, 'Color', 'k');
340     text(meta.psus, textY + 3150, saText, 'FontSize', 7, 'Color', 'r');
341     text(meta.peaks(1:end-1), textY(1:end-1) + 3300, p2pText, 'FontSize', 7, 'Color', 'b');
342     text(meta.peaks(1:end-1), textY(1:end-1) + 3450, pvText, 'FontSize', 7, 'Color',
343 varianceColor);
344
345
346     text(meta.psus, textY + 1600, sitText, 'FontSize', 7, 'Color', 'k');
347     text(meta.psus, textY + 1670, siaText, 'FontSize', 7, 'Color', 'r');
348     text(meta.peaks(1:end-1), textY(1:end-1) + 1740, s2sText, 'FontSize', 7, 'Color', 'b');
349     text(meta.peaks(1:end-1), textY(1:end-1) + 1810, svText, 'FontSize', 7, 'Color',
350 varianceColor);
351     end
352   end
353
354
355   function displayPttText(meta)
356     if displayPlot && displayText
357       subplot(2,1,2);
358       textY = zeros(1,pttBeats);
359
360
361       pttPeakText = strtrim(cellstr(num2str(pttPeak(:), '%.2f')));
362       pttMsusText = strtrim(cellstr(num2str(pttMsus(:), '%.2f')));
363       pttSsusText = strtrim(cellstr(num2str(pttSsus(:), '%.2f')));
364       pttPsusText = strtrim(cellstr(num2str(pttPsus(:), '%.2f')));
365
366       text(meta.peaks(1:pttBeats), textY + 4000, pttPeakText, 'FontSize', 7, 'Color', 'k');
367       text(meta.msus(1:pttBeats), textY + 3900, pttMsusText, 'FontSize', 7, 'Color', 'k');
368       text(meta.ssus(1:pttBeats), textY + 3800, pttSsusText, 'FontSize', 7, 'Color', 'k');
369       text(meta.psus(1:pttBeats), textY + 3700, pttPsusText, 'FontSize', 7, 'Color', 'k');
370
371       text(-500, 4000, 'PTTpeak', 'FontSize', 7, 'Color', 'k');
372       text(-500, 3900, 'PTTmsus', 'FontSize', 7, 'Color', 'k');
373       text(-500, 3800, 'PTTsus', 'FontSize', 7, 'Color', 'k');
374       text(-500, 3700, 'PTTpss', 'FontSize', 7, 'Color', 'k');
375     end
376   end
377
378
379
380
381
382   function meta = manualRemoval(testId, iChannel, meta)
383
384     switch testId
385       case 12
386         if iChannel == 2
387           meta.ssus = meta.ssus(meta.ssus ~= 1518);
388         end
389
390
391       case 22
392         % removes a large dicrotic notch that is detated as a peak
393         if iChannel == 2
394           meta.peaks = meta.peaks(meta.peaks ~= 22476);
395           meta.msus = meta.msus(meta.msus ~= 22393);
396           meta.ssus = meta.ssus(meta.ssus ~= 22342);
397           meta.psus = meta.psus(meta.psus ~= 22320);
398         end
399
400       case 28

```

```

401             % removes a SUSS from channel 2 - chan swapped finger < ear
402             if iChannel == 2
403                 meta.ssus = meta.ssus(meta.ssus ~= 3021);
404
405             end
406
407         case 31
408             % removes a SSUS from channel 1
409             if iChannel == 1
410                 meta.ssus = meta.ssus(meta.ssus ~= 5715);
411             end
412
413         case 55
414             % removes extra SSUS on ch2
415             if iChannel == 2
416                 meta.ssus = meta.ssus(meta.ssus ~= 2456);
417             end
418
419         case 57
420             if iChannel == 1
421                 meta.psus = meta.psus(meta.psus ~= 411);
422                 meta.ssus = meta.ssus(meta.ssus ~= 507);
423                 meta.msus = meta.msus(meta.msus ~= 528);
424                 meta.peaks = meta.peaks(meta.peaks ~= 572);
425             end
426
427             if iChannel == 2
428                 meta.psus = meta.psus(meta.psus ~= 410);
429                 meta.msus = meta.msus(meta.msus ~= 525);
430                 meta.peaks = meta.peaks(meta.peaks ~= 572);
431                 meta.ssus = meta.ssus(meta.ssus ~= 722);
432             end
433
434         end
435     end
436
437
438 end

```

MATLAB code: locatePpgChannelPeaks.m

```

1 function [ReturnedValue] = locatePpgChannelPeaks(Data, channel)
2 % locatePpgChannelPeaks located the peaks in a PPG signal
3 % Data param can either be a LogEntry or PpgData struct
4 % The channel param is used to indicate which channel to process as this
5 % script only works on one channel at a time.
6 % The ReturnedValue struct is an updated version of the struct passed in as
7 % the Data param.
8 %
9 % See also parsePpgData, extractPpgData, createLogStructs
10
11
12 % validate Data param and extract PpgData
13 [PpgData, dataParamIsLogEntry] = extractPpgData(Data);
14
15 % validate param - channel
16 % valid channel depends on what was passed in as Data param
17 % if Data was test log row can be 1 or 2
18 % if Data was PPG Data can have 2 channels, but might only be 1
19 nChannels = length(PpgData);
20
21 if nChannels == 1
22     channel = 1;
23 elseif nChannels == 2 && (~exist('channel', 'var') || channel ~= 2)
24     channel = 1;
25 elseif channel ~= 2
26     error('PPG:locatePpgChannelPeaks:invalidChannel', 'Invalid PPG Channel setting %d.', channel);

```


MATLAB code: parseCircTests.m

```
1 function LogEntry = parseCircTests()
2
3 %%%
4 clear
5 basePath = 'G:\PPG Master Research - Ross\Circadian Experiments\Day%d\' ;
6
7 totalTests = 0;
8 allFiles = [];
9 for iDay = 1 : 7
10     filepath = sprintf(basePath, iDay);
11
12     % create a list of .mat filenames from the path
13     files = dir(fullfile(filepath, '*.mat'));
14     clear filenames;
15     [filenames{1:numel(files)}] = deal(files.name);
16     numTests = numel(filenames);
17     disp(['There are ', num2str(numTests) , ' tests in path ', filepath]);
18
19     totalTests = totalTests + numTests;
20     allFiles = [allFiles; files];
21 end
22 %%
23
24 %load('LP10.mat');
25 Hd = '';
26 LogEntry(totalTests) = createLogStructs();
27
28
29
30
31 %%
32 for iFile = 1: length(allFiles)
33     fprintf('%s %s\n', allFiles(iFile).folder, allFiles(iFile).name);
34     LogEntry(iFile) = parseInitialTest(allFiles(iFile).name, [allFiles(iFile).folder '\'], Hd);
35
36 end
37
38 end
```

MATLAB code: parseInitialTest.m

```
1 function [LogEntry] = parseInitialTest(testLogFile, pathname, Hd)
2 % parseTestLog Parsed the initial test data (eg circadian) to create more usesable
3 % NewLog = parseTestLog() uses LP10.mat
4 % NewLog = parseTestLog(testLogFile, excludedFile, filterFile)
5 %
6 % See also createLogStructs
7
8 %     % load the file - use default if param not specified
9 %     try
10 %         if ~exist("testLog", "var")
11 %             if ~exist("testLogFile", "var")
12 %                 [testLogFile, pathname] = uigetfile('.mat', 'Load single test .mat file');
13 %
14 %                 if ~testLogFile
15 %                     error('PPG:parseInitialTest: loadData: fileNotFound', 'No data file');
16 %                 end
17 %             end
18 %             fprintf("loading %s\n", [pathname, testLogFile]);
19 %             load([pathname, testLogFile]);
```

```

20      fprintf("Test log %s loaded\n", testLogFile);
21    end
22  catch
23    error('PPG:parseInitialTest:loadData:fileNotFound', 'The data file %s was not found',
24 testLogFile);
25  end
26 %
27 % load the filter - use default if param not specified
28 try
29   if ~exist("filterFile", "var")
30     filterFile = "LP10.mat";
31   end
32   load(filterFile, "Hd");
33 %
34 catch
35   error('PPG:parseTestLog:loadFilter:fileNotFound', 'The filter file %s was not found',
36 filterFile);
37 end
38
39 load([pathname, testLogFile]);
40 load('LP10.mat');
41
42 %
43 % create data structures
44 [LogEntryStruct, PpgChannel] = createLogStructs();
45
46 %
47 % set basic log entry data
48 LogEntry = LogEntryStruct;
49 LogEntry.testId = generateTestId(timestamp);
50 LogEntry.timestamp = string(datestr(timestamp));
51
52 if strncmpi(test_subject,'Ross',1)
53   LogEntry.userId = 01;
54   LogEntry.armLength = 70;
55
56 elseif strncmpi(test_subject,'Zhe',1)
57   LogEntry.userId = 02;
58   LogEntry.armLength = 72;
59
60 else
61   LogEntry.userId = 99;
62   LogEntry.valid = 0;
63   LogEntry.comment = 'Unknown test subject';
64
65 end
66
67 LogEntry.userName = string(test_subject);
68 LogEntry.recSBP = omron_sbp;
69 LogEntry.recDBP = omron_dbp;
70 LogEntry.reCHR = omron_hr;
71
72
73 %
74 % create ppg data struct
75 LogEntry.PpgData(2) = PpgChannel;
76
77 if ischar(dataS1)
78   % parse raw data from single text strings into vectors
79   len = length(dataS1);
80   dataS1double(len) = 0;
81   dataS2double(len) = 0;
82   for ii = 1 : len
83     dataS1double(ii) = uint16(dataS1(ii));
84     dataS2double(ii) = uint16(dataS2(ii));
85   end
86
87   if size(dataS1double,1) == 1
88     dataS1double = dataS1double';
89   end
90   if size(dataS2double,1) == 1
91     dataS2double = dataS2double';
92   end
93

```

```

94
95     LogEntry.PpgData(1).raw = dataS1double;
96     LogEntry.PpgData(2).raw = dataS2double;
97 else
98 % data already in vectors
99 if size(dataS1,1) == 1
100     dataS1 = dataS1';
101 end
102 if size(dataS2,1) == 1
103     dataS2 = dataS2';
104 end
105
106 LogEntry.PpgData(1).raw = dataS1;
107 LogEntry.PpgData(2).raw = dataS2;
108 end
109
110 % check number of samples (raw data length) matches
111 nSamples = length(LogEntry.PpgData(1).raw);
112 if(nSamples ~= length(LogEntry.PpgData(2).raw))
113     fprintf("testId %d invalid. Ch1 and Ch2 are ''t the same length\n", iTest);
114     LogEntry.valid = false;
115     LogEntry.comment = 'Ch1 and Ch2 are ''t the same length';
116 end
117
118 %set sample time (1kHz)
119 LogEntry.sampleTime = nSamples / 1000;
120
121 % filter and get 1st and 2nd derivatives for each channel
122 for iCh = 1 : 2
123     LogEntry.PpgData(iCh).filtered = filter(Hd,LogEntry.PpgData(iCh).raw);
124     LogEntry.PpgData(iCh).d1 = diff(LogEntry.PpgData(iCh).filtered);
125     LogEntry.PpgData(iCh).d2 = diff(LogEntry.PpgData(iCh).d1);
126 end
127
128
129
130
131 end
132
133 %%
134 function did = generateTestId(timestamp)
135     dv = datevec(timestamp);
136     di = 2 : length(dv) - 1;
137     did = dv(di) .* 10 .^ (10 - (di *2));
138     did = sum(did);
139 end

```

MATLAB code: parseInitialTests.m

```
1 function LogEntry = parseInitialTests()
2
3     %% 
4     filepath = 'G:\\PPG Master Research - Ross\\PPG-BP Project Folder\\PTT-BP Experiment\\Old Data\\';
5
6     % create a list of .mat filenames from the path
7     files = dir(fullfile(filepath, '*.mat'));
8     clear filenames;
9     [filenames{1:numel(files)}] = deal(files.name);
10    numTests = numel(filenames);
11    disp(['There are ', num2str(numTests), ' tests in path ', filepath]);
12
13
14    %%
15
16    %load('LP10.mat');
17    Hd = '';
18    LogEntry(numTests) = createLogStructs();
19
20
21
22
23    %%
24    for iFile = 1: length(files)
25        fprintf('%s %s\n', files(iFile).folder, files(iFile).name);
26        try
27            LogEntry(iFile) = parseInitialTest(files(iFile).name, [files(iFile).folder '\\'], Hd);
28        catch me
29            fprintf('Error processing %s\n%', files(iFile).name, me.message');
30        end
31    end
32
33
34    %% remove empty rows
35    ii = 1;
36    while ii < length(LogEntry)
37        if isempty(LogEntry(ii).testId)
38            LogEntry(ii) = [];
39        else
40            ii = ii + 1;
41        end
42    end
43 end
```

MATLAB code: parsePpgData.m

```
1 function [ReturnedStruct] = parsePpgData(Data)
2 % parsePpgData parses both PPG channels provided in Data param
3 % Parses data to locate the PPG peaks on each channel as well as the start
4 % of the systolic up stroke (SSUS)
5 % The Data param can be either a LogEntry or PpgData struct
6 % The returned struct is the same type as what was supplied as a param
7 %
8 % See also locatePpgChannelPeaks, extractPpgData, createLogStructs
9
10
11 % validate Data param and extract PpgData
12 [PpgData, dataParamIsLogEntry] = extractPpgData(Data);
13
14
15 % validate param - channel
16 % Must have 2 channels to work correctly.
17 nChannels = length(PpgData);
```

```

18
19     if nChannels ~= 2
20         error('PPG:parsePpgData:invalidChannel', 'Invalid PPG channels %d.', nChannels);
21     end
22
23
24
25     % call other script to actually locate and set the peaks in PpgData
26     for iChannel = 1 : nChannels
27         PpgData = locatePpgChannelPeaks(PpgData, iChannel);
28     end
29
30
31     % signal is not normalised and sits above bias of around 1500
32     bias = 1500;
33
34     meanPpgPeak(1) = mean(PpgData(1).sysPeakVals - bias);
35     meanPpgPeak(2) = mean(PpgData(2).sysPeakVals - bias);
36
37
38     % check PPG channels are "numbered" correctly, swapping if req.
39     % channel 1 should be ear, channel 2 should be finger
40     % finger has higher peak values, so it's mean will be higher
41     if meanPpgPeak(1) > meanPpgPeak(2)
42         % fprintf("swapping channels");
43
44         temp = PpgData(1);
45         PpgData(1) = PpgData(2);
46         PpgData(2) = temp;
47
48         temp = meanPpgPeak(1);
49         meanPpgPeak(1) = meanPpgPeak(2);
50         meanPpgPeak(2) = temp;
51         clear temp;
52
53     end
54
55     % remove any peaks from start and end that are < 50% of mean
56     % these are falsely detected peaks
57     for iChannel = 1 : nChannels
58
59         while PpgData(iChannel).sysPeakVals(1) - bias < meanPpgPeak(iChannel) * 0.5
60             PpgData(iChannel) = removeFirstPeak(PpgData(iChannel));
61         end
62
63         while PpgData(iChannel).sysPeakVals(end) - bias < meanPpgPeak(iChannel) * 0.5
64             PpgData(iChannel) = removeLastPeak(PpgData(iChannel));
65         end
66
67     end
68
69
70     % Ensure the same number of peaks found on each channel
71     % Two ways number of peaks could be different, both relate to when
72     % sampling finished and ended.
73     % If sampling / peak detection started between Ear Peak and its
74     % corresponding Finger Peak then Finger Peak will be detected first
75     % and Finger channel will have more samples.
76     % If sampling ended after last Ear Peak, but before corresponding
77     % corresponding Finger Peak occurred Ear channel will have more samples
78     nPeaksPpg1 = length(PpgData(1).sysPeakVals);
79     nPeaksPpg2 = length(PpgData(2).sysPeakVals);
80
81     breakOutOfLoop = 10;
82     while nPeaksPpg1 ~= nPeaksPpg2 && breakOutOfLoop > 0
83         % fprintf("Number of peaks on each channel do not match\n");
84         breakOutOfLoop = breakOutOfLoop - 1;
85
86         % if there are 2 peaks on ch1 before 1st peak on ch2 - testId 35
87         if nPeaksPpg1 > nPeaksPpg2 && PpgData(1).sysPeakSamp(1) < PpgData(2).sysPeakSamp(1) &&
88             PpgData(1).sysPeakSamp(2) < PpgData(2).sysPeakSamp(1)
89                 PpgData(1) = removeFirstPeak(PpgData(1));
90                 nPeaksPpg1 = nPeaksPpg1 - 1;
91

```

```

92     % if num peaks on ch1 > ch2 and last sample time of ch1 > ch2
93     elseif nPeaksPpg1 > nPeaksPpg2 && PpgData(1).sysPeakSamp(end) > PpgData(2).sysPeakSamp(end)
94         % fprintf("More peaks on channel 1. Removing extra sample from end of channel 1\n");
95
96     %PpgData(1).sysPeakVals = PpgData(1).sysPeakVals(1 : end - 1);
97     %PpgData(1).sysPeakSamp = PpgData(1).sysPeakSamp(1 : end - 1);
98
99     PpgData(1) = removeLastPeak(PpgData(1));
100    nPeaksPpg1 = nPeaksPpg1 - 1;
101
102
103    elseif nPeaksPpg1 < nPeaksPpg2 && PpgData(2).sysPeakSamp(1) < PpgData(1).sysPeakSamp(1)
104        % fprintf("More peaks on channel 2. Removing extra sample from start of channel 2\n");
105        % PpgData(2).sysPeakVals = PpgData(2).sysPeakVals(2 : end);
106        % PpgData(2).sysPeakSamp = PpgData(2).sysPeakSamp(2 : end);
107
108        PpgData(2) = removeFirstPeak(PpgData(2));
109        nPeaksPpg2 = nPeaksPpg2 - 1;
110
111    else
112        % error('PPG:parsePpgData:invalidPeaks', 'Invaild PPG peaks calculated');
113    end
114 end
115
116 if breakOutOfLoop == 0
117     if dataParamIsLogEntry
118         warning('Had to break out of loop on testId = %d', Data.testId);
119     else
120         warning('Had to break out of loop');
121     end
122 end
123
124
125
126
127
128 % if Finger PPG still has 1st detected peak remove 1st peak from both
129 % % channels (test log #)
130 % while PpgData(1).sysPeakSamp(1) > PpgData(2).sysPeakSamp(1) && PpgData(2).sysPeakVals(1) >
131 PpgData(1).sysPeakVals(1)
132     % PpgData(1) = removeFirstPeak(PpgData(1));
133     % PpgData(2) = removeFirstPeak(PpgData(2));
134 % end
135
136
137
138 % %% Locates the knee before each peak (1st derivative 0-crossing)
139 % i=1:length(diffPPG)-1;
140 % k=find ((diffPPG(i)>0 & diffPPG(i+1)<0) | (diffPPG(i)<0 & diffPPG(i+1)>0));
141 % k = k + 1;
142 % [~,k1] = intersect(k,plocPPG,'stable');
143 % k1 = k1 - 1;
144 % knee = k(k1);
145
146 for iChannel = 1 : nChannels
147     diffPPG = PpgData(iChannel).d1;
148     plocPPG = PpgData(iChannel).sysPeakSamp;
149     i = 1 : length(diffPPG) - 1;
150     zeroCrossingD1 = find ((diffPPG(i)>0 & diffPPG(i+1)<0) | (diffPPG(i)<0 & diffPPG(i+1)>0));
151     zeroCrossingD1 = zeroCrossingD1 + 2; % I don't know why this is two rather than one now.
152     [~,k1] = intersect(zeroCrossingD1,plocPPG,'stable');
153     k1 = k1 - 1;
154     knee = zeroCrossingD1(k1);
155     PpgData(iChannel).sysStartSamp = knee;
156     PpgData(iChannel).sysStartVals = PpgData(iChannel).filtered(knee);
157
158 end
159
160
161
162
163
164
165 if dataParamIsLogEntry

```

```

166     Data.PpgData = PpgData;
167     ReturnedStruct = Data;
168 else
169     ReturnedStruct = PpgData;
170 end
171
172
173
174
175 function PpgDataChannel = removeFirstPeak(PpgDataChannel)
176     PpgDataChannel.sysPeakVals = PpgDataChannel.sysPeakVals(2 : end);
177     PpgDataChannel.sysPeakSamp = PpgDataChannel.sysPeakSamp(2 : end);
178 end
179
180 function PpgDataChannel = removeLastPeak(PpgDataChannel)
181     PpgDataChannel.sysPeakVals = PpgDataChannel.sysPeakVals(1 : end - 1);
182     PpgDataChannel.sysPeakSamp = PpgDataChannel.sysPeakSamp(1 : end - 1);
183 end
184
185
186 end

```

MATLAB code: parseTestLog.m

```

1 function [NewLog] = parseTestLog(testLogFile, excludedFile, filterFile)
2 % parseTestLog Parsed the original test data to create more usesable
3 % data file marking those deemed to be excluded as invalid
4 % NewLog = parseTestLog() uses defaults testLog.mat, exclude.mat & LP10.mat
5 % NewLog = parseTestLog(testLogFile, excludedFile, filterFile)
6 %
7 % See also createLogStructs
8
9 % load the file - use default if param not specified
10 try
11     if ~exist("testLog", "var")
12         if ~exist("testLogFile", "var")
13             testLogFile = "testLog.mat";
14         end
15         load(testLogFile, "testLog");
16         nTests = length(testLog);
17         fprintf("Test log %s with %d entries loaded\n", testLogFile, nTests);
18     else
19         nTests = length(testLog);
20     end
21 catch
22     error('PPG:parseTestLog:loadData:fileNotFound', 'The log file %s was not found', testLogFile);
23 end
24
25 % load the excluded tests file doesn't already exist - use default if param not specified
26 try
27     if ~exist("excluded", "var")
28         excluded = {};
29         if ~exist("excludedFile", "var")
30             excludedFile = "excludedTests.mat";
31         end
32
33         load(excludedFile, "excluded");
34         nExcluded = length(excluded);
35         fprintf("Excluded test list %s with %d entries loaded\n", excludedFile, nExcluded);
36     end
37
38 catch
39     error('PPG:parseTestLog:loadData:fileNotFound', 'The excluded test file %s was not found',
40 excludedFile);
41 end
42
43
44 % load the filter - use default if param not specified
45 try

```

```

46         if ~exist("filter", "var")
47             filterFile = "LP10.mat";
48         end
49         load(filterFile, "Hd");
50
51     catch
52         error('PPG:parseTestLog:loadFilter:fileNotFound', 'The filter file %s was not found',
53     filterFile);
54     end
55
56
57     % create data structures
58     [LogEntryStruct, PpgChannel] = createLogStructs();
59
60     NewLog(nTests) = LogEntryStruct;
61
62     % loop over all the tests to create new log
63     for iTest = 1:nTests
64
65         % set basic log entry data
66         testData = testLog(iTest);
67         LogEntry = LogEntryStruct;
68         LogEntry.testId = iTest;
69         LogEntry.timestamp = string(testData.mTime);
70         LogEntry.userId = int64(str2double(testData.userID));
71         LogEntry.userName = string(testData.userName);
72         LogEntry.recSBP = str2double(testData.systolic);
73         LogEntry.recDBP = str2double(testData.diastolic);
74         LogEntry.recHR = str2double(testData.heartRate);
75
76
77         % check is test excluded
78         iExcluded = find([excluded{:,1}] == iTest);
79         if iExcluded
80             LogEntry.valid = false;
81             fprintf("testID #%2d excluded (e-list %#2d): %s\n", ...
82                     iTest, iExcluded, string(excluded(iExcluded,2)));
83             LogEntry.comment = string(excluded(iExcluded,2));
84             % logEntry.ppgData = [];
85             % newLog(i) = logEntry;
86             % continue;
87         else
88             LogEntry.valid = true;
89         end
90
91         % create ppg data struct
92         LogEntry.PpgData(2) = PpgChannel;
93
94         % parse raw data from single text strings into vectors
95         LogEntry.PpgData(1).raw = sscanf(testData.rPPG1, "%d");
96         LogEntry.PpgData(2).raw = sscanf(testData.rPPG2, "%d");
97
98         % check number of samples (raw data length) matches
99         nSamples = length(LogEntry.PpgData(1).raw);
100        if(nSamples ~= length(LogEntry.PpgData(2).raw))
101            fprintf("testId %d invalid. Number of sample don't match\n", iTest);
102            LogEntry.valid = false;
103        end
104
105        %set sample time (1kHz)
106        LogEntry.sampleTime = nSamples / 1000;
107
108        % filter and get 1st and 2nd derivatives for each channel
109        for iCh = 1 : 2
110            LogEntry.PpgData(iCh).filtered = filter(Hd,LogEntry.PpgData(iCh).raw);
111            LogEntry.PpgData(iCh).d1 = diff(LogEntry.PpgData(iCh).filtered);
112            LogEntry.PpgData(iCh).d2 = diff(LogEntry.PpgData(iCh).d1);
113        end
114
115        NewLog(iTest) = LogEntry;
116
117    end
118
119 end

```

MATLAB code: plotPpgSignals.m

```
1 function plotPpgSignals(Data, gHandle, signalsToPlot, channelsToPlot)
2 % plotPpgSignals plots specified PPG data to figure/subplot
3 % Data param can be either LogEntry or PpgData struct
4 % The gHandle param can be a figure or axes (subplot) graphics handle to
5 % where the plot will be drawn.
6 % The signalsToPlot param is a character vector nominating what to plot
7 % rfdwpkPK - raw, filtered, derivative, 2nd derivative, peak, ssuss, Peak
8 % PTT bands, SUSS PTT bands
9 % The channelsToPlot is a vector 1, 2 or [1 2] nominating the channels to
10 % plot
11 %
12 % See also createLogStructs
13
14
15
16
17 % validate Data param and extract PpgData
18 PpgData = extractPpgData(Data);
19
20
21
22 % validate / set gHandle. If not param then use current figure
23 % can be figure handle or subplot handle
24 if ~exist('gHandle', 'var')
25     % don't have to do anything here.. just uses gcf
26 elseif ~isgraphics(gHandle, 'figure') || isgraphics(gHandle, 'axes')
27     error('PPG:plotPpgSignals:invalidGHandle', 'Invalid graphics handle. Cannot create axis');
28 elseif isgraphics(gHandle, 'figure')
29     figure(gHandle);
30 elseif isgraphics(gHandle, 'axes')
31     subplot(gHandle);
32 end
33
34
35 % validate / create signals to plot
36 ALL_SIGNALS_TO_PLOT = 'rfdwpkPK'; %'rfdwpk';
37 if ~exist('signalsToPlot', 'var')
38     signalsToPlot = ALL_SIGNALS_TO_PLOT;
39 elseif ~all(ismember(signalsToPlot, ALL_SIGNALS_TO_PLOT))
40     error('PPG:plotPpgSignals:invalidSignalsToPlot', 'Invalid signals selection. Cannot plot');
41 else
42 end
43
44
45 if ~exist('channelsToPlot', 'var')
46     if length(PpgData) == 2
47         channelsToPlot = [1 2];
48     else
49         channelsToPlot = 1;
50     end
51 elseif (ischar(channelsToPlot))
52     channelsToPlot = str2num(channelsToPlot); %#ok<ST2NM>
53 end
54
55 if (isfloat(channelsToPlot)) && ~all(ismember(channelsToPlot, [1 2]))
56     error('PPG:plotPpgSignals:invalidChannelsToPlot', 'Invalid channels selection (double). Cannot
57 plot');
58 end
59
60 if length(channelsToPlot) > length(PpgData)
61     error('PPG:plotPpgSignals:tooManyChannelsToPlot', 'More channels to plot than there are
62 channels.');
63 end
64
65 hold on;
```

```

69     if contains(signalsToPlot, 'P')
70         plotPeakPttShading();
71         signalsToPlot = erase(signalsToPlot, 'P');
72     end
73
74
75     if contains(signalsToPlot, 'K')
76         plotSsusPttShading();
77         signalsToPlot = erase(signalsToPlot, 'K');
78     end
79
80
81
82     nPlotObjects = length(signalsToPlot) * length(channelsToPlot);
83
84     plotHandles = gobjects(nPlotObjects,1);
85     legendNames = strings(nPlotObjects,1);
86
87     iLegend = 1;
88
89     for iChannel = channelsToPlot
90         for iSignal = signalsToPlot
91             switch(iSignal)
92                 case 'r'
93                     plotSignal = PpgData(iChannel).raw;
94                     plotColors = {'y','c'};
95                     legendId = "PPG%d Raw";
96
97                 case 'f'
98                     plotSignal = PpgData(iChannel).filtered;
99                     plotColors = {'r','b'};
100                    legendId = "PPG%d Filtered";
101
102                 case 'd'
103                     offset = 1700 + (length(channelsToPlot) - 1) * 50;
104                     plotSignal = PpgData(iChannel).d1 * 10 + offset;
105                     plotColors = {'m','g'};
106                     legendId = "PPG%d 1st derivative";
107
108                 case 'w'
109                     offset = 1700 + (length(channelsToPlot) - 1) * 50;
110                     plotSignal = PpgData(iChannel).d2 * 20 + offset;
111                     plotColors = {'k','y'};
112                     legendId = "PPG%d 2nd derivative";
113
114                 case 'p'
115                     plotSignal = PpgData(iChannel).sysPeakVals;
116                     plotSample = PpgData(iChannel).sysPeakSamp;
117                     plotColors = {'r--^','b--^'};
118                     legendId = "PPG%d Peak";
119                     %plotHandles(iLegend) = plot(l,p,char(plotColors(iChannel)));
120
121
122                 case 'k'
123                     plotSignal = PpgData(iChannel).sysStartVals;
124                     plotSample = PpgData(iChannel).sysStartSamp;
125                     plotColors = {'r--v','b--v'};
126                     legendId = "PPG%d SSUS";
127                     %k = kneePPG2;
128                     %f = filterPPG2;
129                     %plotHandles(iLegend) = plot(k,f(k),char(plotColors(iChannel)));
130
131             otherwise
132                 fprintf("Some unknown signal, %s, unprocessed.\n", iSignal);
133             end
134
135             % peaks / sys have to be plotted differntly use plotSample to
136             % check is it's one of those or "ordinary" plot.
137             if exist('plotSample','var')
138                 plotHandles(iLegend) = plot(plotSample,plotSignal,char(plotColors(iChannel)));
139                 clear plotSample;
140             else
141                 plotHandles(iLegend) = plot(plotSignal,char(plotColors(iChannel)));
142             end

```

```

143
144
145     legendNames(iLegend) = sprintf(legendId, iChannel);
146     iLegend = iLegend + 1;
147 end
148
149 end
150
151 %plotHandles = plotHandles(1:iLegend-1);
152 %legendNames = legendNames(1:iLegend-1);
153
154 legend(plotHandles,legendNames);
155
156
157
158 % plot grey PTT blocks (patches) and write (raw) PTT to plot for PEAK TO PEAK calculation
159 function plotPeakPttShading()
160     fillColor = [0.9 0.9 0.9];
161     textHeight = 3450;
162     text(0,textHeight,cellstr('PTTpeak'));
163
164     plocPPG1 = PpgData(1).sysPeakSamp;
165     plocPPG2 = PpgData(2).sysPeakSamp;
166     peakPTT = plocPPG2 - plocPPG1;
167
168     for n = 1:length(plocPPG1)
169         patch([plocPPG1(n) plocPPG2(n) plocPPG2(n) plocPPG1(n)], [0 0 4095 4095], fillColor);
170         text(plocPPG2(n)+20, textHeight, cellstr(num2str(peakPTT(n)))); 
171     end
172     clear fillColor n;
173 end
174
175
176 % plot grey PTT blocks (patches) and write (raw) PTT to plot for SSUS TO SSUS calculation
177 function plotSsusPttShading()
178     fillColor = [0.9 0.7 0.6];
179     textHeight = 3350;
180     text(0, textHeight, cellstr('PTTsus'));
181
182     ssusPPG1 = PpgData(1).sysStartSamp;
183     ssusPPG2 = PpgData(2).sysStartSamp;
184     ssusPTT = ssusPPG2 - ssusPPG1;
185
186
187     for n = 1:length(ssusPPG1)
188         patch([ssusPPG1(n) ssusPPG2(n) ssusPPG2(n) ssusPPG1(n)], [0 0 4095 4095], fillColor);
189         text(ssusPPG2(n)+20, textHeight, cellstr(num2str(ssusPTT(n)))); 
190     end
191     clear fillColor n;
192 end
193
194 end

```

MATLAB code: ppppg.m

```
1 function LogEntry = ppppg(LogEntry, startId, endId, displayPlot)
2
3 % validate params
4 if ~exist('LogEntry', 'var')
5     if exist('NewLog', 'var')
6         LogEntry = NewLog;
7     else
8         error('PPG:ppppg:NoLog', 'No log to read from');
9     end
10 end
11
12 if ~exist('startId', 'var')
13     startId = 1;
14 end
15 if ~exist('endId', 'var')
16     endId = startId;
17 end
18
19 if length(LogEntry) < startId || startId <= 0
20     error('PPG:ppppg:InvalidStartId', 'Invalid startId %d', startId);
21 end
22
23 if length(LogEntry) < endId || endId <= 0 || endId < startId
24     error('PPG:ppppg:InvalidEndId', 'Invalid endId %d', endId);
25 end
26
27 if ~exist('displayPlot', 'var')
28     displayPlot = true;
29 end
30
31
32
33
34 % loop over log entries and parse, and plot
35 for iLogEntry = startId : endId
36
37     LogEntry(iLogEntry) = parsePpgData(LogEntry(iLogEntry));
38     disp(LogEntry(iLogEntry));
39
40     if ~displayPlot
41         continue;
42     end
43
44     figure();
45
46     LE = LogEntry(iLogEntry);
47
48
49     i = createEmptyPpgAxis(LE, subplot(4,1,1));
50     j = createEmptyPpgAxis(LE, subplot(4,1,2));
51     k = createEmptyPpgAxis(LE, subplot(4,1,3));
52     l = createEmptyPpgAxis(LE, subplot(4,1,4));
53     plotPpgSignals(LogEntry(iLogEntry), i, 'fpk',1);
54
55     %title(sprintf('testId: %d\nuserId: %d\ntimestamp %s', LE.testId, LE.userId, LE.timestamp));
56     titleText = sprintf('testId: %d userId: %d timestamp: %s rec HR: %d bpm', LE.testId,
57     LE.userId, LE.timestamp, LE.recHR );
58     title(titleText);
59
60     plotPpgSignals(LogEntry(iLogEntry), j, 'fpk',2);
61     plotPpgSignals(LogEntry(iLogEntry), k, 'fP',[1 2]);
62     plotPpgSignals(LogEntry(iLogEntry), l, 'fK',[1 2]);
63
64
65     end
66 end
```

MATLAB code: swapChannelData.m

```
1 function LogEntry = swapChannelData(LogEntry)
2
3 [swapped1, swapped2] = swapChannelBytes (LogEntry.PpgData(1).raw, LogEntry.PpgData(2).raw);
4
5 LogEntry.comment = 'Bytes swapped. ';
6 LogEntry = filterAndDerivatives(LogEntry, swapped1, swapped2);
7 LogEntry = fppp(LogEntry, true, true);
8
9 if LogEntry.pttMsusMean < 0
10    close;
11
12    LogEntry.comment = 'Bytes swapped. Channels swapped. ';
13    LogEntry = filterAndDerivatives(LogEntry, swapped2, swapped1);
14    LogEntry = fppp(LogEntry, true, true);
15 end
16
17 a = inputdlg({'Valid','Comment'},sprintf('Results
18 %d',LogEntry.testId),1,{num2str(LogEntry.valid),char(LogEntry.comment)}));
19 close;
20 LogEntry.valid = logical(str2num(a{1,:})); %#ok<ST2NM>
21 LogEntry.comment = a{2,:};
22
23 end
```

MATLAB code: validatePpgPoints.m

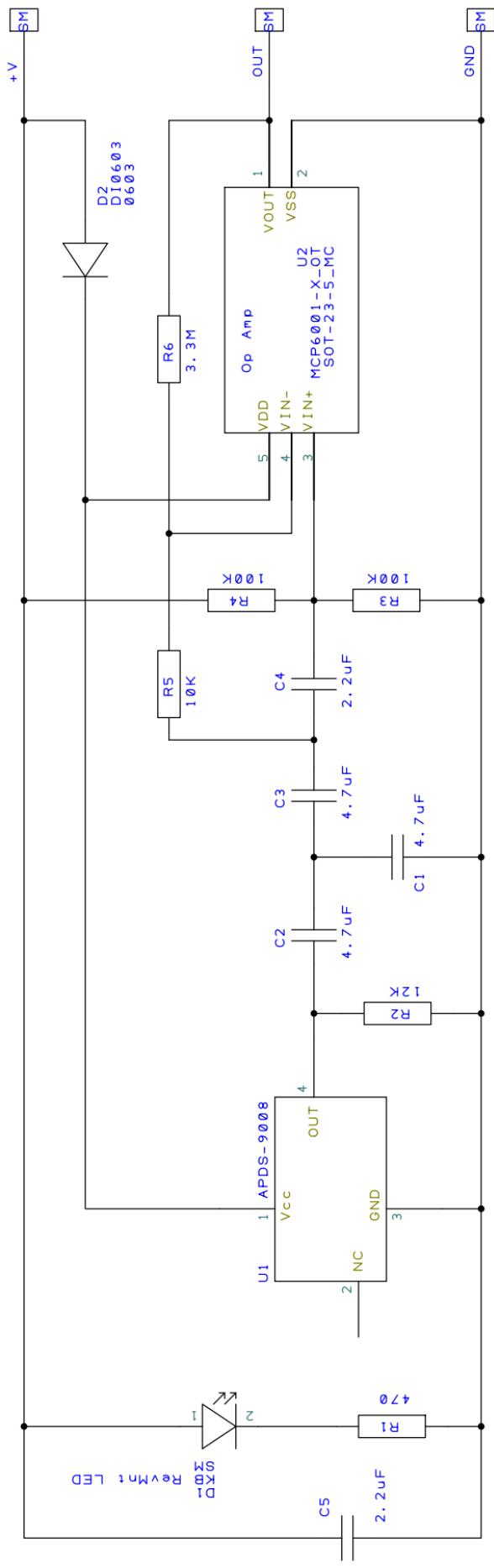
```
1 function [PpgMetaValidated, valid] = validatePpgPoints(PpgMeta)
2
3
4 nPsus = length(PpgMeta.psus);
5 nSsus = length(PpgMeta.ssus);
6 nMsus = length(PpgMeta.msus);
7 nPeaks = length(PpgMeta.peaks);
8
9
10 % ensure the same number of points detected for each point type
11 % and ensure that points are in order psus -> ssus -> msus -> peak
12
13
14 if ~(PpgMeta.psus(1) < PpgMeta.ssus(1) && PpgMeta.ssus(1) < PpgMeta.msus(1) && PpgMeta.msus(1) <
15 PpgMeta.peaks(1)) && ...
16     (PpgMeta.psus(2) < PpgMeta.ssus(1) && PpgMeta.ssus(1) < PpgMeta.msus(2) && PpgMeta.msus(2) <
17 PpgMeta.peaks(2))
18
19     % First PSUS, MSUS and Peak detected without corresponding SUSS,
20     % but 1st SUSS fits into 2nd set of PSUS, MSUS and Peak values
21     % Discard the 1st PSUS, MSUS and Peak
22     warning('PPG:validatePpgPoints:firstPointsNotOrdered', 'First SUSS not aligned to PSUS, MSUS
23 and Peak. Discard them.');
24
25 PpgMeta.psus(1) = [];
26 PpgMeta.msus(1) = [];
27 PpgMeta.peaks(1) = [];
28
29 nPsus = length(PpgMeta.psus);
30 nSsus = length(PpgMeta.ssus);
31 nMsus = length(PpgMeta.msus);
32 nPeaks = length(PpgMeta.peaks);
33
34
35 if PpgMeta.ssus(1) < PpgMeta.psus(1) && nSsus > 1
36     PpgMeta.ssus(1) = [];
37     nSsus = length(PpgMeta.ssus);
```

```

38     warning('PPG:validatePpgPoints:firstPointsNotOrdered', 'First SUSS before PSUS. Discarded.');
39 end
40
41
42
43 minBeats = min([nPsus nSsus nMsus nPeaks]);
44 i = 1 : minBeats;
45
46 if minBeats <= 1
47     warning('PPG:validatePpgPoints:zeroMinBeats', 'Min beats <= 1');
48 end
49
50 if all(PpgMeta.psus(i) < PpgMeta.ssus(i)) && all(PpgMeta.ssus(i) < PpgMeta.msus(i)) &&
51 all(PpgMeta.msus(i) < PpgMeta.peaks(i))
52     valid = true;
53 else
54     valid = false;
55     warning('PPG:validatePpgPoints:pointsNotOrdered', 'Points detected not in order');
56 end
57
58 PpgMetaValidated.psus = PpgMeta.psus(1:minBeats);
59 PpgMetaValidated.ssus = PpgMeta.ssus(1:minBeats);
60 PpgMetaValidated.msus = PpgMeta.msus(1:minBeats);
61 PpgMetaValidated.peaks = PpgMeta.peaks(1:minBeats);
62 PpgMetaValidated.d1zc = PpgMeta.d1zc;
63 PpgMetaValidated.beats = minBeats;
64
65 % testData.hrPloc1 = (length(plocPPG1) - 1 ) * (60000 / (plocPPG1(end) - plocPPG1(1)));
66 % changed to using msus instead of peaks 29.11
67 % this if is to stop it crashing when mBeats = 0
68 if minBeats >= 1
69     % circadian process with this wrong.. was (end) - (1)
70     PpgMetaValidated.bpm = (minBeats - 1) * (60000 / (PpgMetaValidated.msus(minBeats) -
71 PpgMetaValidated.msus(1)));
72 else
73     PpgMetaValidated.bpm = (nPsus - 1) * (60000 / (PpgMeta.msus(end) - PpgMeta.msus(1)));
74 end
75
76 end

```

Open Hardware PPG sensor design [57]



Pulse Sensor Amplified
Designed by Joel Murphy
Spring 2012
Licensed under the TAPR Open Hardware License (www.tapr.org/OHL)