

University of Louisville

ThinkIR: The University of Louisville's Institutional Repository

Electronic Theses and Dissertations

7-2011

Feasibility of subcutaneous ECG leads for synchronized timing of a counterpulsation device.

Sean P. Warren
University of Louisville

Follow this and additional works at: <https://ir.library.louisville.edu/etd>

Recommended Citation

Warren, Sean P., "Feasibility of subcutaneous ECG leads for synchronized timing of a counterpulsation device." (2011). *Electronic Theses and Dissertations*. Paper 1533.
<https://doi.org/10.18297/etd/1533>

This Master's Thesis is brought to you for free and open access by ThinkIR: The University of Louisville's Institutional Repository. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of ThinkIR: The University of Louisville's Institutional Repository. This title appears here courtesy of the author, who has retained all other copyrights. For more information, please contact thinkir@louisville.edu.

FEASIBILITY OF SUBCUTANEOUS ECG LEADS FOR SYNCHRONIZED TIMING
OF A COUNTERPULSATION DEVICE

By

Sean P. Warren, B.S.
University of Louisville, 2010

A Thesis
Submitted to the Faculty of the
Speed School of Engineering of the University of Louisville
As Partial Fulfillment of the Requirements
For the Professional Degree of

MASTER OF ENGINEERING

Department of Bioengineering
University of Louisville
Louisville, Kentucky

July 29, 2011

FEASIBILITY OF SUBCUTANEOUS ECG LEADS FOR SYNCHRONIZED TIMING
OF A COUNTERPULSATION DEVICE

By

Sean P. Warren, B.S.
University of Louisville, 2010

A Thesis Approved on

7/21/2011
(DATE)

By the following Thesis Committee

Steven C. Koenig, PhD, Thesis Director

Guruprasad A. Giridharan,, PhD – Thesis Advisor

Robert S. Keynton, PhD – Thesis Committee

William D. Ehringer, PhD – Thesis Committee

DEDICATION

This thesis is dedicated to my fiancée,

Rachael E. Rice.

ACKNOWLEDGMENTS

This study was funded, in part, by NIH SBIR phase I grant (1R43HL102981-01). I would like to acknowledge and thank Dr. Paul Spence, Dr. Robert Dowling, and Daiga Koenig for allowing me the opportunity to work with them at SCR Inc. on this thesis topic. You all have opened so many doors for me, and I wish you the best of luck with your projects in the future. I would also like to thank Dr. Steven Koenig and Dr. Guruprasad Giridharan for being my mentors on this project, and for being a constant source of help and advice throughout my Bioengineering studies. I would like to thank Dr. Keynton for his guidance throughout my undergraduate career and for all of the opportunities that he has provided me. I want to thank Scott Berry for giving me a taste of graduate research and PhD Comics. I would like to thank the students and staff of the Cardiovascular Innovations Institute, among whom Cary Woolard, Mary Anne Hauck, and Laura and Karen Lott provided immense help. I would like to also thank Landon Tompkins for his help running the Matlab code – you will never get those countless hours back. I would like to thank my parents who have always been there for me; I hope I've made you proud! Lastly, I would like to thank my wonderful fiancée Rachael Rice. Without your love and support, I wouldn't have made it through Speed School!

ABSTRACT

Background: Emerging circulatory support devices that operate in counterpulsation to the native heart require synchronized timing of device ejection and filling on a beat-to-beat basis with the native heart using a patient's ECG. Surface leads are commonly used for short-term patient monitoring but not appropriate for long-term use, and epicardial and non-thoracotomy leads increase the complexity of the device implant/explant procedures. Subcutaneous leads have been shown to be less susceptible to artifacts than surface leads, require less invasive surgery, and have recently been used successfully with a long-term subcutaneous implantable cardioverter-defibrillator. The objective of this study was to develop subcutaneous ECG leads for synchronized timing (filling and ejection cycles) of the Symphony device and wearable pneumatic driver for chronic counterpulsation therapy. To demonstrate feasibility, we tested the hypothesis that subcutaneous ECG leads provide equivalent QRS detection, lead migration, and durability as clinical-standard epicardial leads.

Methods: Two epicardial and six subcutaneous leads were implanted in bovine (7 days, n=4 and 14 days, n=2). Epicardial and subcutaneous ECGs were recorded simultaneously in 30 second epochs every hour and in 20 minute epochs during daily treadmill exercise. Landmark features and R-wave triggering detection rates for each lead configuration were calculated and compared. Lead placement, migration, and durability were quantified using fluoroscopy and evaluated at necropsy.

Results: There were 2,818 data epochs recorded at rest and 24 data epochs recorded during treadmill exercise. Using a simple adaptive-thresholding R-wave detection algorithm with the epicardial signal as the control, the triggering rate for the subcutaneous ECG leads demonstrated $99.1\pm 0.4\%$ positive predictive value and $96.8\pm 1.5\%$ sensitivity during normal daily activity, and $98.0\pm 10.2\%$ positive predictive value and $93.3\pm 5.6\%$ sensitivity during treadmill exercise. The average QRS voltage was 818 ± 99 μV with a T/R ratio of 0.44 ± 0.05 . There was no significant waveform distortion or artifacts observed in ECG waveforms measured with the subcutaneous leads. Upon study endpoint, all subcutaneous leads (36/36) were within 1.4 cm of initial placement with an average migration distance of 0.52 ± 0.10 cm. There was no lead fracture or permanent signal loss for all leads (36/36).

Conclusion: In this study, the subcutaneous leads provided equivalent performance to epicardial leads in bovine during normal daily activity and treadmill exercise. The leads were easily implantable using standard surgical instruments, provided high reliability of QRS detection, and showed no significant migration from the initial placement site. These findings demonstrate the feasibility of using subcutaneous leads for synchronized timing of mechanical circulatory support while offering the advantage of less invasive surgery and associated risk factors.

TABLE OF CONTENTS

	<u>Page</u>
APPROVAL PAGE	iii
DEDICATION.....	iv
ACKNOWLEDGEMENTS.....	v
ABSTRACT	vi
LIST OF FIGURES.....	ix
I. BACKGROUND.....	1
II. METHODS.....	12
III. RESULTS.....	25
IV. DISCUSSION, LIMITATIONS AND CONCLUSION.....	30
V. WORKS CITED	34
VI. CURRICULUM VITAE.....	37
VII. APPENDIX.....	43

LIST OF FIGURES

- Figure 1. The Symphony is implanted in the patient's right side via a pacemaker-like pocket, anastomosed to the right axillary artery, and the driveline exits through the skin (left). The pump (center) fills during systole and empties during diastole through a valveless, silicon-coated ePTFE cannula coated. The pump is actuated by a small, lightweight, pneumatic driver (right).....6
- Figure 2. The 32-ml Symphony provided a greater reduction in left ventricular myocardial oxygen consumption ($LV O_2$) and a greater increase in mean diastolic coronary artery flow (CAF_d) per unit left ventricular external work (LVEW). These data demonstrate the 32-ml Symphony improves myocardial perfusion (supply) more than a 40-ml IABP device while reducing cardiac work (demand) [1, 2].....7
- Figure 3. Measured surface voltages during ventricular depolarization show that the best change in voltage can be achieved with vertical lead pairs across the right chest (Left). An analog lead configuration was tested in a calf model (Right).....10
- Figure 4: Intracardiac, subcutaneous, and surface ECG leads during acute ECG mapping study conducted on 6-2-09 (1-left) and 6-25-09 (2-right).....10
- Figure 5: A comparison of intracardiac, subcutaneous, and surface ECG recordings during acute animal ECG mapping experiments conducted on 6-24-09 (1, left) and 6-25-09 (2, right). The subcutaneous and surface ECG signals show comparable fidelity to the intracardiac ECG waveform.....11
- Figure 6. Illustration of experimental protocol for testing and comparison of subcutaneous and epicardial leads. ECG waveforms were recorded simultaneously in bovine during normal daily activity and treadmill exercise.....12
- Figure 7. Subcutaneous ECG lead with extended sensing wire. Enlarged view shows the coiled sensing wire.13
- Figure 8. Ventral-dorsal lead pair configurations on the right side of the calf. All leads were placed through the single incision.....15
- Figure 9. Summary of post-processing analysis of subcutaneous and epicardial ECG waveforms recorded in calves during normal activity and treadmill exercise. After the QRS identification, the R-R interval was calculated, allowing for time-based isolation of the P, QRS, and T waves for further analysis.....19
- Figure 10. Average FFT of a subcutaneous lead pair over seven days.....20
- Figure 11. Sample ECG waveforms measured with epicardial (top) and subcutaneous (bottom) leads plotted over the stem plot of the detection matrix. The vertical thick line indicates a detected QRS complex.....23

Figure 12. Subcutaneous versus epicardial ECG signals for both normal calf activity (left) and during treadmill exercise (right).....26

Table 1. Positive Predictive Value (PPV), Sensitivity, and ECG landmark voltages for all ECG lead pairs at rest and during treadmill exercise.....27

Figure 13. Fluoroscopic digital image of implanted ECG subcutaneous leads at 14-days post-implant. Staples were placed on the skin at time of implant to mark placement of the subcutaneous leads (left). Image reveals no significant migration of the subcutaneous leads (right), which was quantified using ImageJ (NIH, Bethesda MD) analysis software.....28

CHAPTER I - BACKGROUND

Heart Failure

Heart failure (HF) is one of the largest unsolved problems in cardiac care today. There are more than 5.8 million patients in the US alone with this diagnosis and the number of patients is expected to double over the next ten years. Among the three major forms of cardiovascular disease (HF, coronary artery disease and stroke), only HF has shown a significant increase in hospitalization rates. Between 1980 and 2006, the number of patients over age 65 that were hospitalized for HF increased 131% [3]. In 2008, the direct and indirect costs of HF in the US were estimated to be 34.8 billion dollars [4]. Globally, the incidence of HF is also increasing with over one million new cases diagnosed annually. Current treatment options for HF include pharmacological therapy, cardiac resynchronization therapy (CRT), mechanical heart assist, and heart transplantation. Despite improvements in survival with medical therapy and CRT, the prognosis with heart failure remains poor. In a recent longitudinal study, mortality at one year was 15.0% and 28.0% respectively for NYHA Class III and IV [5]. Two-year mortality in the RALES trial was approximately 30% [6]. In advanced stages of HF (NYHA Class IV), heart transplantation offers the best opportunity for long-term survival. However, heart transplantation is restricted to select patients based on multiple factors including age, prior operations, end-organ function, and even appropriate insurance coverage. Furthermore, the number of available donor organs (~2,300/yr in the US according to the Organ Procurement and Transplantation Network as of 2011) cannot meet the growing demand. The shortage of appropriate donor hearts is worse in other countries such as Canada (<200 heart donors/year) and Japan (11 heart transplants in 2009) [7, 8].

The large patient population with advanced HF and the limited number of donor organs has stimulated the development of mechanical assist devices both as a bridge to transplant and as permanent support devices (destination therapy). Despite years of research and hundreds of millions of dollars in expenditure, only approximately 4,000 patients receive mechanical circulatory support each year. Current mechanical circulatory support devices require a major surgical procedure (sternotomy or thoracotomy) and the use of cardiopulmonary bypass for implantation. Further, postoperative management of these patients can be challenging and costly. Due to the complexity of the devices and the major surgical procedure required for implantation, complications are not uncommon and include device failure, thromboembolism, stroke, infection, and bleeding. For these reasons, long term mechanical support is reserved for patients with advanced heart failure who are unlikely to recover ventricular function. Additionally, the high cost of these devices, over \$67,000 for the left ventricular assist devices (LVADs) alone [9], has limited adoption in large, global markets such as India and China and has contributed to devices not being approved for long-term use by government regulatory agencies (UK) [10]. The NHLBI has recognized the need for innovative approaches to the treatment of HF. In 2003, the NHLBI “Recovery from Heart Failure with Circulatory Assist” working group recommended the development of novel therapies for myocardial recovery, and elucidation of the mechanisms leading to reverse remodeling and myocardial recovery [11].

Counterpulsation Therapy

Introduced in the 1960's, counterpulsation is considered a proven method of providing mechanical circulatory support. Basically, counterpulsation is the removal or displacement of arterial blood during ventricular systole to provide afterload reduction and "return" of the blood during diastole to augment coronary and systemic perfusion. Counterpulsation has many important clinical benefits for the heart, end organs and the peripheral circulation for patients with moderate myocardial dysfunction. Counterpulsation therapy increases the diastolic aortic pressure by 30-70% improving end-organ perfusion and coronary perfusion [12-14]. The ejection pressure of the native ventricle is diminished, reducing afterload and left ventricular external work [12, 15-18]. The peak systolic pressure is reduced by up to 15% [12, 16, 17], while the end diastolic pressure is decreased by up to 30% [12, 16, 17]. Cardiac output and stroke volume have been found to increase by up to 20% [13, 19]. Counterpulsation therapy also decreases the native heart rate by 10% [15], and may reduce the left ventricular end diastolic volume by 10-15% [20]. The hemodynamic benefits of counterpulsation support also translate into improved metabolic function and often assist in recovery of end organ function [15]. Counterpulsation therapy improves the cardiac energy balance by increasing the oxygen supply to the myocardium through increased coronary perfusion [21, 22] and by reducing myocardial oxygen consumption through a decrease in afterload and left ventricular work[15]. Counterpulsation therapy has been shown to augment cerebral, renal, mesenteric and pulmonary blood flows [23-25]. Increases in urine output, decreases in lactic acidosis and enhancement of venous oxygen saturation have also been observed in patients [15], demonstrating improved end organ function.

Counterpulsation is currently achieved with placement of an intra-aortic balloon pump (IABP), which is used clinically for limited time periods of several hours to several days. Each year more than 160,000 patients worldwide receive IABP therapy and, as a result, clinicians have developed a great deal of confidence in the therapeutic effectiveness of IABP counterpulsation [12, 15]. Placement of an IABP simply requires percutaneous access to the femoral artery. While the percutaneous access used for IABPs is less invasive than with LVADs, complication rates increase with duration of IABP support, and patients are unable to ambulate. These issues along with the limited durability of the balloon itself preclude the use of IABP for chronic support.

Given the proven benefits of counterpulsation, it is not surprising that a number of investigators have tried to develop devices that can provide chronic counterpulsation (LVAD Technologies, and Sunshine Medical). Jeevanandam et al. reported on clinical outcomes of a counterpulsation device (CardioPlus, LVAD Technologies) that was sewn to the descending aorta [26]. After implantation, these patients had significantly increased cardiac output and significantly decreased right atrial and pulmonary capillary wedge pressures. Improved cardiac output and end organ perfusion were also evidenced by a significant improvement in serum creatinine. Additionally, there was a significant decrease (improvement) in NYHA functional class [26]. The CardioPlus did not achieve clinical acceptance due to the need to perform a full thoracotomy and the use of cardiopulmonary bypass to place the device. However, this trial clearly demonstrated the hemodynamic and clinical benefits, and the potential for improved native heart function with chronic counterpulsation in patients with advanced heart failure. The Sunshine Heart

device is a counterpulsation device that is wrapped around the ascending aorta and therefore requires a thoracotomy or a sternotomy for placement. The presence of previous heart surgeries, and particularly the presence of coronary artery bypass grafts, will limit or preclude the use of this device. Also, placement of this device may significantly complicate further therapies such as LVAD placement or heart transplantation. The Sunshine Heart trial is currently enrolling patients in an IDE trial. Preliminary data suggests significant hemodynamic benefits [27].

Symphony

“Symphony” is the name given by Abiomed in late 2009 to a counterpulsation device developed in conjunction with SCR Inc (Louisville, KY) and the University of Louisville. The Symphony device (currently in pre-clinical trials) is a 30-mL stroke volume polyurethane-lined pumping chamber, which is designed to fit comfortably in a pacemaker-like pocket on the right side of a patient [2, 28]. The pumping chamber is connected to the systemic circulation by a short, silicon-coated ePTFE graft anastomosed to the right axillary artery using a simple surgical procedure. A percutaneous driveline runs from the pumping chamber, exits the skin near the costal margin, and connects to a small pneumatic external driver. During systole, the driver evacuates air from the pumping chamber, thus removing blood from the circulation and reducing cardiac work. During diastole, the Symphony ejects the blood into the circulation providing diastolic augmentation and improving coronary perfusion.



Figure 1. The Symphony is implanted in the patient’s right side via a pacemaker-like pocket, anastomosed to the right axillary artery, and the driveline exits through the skin (left). The pump (center) fills during systole and empties during diastole through a valveless, silicon-coated ePTFE cannula coated. The pump is actuated by a small, lightweight, pneumatic driver (right).

Investigators have demonstrated, in computer simulations, in vitro and multiple in vivo experiments, that the Symphony device is as effective or better than an IABP (Figure 2) [1, 2]. Placement of the Symphony device is achieved through a simple infraclavicular incision similar to that used for placement of a pacemaker. The device itself is placed superficial to the pectoralis muscle (i.e. in the subcutaneous tissue). With this operative approach, Symphony gains the proven hemodynamic benefits of chronic counterpulsation while avoiding the surgical trauma associated with a thoracotomy or a sternotomy. Symphony may be uniquely suitable to deliver counterpulsation therapy to patients that have earlier stage HF in order to improve functional capacity, restore quality of life, allow for improved medical therapy, and promote myocardial recovery.

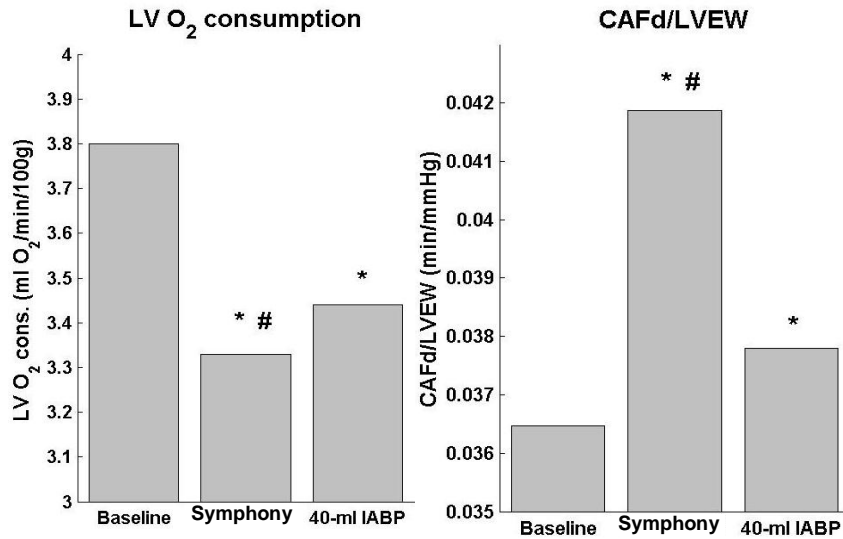


Figure 2. The 32-ml Symphony provided a greater reduction in left ventricular myocardial oxygen consumption (LV O₂) and a greater increase in mean diastolic coronary artery flow (CAFd) per unit left ventricular external work (LVEW). These data demonstrate the 32-ml Symphony improves myocardial perfusion (supply) more than a 40-ml IABP device while reducing cardiac work (demand) [1, 2].

ECG Acquisition and the Clinical Need for Subcutaneous Leads

Counterpulsation devices require precise synchronized timing of device ejection and filling on a beat-to-beat basis with the native heart using a patient’s ECG. Specifically, effectiveness of counterpulsation therapy is due to the synchronous timing of the ejection and filling of the circulatory support device with the filling and ejection of the native ventricles to reduce left ventricular external work and increase coronary perfusion. If the actions of the device are not appropriately timed dangerous hemodynamic conditions can result. For example, early device ejection causes the premature closure of the aortic valve, decreasing stroke volume and cardiac output.

Current counterpulsation therapy utilizes an IABP to provide acute circulatory support, and the inflation and deflation of this device is timed primarily by the patient’s ECG

(although aortic pressure can be used as an alternative [29]). The 12-lead surface electrocardiogram remains the current gold standard for short-term device timing, patient monitoring and diagnosis. While technologies are being developed for chronic monitoring [30], difficulties associated with the skin-electrode interface including poor contact, skin movement, and patient wire management currently preclude surface ECGs for long-term use. Non-thoracotomy lead systems, specifically transvenous right ventricular leads, are frequently used in cardiac pacing and defibrillation devices and provide quality ECG signals. Unfortunately, the implantation of these leads has a significant learning curve and increases the complexity of counterpulsation device implant/explant procedures by requiring the use of fluoroscopy [31]. Postoperative complications with these lead systems have also been reported and include lead dislodgment, fracture, vein subclavian vein thrombosis, sensing abnormalities, and system infection requiring full explantation [32, 33].

Recently, subcutaneous lead systems (a term broadly used to indicate lead positioning in the space external to the pericardium but below the dermis) have been developed for use in cardiac rhythm monitors [34-36]. Subcutaneous ECGs have long been theorized to be an adequate surrogate for surface signals [37] and QRS detection [38], and have recently been shown to be less susceptible to motion artifacts than surface leads [39]. In relation to the Symphony device, subcutaneous leads provide the benefit of minimally invasive surgery for implantation. Furthermore, when integrated with a device, they can remain fully implanted in the body – ideal for chronic counterpulsation therapy. Nevertheless,

implantable loop recorders with subcutaneous lead systems have had difficulties with isolation and detection of the QRS complex [40].

Early Investigation of ECG Lead Systems

SCR initiated preliminary studies to examine the feasibility of subcutaneous ECGs for device triggering. Preliminary subcutaneous ECG studies were first performed as an adjunct to ongoing acute Symphony implantations in male Jersey calves. The Symphony device is implanted on the right subclavian artery, which requires the ECG leads to be placed over the right chest. Ideally, leads should not cross the sternum in order to avoid issues with any future cardiac procedures that require a sternotomy. Surface voltage maps of the thorax during ventricular depolarization (ECGSIM, Radboud University medical Center, Nijmegen, The Netherlands) show that vertical lead-pairs over the right chest offer the largest change in measured voltage (Figure 3, left). Several lead-pair configurations were placed across the right chest of a calf model in order to best mimic the human analog configuration (Figure 3, right). These experiments demonstrated that the subcutaneous ECG leads were comparable to intra-cardiac leads, and the lead configuration with the highest fidelity resulted when the distance between the leads was at a maximum in a parallel orientation to the mean electrical axis of the heart. Surface ECG and subcutaneous ECG mapping tests (Figure 4, 5) were conducted to determine the ECG lead configurations with the highest fidelity data (magnitude and morphology). Based upon these early experiments, development of subcutaneous ECG leads and chronic testing in bovine were initiated to demonstrate equivalency to epicardial leads for QRS detection and integration with Symphony device and driver for long-term support.

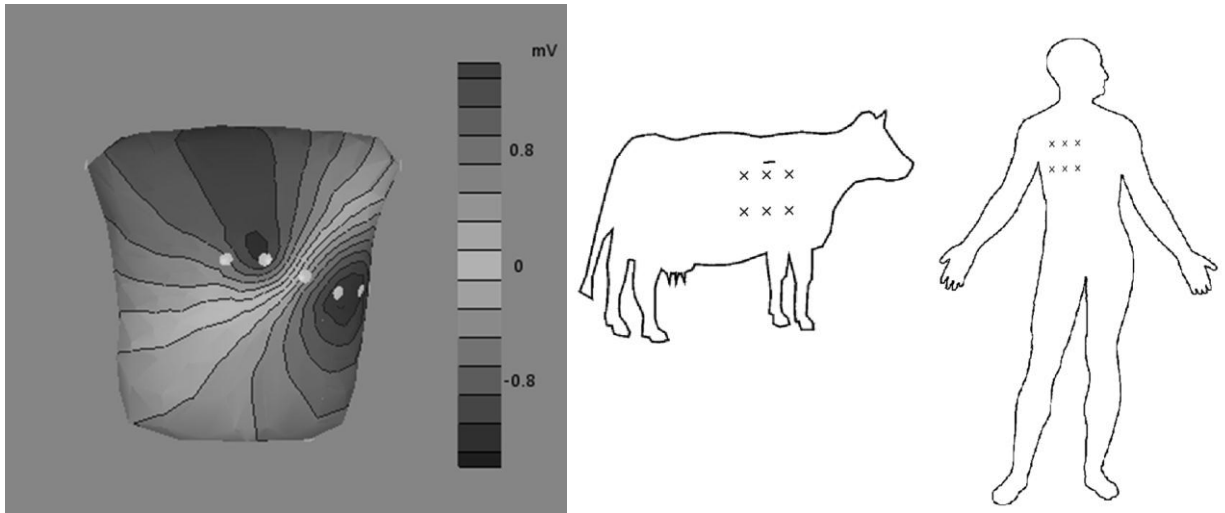


Figure 3. Measured surface voltages during ventricular depolarization show that the best change in voltage can be achieved with vertical lead pairs across the right chest (Left). An analog lead configuration was tested in a calf model (Right).

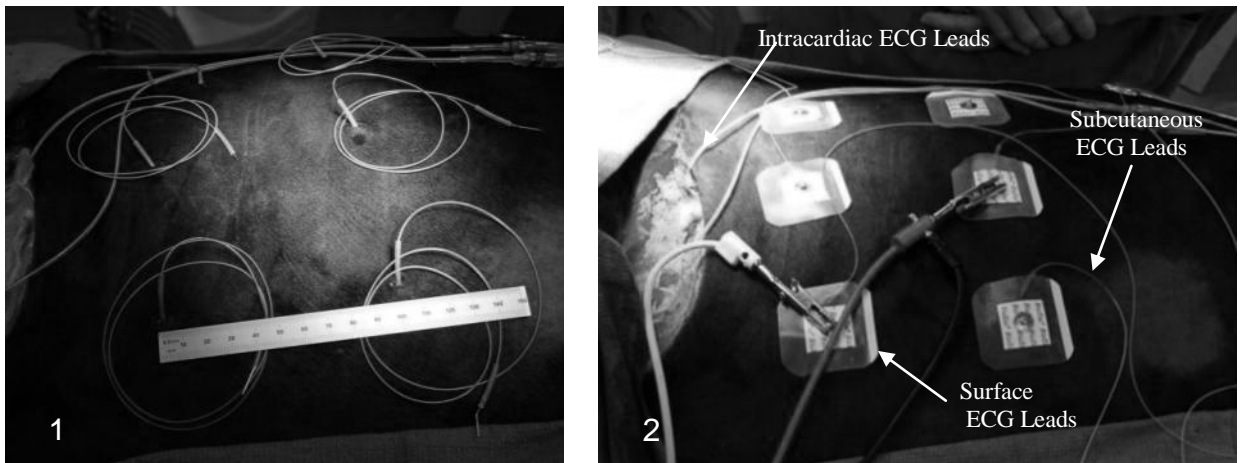


Figure 4: Intracardiac, subcutaneous, and surface ECG leads during acute ECG mapping study conducted on 6-2-09 (1-left) and 6-25-09 (2-right).

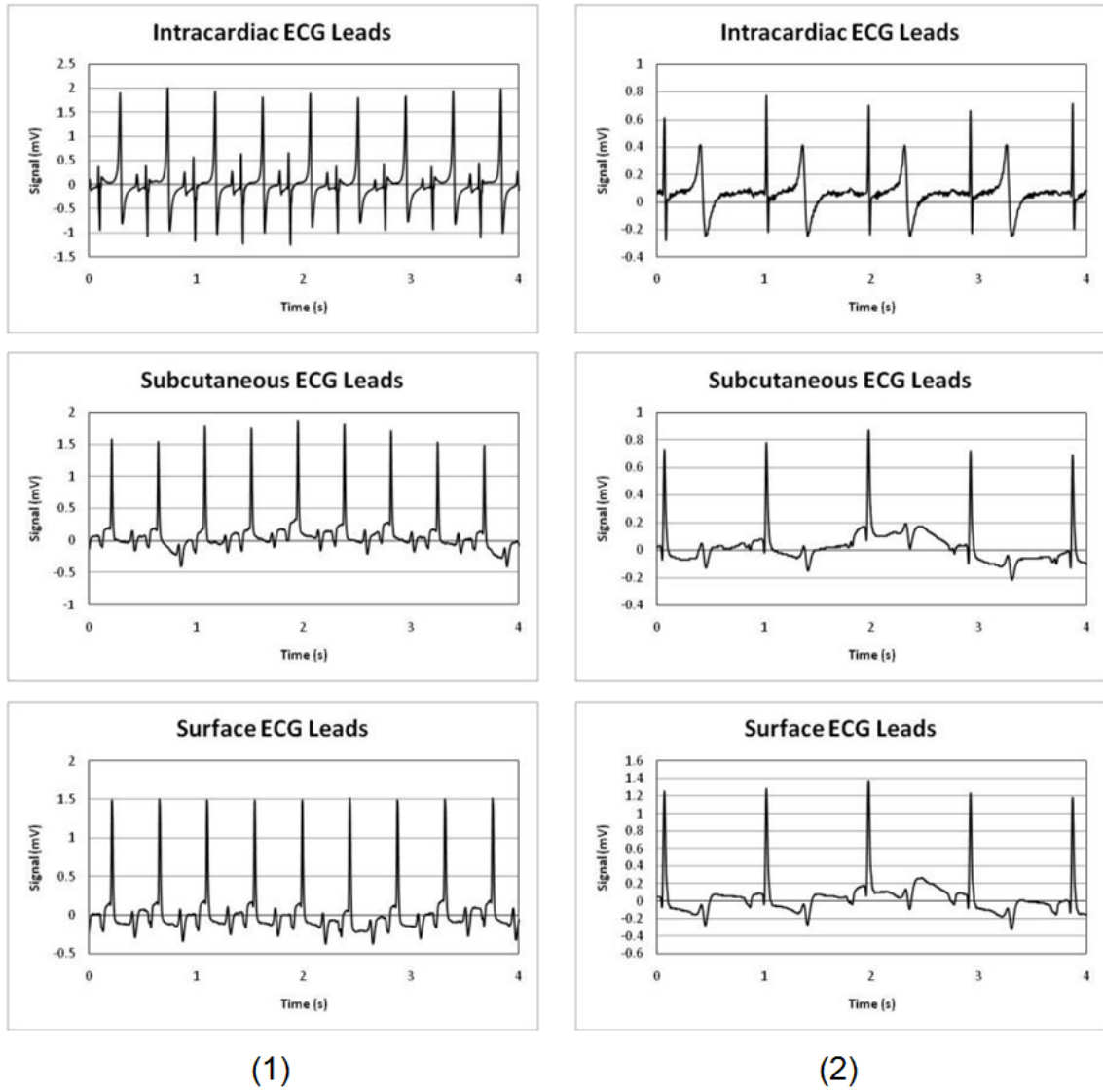


Figure 5: A comparison of intracardiac, subcutaneous, and surface ECG recordings during acute animal ECG mapping experiments conducted on 6-24-09 (1, left) and 6-25-09 (2, right). The subcutaneous and surface ECG signals show comparable fidelity to the intracardiac ECG waveform.

CHAPTER II – METHODS

Study Design and Experimental Protocol

The objective of this study was to develop subcutaneous ECG leads for synchronized timing (filling and ejection cycles) of the Symphony device and wearable pneumatic driver for chronic counterpulsation therapy. To demonstrate feasibility, we tested the hypothesis that subcutaneous ECG leads provide equivalent QRS detection, lead migration, and durability as clinical-standard epicardial leads (Figure 6).

7 Day Studies (N=4)

Surgery				Survival	Endpoint		
SubQ Leads (n=6)	Fluoroscopy	Epicardial Leads	Symphony System Implant	7 Days ECGs recorded	Animal Euthanized	Fluoroscopy Lead Migration	Necropsy Lead Durability

14 Day Studies (N=2)

Surgery				Survival		Endpoint		
SubQ Leads (n=6)	Fluoroscopy	Epicardial Leads	Symphony System Implant	7 Days ECGs recorded	7 Days with Treadmill ECGs recorded	Animal Euthanized	Fluoroscopy Lead Migration	Necropsy Lead Durability

Figure 6. Illustration of experimental protocol for testing and comparison of subcutaneous and epicardial leads. ECG waveforms were recorded simultaneously in bovine during normal daily activity and treadmill exercise.

In bovine, ECG was measured and recorded with subcutaneous (6 leads) and clinical-standard epicardial (2 leads) for 7-days (n=4) and 14-days (n=2) during normal daily activity and treadmill exercise. Recorded ECG waveforms with subcutaneous and epicardial lead configurations were analyzed to quantify QRS detection (Matlab, The

Mathworks, Natick MA). Fluoroscopy images (GE Innova 3100, Waukesha WI) of leads during placement (implant) were recorded, analyzed using ImageJ (NIH, Bethesda MD), and compared at necropsy (explant) to quantify lead migration. At necropsy, subcutaneous and epicardial leads were photographed and examined for any signs of fracture, damage, or infection.

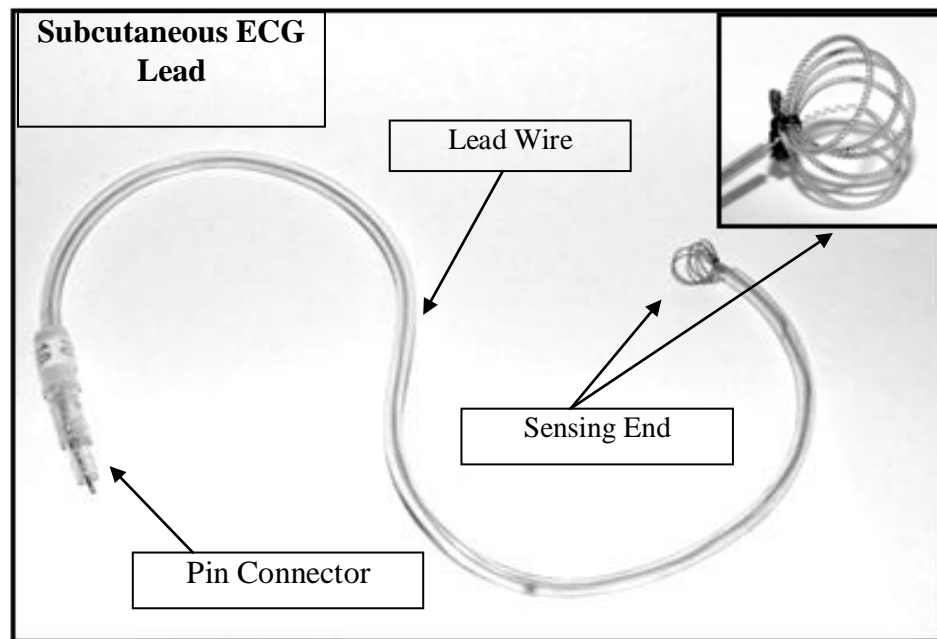


Figure 7. Subcutaneous ECG lead with extended sensing wire. Enlarged view shows the coiled sensing wire.

Lead Fabrication

The subcutaneous ECG leads were fabricated from commercially-available ECG lead coupler kits (Data Systems International, St. Paul, MN), Tygon tubing (Fischer Scientific, Pittsburg, PA), and electronic pin and socket connectors (Allied Electronics Inc., Ft. Worth, TX) (Figure 7). These leads are suitable for placement in large quadrupeds and were fabricated using a modified version of the technique described in the Data Systems

International technical bulletin [41]. Seven centimeters of lead was exposed, looped around a clamp, and secured with a suture. Leads were sterilized using ethylene oxide. Signals were recorded using a custom data acquisition system that is in compliance with Good Laboratory Practices, as has been previously reported [42].

Delivery and Placement of Subcutaneous Leads.

Several operative approaches and devices to deliver the subcutaneous ECG leads were developed using human cadavers (Fresh Tissue Laboratory, University of Louisville), including tunneling tools, modified endoscopic clip appliers, and laparoscopic instruments. The selected method was to simply pull the lead to the desired position using a standard straight needle. A 2-0 nylon suture was passed through the loops of wire at the end of the subcutaneous leads and then through the eye of a four-inch needle. A standard needle holder was then used to control the needle. Working through the device implantation wound, the needle was advanced to the desired position on the chest and then the needle and attached suture were passed through the skin. The needle and suture were removed leaving the ECG lead in place beneath the skin. This method does not require any additional incision(s) or special training and can be accomplished with standard surgical instruments.

Preliminary studies in bovine demonstrated that a lead configuration of ventral and dorsal lead pairs produced high-quality signals (Figure 8). Specifically, robust signals were obtained when a pair of leads was at least 15 cm apart and within 25 cm of the heart. An adequate human analogue configuration is a pair of ECG leads placed 15 cm apart and

parallel to the electrical axis of the heart. Similar subcutaneous configurations have been reported with implantable cardioverter-defibrillators using a parasternal electrode [34].

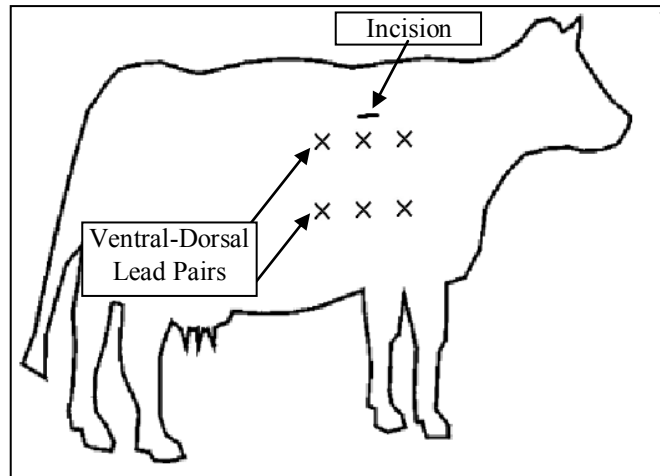


Figure 8. Ventral-dorsal lead pair configurations on the right side of the calf. All leads were placed through the single incision.

Animal Study

All animals received humane care in accordance with the Guide for Care and Use of Laboratory Animals (NIH publication 86--23, revised 1996) and the guidelines determined by the Institutional Animal Care and Use Committee (IACUC) of the University of Louisville (UofL). The study was completed in compliance with a UofL approved IACUC protocol.

The ECG study was performed concurrently with a study involving a circulatory support device implantation in a Holstein calf model (80-100 kg). Four animals were monitored with implanted devices for 7 days and two animals were monitored with implanted devices for 14 days. After initial physical examination and a 5–7 day quarantine period, animals were preanesthetized with Atropine (30 mg) and then anesthetized with

isoflurane (3%–5%) *via* nose cone. After intubation, anesthesia was maintained with isoflurane (1%–3%) mixed with oxygen and room air. Animals were then prepared for surgery on the right side. Six leads were placed subcutaneously through a single incision over the right side of each animal (n=6 animals, 36 leads) in three ventral-dorsal pairs. The placement of all six leads was timed. Subcutaneous ECG signal was verified for each vertical lead-pair. Surgical staples were placed on the skin to mark the initial site of the subcutaneous lead. Fluoroscopy was done at the time of implant to verify that the skin staples were in immediate proximity to the subcutaneous leads.

The animal was placed on its right side and the left neck and chest were prepared and draped. An incision was made over the fourth rib and carried down to the 4th intercostal space. The chest was entered and a standard rib retractor was placed. The pericardium was identified and opened and two epicardial screw-in leads were placed and connected to the circulatory support device driver. Once the quality of the epicardial signal was verified the leads were tunneled transcutaneously and secured in position. The chest was closed and the circulatory support device was implanted onto the left carotid artery using previously described methods [8].

Data Collection

After the animals were recovered, the epicardial and 3 subcutaneous ECG signals were simultaneously measured and recorded with a GLP compliant data acquisition system, as previously reported [42]. Briefly, three pairs of subcutaneous leads were signal conditioned with a clinical-standard ECG amplifier (Gould Biotach 6600, Cleveland

OH). The epicardial ECG waveforms were signal conditioned by the Symphony pneumatic driver (iPulse, Abiomed, Danvers MA). All signal conditioned ECG waveforms were sampled at 400 Hz (National Instruments, AT-MIO-16E-10, Austin, TX), filtered (60 Hz notch, Frequency Devices, Ottawa IL), and displayed real-time on a computer monitor using custom software [43] developed with LabVIEW (National Instruments, Austin TX). Specifically, thirty-second data epochs were recorded every hour for the duration of the study. All recorded ECG waveforms were saved for post-processing and analysis.

Treadmill Testing

For each of the 14-day animal experiments (n=2), treadmill testing was initiated after the seventh postoperative day and repeated daily for the duration of the study (i.e. 7 exercise days). Calves were titrated to a walking speed of 30 meters/minute on the treadmill (Safe-T-Mill, Good Horsekeeping, Inc, Ash Grove, MO) and then maintained at that rate for up to 20 minutes or until the cow showed signs of fatigue (as evaluated and recommended by study veterinarian, Leslie Sherwood, DVM). .

Lead Migration

Subcutaneous ECG lead placement at time of Symphony device implant and explant were digitally photographed and recorded using fluoroscopy (GE Innova 3100, Waukesha WI). Surgical staples were placed at device implant as a landmark for lead placement. Commercial image processing software (ImageJ, NIH, Bethesda MD) was used to quantify lead migration by overlaying digital photographs and measuring migration

distance. In addition, during all leads were surgical exposed, examined, and gross lead migration measured as a back-up technique.

ECG Analysis (subcutaneous and epicardial)

In analyzing the recorded subcutaneous and epicardial ECG waveforms, an algorithm was developed to meet the following design criteria: (1) use a standard QRS detection strategy, (2) robust to handle noise and artifacts, (3) handle large volumes of ECG data, and (4) feasible for integration into real-time driver software, as shown in Figure 9. Additionally, the analysis software has to allow for characterization of the subcutaneous signal by isolation of ECG landmarks.

The software code for data processing and analysis of all recorded ECG waveforms was developed in Matlab programming software (MathWorks, Natick MA). A preliminary program extracted the ECG data from the raw LabVIEW data files (*.dat), which were then passed to the SignalFiltv7 script. The code handled ECG signal alignment, signal filtering, QRS detection for the epicardial and subcutaneous signals, comparison and analysis of the two detection rates, and subcutaneous landmark isolation and characterization. It was written so that it could be run on an entire data set and save all important data into matrices that could be accessed by a later script. A description of each of the script's processes and algorithms is presented next. A complete line-by-line listing of the data processing and analysis software is presented in the Appendix.

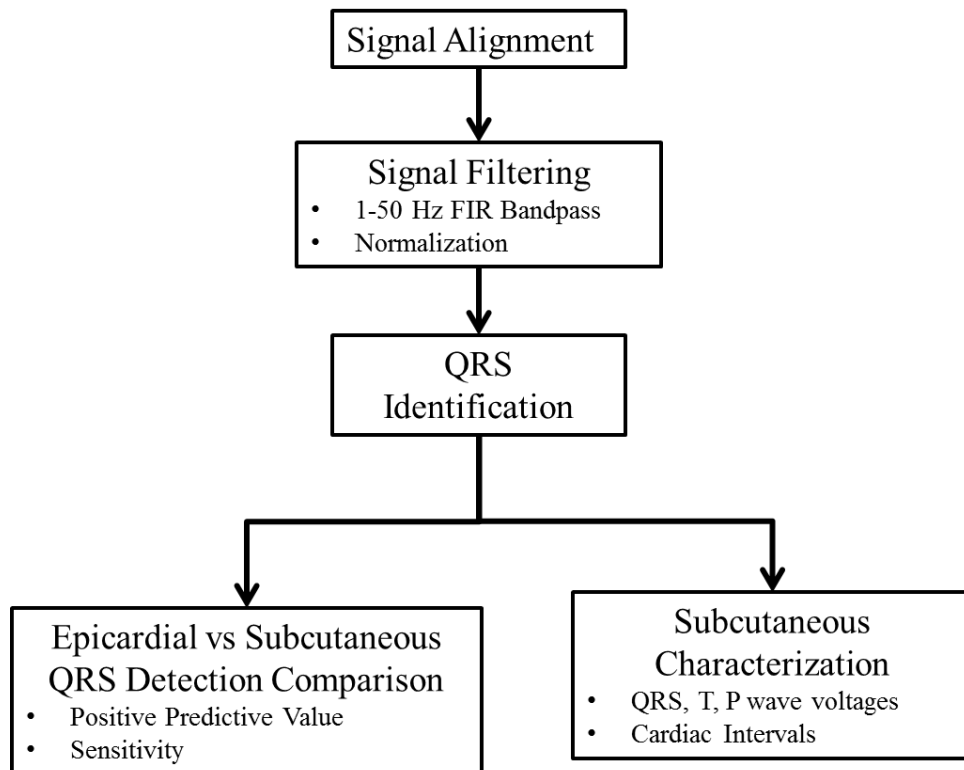


Figure 9. Summary of post-processing analysis of subcutaneous and epicardial ECG waveforms recorded in calves during normal activity and treadmill exercise. After the QRS identification, the R-R interval was calculated, allowing for time-based isolation of the P, QRS, and T waves for further analysis.

Signal Alignment

First, the program aligns all three subcutaneous signals with the epicardial signal. This is essential because time delay between the driver output signal and the subcutaneous lead signals can cause a mismatch between the QRS peaks, hampering the detection algorithm.

The alignment subroutine first prompted the user to indicate the time frame of the first epicardial QRS wave. Because of the nature of the processed epicardial signal, the user is

given several options to manipulate the signal for best QRS identification. These include inversion, second and third derivative functions. It then identifies the first QRS complex by performing a local maximum search for all three signals in the indicated time frame and aligns the signals based upon the matching of the peaks. The signals are then truncated to the same length in order to ease later processing and comparison. If, at any time, the program does not detect the epicardial signal or all three subcutaneous signals, it skips that data set and logs it as useless for QRS detection purposes.

Signal Processing

Initially, the author tried several approaches to filter the ECG signal for best QRS isolation. Combinations of a finite impulse response bandpass and Savitzky-Golay smoothing filter were attempted on raw data until the best combination with the smallest acceptable time delay was found in accordance with the average signal fast Fourier transform (FFT) (Figure 10).

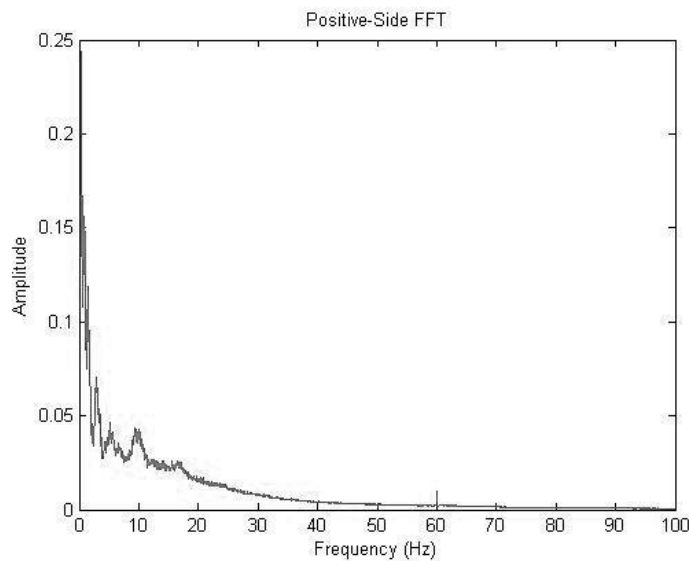


Figure 10. Average FFT of a subcutaneous lead pair over seven days.

Ultimately, the Savitzky-Golay smoothing filter was unnecessary, and a simple 1-50 Hz (N=80) FIR bandpass filter was used to reject noise. Typical QRS detection filters often have bandpass filters with their upper limit in the range of 40-80 Hz for optimal detection [44, 45] with lower limits in the range of 0.5-1 Hz to reject low-frequency baseline drift [46]. The implemented filter only causes a time delay of 100 ms, meaning that it could be theoretically realized in real-time driver software.

QRS Detection

The most intensive segment of the algorithm involved the coding of the QRS detection. The following is a summary of the implemented strategy which is a modified version of the SQRS method described by Pino et al [47]. First, the script asks for an initial threshold and blanking period as an input. It should be noted that the current Symphony driver requires a similar user input to initialize the QRS detection. The user inputs the best determined threshold for the first three QRS complexes along with an estimated blanking period in which, for the duration after successful QRS detection, the algorithm will ignore any subsequent incidents that reach the QRS threshold. After the program uses the bootstrap input to detect the first three complexes, it continues detecting new QRS complexes using 80% of the average of the past three thresholds and a blanking period of 80% of the average of the past three R-R intervals. If the next QRS amplitude or R-R interval varies from this average by more than 30%, the algorithm ignores it in its average calculations. Detected QRS complexes are represented in an empty “detection matrix” with 1’s. The script does this for all lead pairs and then compares their detection matrices against that of the epicardial signal to calculate true positives, false positives,

and false negatives (Figure 11). Complexes that are detected within 170 ms of each other (less than twice the average length of a QRS complex) are considered “true positives”; all other detection combinations are categorized as either “false positives” (QRS detected in the subcutaneous signal but not in the epicardial signal) or “false negatives” (QRS detected in the epicardial signal but not in the subcutaneous signal). These were used to calculate the positive predictive value and sensitivity of each subcutaneous lead pair according to the following equations:

$$\text{Positive Predictive Value} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

$$\text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

It is important to note that positive predictive value is an indication of the rate of a Type I error (false positive). While on counterpulsation therapy, a Type I error may result in an inappropriately timed device diastole, creating unstable and non-ideal hemodynamic conditions. Sensitivity indicates the rate of a Type II error (False Negative). A Type I error would simply result in a “skipped” device diastole, meaning that the patient has no counterpulsation support on that cardiac beat. Since the emerging devices are designed for partial circulatory support, the temporary absence of this support reduces the significance of this type of error when compared to the Type II error.

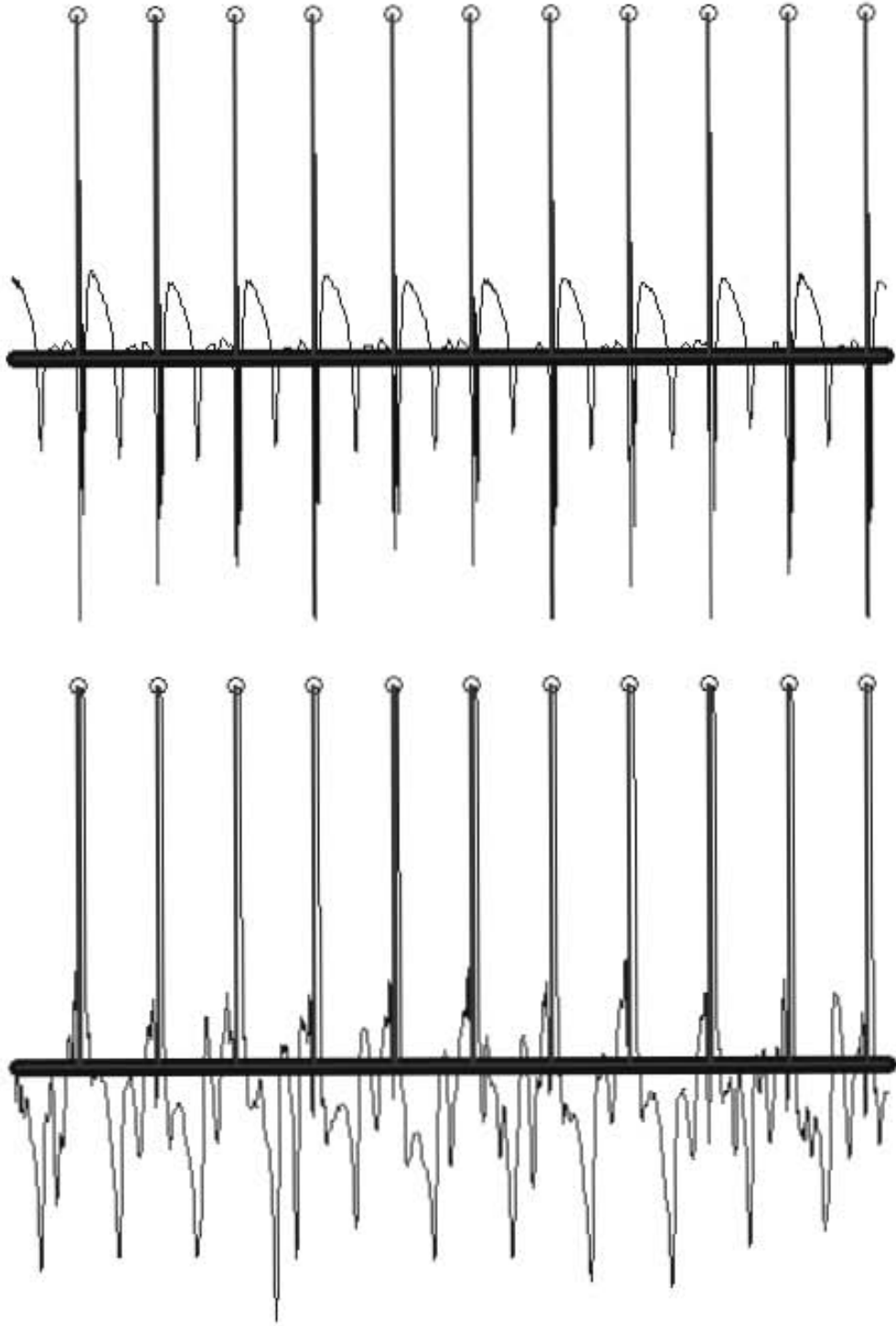


Figure 11. Sample ECG waveforms measured with epicardial (top) and subcutaneous (bottom) leads plotted over the stem plot of the detection matrix. The vertical thick line indicates a detected QRS complex.

Subcutaneous Characterization

Once the program has detected a “normal” R-R interval, it uses the time capture method to isolate the P, QRS and T waves. Basically, it takes specified percentages of the R-R interval and labels them as each cardiac wave. Once it has done this, it calculates the voltage change for each cardiac wave and attempts to locate the peaks using a local maximum or minimum method. For the QRS complex, a “moving average” detection algorithm “rides” the peak up and down, marking the length of the peak in order to calculate QRS length. Once all of these values are accounted for, the major cardiac intervals are calculated using a peak-to-peak method. All statistics were performed in Minitab (Minitab Inc, State College, PA).

CHAPTER III – RESULTS

Experimental Bovine Data

2,818 thirty-second data epochs were recorded at rest and 24 data epochs recorded during treadmill exercise (15-20 minutes each) for an aggregated total of 32 hours of recorded subcutaneous ECG data. Data sets were excluded when the ECG waveforms measured with the epicardial or any of the three subcutaneous leads were missing due to unintended disconnection from the data acquisition system. Sample ECG waveforms for epicardial and subcutaneous lead configurations during normal activity and treadmill exercise are shown in Figure 12.

Epicardial and Subcutaneous QRS Triggering Comparison

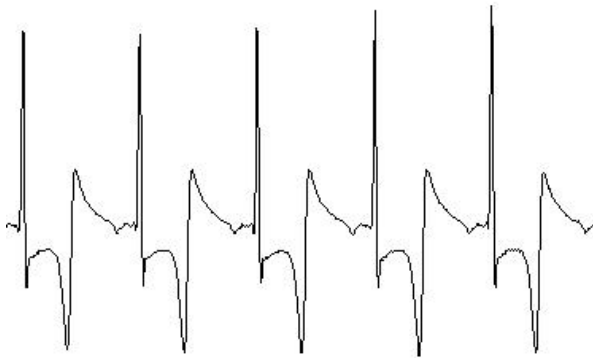
Using the described QRS detection algorithm with the epicardial signal as the control, the triggering rate for the subcutaneous ECG leads demonstrated $99.1 \pm 0.4\%$ positive predictive value and $96.8 \pm 1.5\%$ sensitivity during normal daily activity, and $98.0 \pm 10.2\%$ positive predictive value and $93.3 \pm 5.6\%$ sensitivity during treadmill exercise (Table 1). No set of lead pairs showed a statistically significant difference in positive predictive value or sensitivity compared to the other configurations ($\alpha = .05$). Triggering rate was not significantly affected by QRS, T, or P voltage as determined by multivariate regression and best-subset analysis.

The mean QRS voltage was $818 \pm 99 \mu\text{V}$, the mean T-wave voltage was $371 \pm 55 \mu\text{V}$, and the mean-P wave voltage $256 \pm 48 \mu\text{V}$. No abnormalities were observed in any of the standard cardiac intervals.

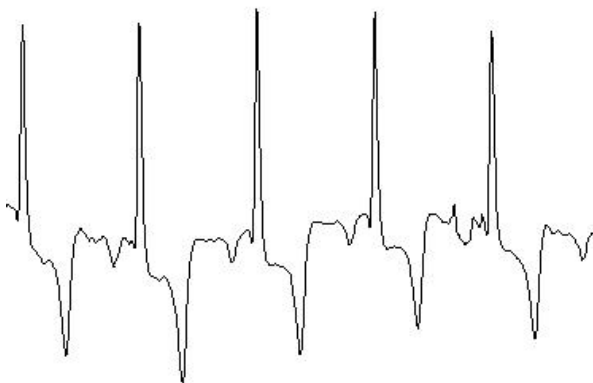
At Rest



Processed Epicardial ECG



Subcutaneous ECG



Treadmill Exercise



Processed Epicardial ECG



Subcutaneous ECG

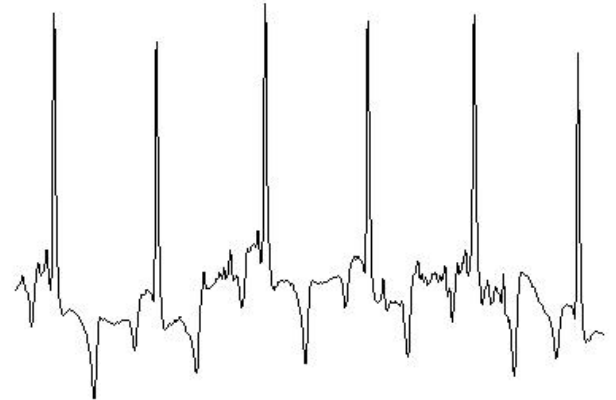


Figure 12. Subcutaneous versus epicardial ECG signals for both normal calf activity (left) and during treadmill exercise (right).

At Rest Data (30 second clips, 400 Hz)

N	SubQ Pair	Epochs	PPV	Sensitivity	R Wave (mV)	T Wave (mV)	P Wave (mV)
1	Pair 1	137	0.999 ± .001	0.996 ± .002	1.123	0.524	0.534
	Pair 2	125	0.971 ± .020	0.925 ± .021	0.964	0.597	0.405
	Pair 3	135	0.997 ± .002	0.972 ± .006	0.549	0.175	0.125
2	Pair 1	158	0.999 ± .001	0.998 ± .001	1.147	0.478	0.234
	Pair 2	157	0.998 ± .001	0.994 ± .003	0.845	0.337	0.169
	Pair 3	151	0.995 ± .002	0.987 ± .006	0.639	0.342	0.216
3	Pair 1	101	0.986 ± .012	0.962 ± .013	0.202	0.243	0.206
	Pair 2	158	0.995 ± .007	0.984 ± .008	0.691	0.207	0.147
	Pair 3	95	0.993 ± .012	0.992 ± .012	0.831	0.209	0.159
4	Pair 1	160	0.996 ± .002	0.992 ± .003	1.035	0.320	0.220
	Pair 2	83	0.994 ± .003	0.932 ± .009	0.828	0.334	0.239
	Pair 3	83	0.969 ± .014	0.890 ± .033	0.591	0.256	0.161
5	Pair 1	254	0.982 ± .012	0.969 ± .011	0.360	0.258	0.135
	Pair 2	263	0.999 ± .001	0.997 ± .001	0.880	0.237	0.157
	Pair 3	249	0.985 ± .005	0.968 ± .013	0.717	0.328	0.202
6	Pair 1	201	0.996 ± .004	0.986 ± .007	0.944	0.475	0.374
	Pair 2	153	0.991 ± .004	0.935 ± .016	0.953	0.458	0.341
	Pair 3	155	0.990 ± .005	0.950 ± .012	0.866	0.418	0.284

Treadmill Data (10-20 minute clips, 400 Hz)

N	SubQ Pair	Epochs	PPV	Sensitivity	R Wave (mV)	T Wave (mV)	P Wave (mV)
5	Pair 1	5	0.965 ± .076	0.972 ± .035	0.660	0.283	0.173
	Pair 2	4	0.992 ± .016	0.985 ± .021	1.045	0.424	0.259
	Pair 3	3	0.986 ± .064	0.922 ± .110	1.003	0.383	0.221
6	Pair 1	6	0.974 ± .012	0.919 ± .039	0.923	0.664	0.486
	Pair 2	3	0.969 ± .067	0.839 ± .290	0.778	0.452	0.278
	Pair 3	3	0.992 ± .023	0.960 ± .067	1.060	0.492	0.418

Note: When applicable, 95% confidence interval upper bound does not exceed 1.

Table 1. Positive Predictive Value (PPV), Sensitivity, and ECG landmark voltages for all ECG lead pairs at rest and during treadmill exercise.

Lead Implantation and Migration

The surgical technique for implanting the subcutaneous ECG leads was developed using human cadavers and during acute animal experiments, as previously described. Subcutaneous leads were accurately delivered at distances up to 15 cm from the incision using the straight-needle method. The average time for placement of a single lead was 35 ± 6 seconds. For placement of a full set of six subcutaneous leads, the average total time was 9 ± 2 minutes. All leads (36/36) were within 1.4 cm of their initial placement with an average migration of 0.52 ± 0.10 cm (Figure 13). Surgical exposure at necropsy confirmed that none of the subcutaneous leads had significant migration within the subcutaneous space. Electrode position did not significantly increase lead migration, and average lead migration did not vary between animals ($\alpha=0.05$).

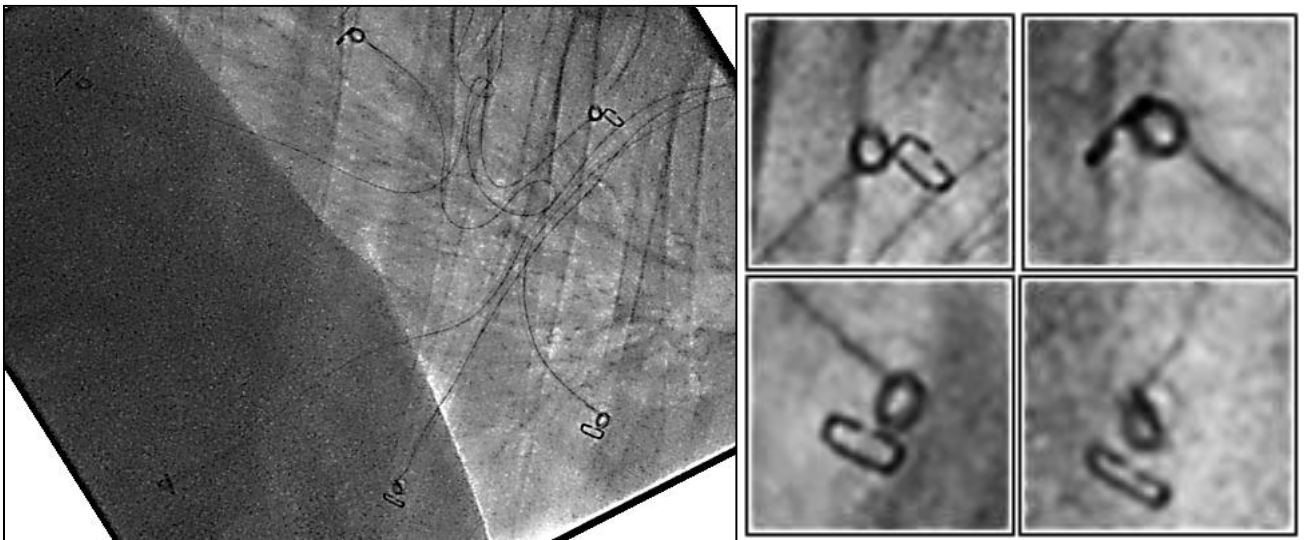


Figure 13. Fluoroscopic digital image of implanted ECG subcutaneous leads at 14-days post-implant. Staples were placed on the skin at time of implant to mark placement of the subcutaneous leads (left). Image reveals no significant migration of the subcutaneous leads (right), which was quantified using ImageJ (NIH, Bethesda MD) analysis software.

Lead Durability

There were no lead fractures, sensing wire damage, or permanent signal loss for all tested leads (36/36, 100%). In all calves, no sign of lead infection was observed or identified at necropsy. Temporary loss of signal occasionally resulted from the disconnection of the external pin and socket connectors. This was intended, by design, to prevent the animals from accidentally pulling out the subcutaneous leads.

CHAPTER IV – DISCUSSION, LIMITATIONS AND CONCLUSION

Discussion

Current counterpulsation therapies utilize surface ECGs for triggering of device systole and diastole. However, with the emergence of chronic counterpulsation devices, a more robust, long-term ECG strategy is necessary. It is essential that these devices are triggered accurately and precisely in order to prevent unstable hemodynamic conditions.

While studies have examined the R-wave detection performance between subcutaneous and surface ECGs in an implantable loop recorder [39], surface leads are not optimal for long-term triggering of a chronic counterpulsation device. Epicardial and endocardial leads offer quality signals for QRS detection, but unwantedly increase operative complexity and postoperative complications. In order for subcutaneous leads to act as adequate surrogates for cardiac leads, they must be easily implantable, demonstrate no long-term migration, and offer robust ECG signals for QRS detection. The fabricated subcutaneous leads were easily implanted using a minimally invasive technique and standard surgical equipment. Fluoroscopy confirmed that the leads did not migrate over time, demonstrating potential for use with a chronic device.

Uniquely positioned in the subcutaneous space, treadmill exercise of the monitored animals introduces significant motion and muscular (electric) artifacts into the subcutaneous signals. These artifacts did not significantly impair QRS detection and were

minimized with signal processing. QRS detection analysis demonstrated that exercise did not significantly impair Symphony triggering rates; positive predictive values remained above 95% and sensitivity values remained above 84% (mean $93.3 \pm 5.6\%$). The location of the subcutaneous leads provided excellent R wave and moderate T wave amplitudes ensuring that “oversensing” was not a problem.

Limitations

This feasibility study has several limitations. Due to the nature of the concomitant circulatory support device implantation, the epicardial signal was first sent through the circulatory support device driver to serve as the trigger for the device ejection. An output from this was fed into the data acquisition system and recorded by LabVIEW. Epicardial signal was heavily processed by the driver algorithm and the original signal could not be recovered. This meant that important ECG landmark and statistical comparisons (such as root mean square noise and signal subtraction) could not be made. Regardless, the QRS triggering analysis remains the most essential element of the comparison. Due to the volume of data, epicardial leads were used as the control and QRS complexes were not visually identified for each epoch. Any false positives or false negatives in the epicardial signal cannot be accounted for. While the fabricated leads were sufficient for use in large quadrupeds, they remain inappropriate for human implantation without further development and evaluation. The chronic study end point of 14 days remains insufficient to quantify long-term lead migration (6 months – 2 years), durability and histological response.

The QRS detection algorithm has several limitations. First, the algorithm utilizes an adaptive thresholding technique for identifying QRS complexes. It should be noted that while this is an accepted algorithm and still often used in the industry, it is a fairly dated strategy and does not take into account advances in ECG wavelet analysis and other emerging techniques. However, these new strategies can be implemented in future versions of the script.

Another limitation is that the ECG landmark analysis is not comprehensive; the script performs “peak-to-peak” interval analysis which does not give, for example, the exact PR interval or PR segment (Note: The PR interval is the distance from the start of the P wave to the beginning of the QRS complex. The PR Segment is the distance between the end of the P wave and the beginning of the QRS segment). Instead, it gives the distance from the peak of the QRS complex to the peak of the P wave – a useful number, but not a clinical standard. Improvements in P and T wave isolation through peak and valley detection methods are possible solutions to this problem.

Finally, the bootstrapping of the program for each ECG epoch means that it takes a significant amount of time to analyze a large number of signals. While the program does record the bootstrap of every epoch it has analyzed in case a segment needs to be reviewed, it still needs an initial user input. This can be remedied by a system that uses the average bootstrapping threshold and blanking period to initialize the next epoch. This would have to be carefully implemented to reduce “bad” bootstrapping and subsequent poor QRS detection.

Conclusion

The study objective to develop subcutaneous ECG leads for use with Symphony device and wearable pneumatic drier was achieved as evidenced by successful implantation of leads and completion of chronic bovine experiments. The hypothesis that subcutaneous leads provide equivalent QRS detection, lead migration, and durability as clinical-standard epicardial leads was also proven as evidenced by (1) equivalent QRS detection rates at rest (>95% positive predictive value and sensitivity) and exercise (>90% positive predictive value and sensitivity), (2) lead migration of less than 1.4 cm in all leads, and (3) no lead fracture, permanent signal loss, or infection. These findings demonstrate the feasibility of subcutaneous leads for synchronized timing of mechanical circulatory support while offering the advantages of less invasive surgery and associated risk factors.

CHAPTER V – Works Cited

1. Koenig, S.C., et al., *Development and early testing of a simple subcutaneous counterpulsation device*. ASAIO J, 2006. **52**(4): p. 362-7.
2. Giridharan, G.A., et al., *Predicted hemodynamic benefits of counterpulsation therapy using a superficial surgical approach*. ASAIO J, 2006. **52**(1): p. 39-46.
3. Lloyd-Jones, D., et al., *Heart disease and stroke statistics—2010 update*. Circulation, 2010. **121**(7): p. e46-e215.
4. Rosamond, W., et al., *Heart disease and stroke statistics--2008 update: a report from the American Heart Association Statistics Committee and Stroke Statistics Subcommittee*. Circulation, 2008. **117**(4): p. e25.
5. Muntwyler, J., et al., *One-year mortality among unselected outpatients with heart failure*. Eur Heart J, 2002. **23**(23): p. 1861-6.
6. Pitt, B., et al., *The effect of spironolactone on morbidity and mortality in patients with severe heart failure*. Randomized Aldactone Evaluation Study Investigators. N Engl J Med, 1999. **341**(10): p. 709-17.
7. *e-Statistics Report on Transplant, Waiting List and Donor Statistics, in 2008 Summary Statistics, January 1 to December 31st, 2008*. 2008, Canadian Institute for Health Information.
8. Koenig, S.C., et al., *Acute hemodynamic efficacy of a 32-ml subcutaneous counterpulsation device in a calf model of diminished cardiac function*. ASAIO journal (American Society for Artificial Internal Organs: 1992), 2008. **54**(6): p. 578.
9. Moskowitz, A.J., E.A. Rose, and A.C. Gelijns, *The cost of long-term LVAD implantation*. The Annals of thoracic surgery, 2001. **71**(3): p. S195-S198.
10. Westaby, S. and P. Poole-Wilson, *Mechanical circulatory support in the UK*. BMJ: British Medical Journal, 2007. **334**(7586): p. 167.
11. Reinlib, L. and W. Abraham, *Recovery from heart failure with circulatory assist: a working group of the National, Heart, Lung, and Blood Institute*. J Card Fail, 2003. **9**(6): p. 459-63.
12. Nanas, J.N. and S.D. Moulopoulos, *Counterpulsation: historical background, technical improvements, hemodynamic and metabolic effects*. Cardiology, 1994. **84**(3): p. 156-67.
13. Scheidt, S., et al., *Intra-aortic balloon counterpulsation in cardiogenic shock. Report of a co-operative clinical trial*. N Engl J Med, 1973. **288**(19): p. 979-84.
14. Lefemine, A.A., et al., *Assisted circulation. III. The effect of synchronized arterial counterpulsation on myocardial oxygen consumption and coronary flow*. Am Heart J, 1962. **64**: p. 789-95.
15. Papaioannou, T.G. and C. Stefanadis, *Basic principles of the intraaortic balloon pump and mechanisms affecting its performance*. ASAIO J, 2005. **51**(3): p. 296-300.
16. Bolooki, H., ed. *Clinical Application of the Intra-Aortic Balloon Pump*. 3rd ed. 1998, Futura: New York.

17. Santa-Cruz, R.A., M.G. Cohen, and E.M. Ohman, *Aortic counterpulsation: a review of the hemodynamic effects and indications for use*. Catheter Cardiovasc Interv, 2006. **67**(1): p. 68-77.
18. Tsagalou, E.P., M.I. Anastasiou-Nana, and J.N. Nanas, *Intra-aortic balloon counterpulsation for the treatment of myocardial infarction complicated by acute severe heart failure*. Congest Heart Fail, 2009. **15**(1): p. 35-40.
19. Dunkman, W.B., et al., *Clinical and hemodynamic results of intraaortic balloon pumping and surgery for cardiogenic shock*. Circulation, 1972. **46**(3): p. 465-77.
20. Weiss, A.T., et al., *Regional and global left ventricular function during intra-aortic balloon counterpulsation in patients with acute myocardial infarction shock*. Am Heart J, 1984. **108**(2): p. 249-54.
21. Kern, M.J., et al., *Enhanced coronary blood flow velocity during intraaortic balloon counterpulsation in critically ill patients*. J Am Coll Cardiol, 1993. **21**(2): p. 359-68.
22. Ishihara, M., et al., *Effects of intraaortic balloon pumping on coronary hemodynamics after coronary angioplasty in patients with acute myocardial infarction*. Am Heart J, 1992. **124**(5): p. 1133-8.
23. Bhayana, J.N., et al., *Effects of intraaortic balloon pumping on organ perfusion in cardiogenic shock*. J Surg Res, 1979. **26**(2): p. 108-13.
24. Sugita, Y., et al., *The effect of intraaortic balloon pumping (IABP) on pulmonary circulation*. Trans Am Soc Artif Intern Organs, 1985. **31**: p. 389-94.
25. Hilberman, M., et al., *Effect of the intra-aortic balloon pump upon postoperative renal function in man*. Crit Care Med, 1981. **9**(2): p. 85-9.
26. Jeevanandam, V., et al., *Circulatory assistance with a permanent implantable IABP: initial human experience*. Circulation, 2002. **106**(12 Suppl 1): p. I183-8.
27. Sun, B. *Sunshine: Chronic Aortic Counterpulsation for Class III Heart Failure*. in *Transcatheter Cardiovascular Therapeutics*. 2009. San Francisco, CA.
28. Koenig, S.C., et al., *Development and early testing of a simple subcutaneous counterpulsation device*. ASAIO journal (American Society for Artificial Internal Organs: 1992), 2006. **52**(4): p. 362.
29. Goldich, G., *Getting in sync with intra-aortic balloon pump therapy*. Nursing2011 Critical Care, 2011. **6**(3): p. 14.
30. Lobodzinski, S.S. and M.M. Laks, *Comfortable textile-based electrocardiogram systems for very long-term monitoring*. Cardiology journal, 2008. **15**(5): p. 477.
31. Kautzner, J., et al., *Technical aspects of implantation of LV lead for cardiac resynchronization therapy in chronic heart failure*. Pacing and clinical electrophysiology, 2004. **27**(6p1): p. 783-790.
32. Schwartzman, D., et al., *Postoperative lead-related complications in patients with nonthoracotomy defibrillation lead systems*. Journal of the American College of Cardiology, 1995. **26**(3): p. 776-786.
33. Lawton, J.S., et al., *Sensing lead-related complications in patients with transvenous implantable cardioverter-defibrillators*. The American Journal of Cardiology, 1996. **78**(6): p. 647-651.
34. Bardy, G.H., et al., *An entirely subcutaneous implantable cardioverter-defibrillator*. N Engl J Med, 2010. **363**(1): p. 36-44.

35. Grace, A., et al., *Evaluation of four distinct subcutaneous implantable defibrillator (S-ICD®) lead systems in humans*. Heart Rhythm, 2006. **3**: p. S128-S129.
36. Iglesias, J.F., et al., *The implantable loop recorder: a critical review*. Kardiovaskuläre Medizin, 2009. **12**(3): p. 85-93.
37. Ho, C. and S. Kurtzman, *Three perspectives of cardiac electrical activity*. Biomedical sciences instrumentation, 2001. **37**: p. 325.
38. Fotuhi, P., et al., *R Wave Detection by Subcutaneous ECG. Possible Use for Analyzing R R Variability*. Annals of noninvasive electrocardiology, 2001. **6**(1): p. 18-23.
39. Bellardine, B., et al., *Is Surface ECG a Useful Surrogate for Subcutaneous ECG?* Pacing and Clinical Electrophysiology, 2010. **33**(2): p. 135-145.
40. van Dam, P., et al., *Improving sensing and detection performance in subcutaneous monitors*. Journal of Electrocardiology, 2009. **42**(6): p. 580-583.
41. Hassler, C., *ECG lead placement in quadrupeds*. Telemetry Times (Technical Note, 8/3/94). Data Sciences Int, 1994.
42. Koenig, S.C., et al., *Integrated data acquisition system for medical device testing and physiology research in compliance with good laboratory practices*. Biomedical Instrumentation & Technology, 2004. **38**(3): p. 229-240.
43. Drew, G. and S. Koenig, *Biomedical patient monitoring, data acquisition, and playback with LabVIEW*. Olansen JB, Rosow E. Virtual Bio-Instrumentation: Biomedical, Clinical, and Healthcare Applications in LabVIEW, Prentice Hall, Upper Saddle River, NJ, 2002: p. 180–186.
44. Gomes, J.A., et al., *Optimal bandpass filters for time-domain analysis of the signal-averaged electrocardiogram*. The American journal of cardiology, 1987. **60**(16): p. 1290-1298.
45. Thakor, N.V., J.G. Webster, and W.J. Tompkins, *Estimation of QRS complex power spectra for design of a QRS filter*. Biomedical Engineering, IEEE Transactions on, 1984(11): p. 702-706.
46. Frankel, R.A., et al., *A filter to suppress ECG baseline wander and preserve ST-segment accuracy in a real-time environment*. Journal of electrocardiology, 1991. **24**(4): p. 315-323.
47. Pino, E., et al. *Real-Time ECG Algorithms for Ambulatory Patient Monitoring*. 2005: American Medical Informatics Association.

1243 Bourbon Ave
Louisville, KY 40213
(502) 649-3924
spwarren7@gmail.com

EDUCATION

M. Eng. in Bioengineering Expected July 2011
B. S. in Bioengineering Received May 2010
J.B. Speed School of Engineering, UofL, Louisville, KY
GPA 3.98 / 4.0
Dates attended: 8/2006 - present Hours Completed: 133
Graduated Summa Cum Laude

High School Diploma May 2006
North Oldham High School, Goshen, KY
Graduated Summa Cum Laude

ACADEMIC AWARDS AND HONORS

1. **Rolando “Chip” Cheng, Jr. Memorial Scholarship** - 2010
Recipient
Recognizes a bioengineering student in the graduating class whose character reflects humanitarianism, innovative mind, and entrepreneurial spirit.
2. **Jerry and Pat Sturgeon Academic Excellence Award** – 2010
Recipient
Awarded annually to the student achieving the highest scholastic performance after completing their Bachelor of Science degree in Bioengineering.
3. **U of L Trustee’s Scholarship**
Full Tuition, room and board for five years.
4. **Dean’s List** – Fall 2006, Spring 2009, Fall 2009
5. **Dean’s Scholar** – Spring 2007, Summer 2007, Fall 2007, Spring 2008, Summer 2008,
6. **Tau Beta Pi Engineering Honor Society**
7. **2nd Place Undergraduate Poster**, *Development of Microscale Polymer Fibers as Scaffolds for the Engineering of Tissue*, University of Louisville Engineering Exposition, 2008

EMPLOYMENT HISTORY

Spence Cardiac Research Inc, Louisville, KY

Co-op Engineer under Paul Spence, MD

May 2009 – Oct 2010

- Work with an international team of engineers and physicians to develop and test medical products
- Design, prototype and test early-stage medical devices in the fields of atrial fibrillation and sports medicine (preliminary patents filed)
- Design and test components, and assist in pre-GLP animal and cadaver studies for a counterpulsation cardiac assist device
- Assist in writing and editing NIH and state-funded grant applications for the development of medical products

Ethicon, Somerville, NJ

Co-op in R&D Ethicon Products

Aug 2008 – Dec 2008

- Designed, fabricated and tested prototype devices for next-generation wound closure
- Drafted design plan for discovery-phase project
- Participated in a cadaver lab and received OR observation training

University of Louisville, Louisville, KY

Member of the Bio-MEMS Research Team

June 2007 – May 2009

- Collaborated with a team of engineers and professors on university PhD-level and Master-level research projects
- Performed sterile cell culture, fluorescent staining, and ELISA assay analysis
- Operated and programmed micromill machine for direct-write fabrication of polymer nanofibers

Sonopress USA, Louisville, KY

Intern under Eric Jones, M. Eng.

June 2006 - Aug 2006

- Monitored line productivity
- Created production efficiency charts
- Worked with superiors to increase line efficiency

STATE AND FEDERAL EXTRAMURAL FUNDING

NIH SBIR PHASE II GRANT: 2R44HL083586-02A1

Development of a Counterpulsation Therapy Device

August 2008 – July 2010

Award Amount: \$1,437,630

Role: Biomedical Engineer

Assist in device animal studies and the collection and analysis of necropsy data.

NIH SBIR PHASE II GRANT: 2R44HL088760-02

Portable Pneumatic Driver for Counterpulsation Therapy

May 2009 – May 2011

Award Amount: \$1,580,460

Role: Biomedical Engineer

Assist in driver animal studies and the collection and analysis of necropsy data.

NIH SBIR PHASE I GRANT: 1R43HL102981-01

Counterpulsation Device with Integrated ECG Sensing

May 2010 – October 2010

Award Amount: \$261,965

Role: Biomedical Engineer

Provide ECG design and fabrication support, participate in animal studies, and collect and analyze signaling data.

KENTUCKY SCIENCE AND TECHNOLOGY CORPORATION SBIR PHASE II MATCHING FUND: KSTC-184-512-08-054

Development of a Counterpulsation Therapy Device

October 2008 – April 2010

Award Amount: \$500,000

Role: Co-op Student

Responsible for design, fabrication, and benchtop testing of prototype medical devices as well as participating in mock flow loop and animal model experiments by assisting with data collection and analysis.

KENTUCKY SCIENCE AND TECHNOLOGY CORPORATION SBIR PHASE II MATCHING FUND: KSTC-184-512-10-077

Patient Friendly Driver for Counterpulsation Therapy

January 2010 – July 2011

Award Amount: \$430,000

Role: Engineer

Complete the development of the portable driver in collaboration with LB Technology and complete testing and analysis of the display and patient messaging over a range of clinically relevant scenarios.

PUBLISHED IN PEER-REVIEWED JOURNALS

1. Ising M, **Warren SP**, Sobieski MA, Slaughter MS, Koenig SC, Giridharan GA. *Flow Modulation Algorithms for Continuous Flow Left Ventricular Assist Devices to Increase Vascular Pulsatility: A Computer Simulation Study*. Cardiovascular Engineering and Technology, Vol. 2, No. 2, June 2011.
2. Berry SM, **Warren SP**, Hilgart DA, Schworer AT, Gobin AS, Cohn RW, Keynton RS. *Endothelial Cell Scaffolds Generated by 3D Direct Writing of Biodegradable Polymer Microfibers*. Biomaterials, Vol. 32, No. 7, March 2011.

PUBLISHED IN PEER-REVIEWED CONFERENCE PROCEEDINGS

1. **Warren SP**, Dowling R, Tompkins LH, Spence PA, Koenig SC, Giridharan GA, Sobieski MA, Sherwood L, Slaughter MS. *Continued Development of a Long-Term Counterpulsation Device (Symphony) for Partial Circulatory Support*. 21st Century Treatment of Heart Failure: Synchronizing Surgical and Medical Therapies for Better Outcomes, Cleveland, OH, 2010.
2. Ising MS, **Warren SP**, Slaughter MS, Koenig SC, Sobieski MA, Giridharan GA. *Delivering Physiologic Pulse Pressures with Continuous Flow Ventricular Assist Devices*. 56th Annual ASAIO Conference, Baltimore, MD, 2010.
3. Berry SM, Hilgart DA, **Warren SP**, Schworer AT, Gobin AS, Cohn RW, Keynton RS. *Fabrication of Microvascular-Scale Polymer-Fiber Scaffolds for the Growth of Endothelial Cells*. 5th International Conference on Microtechnologies in Medicine and Biology, Quebec City, Quebec, 2009.
4. Berry SM, Cambron SD, **Warren SP**, Pabba S, Cohn RW, Keynton RS. *Direct Write Fabrication of Polymer Fibers for*

Microscale Applications. 17th Biennial UGIM, Louisville, KY, 2008.

PATENT AND INTELLECTUAL PROPERTY

1. **System and Method for Providing Cardiac Support**
Ising MS, Warren SP, Sobieski MA, Slaughter MS, Koenig SC, Giridharan GA
University of Louisville Research Foundation
Provisional Patent Application Submitted, 2010
2. **Ablation Catheter Device and Methods**
Spence PA, Warren SP, et al.
SCR Inc.
Pending Application Submission 2011

LECTURES

1. **Subcutaneous ECG Cooperative Education Project**
Overview of work done at SCR Inc. on subcutaneous ECG animal trials.
Presented to the **Speed School Industrial Board of Advisors**
University of Louisville, November 12, 2010
2. **Pediatric Cardiology**
Overview of pediatric hemodynamics, cardiovascular disorders, medical therapies and surgical procedures.
Presented to graduate-level class, **Cardiovascular Dynamics I**
University of Louisville, July 12, 2010
3. **Embryonic Cardiogenesis**
Overview of embryonic cardiovascular development, fetal vs postnatal circulation, common cardiac congenital defects and therapies, and fetal cardiac surgery.
Presented to graduate-level class, **Cardiovascular Dynamics I**
University of Louisville, June 21, 2010
4. **Ventricular Assist Devices**
Brief introduction to the technologies and applications of ventricular assist devices.
Presented to participants in the **INSPIRE Program**, a summer program targeting minority high school students with interests in engineering

University of Louisville, June 9, 2010.

CONFERENCES ATTENDED

1. 21st Century Treatment of Heart Failure – 2010
2. American Society for Artificial Internal Organs – 2010
3. Kentucky Innovation and Entrepreneurship Conference – 2010
4. Boston Atrial Fibrillation Symposium – 2010
5. International Conference on Microtechnologies in Medicine and Biology – 2009
6. University Government Industry Micro/Nano Symposium – 2008

SOCIETIES

1. American Society for Artificial Internal Organs
2. Biomedical Engineering Society
3. American Association for the Advancement of Science
4. Tau Beta Phi Engineering Honor Society
5. Louisville Astronomical Society
6. Kappa Sigma Fraternity

EXTRACURRICULAR ACTIVITIES

1. University of Louisville Biomedical Engineering Society
 - Vice President (2009-2010)
2. Kappa Sigma Fraternity
 - Greek Week Chair
3. SSI Advanced Scuba Diver
 - Stress and Rescue Certified
 - Over 35 logged dives
4. Louisville Astronomy Society

COMMUNITY SERVICE

1. “Opposition Force” Volunteer, Fort Knox
Zussman Military
Training Center
2. St. James Art Festival Volunteer
3. U of L Spring Football Game Volunteer
4. Arcadia Community Center Childcare Volunteer
5. Louisville Astronomy Society Public Viewing

CHAPTER VII – APPENDIX

I. EXTENDED STATISTICS TABLE

At Rest Data (30 second clips, 400 Hz)

	Epochs	PPV	Sensitivity	R Voltage (mV)	T Voltage (mV)	P Voltage (mV)	P/R	T/R	P-R (s)	T-R (s)	R-R (s)	T-T (s)	P-T (s)	QRS Length (s)	Mean V (mV)	
J066 (7 day)	Subq1	137	0.999 ± .001	0.996 ± .002	1.12	0.52	0.534	0.469	0.464	0.139	0.254	0.648	0.655	0.651	0.0502	0.00435
	Subq2	125	0.971 ± .020	0.925 ± .021	0.96	0.60	0.405	0.423	0.628	0.168	0.251	0.645	0.693	0.693	0.0536	0.02861
	Subq3	135	0.997 ± .002	0.972 ± .006	0.55	0.17	0.125	0.254	0.354	0.174	0.230	0.637	0.722	0.720	0.0516	0.02308
J067 (7 day)	Subq1	158	0.999 ± .001	0.998 ± .001	1.15	0.48	0.234	0.204	0.430	0.159	0.253	0.593	0.593	0.594	0.0565	0.00401
	Subq2	157	0.998 ± .001	0.994 ± .003	0.85	0.34	0.169	0.201	0.407	0.165	0.251	0.592	0.675	0.680	0.0557	0.00357
	Subq3	151	0.995 ± .002	0.987 ± .006	0.64	0.34	0.216	0.329	0.542	0.167	0.247	0.596	0.600	0.599	0.0563	0.00311
J068 (7 day)	Subq1	101	0.986 ± .012	0.962 ± .013	0.20	0.24	0.206	0.349	0.433	0.148	0.213	0.595	0.554	0.557	0.0183	0.00220
	Subq2	158	0.995 ± .007	0.984 ± .008	0.69	0.21	0.147	0.211	0.300	0.155	0.256	0.629	0.633	0.633	0.0513	-0.00383
	Subq3	95	0.995 ± .012	0.992 ± .012	0.83	0.21	0.159	0.191	0.252	0.142	0.271	0.619	0.621	0.620	0.0514	0.00215
J069 (7 day)	Subq1	160	0.996 ± .002	0.992 ± .003	1.04	0.32	0.220	0.213	0.316	0.139	0.231	0.567	0.571	0.571	0.0248	0.01963
	Subq2	83	0.994 ± .003	0.952 ± .009	0.83	0.33	0.239	0.284	0.399	0.149	0.202	0.552	0.561	0.558	0.0526	0.13096
	Subq3	83	0.969 ± .014	0.890 ± .033	0.59	0.26	0.161	0.266	0.446	0.143	0.228	0.587	0.837	0.841	0.0439	-0.00860
J070 (14 day)	Subq1	254	0.982 ± .012	0.969 ± .011	0.36	0.26	0.135	0.183	0.353	0.158	0.261	0.523	0.669	0.668	0.0206	0.00125
	Subq2	263	0.999 ± .001	0.997 ± .001	0.88	0.24	0.157	0.179	0.269	0.142	0.263	0.695	0.697	0.697	0.0534	-0.00258
	Subq3	249	0.985 ± .005	0.968 ± .013	0.72	0.33	0.202	0.298	0.486	0.178	0.266	0.741	0.766	0.768	0.0598	0.00077
J071 (14 day)	Subq1	201	0.996 ± .004	0.986 ± .007	0.94	0.48	0.374	0.396	0.502	0.143	0.276	0.656	0.656	0.658	0.0227	0.01487
	Subq2	153	0.991 ± .004	0.935 ± .016	0.95	0.46	0.341	0.357	0.491	0.147	0.276	0.654	0.853	0.854	0.0542	0.04851
	Subq3	155	0.990 ± .005	0.950 ± .012	0.87	0.42	0.284	0.332	0.483	0.145	0.266	0.623	0.630	0.625	0.0533	-0.00910

Treadmill Data (10-20 minute clips, 400 Hz)

	Epochs	PPV	Sensitivity	R Voltage (mV)	T Voltage (mV)	P Voltage (mV)	P/R	T/R	P-R (s)	T-R (s)	R-R (s)	T-T (s)	P-T (s)	QRS Length (s)	Mean V (mV)	
J070 (14 day)	Subq1	5	0.965 ± .076	0.972 ± .035	0.660	0.283	0.173	0.262	0.429	0.120	0.246	0.607	0.616	0.652	0.0408	0.0134
	Subq2	4	0.992 ± .016	0.985 ± .021	1.045	0.424	0.259	0.246	0.407	0.128	0.265	0.631	0.632	0.632	0.0551	0.0035
	Subq3	3	0.986 ± .064	0.922 ± .110	1.003	0.383	0.221	0.220	0.382	0.134	0.275	0.654	0.659	0.661	0.0543	0.0021
J071 (14 day)	Subq1	6	0.974 ± .012	0.919 ± .039	0.923	0.664	0.486	0.522	0.751	0.134	0.262	0.570	0.576	0.576	0.0281	0.0027
	Subq2	3	0.969 ± .067	0.839 ± .290	0.778	0.432	0.278	0.357	0.580	0.139	0.265	0.577	0.583	0.586	0.0545	0.1513
	Subq3	3	0.992 ± .023	0.960 ± .067	1.060	0.492	0.418	0.397	0.473	0.138	0.264	0.575	0.576	0.577	0.0534	0.0155

- Notes:**
1. Due to the nature of the P and T wave detection algorithms, the intervals are measured peak to peak
 2. Clips that had no or incomplete lead connections (the cow pulled out the connections, the wire became untaped and strained, etc) were removed from the data set, hence why the number of clips vary between lead configurations and animals.
 3. The upper bound of the 95% confidence interval does not exceed 1.

Lead Migration Statistics

	Animal						Average	95% CI	Std. Dev	
	1	2	3	4	5	6				
Lead Number	1	8.6	4.2	3.8	5.9	5.2	7.7	5.9	2.0	1.92
	2	3.2	2.4	9.7	8.6	3.5	8.3	5.9	3.4	3.26
	3	2.6	1.9	0.9	3.5	3.9	8.9	3.6	2.9	2.80
	4	10.2	2.5	13.7	1.1	2.0	5.1	5.8	5.3	5.09
	5	7.7	5.3	10.5	4.0	4.0	3.5	5.8	2.9	2.76
	6	2.3	2.9	6.5	5.7	4.6	4.5	4.4	1.7	1.60
Average	5.8	3.2	7.5	4.8	3.9	6.3				
95% CI	3.6	1.4	5.0	2.7	1.1	2.3				
Std. Dev	3.47	1.30	4.72	2.55	1.07	2.23				
Overall	Average	95% CI	Std. Dev							
	5.24	3.02	1.02							

II. CODE APPENDIX (Printed in columns)

A. Master Algorithm Code

```

%Test version of filtering for ECG signal
%By Sean Warren
%Last modified: 6/6/2011
%Make sure you have an empty stats matrix
(totalfiles x 65) of zeros

%USE THESE COMMANDS BEFORE
STARTING A FRESH SET (without the %)
% stats=zeros(TOTALFILES,65);
% save('stats.mat','stats');

clc
clear all

diary('J066_log.txt')

ampfactor=10;
counter=273;
fileend=317;
cow='J071';
hz=400;
load('signalcheck.mat');
load('stats.mat');

for i=counter+1:1:fileend+1
save('stats.mat','stats')
    hold off
    filename=sprintf('%s%s%i%s',cow,'_',i-
1, '.mat'); % Treadmill_ for treadmill

    disp(sprintf('%s',' '))
    disp(sprintf('%s','-----'))
    disp(sprintf('%s%s','File ',filename))

        if signalcheck(i,1)==0 ||
sum(signalcheck(i,2:1:4))==0
            disp(sprintf('%s%s','Not enough data
for signal on file ',filename))
        else

%Load signal
clear subq1
clear subq2
clear subq3
clear screw
clear t

load(filename);

clear t
t=(1/hz:1/hz:length(subq1)/hz);

%Screw Normalize
%Normalize
subq1=subq1(1:1:end-1);
subq2=subq2(1:1:end-1);
subq3=subq3(1:1:end-1);
subq1=subq1/(max(abs(subq1)));
subq2=subq2/(max(abs(subq2)));
subq3=subq3/(max(abs(subq3)));
dscrew=diff(screw);
screw=screw(1:1:end-1);
dscrew=dscrew/(max(abs(dscrew)));
screw=screw/(max(abs(screw)));
t=t(1:1:end-1);
t1=t;
t2=t;
t3=t;

%-----
%Align

screwseconds11=0;
screwseconds22=0;

%Timeframe for finding alignment maxima

if stats(i,28)==1 && stats(i,29)==1 &&
stats(i,30)==1 && stats(i,31)==1
else
disp(sprintf('%s','NOTE: -1 at any time for
the SubQ to align via Xth beat, -10 to skip
data set, -7 to invert data set'))
end

signalfilt=screw;

%Signal Drift, FIR method
fs=400;           % sampling rate
fch=50;           % high pass cutoff freq
fcl=1;           % low pass cutoff freq
N=700;           % filter order
filt=fir1(N,[fcl/(fs/2) fch/(fs/2)]);
%Design high pass filter
signalfilt=filtfilt(filt,1,signalfilt);

%SG Filter
k=10;             %polynomial order for
SG filter
f=50+1;          % frame length (must
be odd)

```



```

signalfilt=sgolayfilt(signalfilt,k,f);
signal=signalfilt;

stdev=2.5*std(signalfilt(:));
m=mean(signalfilt(:));

for j=1:length(signalfilt);
    if signalfilt(j) > (m+stdev)
        signalfilt(j)=m+stdev;
    end
    if signalfilt(j) < (m-stdev)
        signalfilt(j)=m-stdev;
    end
end

screwfiltered=signalfilt;

if stats(i,28)==1
    usediff=stats(i,32);
else

plot(t,screw,t1,subq1)
legend('screw','subq1');
hold
xlim([0 2]);
pause
plot(t,dscrew,'color','red')
legend('screw','subq1','screwder');
pause
plot(t,screwfiltered,'color','magenta')
legend('screw','subq1','screwder','filtered');
hold
usediff=input('Use screw (0), screw
derivative (1), or filtered (2)?');
stats(i,32)=usediff;
end
if usediff==1
    screw=dscrew;
elseif usediff==2;
    screw=screwfiltered;
end
if stats(i,28)==1
    usediff=stats(i,33);
else
plot(t,screw,t1,subq1)
xlim([0 1]);
usediff=input('Invert (1) or not (0)?');
stats(i,33)=usediff;
end
if usediff==1
    screw=-screw;
end
if stats(i,28)==1
    screwseconds1=stats(i,34);
        screwseconds2=stats(i,35);
    else
plot(t,screw,t1,subq1)
xlim([0 1]);
screwseconds1=input('Epicardial first R
wave timesearch(suggest=0,.5):');
screwseconds2=input(' ');
stats(i,34)=screwseconds1;
stats(i,35)=screwseconds2;
end
screwseconds1=round(400*screwseconds1);
screwseconds2=round(400*screwseconds2);
screwseconds11=screwseconds1;
screwseconds22=screwseconds2;
[screwmax,screwmaxindex]=max(screw(1,s
crewseconds1+1:1:screwseconds2+1));
screwmax11=screwmax;
screwmaxindex11=screwmaxindex;
%Ethreshold=input('Epicardial R wave
threshold(suggest >.8):');

suggest=sprintf('%s%4.2f%s','SubQ first R
wave timeframe (>',screwmaxindex/hz,'
sec):');
suggest2=sprintf('%s','SubQ X R wave
timeframe:');

if signalcheck(i,2)==1
%Subq1 alignment
if stats(i,29)==1
    qseconds1=stats(i,36);
    qseconds2=stats(i,37);
else
plot(t,screw,t,subq1)
legend('screw','subq')
xlim([0 1]);
qseconds1=input(suggest);
qseconds2=input(' ');
stats(i,36)=qseconds1;
stats(i,37)=qseconds2;
end

if qseconds1== -7
    subq1=-subq1;
    if stats(i,29)==1
        qseconds1=stats(i,38);
        qseconds2=stats(i,39);
    else
        plot(t,screw,t,subq1)
        legend('screw','subq')
        xlim([0 1]);
        qseconds1=input(suggest);
        qseconds2=input(' ');
        stats(i,38)=qseconds1;
        stats(i,39)=qseconds2;
    end
end

```

```

end

if qseconds1==-1
    if stats(i,29)==1
        screwseconds1=stats(i,38);
        screwseconds2=stats(i,39);
    else
        plot(t,screw,t,subq1)
        legend('screw','subq')
        xlim([0 2.5]);
        screwseconds1=input('Epicardial X R wave
timesearch:');
        screwseconds2=input(' ');
        stats(i,38)=screwseconds1;
        stats(i,39)=screwseconds2;
    end
    screwseconds1=round(400*screwseconds1);
    screwseconds2=round(400*screwseconds2);
    [screwmax,screwmaxindex]=max(screw(1,s
crewseconds1+1:1:screwseconds2+1));
    %Ethreshold=input('Epicardial R wave
threshold(suggest >.8):');
    if stats(i,29)==1
        qseconds1=stats(i,66);
        qseconds2=stats(i,67);
    else
        qseconds1=input(suggest2);
        qseconds2=input(' ');
        stats(i,66)=qseconds1;
        stats(i,67)=qseconds2;
    end
elseif qseconds1==-10
    signalcheck(i,2)=0;
end

qseconds1=round(400*qseconds1);
qseconds2=round(400*qseconds2);
%Qthreshold=input('SubQ R wave
threshold(suggest >.8):');
end

if signalcheck(i,2)==1
%%Search for R wave index
[qmax,qmaxindex]=max(subq1(1,qseconds1
+1:1:qseconds2+1));
QEdiff=(qmaxindex+qseconds1)-
(screwmaxindex+screwseconds1);
if QEdiff<0;
    if QEdiff>-20
        else
            disp(sprintf('%s','Bad matchup please
redo'));
        end
    else
        subq1=subq1(1,QEdiff+1:1:end);
        t1=t(1,1:1:end-QEdiff);
    end
else
    subq1=zeros(1,length(screw));
end

screwseconds1=screwseconds11;
screwseconds2=screwseconds22;
screwmax=screwmax11;
screwmaxindex=screwmaxindex11;

if signalcheck(i,3)==1
%%Subq2 alignment
if stats(i,30)==1
    qseconds1=stats(i,40);
    qseconds2=stats(i,41);
else
    plot(t,screw,t,subq2)
    legend('screw','subq')
    xlim([0 1]);
    qseconds1=input(suggest);
    qseconds2=input(' ');
    stats(i,40)=qseconds1;
    stats(i,41)=qseconds2;
end

if qseconds1== -7
    subq2=-subq2;
    if stats(i,30)==1
        qseconds1=stats(i,42);
        qseconds2=stats(i,43);
    else
        plot(t,screw,t,subq2)
        legend('screw','subq')
        xlim([0 1]);
        qseconds1=input(suggest);
        qseconds2=input(' ');
        stats(i,42)=qseconds1;
        stats(i,43)=qseconds2;
    end
end

if qseconds1== -1
    if stats(i,30)==1
        screwseconds1=stats(i,42);
        screwseconds2=stats(i,43);
    else
        plot(t,screw,t,subq2)
        legend('screw','subq')
        xlim([0 2.5]);

        screwseconds1=input('Epicardial X R wave
timesearch:');
        screwseconds2=input(' ');
        screwseconds1=round(400*screwseconds1);
        screwseconds2=round(400*screwseconds2);
        stats(i,42)=screwseconds1;
    end
end

```

```

    stats(i,43)=screwseconds2;
    end
[screwmax,screwmaxindex]=max(screw(1,s
crewseconds1+1:1:screwseconds2+1));
%Ethreshold=input('Epicardial R wave
threshold(suggest >.8):');
    if stats(i,30)==1
        qseconds1=stats(i,68);
        qseconds2=stats(i,69);
    else
qseconds1=input(suggest2);
qseconds2=input(' ');
        stats(i,68)=qseconds1;
        stats(i,69)=qseconds2;
    end
elseif qseconds1==-10
    signalcheck(i,3)=0;
end

qseconds1=round(400*qseconds1);
qseconds2=round(400*qseconds2);
%Qthreshold=input('SubQ R wave
threshold(suggest >.8):');
end

if signalcheck(i,3)==1
%%Search for R wave index
[qmax,qmaxindex]=max(subq2(1,qseconds1
+1:1:qseconds2+1));
QEdiff=(qmaxindex+qseconds1)-
(screwmaxindex+screwseconds1);
if QEdiff<0;
    if QEdiff>-20
        else
            disp(sprintf('%s','Bad matchup please
redo'));
        end
    else
        subq2=subq2(1,QEdiff+1:1:end);
        t2=t(1,1:1:end-QEdiff);
    end
    else
        subq2=zeros(1,length(screw));
    end

screwseconds1=screwseconds11;
screwseconds2=screwseconds22;
screwmax=screwmax11;
screwmaxindex=screwmaxindex11;

if signalcheck(i,4)==1
%Subq3 alignment
if stats(i,31)==1
    qseconds1=stats(i,44);
    qseconds2=stats(i,45);
else
    plot(t,screw,t,subq3)
    legend('screw','subq')
    xlim([0 2]);
    qseconds1=input(suggest);
    qseconds2=input(' ');
    stats(i,44)=qseconds1;
    stats(i,45)=qseconds2;
end

if qseconds1==-7
    subq3=-subq3;
    if stats(i,31)==1
        qseconds1=stats(i,46);
        qseconds2=stats(i,47);
    else
        plot(t,screw,t,subq3)
        legend('screw','subq')
        xlim([0 2]);
        qseconds1=input(suggest);
        qseconds2=input(' ');
        stats(i,46)=qseconds1;
        stats(i,47)=qseconds2;
    end
end

if qseconds1==-1
    if stats(i,31)==1
        screwseconds1=stats(i,46);
        screwseconds2=stats(i,47);
    else
        plot(t,screw,t,subq3)
        legend('screw','subq')
        xlim([0 10]);

        screwseconds1=input('Epicardial X R wave
timesearch:');
        screwseconds2=input(' ');
        screwseconds1=round(400*screwseconds1);
        screwseconds2=round(400*screwseconds2);
        stats(i,46)=screwseconds1;
        stats(i,47)=screwseconds2;
    end
    [screwmax,screwmaxindex]=max(screw(1,s
crewseconds1+1:1:screwseconds2+1));
%Ethreshold=input('Epicardial R wave
threshold(suggest >.8):');
    if stats(i,31)==1
        qseconds1=stats(i,70);
        qseconds2=stats(i,71);
    else
qseconds1=input(suggest2);
qseconds2=input(' ');
        stats(i,70)=qseconds1;
        stats(i,71)=qseconds2;
    end
elseif qseconds1==-10

```

```

    signalcheck(i,4)=0;
end

qseconds1=round(400*qseconds1);
qseconds2=round(400*qseconds2);
% Qthreshold=input('SubQ R wave
threshold(suggest >.8):');
end
if signalcheck(i,4)==1
%% Search for R wave index
[qmax,qmaxindex]=max(subq3(1,qseconds1
+1:1:qseconds2+1));
QEdiff=(qmaxindex+qseconds1)-
(screwmaxindex+screwseconds1);
if QEdiff<0;
    if QEdiff>-.20
        else
            disp(sprintf('%s','Bad matchup please
redo'));
        end
    else
        subq3=subq3(1,QEdiff+1:1:end);
        t3=t(1,1:1:end-QEdiff);
    end
else
    subq3=zeros(1,length(screw));
end

%-----
%-----

%Begin Fmin Filter Search

%-----
%-----

%Filtering

%Cut to same length
cutmatrix=[length(subq1),length(subq2),len
gth(subq3),length(screw)];
screw=screw(1,1:1:min(cutmatrix));
subq1=subq1(1,1:1:length(screw));
subq2=subq2(1,1:1:length(screw));
subq3=subq3(1,1:1:length(screw));
t=t(1,1:1:min(cutmatrix));
t1=t1(1,1:1:length(t));
t2=t2(1,1:1:length(t));
t3=t3(1,1:1:length(t));

%Add blank period at end (2 sec long)
subq1(1,(length(subq1)+1):1:(length(subq1)
+800))=zeros(1,800);
subq2(1,(length(subq2)+1):1:(length(subq2)
+800))=zeros(1,800);

```

```

subq3(1,(length(subq3)+1):1:(length(subq3)
+800))=zeros(1,800);
screw(1,(length(screw)+1):1:(length(screw)
+800))=zeros(1,800);
t((end+1):(end+800))=(t(end)+t(2)):(t(2)):(t(
end)+t(2)*800);
t1((end+1):(end+800))=(t1(end)+t1(2)):(t1(2)
):(t1(end)+t1(2)*800);
t2((end+1):(end+800))=(t2(end)+t2(2)):(t2(2)
):(t2(end)+t2(2)*800);
t3((end+1):(end+800))=(t3(end)+t3(2)):(t3(2)
):(t3(end)+t3(2)*800);

clear subq1o;
clear subq2o;
clear subq3o;
subq1o=subq1;
subq2o=subq2;
subq3o=subq3;

%subq1

signalfilt=subq1;

%Signal Drift, FIR method
fs=400;           %sampling rate
fch=50;           %high pass cutoff freq
fcl=1;           %low pass cutoff freq
N=700;           %filter order
filt=fir1(N,[fcl/(fs/2) fch/(fs/2)]);
%Design high pass filter
signalfilt=filtfilt(filt,1,signalfilt);

%SG Filter
k=10;             %polynomial order for
SG filter
f=50+1;          %frame length (must
be odd)
signalfilt=sgolayfilt(signalfilt,k,f);
signal=signalfilt;

stdev=2.5*std(signalfilt(:));
m=mean(signalfilt(:));

for j=1:1:length(signalfilt);
    if signalfilt(j) > (m+stdev)
        signalfilt(j)=m+stdev;
    end
    if signalfilt(j) < (m-stdev)
        signalfilt(j)=m-stdev;
    end
end

subq1=signalfilt;

%subq2

```

```

signalfilt=subq2;

%Signal Drift, FIR method
fs=400;           %sampling rate
fch=50;          %high pass cutoff freq
fcl=1;           %low pass cutoff freq
N=700;           %filter order
filt=fir1(N,[fcl/(fs/2) fch/(fs/2)]);
%Design high pass filter
signalfilt=filtfilt(filt,1,signalfilt);

%SG Filter
k=10;             %polynomial order for
SG filter
f=50+1;          %frame length (must
be odd)
signalfilt=sgolayfilt(signalfilt,k,f);
signal=signalfilt;

stdev=2.5*std(signalfilt(:));
m=mean(signalfilt(:));

for j=1:1:length(signalfilt);
    if signalfilt(j) > (m+stdev)
        signalfilt(j)=m+stdev;
    end
    if signalfilt(j) < (m-stdev)
        signalfilt(j)=m-stdev;
    end
end

subq2=signalfilt;

%subq3

signalfilt=subq3;

%Signal Drift, FIR method
fs=400;           %sampling rate
fch=50;          %high pass cutoff freq
fcl=1;           %low pass cutoff freq
N=700;           %filter order
filt=fir1(N,[fcl/(fs/2) fch/(fs/2)]);
%Design high pass filter
signalfilt=filtfilt(filt,1,signalfilt);

%SG Filter
k=10;             %polynomial order for
SG filter
f=50+1;          %frame length (must
be odd)
signalfilt=sgolayfilt(signalfilt,k,f);
signal=signalfilt;

stdev=2.5*std(signalfilt(:));

```

```

m=mean(signalfilt(:));

for j=1:1:length(signalfilt);
    if signalfilt(j) > (m+stdev)
        signalfilt(j)=m+stdev;
    end
    if signalfilt(j) < (m-stdev)
        signalfilt(j)=m-stdev;
    end
end

subq3=signalfilt;

%-----
%Normalization
subq1=subq1/(max(abs(subq1)));
subq2=subq2/(max(abs(subq2)));
subq3=subq3/(max(abs(subq3)));
screw=screw/(max(abs(screw)));

if stats(i,28)==1 && stats(i,29)==1 &&
stats(i,30)==1 && stats(i,31)==1
else
plot(t,screw,t1,subq1,t2,subq2,t3,subq3)
legend('screw','subq1','subq2','subq3')
xlim([0 10])

pause
end

%-----
%Threshold Detection Program

resolution=.17*400; %Detection
Resolution

% screw R wave detection

clear signalfilt;
clear trigger;
clear ts;

ts=t;
signalfilt=screw;

if stats(i,28)==1
    ithresh=stats(i,48);
    iblank=stats(i,49);
    spblank=stats(i,50);
else
plot(t,screw);
xlim([0 2.5]);

```

```

ithresh=input('First 3 R wave detection
threshold:');
iblack=input('Initial blanking period in
sec:');
spblank=input('Ignore period in sec:');
    stats(i,48)=ithresh;
    stats(i,49)=iblack;
    stats(i,50)=spblank;
end
iblack=iblack*400;
RRT1=0;
RRT2=0;
RRT3=0;
RRV1=0;
RRV2=0;
RRV3=0;
blankpercent=.80;
initialR=1;
initialrsec=.050;
ramppercnt=.80;
trigger=zeros(1,length(screw));
lastR=0;
RRTvariabilityfactor=.3;
RRVvariabilityfactor=.3;

blankpercent=.80;
initialR=1;
ppercent=.35;
tpercent=.50;
rpercentforward=.10;
rpercentbackward=.05;
initialrsec=.050;
ramppercnt=.80;
lastR=0;
RRTvariabilityfactor=.3;
RRVvariabilityfactor=.3;
clear pstats
clear rstats
clear tstats
pz=1;
rz=1;
tz=1;

for k=(1+spblank*400):1:(length(signalfilt)-
800);
    if initialR==3
        if signalfilt(k)>=ithresh && (k-
RRT1)>iblack
            trigger(k)=1;
            RRT3=RRT2;
            RRT2=RRT1;
            RRT1=k;
            lastR=k;
            RRV3=RRV2;
            RRV2=RRV1;

```

```

RRV1=max(signalfilt(k:(k+initialrsec*400))
);
        initialR=initialR+1;
        RRT3=RRT2-RRT3;
        RRT2=RRT1-RRT2;
        RRT1=RRT2;
        end
    elseif initialR==2
        if signalfilt(k)>=ithresh && (k-
RRT1)>iblack
            trigger(k)=1;
            RRT3=RRT2;
            RRT2=RRT1;
            RRT1=k;
            RRV3=RRV2;
            RRV2=RRV1;

RRV1=max(signalfilt(k:(k+initialrsec*400))
);
        initialR=initialR+1;
        end
    elseif initialR==1
        if signalfilt(k)>=ithresh
            trigger(k)=1;
            RRT3=RRT2;
            RRT2=RRT1;
            RRT1=k;
            RRV3=RRV2;
            RRV2=RRV1;

RRV1=max(signalfilt(k:(k+initialrsec*400))
);
        initialR=initialR+1;
        end
    else
        if signalfilt(k)>=ithresh && (k-
lastR)>(((RRT1+RRT2+RRT3)/3)*blankper
cent)
            trigger(k)=1;
            RRT3=RRT2;
            RRT2=RRT1;
            if (k-
lastR)>(((RRT1+RRT2+RRT3)/3)*(1+RRTv
ariabilityfactor) || (k-
lastR)<(((RRT1+RRT2+RRT3)/3)*(1-
RRTvariabilityfactor)
                else
                    RRT1=k-lastR;
                    %           %p,t,r wave capture and voltage
indexing
                    %           pwave(round(k-(k-
lastR)*(ppercnt+rpercentbackward)):round(
k-(k-
lastR)*rpercentbackward))=ECG(round(k-
(k-

```

```

lastR)*(ppercent+rpercentbackward)):round(
k-(k-lastR)*rpercentbackward));
%
pstats(1,pz)=abs(max(pwave(round(k-(k-
lastR)*(ppercent+rpercentbackward)):round(
k-(k-lastR)*rpercentbackward)))-
min(pwave(round(k-(k-
lastR)*(ppercent+rpercentbackward)):round(
k-(k-lastR)*rpercentbackward))));
%
twave(round(k-(k-lastR)*(1-
rpercentforward)):round(k-(k-lastR)*(1-
rpercentforward-tpercent)))=ECG(round(k-
(k-lastR)*(1-rpercentforward)):round(k-(k-
lastR)*(1-rpercentforward-tpercent)));
%
tstats(1,tz)=abs(max(twave(round(k-(k-
lastR)*(1-rpercentforward)):round(k-(k-
lastR)*(1-rpercentforward-tpercent))))-
min(twave(round(k-(k-lastR)*(1-
rpercentforward)):round(k-(k-lastR)*(1-
rpercentforward-tpercent)))));
%
rwave(round(k-(k-
lastR)*(rpercentbackward)):round(k+(k-
lastR)*(rpercentforward)))=ECG(round(k-
(k-lastR)*(rpercentbackward)):round(k+(k-
lastR)*(rpercentforward)));
%
rstats(1,rz)=abs(max(rwave(round(k-(k-
lastR)*(rpercentbackward)):round(k+(k-
lastR)*(rpercentforward))))-
min(rwave(round(k-(k-
lastR)*(rpercentbackward)):round(k+(k-
lastR)*(rpercentforward)))));
%
% p,t wave / r wave ratio
indexing
%
pstats(2,pz)=pstats(1,pz)/rstats(1,rz);
%
tstats(2,tz)=tstats(1,tz)/rstats(1,rz);
%
[max v,prt1]=max(rwave(round(k-(k-
lastR)*(rpercentbackward)):round(k+(k-
lastR)*(rpercentforward))));
%
[max v,prt2]=max(abs(diff(pwave(round(k-
(k-
lastR)*(ppercent+rpercentbackward)):round(
k-(k-lastR)*rpercentbackward))));
%
pstats(3,pz)=(prt1+round(k-(k-
lastR)*(rpercentbackward))-prt2-round(k-(k-
lastR)*(ppercent+rpercentbackward)))/400;
%
[max v,trt2]=max(abs(diff(twave(round(k-(k-
lastR)*(1-rpercentforward)):round(k-(k-
lastR)*(1-rpercentforward-tpercent))))));
%
tstats(3,tz)=(trt2+round(k-(k-
lastR)*(1-rpercentforward))-lastR)/400;
%
rz=rz+1;
%
pz=pz+1;
%
tz=tz+1;
end
RRV3=RRV2;
RRV2=RRV1;
if
max(signalfilt(k:(k+initialrsec*400))>((RR
V1+RRV2+RRV3)/3)*(1+RRVvariabilityfa
ctor) ||
max(signalfilt(k:(k+initialrsec*400))<((RR
V1+RRV2+RRV3)/3)*(1-
RRVvariabilityfactor)
else
RRV1=max(signalfilt(k:(k+initialrsec*400))
);
end
lastR=k;
end
end
if stats(i,28)==1
else
stem(t,trigger,'linewidth',1);
pause
% hold;
% plot(t,sigalfilt,'color','blue');
% plot(t,trigger,'linewidth',2,'color','red');
% hold;
% for l=0:3:27
% xlim([l (l+3)])
% pause
% end
stats(i,28)=input('Save epicardial trigger
data (1 yes 0 no)?');
if stats(i,28)==1
save('stats.mat','stats')
else
stats(i,28)=0;
stats(i,32:35)=0;
stats(i,48:50)=0;
end
end
screwd=trigger;
%-----
%-----

clear twave
clear pwave
clear rwave
twave=zeros(1,length(t1));

```

```

twave(:,:)=NaN;
pwave=zeros(1,length(t1));
pwave(:,:)=NaN;
rwave=zeros(1,length(t1));
rwave(:,:)=NaN;

%subq1
if signalcheck(i,2)==0
    stats(i,1)=0;
else
    stats(i,1)=1;

%subq1 R wave detection

clear signalfilt;
clear trigger;
clear ts;
clear ECG;

ts=t1;
signalfilt=subq1;
ECG=subq1o;

if stats(i,29)==1
    itwave=stats(i,51);
    ipwave=stats(i,52);
    ithresh=stats(i,53);
    iblank=stats(i,54);
    spblank=stats(i,55);
else
    plot(ts,signalfilt);
    xlim([0 2.5]);
    itwave=input('Is the T wave inverted? (0 no
1 yes 2 indiscernable:');
    ipwave=input('Is the P wave inverted? (0 no
1 yes 2 indiscernable:');
    if itwave==2
        stats(i,3)=0;
    else
        stats(i,3)=1;
    end
    if ipwave==2
        stats(i,2)=0;
    else
        stats(i,2)=1;
    end
    ithresh=input('First 3 R wave detection
threshold:');
    iblank=input('Initial blanking period in
sec:');
    spblank=input('Ignore period in sec:');
    stats(i,51)=itwave;
    stats(i,52)=ipwave;
    stats(i,53)=ithresh;
    stats(i,54)=iblank;
    stats(i,55)=spblank;

end
iblank=iblank*400;
RRT1=0;
RRT2=0;
RRT3=0;
RRV1=0;
RRV2=0;
RRV3=0;
blankpercent=.80;
initialR=1;
ppercent=.35;
tpercent=.50;
rpercentforward=.10;
rpercentbackward=.05;
initialrsec=.050;
ramppercnt=.80;
trigger=zeros(1,length(signalfilt));
lastR=0;
RRTvariabilityfactor=.3;
RRVvariabilityfactor=.3;
clear pstats
clear rstats
clear tstats
pz=1;
rz=1;
tz=1;

for k=(1+spblank*400):1:(length(signalfilt)-
800);
    if initialR==3
        if signalfilt(k)>=ithresh && (k-
RRT1)>iblank
            trigger(k)=1;
            RRT3=RRT2;
            RRT2=RRT1;
            RRT1=k;
            lastR=k;
            RRV3=RRV2;
            RRV2=RRV1;

RRV1=max(signalfilt(k:(k+initialrsec*400))
);
            initialR=initialR+1;
            RRT3=RRT2-RRT3;
            RRT2=RRT1-RRT2;
            RRT1=RRT2;
        end
        elseif initialR==2
            if signalfilt(k)>=ithresh && (k-
RRT1)>iblank
                trigger(k)=1;
                RRT3=RRT2;
                RRT2=RRT1;
                RRT1=k;
                RRV3=RRV2;
                RRV2=RRV1;
            end
        end
    end
end

```



```

RRV1=max(signalfilt(k:(k+initialrsec*400))
);
    initialR=initialR+1;
    end
elseif initialR==1
    if signalfilt(k)>=ithresh
        trigger(k)=1;
        RRT3=RRT2;
        RRT2=RRT1;
        RRT1=k;
        RRV3=RRV2;
        RRV2=RRV1;

RRV1=max(signalfilt(k:(k+initialrsec*400))
);
    initialR=initialR+1;
    end
    else
    if
signalfilt(k)>=((RRV1+RRV2+RRV3)/3*ramppercnt) && (k-
lastR)>(((RRT1+RRT2+RRT3)/3)*blankper
cent)
        trigger(k)=1;
        if (k-
lastR)>(((RRT1+RRT2+RRT3)/3)*(1+RRTv
ariabilityfactor) || (k-
lastR)<(((RRT1+RRT2+RRT3)/3)*(1-
RRTvariabilityfactor)
            pstats(1:4,pz)=0;
            tstats(1:4,tz)=0;
            rstats(1:4,rz)=0;
            rz=rz+1;
            pz=pz+1;
            tz=tz+1;
        else
            RRT3=RRT2;
            RRT2=RRT1;
            RRT1=k-lastR;
            %p,t,r wave capture and voltage
indexing
            pwave(round(k-(k-
lastR)*(ppercent+rpercentbackward)):round(
k-(k-
lastR)*rpercentbackward))=ECG(round(k-
(k-
lastR)*(ppercent+rpercentbackward)):round(
k-(k-lastR)*rpercentbackward));

pstats(1,pz)=abs(max(pwave(round(k-(k-
lastR)*(ppercent+rpercentbackward)):round(
k-(k-lastR)*rpercentbackward)))-
min(pwave(round(k-(k-
lastR)*(ppercent+rpercentbackward)):round(
k-(k-lastR)*rpercentbackward))));

```

```

twave(round(k-(k-lastR)*(1-
rpercentforward)):round(k-(k-lastR)*(1-
rpercentforward-tpercent)))=ECG(round(k-
(k-lastR)*(1-rpercentforward)):round(k-(k-
lastR)*(1-rpercentforward-tpercent)));

tstats(1,tz)=abs(max(twave(round(k-(k-
lastR)*(1-rpercentforward)):round(k-(k-
lastR)*(1-rpercentforward-tpercent)))-
min(twave(round(k-(k-lastR)*(1-
rpercentforward)):round(k-(k-lastR)*(1-
rpercentforward-tpercent))));
    rwave(round(k-(k-
lastR)*(rpercentbackward)):round(k+(k-
lastR)*(rpercentforward)))=ECG(round(k-
(k-lastR)*(rpercentbackward)):round(k+(k-
lastR)*(rpercentforward)));

rstats(1,rz)=abs(max(rwave(round(k-(k-
lastR)*(rpercentbackward)):round(k+(k-
lastR)*(rpercentforward)))-
min(rwave(round(k-(k-
lastR)*(rpercentbackward)):round(k+(k-
lastR)*(rpercentforward))));
    %p,t wave / r wave ratio indexing

pstats(2,pz)=pstats(1,pz)/rstats(1,rz);
tstats(2,tz)=tstats(1,tz)/rstats(1,rz);
    [maxv,prt1]=max(rwave(round(k-
(k-lastR)*(rpercentbackward)):round(k+(k-
lastR)*(rpercentforward))));
    if ipwave==1

    [maxv,prt2]=min((pwave(round(k-(k-
lastR)*(ppercent+rpercentbackward)):round(
k-(k-lastR)*rpercentbackward))));
    else

    [maxv,prt2]=max((pwave(round(k-(k-
lastR)*(ppercent+rpercentbackward)):round(
k-(k-lastR)*rpercentbackward))));
    end
    pstats(3,pz)=(prt1+round(k-(k-
lastR)*(rpercentbackward))-prt2-round(k-(k-
lastR)*(ppercent+rpercentbackward)))/400;
    if pz>1;
        if pstats(1:4,pz-1)==0

pstats(4,pz)=(prt2+round(k-(k-
lastR)*(ppercent+rpercentbackward)))/400;
        psave=pstats(4,pz);
        else
            pstats(4,pz)=(prt2+round(k-
(k-
lastR)*(ppercent+rpercentbackward)))/400-
psave;

```

```

        psave=(prt2+round(k-(k-
lastR)*(ppercent+rpercentbackward)))/400;
        end
        else
            pstats(4,pz)=(prt2+round(k-(k-
lastR)*(ppercent+rpercentbackward)))/400;
            psave=pstats(4,pz);
            end
            if itwave==1

[ maxv,trt2]=min((twave(round(k-(k-
lastR)*(1-rpercentforward)):round(k-(k-
lastR)*(1-rpercentforward-tpercent)))));
            else

[ maxv,trt2]=max((twave(round(k-(k-
lastR)*(1-rpercentforward)):round(k-(k-
lastR)*(1-rpercentforward-tpercent)))));
            end
            tstats(3,tz)=(trt2+round(k-(k-
lastR)*(1-rpercentforward))-lastR)/400;
            if tz>1;
                if tstats(1:4,tz-1)==0

tstats(4,tz)=(trt2+round(k-(k-lastR)*(1-
rpercentforward)))/400;
                tsave=tstats(4,tz);
                else
                    tstats(4,tz)=(trt2+round(k-(k-
lastR)*(1-rpercentforward)))/400-tsave;
                    tsave=(trt2+round(k-(k-
lastR)*(1-rpercentforward)))/400;
                    end
                    else
                        tstats(4,tz)=(trt2+round(k-(k-
lastR)*(1-rpercentforward)))/400;
                        tsave=tstats(4,tz);
                        end
                        %qrslength routine
                        rmstart=round(k-(k-
lastR)*(rpercentbackward));
                        rmend=round(k+(k-
lastR)*(rpercentforward));
                        qrsseg=ECG(1,rmstart:rmend);
                        qrsseg=qrsseg+(0-min(qrsseg));
                        qrsseg=qrsseg/max(qrsseg);
                        [qrsmax
ignoreindex]=max(qrsseg);
                        averaging=6;
                        qrswidthstart=0;
                        qrsdiffpercent=.8;
                        for jkl=averaging:-1:1
                            clear averagingm
                            iii=0;
                            if qrswidthstart==0
                                for ii=(ignoreindex):-1:1;

```

```

                                if qrswidthstart==0
                                    if iii<jkl
                                        averagingm(iii+1)=qrsseg(ii);
                                        iii=iii+1;
                                    else
                                        for
                                            ll=1:1:(length(averagingm)-1)

averagingm(ll)=averagingm(ll+1);
                                            end
                                            averagingm(jkl)=qrsseg(ii);
                                            if (qrsmax-
mean(averagingm))>=qrsdiffpercent*(qrsma
x-min(qrsseg(1:ignoreindex)))
                                                qrswidthstart=ii;
                                            end
                                            end
                                            end
                                            end
                                            averaging=averaging-1;
                                            end

                                averaging=6;
                                qrswidthend=0;
                                for jkl=averaging:-1:1
                                    clear averagingm
                                    iii=0;
                                    if qrswidthend==0
                                        for
                                            ii=(ignoreindex):1:length(qrsseg);
                                                if qrswidthend==0
                                                    if iii<jkl
                                                        averagingm(iii+1)=qrsseg(ii);
                                                        iii=iii+1;
                                                    else
                                                        for
                                                            ll=1:1:(length(averagingm)-1)

averagingm(ll)=averagingm(ll+1);
                                                            end
                                                            averagingm(jkl)=qrsseg(ii);
                                                            if (qrsmax-
mean(averagingm))>=qrsdiffpercent*(qrsma
x-min(qrsseg(ignoreindex:end)))
                                                                qrswidthend=ii;
                                                            end
                                                            end
                                                            end
                                                            end
                                                            averaging=averaging-1;
                                                            end

                                %
                                qrsindices=(1:1:length(qrsseg));

```

```

        %
plot(qrsindices,qrsseg)
        % hold;
        %
plot(qrsindices(qrswidthstart),qrsseg(qrswidthstart),'--rs','LineWidth',2,...
        %
'MarkerEdgeColor','k',...
        %
'MarkerFaceColor','g',...
        % 'MarkerSize',10)
        %
plot(qrsindices(qrswidthend),qrsseg(qrswidthend),'--rs','LineWidth',2,...
        %
'MarkerEdgeColor','k',...
        %
'MarkerFaceColor','g',...
        % 'MarkerSize',10)
        % hold;
        % pause
        rzstats(2,rz)=(qrswidthend-
qrswidthstart)/400;
        rzstats(3,rz)=RRT1/400;
        rzstats(4,rz)=mean(ECG(lastR:k));
        rz=rz+1;
        pz=pz+1;
        tz=tz+1;
        end
        RRV3=RRV2;
        RRV2=RRV1;
        if
max(signalfilt(k:(k+initialrsec*400)))>((RRV1+RRV2+RRV3)/3)*(1+RRVvariabilityfactor) ||
max(signalfilt(k:(k+initialrsec*400)))<((RRV1+RRV2+RRV3)/3)*(1-
RRVvariabilityfactor)
                else
RRV1=max(signalfilt(k:(k+initialrsec*400))
);
                end
                lastR=k;
        end
        end

end

if stats(i,29)==1
else
% stem(ts,trigger,'linewidth',1);
% pause
% hold;
% plot(ts,signalfilt,'color','blue');
% plot(ts,trigger,'linewidth',2,'color','red');
        % plot(t,-
        1*screwd,'linewidth',2,'color','magenta');
        % hold;
        % for l=0:3:27
        % xlim([l (l+3)])
        % pause
        % end
        end
        %
        plot(ts,ECG(:),'color','black','linewidth',1);
        % hold;
        % plot(ts,pwave,'color','red','linewidth',2)
        % plot(ts,twave,'color','blue','linewidth',2)
        % plot(ts,rwave,'color','green','linewidth',2)
        % for l=0:3:27
        % xlim([l (l+3)])
        % pause
        % end
        % hold;
        %
        subq1d=trigger;
        save(sprintf('%s%s%s%i%s',cow,'_',PQRST
        _',i-
        1,_1.mat'),'pstats','rstats','tstats','pwave','twave',
        'rwave');
        disp('PQRST Matrices saved')

        %Step-through Trigger Analysis
        truepositive=0;
        falsepositive=0;
        falsenegative=0;

        % Truepositive and Falsenegative Search
        for j=1:(length(subq1d)-resolution)
            if screwd(1,j)==1
                bug1=(j-resolution);
                if (j-resolution)<=0
                    bug1=1;
                end
            end
            if
sum(subq1d(1,bug1:1:j+resolution))==1
                truepositive=truepositive+1;
            elseif
sum(subq1d(1,bug1:1:j+resolution))==0
                falsenegative=falsenegative+1;
            else
                disp('Triggering error noted, check
                resolution vs blanking period')
            end
            end
        end

        % Falsepositive Search
        for j=1:(length(subq1d)-resolution)
            if subq1d(1,j)==1
                bug2=(j-resolution);

```

```

    if (j-resolution)<=0
        bug2=1;
    end
    if
        sum(screw(1,bug2:1:j+resolution))==0
            falsepositive=falsepositive+1;
        end
    end
end

disp('*****Subq1 Triggering
Statistics*****')
disp(sprintf('%s%i','Total Beats:
',sum(screw(1,1:1:length(subq1)-
resolution))))
disp(sprintf('%s%i','Truepositive:
',truepositive))
disp(sprintf('%s%i','Falsepositive:
',falsepositive))
disp(sprintf('%s%i','Falsenegative:
',falsenegative)) % #ok<*DSPS>
disp(sprintf('%s%3.2f','Positive Predictive
Value:
',truepositive/(truepositive+falsepositive)))
disp(sprintf('%s%3.2f','Sensitivity:
',truepositive/(truepositive+falsenegative)))

stats(i,10:15)=[sum(screw(1,1:1:length(sub
q1d)-resolution))
truepositive/(truepositive+falsepositive)
truepositive/(truepositive+falsenegative)
truepositive falsepositive falsenegative];
if stats(i,29)==0;
stats(i,29)=input('Save subq1 trigger data (1
yes 0 no)?');
if stats(i,29)==1
    save('stats.mat','stats')
else
    stats(i,29)=0;
    stats(i,36:39)=0;
    stats(i,51:55)=0;
    stats(i,66:67)=0;
end
end
end

%-----
-----

clear twave
clear pwave
clear rwave

twave=zeros(1,length(t2));
twave(:,:)=NaN;
pwave=zeros(1,length(t2));
pwave(:,:)=NaN;
rwave=zeros(1,length(t2));
rwave(:,:)=NaN;

%subq2
if signalcheck(i,3)==0
    stats(i,4)=0;
else
    stats(i,4)=1;
%subq2 R wave detection

clear signalfilt;
clear trigger;
clear ts;
clear ECG;

ts=t2;
signalfilt=subq2;
ECG=subq2o;

if stats(i,30)==1
    itwave=stats(i,56);
    ipwave=stats(i,57);
    ithresh=stats(i,58);
    iblack=stats(i,59);
    spblank=stats(i,60);
else
    plot(ts,signalfilt);
xlim([0 2.5]);
itwave=input('Is the T wave inverted? (0 no
1 yes 2 indiscernable:');
ipwave=input('Is the P wave inverted? (0 no
1 yes 2 indiscernable:');
if itwave==2
    stats(i,6)=0;
else
    stats(i,6)=1;
end
if ipwave==2
    stats(i,5)=0;
else
    stats(i,5)=1;
end
ithresh=input('First 3 R wave detection
threshold:');
iblack=input('Initial blanking period in
sec:');
spblank=input('Ignore period in sec:');
stats(i,56)=itwave;
stats(i,57)=ipwave;
stats(i,58)=ithresh;
stats(i,59)=iblack;
stats(i,60)=spblank;

```

```

end
iblack=iblack*400;
RRT1=0;
RRT2=0;
RRT3=0;
RRV1=0;
RRV2=0;
RRV3=0;
blankpercent=.80;
initialR=1;
pppercent=.35;
tpercent=.50;
rpercentforward=.10;
rpercentbackward=.05;
initialrsec=.050;
rampppercent=.80;
trigger=zeros(1,length(signalfilt));
lastR=0;
RRTvariabilityfactor=.3;
RRVvariabilityfactor=.3;
clear pstats
clear rstats
clear tstats
pz=1;
rz=1;
tz=1;

for k=(1+sblank*400):1:(length(signalfilt)-
800);
    if initialR==3
        if signalfilt(k)>=ithresh && (k-
RRT1)>iblack
            trigger(k)=1;
            RRT3=RRT2;
            RRT2=RRT1;
            RRT1=k;
            lastR=k;
            RRV3=RRV2;
            RRV2=RRV1;

RRV1=max(signalfilt(k:(k+initialrsec*400))
);
            initialR=initialR+1;
            RRT3=RRT2-RRT3;
            RRT2=RRT1-RRT2;
            RRT1=RRT2;
        end
    elseif initialR==2
        if signalfilt(k)>=ithresh && (k-
RRT1)>iblack
            trigger(k)=1;
            RRT3=RRT2;
            RRT2=RRT1;
            RRT1=k;
            RRV3=RRV2;
            RRV2=RRV1;

RRV1=max(signalfilt(k:(k+initialrsec*400))
);
            initialR=initialR+1;
        end
    elseif initialR==1
        if signalfilt(k)>=ithresh
            trigger(k)=1;
            RRT3=RRT2;
            RRT2=RRT1;
            RRT1=k;
            RRV3=RRV2;
            RRV2=RRV1;

RRV1=max(signalfilt(k:(k+initialrsec*400))
);
            initialR=initialR+1;
        end
    else
        if
signalfilt(k)>=((RRV1+RRV2+RRV3)/3*ra
mpppercent) && (k-
lastR)>(((RRT1+RRT2+RRT3)/3)*blankper
cent)
            trigger(k)=1;
            if (k-
lastR)>((RRT1+RRT2+RRT3)/3)*(1+RRTv
ariabilityfactor) || (k-
lastR)<((RRT1+RRT2+RRT3)/3)*(1-
RRTvariabilityfactor)
                pstats(1:4,pz)=0;
                tstats(1:4,tz)=0;
                rstats(1:4,rz)=0;
                rz=rz+1;
                pz=pz+1;
                tz=tz+1;
            else
                RRT3=RRT2;
                RRT2=RRT1;
                RRT1=k-lastR;
                %p,t,r wave capture and voltage
indexing
                pwave(round(k-(k-
lastR)*(pppercent+rpercentbackward)):round(
k-(k-
lastR)*rpercentbackward))=ECG(round(k-
(k-
lastR)*(pppercent+rpercentbackward)):round(
k-(k-lastR)*rpercentbackward));

                pstats(1,pz)=abs(max(pwave(round(k-(k-
lastR)*(pppercent+rpercentbackward)):round(
k-(k-lastR)*rpercentbackward)))-
min(pwave(round(k-(k-
lastR)*(pppercent+rpercentbackward)):round(
k-(k-lastR)*rpercentbackward))));
            end
        end
    end
end

```

```

    twave(round(k-(k-lastR)*(1-
rpercentforward)):round(k-(k-lastR)*(1-
rpercentforward-tpercent)))=ECG(round(k-
(k-lastR)*(1-rpercentforward)):round(k-(k-
lastR)*(1-rpercentforward-tpercent)));

tstats(1,tz)=abs(max(twave(round(k-(k-
lastR)*(1-rpercentforward)):round(k-(k-
lastR)*(1-rpercentforward-tpercent))))-
min(twave(round(k-(k-lastR)*(1-
rpercentforward)):round(k-(k-lastR)*(1-
rpercentforward-tpercent)))));
    rwave(round(k-(k-
lastR)*(rpercentbackward)):round(k+(k-
lastR)*(rpercentforward)))=ECG(round(k-
(k-lastR)*(rpercentbackward)):round(k+(k-
lastR)*(rpercentforward)));

rstats(1,rz)=abs(max(rwave(round(k-(k-
lastR)*(rpercentbackward)):round(k+(k-
lastR)*(rpercentforward))))-
min(rwave(round(k-(k-
lastR)*(rpercentbackward)):round(k+(k-
lastR)*(rpercentforward)))));
    %p,t wave / r wave ratio indexing

pstats(2,pz)=pstats(1,pz)/rstats(1,rz);
    tstats(2,tz)=tstats(1,tz)/rstats(1,rz);
    [maxv,prt1]=max(rwave(round(k-
(k-lastR)*(rpercentbackward)):round(k+(k-
lastR)*(rpercentforward)))));
    if ipwave==1

[maxv,prt2]=min((pwave(round(k-(k-
lastR)*(ppercent+rpercentbackward)):round(
k-(k-lastR)*rpercentbackward))););
    else

[maxv,prt2]=max((pwave(round(k-(k-
lastR)*(ppercent+rpercentbackward)):round(
k-(k-lastR)*rpercentbackward))););
    end
    pstats(3,pz)=(prt1+round(k-(k-
lastR)*(rpercentbackward))-prt2-round(k-(k-
lastR)*(ppercent+rpercentbackward)))/400;
    if pz>1;
        if pstats(1:4,pz-1)==0

pstats(4,pz)=(prt2+round(k-(k-
lastR)*(ppercent+rpercentbackward)))/400;
    psave=pstats(4,pz);
    else
        pstats(4,pz)=(prt2+round(k-
(k-
lastR)*(ppercent+rpercentbackward)))/400-
psave;

```

```

    psave=(prt2+round(k-(k-
lastR)*(ppercent+rpercentbackward)))/400;
    end
    else
        pstats(4,pz)=(prt2+round(k-(k-
lastR)*(ppercent+rpercentbackward)))/400;
        psave=pstats(4,pz);
    end
    if itwave==1

[maxv,trt2]=min((twave(round(k-(k-
lastR)*(1-rpercentforward)):round(k-(k-
lastR)*(1-rpercentforward-tpercent)))));
    else

[maxv,trt2]=max((twave(round(k-(k-
lastR)*(1-rpercentforward)):round(k-(k-
lastR)*(1-rpercentforward-tpercent)))));
    end
    tstats(3,tz)=(trt2+round(k-(k-
lastR)*(1-rpercentforward))-lastR)/400;
    if tz>1;
        if tstats(1:4,tz-1)==0

tstats(4,tz)=(trt2+round(k-(k-lastR)*(1-
rpercentforward)))/400;
    tsave=tstats(4,tz);
    else
        tstats(4,tz)=(trt2+round(k-(k-
lastR)*(1-rpercentforward)))/400-tsave;
        tsave=(trt2+round(k-(k-
lastR)*(1-rpercentforward)))/400;
    end
    else
        tstats(4,tz)=(trt2+round(k-(k-
lastR)*(1-rpercentforward)))/400;
        tsave=tstats(4,tz);
    end
    %qrslength routine
    rmstart=round(k-(k-
lastR)*(rpercentbackward));
    rmend=round(k+(k-
lastR)*(rpercentforward));
    qrsseg=ECG(1,rmstart:rmend);
    qrsseg=qrsseg+(0-min(qrsseg));
    qrsseg=qrsseg/max(qrsseg);
    [qrsmax
ignoreindex]=max(qrsseg);
    averaging=6;
    qrswidthstart=0;
    qrstdiffpercent=.8;
    for jkl=averaging:-1:1
        clear averagingm
        iii=0;
        if qrswidthstart==0
            for ii=(ignoreindex):-1:1;

```

```

        if qrswidthstart==0
        if iii<jkl
            averagingm(iii+1)=qrsseg(ii);
            iii=iii+1;
        else
            for
ll=1:1:(length(averagingm)-1)
averagingm(ll)=averagingm(ll+1);
            end
            averagingm(jkl)=qrsseg(ii);
            if (qrsmax-
mean(averagingm))>=qrsdiffpercent*(qrsma
x-min(qrsseg(1:ignoreindex)))
                qrswidthstart=ii;
            end
            end
        end
    end
    averaging=averaging-1;
end

    averaging=6;
    qrswidthend=0;
    for jkl=averaging:-1:1
    clear averagingm
    iii=0;
    if qrswidthend==0
    for
ii=(ignoreindex):1:length(qrsseg);
        if qrswidthend==0
        if iii<jkl
            averagingm(iii+1)=qrsseg(ii);
            iii=iii+1;
        else
            for
ll=1:1:(length(averagingm)-1)
averagingm(ll)=averagingm(ll+1);
            end
            averagingm(jkl)=qrsseg(ii);
            if (qrsmax-
mean(averagingm))>=qrsdiffpercent*(qrsma
x-min(qrsseg(ignoreindex:end)))
                qrswidthend=ii;
            end
            end
        end
    end
    averaging=averaging-1;
end

    %
qrsindices=(1:1:length(qrsseg));

```

```

        %
        plot(qrsindices,qrsseg)
        % hold;
        %
        plot(qrsindices(qrswidthstart),qrsseg(qrswidt
hstart),'--rs','LineWidth',2,...
        %
'MarkerEdgeColor','k',...
        %
'MarkerFaceColor','g',...
        % 'MarkerSize',10)
        %
        plot(qrsindices(qrswidthend),qrsseg(qrswidt
hend),'--rs','LineWidth',2,...
        %
'MarkerEdgeColor','k',...
        %
'MarkerFaceColor','g',...
        % 'MarkerSize',10)
        % hold;
        % pause
        rstats(2,rz)=(qrswidthend-
qrswidthstart)/400;
        rstats(3,rz)=RRT1/400;
        rstats(4,rz)=mean(ECG(lastR:k));
        rz=rz+1;
        pz=pz+1;
        tz=tz+1;
        end
        RRV3=RRV2;
        RRV2=RRV1;
        if
max(signalfilt(k:(k+initialrsec*400)))>((RR
V1+RRV2+RRV3)/3)*(1+RRVvariabilityfa
ctor) ||
max(signalfilt(k:(k+initialrsec*400)))<((RR
V1+RRV2+RRV3)/3)*(1-
RRVvariabilityfactor)
            else
RRV1=max(signalfilt(k:(k+initialrsec*400))
);
            end
            lastR=k;
        end
    end
end

if stats(i,30)==1
else
% stem(ts,trigger,'linewidth',1);
% pause
% hold;
% plot(ts,signalfilt,'color','blue');
% plot(ts,trigger,'linewidth',2,'color','red');

```

```

% plot(t,-
1*screwd,'linewidth',2,'color','magenta');
% hold;
% for l=0:3:27
%   xlim([l (l+3)])
%   pause
% end
end

%
plot(ts,ECG(:),'color','black','linewidth',1);
% hold;
% plot(ts,pwave,'color','red','linewidth',2)
% plot(ts,twave,'color','blue','linewidth',2)
% plot(ts,rwave,'color','green','linewidth',2)
% for l=0:3:27
%   xlim([l (l+3)])
%   pause
% end
% hold;
%
subq2d=trigger;
save(sprintf('%s%s%i%s',cow,'_',PQRST
_,'i-
1','_2.mat'),'pstats','rstats','tstats','pwave','twa
ve','rwave');
disp('PQRST Matrices saved')

%Step-through Trigger Analysis
truepositive=0;
falsepositive=0;
falsenegative=0;

% Truepositive and Falsenegative Search
for j=1:1:(length(subq2d)-resolution)
    if screwd(1,j)==1
        bug1=(j-resolution);
        if (j-resolution)<=0
            bug1=1;
        end
        if
sum(subq2d(1,bug1:1:j+resolution))==1
            truepositive=truepositive+1;
        elseif
sum(subq2d(1,bug1:1:j+resolution))==0
            falsenegative=falsenegative+1;
        else
            disp('Triggering error noted, check
resolution vs blanking period')
        end
    end
end

% Falsepositive Search
for j=1:1:(length(subq2d)-resolution)
    if subq2d(1,j)==1
        bug2=(j-resolution);
        if (j-resolution)<=0
            bug2=1;
        end
        if
sum(screwd(1,bug2:1:j+resolution))==0
            falsepositive=falsepositive+1;
        end
    end
end

disp('*****Subq2 Triggering
Statistics*****')
disp(sprintf('%s%i','Total Beats:
',sum(screwd(1,1:1:length(subq2)-
resolution))))
disp(sprintf('%s%i','Truepositive:
',truepositive))
disp(sprintf('%s%i','Falsepositive:
',falsepositive))
disp(sprintf('%s%i','Falsenegative:
',falsenegative)) % #ok< *DSPS>
disp(sprintf('%s%3.2f','Positive Predictive
Value:
',truepositive/(truepositive+falsepositive)))
disp(sprintf('%s%3.2f','Sensitivity:
',truepositive/(truepositive+falsenegative)))

stats(i,16:21)=[sum(screwd(1,1:1:length(sub
q2d)-resolution))
truepositive/(truepositive+falsepositive)
truepositive/(truepositive+falsenegative)
truepositive falsepositive falsenegative];

if stats(i,30)==0;
stats(i,30)=input('Save subq2 trigger data (1
yes 0 no)?');
if stats(i,30)==1
    save('stats.mat','stats')
else
    stats(i,30)=0;
    stats(i,40:43)=0;
    stats(i,56:60)=0;
    stats(i,68:69)=0;
end
end
end

%-----
-----

clear twave
clear pwave
clear rwave
twave=zeros(1,length(t3));

```



```

twave(:,:)=NaN;
pwave=zeros(1,length(t3));
pwave(:,:)=NaN;
rwave=zeros(1,length(t3));
rwave(:,:)=NaN;

%subq3
if signalcheck(i,4)==0
    stats(i,7)=0;
else
    stats(i,7)=1;

clear signalfilt;
clear trigger;
clear ts;
clear ECG;

ts=t3;
signalfilt=subq3;
ECG=subq3o;

if stats(i,31)==1
    itwave=stats(i,61);
    ipwave=stats(i,62);
    ithresh=stats(i,63);
    iblank=stats(i,64);
    spblank=stats(i,65);
else
    plot(ts,signalfilt);
    xlim([0 2.5]);
    itwave=input('Is the T wave inverted? (0 no
1 yes 2 indiscernable:');
    ipwave=input('Is the P wave inverted? (0 no
1 yes 2 indiscernable:');
    if itwave==2
        stats(i,9)=0;
    else
        stats(i,9)=1;
    end
    if ipwave==2
        stats(i,8)=0;
    else
        stats(i,8)=1;
    end
    ithresh=input('First 3 R wave detection
threshold:');
    iblank=input('Initial blanking period in
sec:');
    spblank=input('Ignore period in sec:');
    stats(i,61)=itwave;
    stats(i,62)=ipwave;
    stats(i,63)=ithresh;
    stats(i,64)=iblank;
    stats(i,65)=spblank;
end
iblank=iblank*400;

RRT1=0;
RRT2=0;
RRT3=0;
RRV1=0;
RRV2=0;
RRV3=0;
blankpercent=.80;
initialR=1;
ppercent=.35;
tpercent=.50;
rpercentforward=.10;
rpercentbackward=.05;
initialrsec=.050;
ramppercnt=.80;
trigger=zeros(1,length(signalfilt));
lastR=0;
RRTvariabilityfactor=.3;
RRVvariabilityfactor=.3;
clear pstats
clear rstats
clear tstats
pz=1;
rz=1;
tz=1;

for k=(1+spblank*400):1:(length(signalfilt)-
800);
    if initialR==3
        if signalfilt(k)>=ithresh && (k-
RRT1)>iblank
            trigger(k)=1;
            RRT3=RRT2;
            RRT2=RRT1;
            RRT1=k;
            lastR=k;
            RRV3=RRV2;
            RRV2=RRV1;

RRV1=max(signalfilt(k:(k+initialrsec*400))
);
            initialR=initialR+1;
            RRT3=RRT2-RRT3;
            RRT2=RRT1-RRT2;
            RRT1=RRT2;
        end
        elseif initialR==2
            if signalfilt(k)>=ithresh && (k-
RRT1)>iblank
                trigger(k)=1;
                RRT3=RRT2;
                RRT2=RRT1;
                RRT1=k;
                RRV3=RRV2;
                RRV2=RRV1;
            end
        end
    end
end

```

```

RRV1=max(signalfilt(k:(k+initialrsec*400))
);
    initialR=initialR+1;
    end
elseif initialR==1
    if signalfilt(k)>=ithresh
        trigger(k)=1;
        RRT3=RRT2;
        RRT2=RRT1;
        RRT1=k;
        RRV3=RRV2;
        RRV2=RRV1;

RRV1=max(signalfilt(k:(k+initialrsec*400))
);
    initialR=initialR+1;
    end
    else
        if
signalfilt(k)>=((RRV1+RRV2+RRV3)/3*ra
mppercent) && (k-
lastR)>(((RRT1+RRT2+RRT3)/3)*blankper
cent)
            trigger(k)=1;

            if (k-
lastR)>(((RRT1+RRT2+RRT3)/3)*(1+RRTv
ariabilityfactor) || (k-
lastR)<(((RRT1+RRT2+RRT3)/3)*(1-
RRTvariabilityfactor)
                pstats(1:4,pz)=0;
                tstats(1:4,tz)=0;
                rstats(1:4,rz)=0;
                rz=rz+1;
                pz=pz+1;
                tz=tz+1;
            else
                RRT3=RRT2;
                RRT2=RRT1;
                RRT1=k-lastR;
                %p,t,r wave capture and voltage
indexing
                pwave(round(k-(k-
lastR)*(ppercent+rpercentbackward)):round(
k-(k-
lastR)*rpercentbackward))=ECG(round(k-
(k-
lastR)*(ppercent+rpercentbackward)):round(
k-(k-lastR)*rpercentbackward));

                pstats(1,pz)=abs(max(pwave(round(k-(k-
lastR)*(ppercent+rpercentbackward)):round(
k-(k-lastR)*rpercentbackward)))-
min(pwave(round(k-(k-

```

```

lastR)*(ppercent+rpercentbackward)):round(
k-(k-lastR)*rpercentbackward)))));
                twave(round(k-(k-lastR)*(1-
rpercentforward)):round(k-(k-lastR)*(1-
rpercentforward-tpercent)))=ECG(round(k-
(k-lastR)*(1-rpercentforward)):round(k-(k-
lastR)*(1-rpercentforward-tpercent)));

                tstats(1,tz)=abs(max(twave(round(k-(k-
lastR)*(1-rpercentforward)):round(k-(k-
lastR)*(1-rpercentforward-tpercent)))-
min(twave(round(k-(k-lastR)*(1-
rpercentforward)):round(k-(k-lastR)*(1-
rpercentforward-tpercent)))));
                rwave(round(k-(k-
lastR)*(rpercentbackward)):round(k+(k-
lastR)*(rpercentforward)))=ECG(round(k-
(k-lastR)*(rpercentbackward)):round(k+(k-
lastR)*(rpercentforward)));

                rstats(1,rz)=abs(max(rwave(round(k-(k-
lastR)*(rpercentbackward)):round(k+(k-
lastR)*(rpercentforward)))-
min(rwave(round(k-(k-
lastR)*(rpercentbackward)):round(k+(k-
lastR)*(rpercentforward)))));
                %p,t wave / r wave ratio indexing

                pstats(2,pz)=pstats(1,pz)/rstats(1,rz);
                tstats(2,tz)=tstats(1,tz)/rstats(1,rz);
                [maxv,prt1]=max(rwave(round(k-(k-
lastR)*(rpercentbackward)):round(k+(k-
lastR)*(rpercentforward))));
                if ipwave==1

                [maxv,prt2]=min((pwave(round(k-(k-
lastR)*(ppercent+rpercentbackward)):round(
k-(k-lastR)*rpercentbackward))));
                else

                [maxv,prt2]=max((pwave(round(k-(k-
lastR)*(ppercent+rpercentbackward)):round(
k-(k-lastR)*rpercentbackward))));
                end
                pstats(3,pz)=(prt1+round(k-(k-
lastR)*(rpercentbackward))-prt2-round(k-(k-
lastR)*(ppercent+rpercentbackward)))/400;
                if pz>1;
                    if pstats(1:4,pz-1)==0

                pstats(4,pz)=(prt2+round(k-(k-
lastR)*(ppercent+rpercentbackward)))/400;
                psave=pstats(4,pz);
                else
                    pstats(4,pz)=(prt2+round(k-
(k-

```

```

lastR)*(ppercent+rpercentbackward))/400-
psave;
    psave=(prt2+round(k-(k-
lastR)*(ppercent+rpercentbackward))/400;
    end
    else
    pstats(4,pz)=(prt2+round(k-(k-
lastR)*(ppercent+rpercentbackward))/400;
    psave=pstats(4,pz);
    end
    if itwave==1

[maxv,trt2]=min((twave(round(k-(k-
lastR)*(1-rpercentforward)):round(k-(k-
lastR)*(1-rpercentforward-tpercnt)))));
    else

[maxv,trt2]=max((twave(round(k-(k-
lastR)*(1-rpercentforward)):round(k-(k-
lastR)*(1-rpercentforward-tpercnt)))));
    end
    tstats(3,tz)=(trt2+round(k-(k-
lastR)*(1-rpercentforward))-lastR)/400;
    if tz>1;
        if tstats(1:4,tz-1)==0

tstats(4,tz)=(trt2+round(k-(k-lastR)*(1-
rpercentforward)))/400;
        tsave=tstats(4,tz);
        else
        tstats(4,tz)=(trt2+round(k-(k-
lastR)*(1-rpercentforward)))/400-tsave;
        tsave=(trt2+round(k-(k-
lastR)*(1-rpercentforward)))/400;
        end
        else
        tstats(4,tz)=(trt2+round(k-(k-
lastR)*(1-rpercentforward)))/400;
        tsave=tstats(4,tz);
        end
        %qrs length routine
        rmstart=round(k-(k-
lastR)*(rpercentbackward));
        rmend=round(k+(k-
lastR)*(rpercentforward));
        qrsseg=ECG(1,rmstart:rmend);
        qrsseg=qrsseg+(0-min(qrsseg));
        qrsseg=qrsseg/max(qrsseg);
        [qrsmax
ignoreindex]=max(qrsseg);
        averaging=6;
        qrswidthstart=0;
        qrsdiffpercent=.8;
        for jkl=averaging:-1:1
            clear averagingm
            iii=0;

```

```

        if qrswidthstart==0
            for ii=(ignoreindex):-1:1;
                if qrswidthstart==0
                    if iii<jkl
                        averagingm(iii+1)=qrsseg(ii);
                        iii=iii+1;
                    else
                        for
                            ll=1:1:(length(averagingm)-1)
                                averagingm(ll)=averagingm(ll+1);
                                end
                                averagingm(jkl)=qrsseg(ii);
                                if (qrsmax-
                                mean(averagingm))>=qrsdiffpercent*(qrsma
                                x-min(qrsseg(1:ignoreindex)))
                                    qrswidthstart=ii;
                                end
                                end
                                end
                                end
                                averaging=averaging-1;
                                end

                                averaging=6;
                                qrswidthend=0;
                                for jkl=averaging:-1:1
                                    clear averagingm
                                    iii=0;
                                    if qrswidthend==0
                                        for
                                            ii=(ignoreindex):1:length(qrsseg);
                                                if qrswidthend==0
                                                    if iii<jkl
                                                        averagingm(iii+1)=qrsseg(ii);
                                                        iii=iii+1;
                                                    else
                                                        for
                                                            ll=1:1:(length(averagingm)-1)
                                                                averagingm(ll)=averagingm(ll+1);
                                                                end
                                                                averagingm(jkl)=qrsseg(ii);
                                                                if (qrsmax-
                                                                mean(averagingm))>=qrsdiffpercent*(qrsma
                                                                x-min(qrsseg(ignoreindex:end)))
                                                                    qrswidthend=ii;
                                                                end
                                                                end
                                                                end
                                                                end
                                                                averaging=averaging-1;
                                                                end

```

```

%
qrsindices=(1:1:length(qrsseg));
%
plot(qrsindices,qrsseg)
% hold;
%
plot(qrsindices(qrswidthstart),qrsseg(qrswidthstart),'-rs','LineWidth',2,...
%
'MarkerEdgeColor','k',...
%
'MarkerFaceColor','g',...
'MarkerSize',10)
%
plot(qrsindices(qrswidthend),qrsseg(qrswidthend),'-rs','LineWidth',2,...
%
'MarkerEdgeColor','k',...
%
'MarkerFaceColor','g',...
'MarkerSize',10)
% hold;
% pause
rstats(2,rz)=(qrswidthend-
qrswidthstart)/400;
rstats(3,rz)=RRT1/400;
rstats(4,rz)=mean(ECG(lastR:k));
rz=rz+1;
pz=pz+1;
tz=tz+1;
end
RRV3=RRV2;
RRV2=RRV1;
if
max(signalfilt(k:(k+initialrsec*400)))>((RRV1+RRV2+RRV3)/3)*(1+RRVvariabilityfactor) ||
max(signalfilt(k:(k+initialrsec*400)))<((RRV1+RRV2+RRV3)/3)*(1-RRVvariabilityfactor)
else
RRV1=max(signalfilt(k:(k+initialrsec*400)))
);
end
lastR=k;
end
end
end

if stats(i,31)==1
else
% stem(ts,trigger,'linewidth',1);
% pause
% hold;
% plot(ts,signalfilt,'color','blue');
% plot(ts,trigger,'linewidth',2,'color','red');
% plot(t,-
1*screwd,'linewidth',2,'color','magenta');
% hold;
% for l=0:3:27
% xlim([l (l+3)])
% pause
% end
end

%
plot(ts,ECG(:),'color','black','linewidth',1);
% hold;
% plot(ts,pwave,'color','red','linewidth',2)
% plot(ts,twave,'color','blue','linewidth',2)
% plot(ts,rwave,'color','green','linewidth',2)
% for l=0:3:27
% xlim([l (l+3)])
% pause
% end
% hold;

subq3d=trigger;
save(sprintf('%s%s%s%i%s',cow,'_',PQRST
_',i-
1','_3.mat'),'pstats','rstats','tstats','pwave','twave','rwave');
disp('PQRST Matrices saved')

%Step-through Trigger Analysis
truepositive=0;
falsepositive=0;
falsenegative=0;

% Truepositive and Falsenegative Search
for j=1:1:(length(subq3d)-resolution)
if screwd(1,j)==1
bug1=(j-resolution);
if (j-resolution)<=0
bug1=1;
end
if
sum(subq3d(1,bug1:1:j+resolution))==1
truepositive=truepositive+1;
elseif
sum(subq3d(1,bug1:1:j+resolution))==0
falsenegative=falsenegative+1;
else
disp('Triggering error noted, check
resolution vs blanking period')
end
end
end

%Falsepositive Search
for j=1:1:(length(subq3d)-resolution)

```

```

if subq3d(1,j)==1
    bug2=(j-resolution);
if (j-resolution)<=0
    bug2=1;
end
if
sum(screw(1,bug2:1:j+resolution))==0
    falsepositive=falsepositive+1;
end
end
end
end

disp('*****Subq3 Triggering
Statistics*****')
disp(sprintf('%s%i','Total Beats:
',sum(screw(1,1:1:length(subq3)-
resolution))))
disp(sprintf('%s%i','Truepositive:
',truepositive))
disp(sprintf('%s%i','Falsepositive:
',falsepositive))
disp(sprintf('%s%i','Falsenegative:
',falsenegative)) %#ok< *DSPS>
disp(sprintf('%s%3.2f','Positive Predictive
Value:
',truepositive/(truepositive+falsepositive)))
disp(sprintf('%s%3.2f','Sensitivity:
',truepositive/(truepositive+falsenegative)))

stats(i,22:27)=[sum(screw(1,1:1:length(sub
q3d)-resolution))
truepositive/(truepositive+falsepositive)
truepositive/(truepositive+falsenegative)
truepositive falsepositive falsenegative];

if stats(i,31)==0;
stats(i,31)=input('Save subq3 trigger data (1
yes 0 no)?');
if stats(i,31)==1
    save('stats.mat','stats')
else
    stats(i,31)=0;
    stats(i,44:47)=0;
    stats(i,61:65)=0;
    stats(i,70:71)=0;
end
end
end

end
end

save('stats.mat','stats')
disp(sprintf('%s','Stats file saved'))

```

diary off

B. Data Processing Code

```

clc
clear all

ampfactor=10;
cow='J071';

load('stats.mat');
master1=zeros(length(stats),15);
master2=zeros(length(stats),15);
master3=zeros(length(stats),15);
i1=1;
i2=1;
i3=1;
ip1=1;
ip2=1;
ip3=1;
it1=1;
it2=1;
it3=1;
ir1=1;
ir2=1;
ir3=1;

for i=1:1:(length(stats))

%Subq1

%main statistics

if stats(i,1)==1
    triggerstats1(i1,1:3)=stats(i,10:12);
    i1=i1+1;
    master1(i,1:3)=stats(i,10:12);

%PQRS statistics

%PSTATS

if stats(i,2)==1

call=sprintf('%s%s%s%i%s',cow,'_', 'PQRST_',i-
1,'_1.mat');
clear pstats
clear pwave
clear rstats
clear rwave
clear tstats
clear twave

```

```

load(call);

%V
clear Z;
Z=pstats(1,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
pstats1(ip1,1)=O;
master1(i,4)=O;

%P/R
clear Z;
Z=pstats(2,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
pstats1(ip1,2)=O;
master1(i,5)=O;

%P-R
clear Z;
Z=pstats(3,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
pstats1(ip1,3)=O;
master1(i,6)=O;

%P-P
clear Z;
Z=pstats(4,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
pstats1(ip1,4)=O;
master1(i,7)=O;

ip1=ip1+1;

end

%TSTATS
if stats(i,3)==1
call=sprintf('%s%s%s%i%s',cow,'_', 'PQRST_',i-
1,'_1.mat');
clear pstats
clear pwave
clear rstats
clear rwave
clear tstats
clear twave
load(call);

```

```

% V
clear Z;
Z=tstats(1,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
tstats1(it1,1)=O;
master1(i,8)=O;

% T/R
clear Z;
Z=tstats(2,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
tstats1(it1,2)=O;
master1(i,9)=O;

% T-R
clear Z;
Z=tstats(3,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
tstats1(it1,3)=O;
master1(i,10)=O;

% T-T
clear Z;
Z=tstats(4,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
tstats1(it1,4)=O;
master1(i,11)=O;

it1=it1+1;

end

% RSTATS
if stats(i,2)==1 && stats(i,3)==1

% V
clear Z;
Z=rstats(1,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end

```

```

end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
rstats1(ir1,1)=O;
master1(i,12)=O;

%length
clear Z;
Z=rstats(2,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:length(Z);
    if Z(1,ii)==0
        else
            ZO(1,iii)=Z(1,ii);
            iii=iii+1;
        end
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
rstats1(ir2,2)=O;
master1(i,13)=O;

%R-R
clear Z;
Z=rstats(3,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:length(Z);
    if Z(1,ii)==0
        else
            ZO(1,iii)=Z(1,ii);
            iii=iii+1;
        end
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
rstats1(ir1,3)=O;
master1(i,14)=O;

%mean V
clear Z;
Z=rstats(4,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:length(Z);
    if Z(1,ii)==0
        else
            ZO(1,iii)=Z(1,ii);
            iii=iii+1;
        end
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
rstats1(ir1,4)=O;
master1(i,15)=O;

ir1=ir1+1;

end

end

%%%%%%%%%%
%%%%%%%%%%
%%%%%%%%%%
%Subq2

%main statistics

if stats(i,4)==1
    triggerstats2(i2,1:3)=stats(i,16:18);
    i2=i2+1;
    master2(i,1:3)=stats(i,16:18);

%PQRS statistics

%PSTATS

if stats(i,5)==1

call=sprintf('%s%s%s%i%s',cow,'_', 'PQRST_',i-
1, '_2.mat');
clear pstats
clear pwave
clear rstats
clear rwave
clear tstats

```



```

clear twave
load(call);

%V
clear Z;
Z=pstats(1,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
pstats2(ip2,1)=O;
master2(i,4)=O;

%P/R
clear Z;
Z=pstats(2,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
pstats2(ip2,2)=O;
master2(i,5)=O;

%P-R
clear Z;
Z=pstats(3,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;

for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
pstats2(ip2,3)=O;
master2(i,6)=O;

%P-P
clear Z;
Z=pstats(4,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
pstats2(ip2,4)=O;
master2(i,7)=O;

ip2=ip2+1;

end

%TSTATS
if stats(i,6)==1
call=sprintf('%s%s%s%i%s',cow,'_', 'PQRST_',i-
1, '_2.mat');
clear pstats
clear pwave
clear rstats
clear rwave
clear tstats
clear twave
load(call);

```

```

% V
clear Z;
Z=tstats(1,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
tstats2(it2,1)=O;
master2(i,8)=O;

% T/R
clear Z;
Z=tstats(2,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
tstats2(it2,2)=O;
master2(i,9)=O;

% T-R
clear Z;
Z=tstats(3,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
tstats2(it2,3)=O;
master2(i,10)=O;

% T-T
clear Z;
Z=tstats(4,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
tstats2(it2,4)=O;
master2(i,11)=O;

it2=it2+1;
end

% RSTATS
if stats(i,5)==1 && stats(i,6)==1

% V
clear Z;
Z=rstats(1,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
tstats2(it2,5)=O;
master2(i,12)=O;

it2=it2+1;
end

```

```

        iii=iii+1;
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
rstats2(ir2,1)=O;
master2(i,12)=O;

%length
clear Z;
Z=rstats(2,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
rstats2(ir2,2)=O;
master2(i,13)=O;

%R-R
clear Z;
Z=rstats(3,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
rstats2(ir2,3)=O;

        master2(i,14)=O;

        %mean V
        clear Z;
        Z=rstats(4,:);
        %remove zeros, outliers and mean
        clear O
        clear ZO;
        iii=1;
        for ii=1:1:length(Z);
            if Z(1,ii)==0
            else
                ZO(1,iii)=Z(1,ii);
                iii=iii+1;
            end
        end
        if length(ZO)>= 3
            O=mean(removeoutliers(ZO));
        else
            O=mean(ZO);
        end
        %end of zero routine
        rstats2(ir2,4)=O;
        master2(i,15)=O;

        ir2=ir2+1;

    end

end

%Subq3

%main statistics

if stats(i,7)==1
    triggerstats3(i3,1:3)=stats(i,22:24);
    i3=i3+1;
    master3(i,1:3)=stats(i,22:24);

%PQRS statistics

%PSTATS

if stats(i,8)==1

call=sprintf('%s%s%s%i%s',cow,'_', 'PQRST_',i-
1,'_3.mat');
    clear pstats
    clear pwave
    clear rstats
    clear rwave
    clear tstats
    clear twave
    load(call);

```

```

%V
clear Z;
Z=pstats(1,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
pstats3(ip3,1)=O;
master3(i,4)=O;

%P/R
clear Z;
Z=pstats(2,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
pstats3(ip3,2)=O;
master3(i,5)=O;

%P-R
clear Z;
Z=pstats(3,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
pstats3(ip3,3)=O;
master3(i,6)=O;

%P-P
clear Z;
Z=pstats(4,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
pstats3(ip3,4)=O;
master3(i,7)=O;

ip3=ip3+1;

end

%TSTATS

if stats(i,9)==1
call=sprintf('%s%s%s%i%s',cow,'_', 'PQRST_',i-
1,'_3.mat');
clear pstats
clear pwave
clear rstats
clear rwave
clear tstats
clear twave
load(call);

%V
clear Z;
ZO(1,iii)=Z(1,ii);
iii=iii+1;
end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
pstats3(ip3,3)=O;
master3(i,6)=O;

%P-P
clear Z;
Z=pstats(4,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
pstats3(ip3,4)=O;
master3(i,7)=O;

ip3=ip3+1;

end

%TSTATS

if stats(i,9)==1
call=sprintf('%s%s%s%i%s',cow,'_', 'PQRST_',i-
1,'_3.mat');
clear pstats
clear pwave
clear rstats
clear rwave
clear tstats
clear twave
load(call);

%V
clear Z;

```

```

Z=tstats(1,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
tstats3(it3,1)=O;
master3(i,8)=O;

% T/R
clear Z;
Z=tstats(2,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
tstats3(it3,2)=O;
master3(i,9)=O;

% T-R
clear Z;
Z=tstats(3,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
tstats3(it3,3)=O;
master3(i,10)=O;

% T-T
clear Z;
Z=tstats(4,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
tstats3(it3,4)=O;
master3(i,11)=O;

it3=it3+1;

end

% RSTATS
if stats(i,8)==1 && stats(i,9)==1

% V
clear Z;
Z=rstats(1,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:1:length(Z);
    if Z(1,ii)==0
    else
        ZO(1,iii)=Z(1,ii);
        iii=iii+1;
    end
end
end

```

```

if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
rstats3(ir3,1)=O;
master3(i,12)=O;

```

```

%length
clear Z;
Z=rstats(2,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:length(Z);
    if Z(1,ii)==0
        else
            ZO(1,iii)=Z(1,ii);
            iii=iii+1;
        end
    end
end

```

```

if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
rstats3(ir3,2)=O;
master3(i,13)=O;

```

```

%R-R
clear Z;
Z=rstats(3,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:length(Z);
    if Z(1,ii)==0
        else
            ZO(1,iii)=Z(1,ii);
            iii=iii+1;
        end
    end
end

```

```

if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
rstats3(ir3,3)=O;
master3(i,14)=O;

```

```

%mean V

```

```

clear Z;
Z=rstats(4,:);
%remove zeros, outliers and mean
clear O
clear ZO;
iii=1;
for ii=1:length(Z);
    if Z(1,ii)==0
        else
            ZO(1,iii)=Z(1,ii);
            iii=iii+1;
        end
    end
end
if length(ZO)>= 3
    O=mean(removeoutliers(ZO));
else
    O=mean(ZO);
end
%end of zero routine
rstats3(ir3,4)=O;
master3(i,15)=O;

ir3=ir3+1;

```

```

end

```

```

end

```

```

end

```

```

%Master Matrix

```

```

master=zeros(3,14);
master(1,1:2)=mean(triggerstats1(:,2:3));
master(2,1:2)=mean(triggerstats2(:,2:3));
master(3,1:2)=mean(triggerstats3(:,2:3));

```

```

master(1,3)=mean(rstats1(:,1));
master(2,3)=mean(rstats2(:,1));
master(3,3)=mean(rstats3(:,1));

```

```

master(1,4)=mean(tstats1(:,1));
master(2,4)=mean(tstats2(:,1));
master(3,4)=mean(tstats3(:,1));

```

```

master(1,5)=mean(pstats1(:,1));
master(2,5)=mean(pstats2(:,1));
master(3,5)=mean(pstats3(:,1));

```

```

master(1,6)=mean(pstats1(:,2));
master(2,6)=mean(pstats2(:,2));
master(3,6)=mean(pstats3(:,2));

```

```

master(1,7)=mean(tstats1(:,2));
master(2,7)=mean(tstats2(:,2));
master(3,7)=mean(tstats3(:,2));

```

```

master(1,8)=mean(pstats1(:,3));
master(2,8)=mean(pstats2(:,3));
master(3,8)=mean(pstats3(:,3));

master(1,9)=mean(tstats1(:,3));
master(2,9)=mean(tstats2(:,3));
master(3,9)=mean(tstats3(:,3));

master(1,10)=mean(rstats1(:,3));
master(2,10)=mean(rstats2(:,3));
master(3,10)=mean(rstats3(:,3));

master(1,11)=mean(tstats1(:,4));
master(2,11)=mean(tstats2(:,4));
master(3,11)=mean(tstats3(:,4));

master(1,12)=mean(pstats1(:,4));
master(2,12)=mean(pstats2(:,4));
master(3,12)=mean(pstats3(:,4));

master(1,13)=mean(rstats1(:,2));
master(2,13)=mean(rstats2(:,2));
master(3,13)=mean(rstats3(:,2));

master(1,14)=mean(rstats1(:,4));
master(2,14)=mean(rstats2(:,4));
master(3,14)=mean(rstats3(:,4));

xlswrite('Master.xls',master)

xlswrite('Subq1.xls',master1)
xlswrite('Subq2.xls',master2)
xlswrite('Subq3.xls',master3)

xlswrite('triggerstats1.xls',triggerstats1)
xlswrite('triggerstats2.xls',triggerstats2)
xlswrite('triggerstats3.xls',triggerstats3)

xlswrite('pstats1.xls',pstats1)
xlswrite('pstats2.xls',pstats2)
xlswrite('pstats3.xls',pstats3)

xlswrite('tstats1.xls',tstats1)
xlswrite('tstats2.xls',tstats2)
xlswrite('tstats3.xls',tstats3)

xlswrite('rstats1.xls',rstats1)
xlswrite('rstats2.xls',rstats2)
xlswrite('rstats3.xls',rstats3)

disp('Data compiled, excel files written')

```

C. Miscellaneous Code

ECG Extract

```
%ECG Data Extraction Program
%Sean Warren
% Ver 7/14/2010

clear all
clc

%Load Dir
dir=dir('*.*.dat');

%Data Load Parameters
%Change these values to change how the
program takes the data
Cow=sprintf('%s','J066');      %Change
this to Cow #
Tprompt=0;                    %Change this to 1 if
you want prog to prompt for time of data set.
time=30;                      %Time if Tprompt=0
Hzprompt=0;                   %Change this to 1 if
you want prog to prompt for Hz of data set.
hz=400;                       %hz if Hzprompt=1
counter=0;

for i=1:1:length(dir')

    dataascii=importdata(dir(i).name);
    data=dataascii.data;

%Prompts for Channels and Time
% screwC=input('Screw-in Channel:');
% subq1C=input('SubQ 1 Channel:');
% subq2C=input('SubQ 2 Channel:');
% subq3C=input('SubQ 3 Channel:');

screwC=4;
subq1C=2;
subq2C=5;
subq3C=3;

if Tprompt==1;
    time=input('Time of dataset:')
end
if Hzprompt==1;
    hz=input('Hz of dataset:')
end

%Saving Data
```

```
t=[0:1/hz:time-1/hz];
subq1=data(:,subq1C);
subq2=data(:,subq2C);
subq3=data(:,subq3C);
screw=data(:,screwC);

% plot(t,screw,t,subq1,t,subq2,t,subq3)
% xlim([0 5]);
% legend('screw','subq1','subq2','subq3');
% pause

filename=sprintf('%s%c%i%s',Cow,'_',counter,
'.mat');
save(filename,'subq1','subq2','subq3','screw','
t')

counter=counter+1;

end
% filenamevar5=sprintf('%i',VBPM);
%
filename1=char(strcat({'pressures'},{'VADp
w'},{filenamevar3},{ 'ts'},{filenamevar2},{'s
pt'}},{filenamevar4},{'VBPM'}},{filenamevar
5}));

“Play Around” Script

clear all
clc

load('ECG.mat');
t=0:1/400:(length(ECG)-1)/400;
hold off

for i=1:1:length(ECG);

    i=1;
clear ecg
%FFT
% fs=400;
% L=length(ECG);
% NFFT = 2^nextpow2(L); % Next power
of 2 from length of y
% Y = fft(ECG(1,:),NFFT)/L;
% f = fs/2*linspace(0,1,NFFT/2+1);
% figure
% plot(f,2*abs(Y(1:NFFT/2+1)))
% title('Single-Sided Amplitude Spectrum
of y(t)')
% xlabel('Frequency (Hz)')
% ylabel('|Y(f)|')
% xlim([0 100])
% pause
%
% [fft,f]=positiveFFT(ECG(i,:),400);
```



```

%
% plot(f,abs(fft));
% xlim([0 100]);
% pause

signalfilt=ECG(i,:);
%Signal Drift, FIR method
fs=400;           %sampling rate
fch=50;          %high pass cutoff freq
fcl=1;           %low pass cutoff freq
N=80;            %filter order
filt=fir1(N,[fcl/(fs/2) fch/(fs/2)]);
%Design high pass filter fch/(fs/2)
signalfilt=filter(filt,1,signalfilt);

hold off

subplot(2,1,1)

plot(t(1:1:end),ECG(i,:),'color','black')
title('Unfiltered ECG')
xlim([0 5])
subplot(2,1,2)
plot(t(1:1:end),signalfilt,'color','black');
title('Filtered ECG')
xlim([0 5])
pause

%SG Filter
k=10;            %polynomial order for
SG filter
f=50+1;          %frame length (must
be odd)
signalfilt=sgolayfilt(signalfilt,k,f);
signal=signalfilt;

stdev=2.5*std(signalfilt(:));
m=mean(signalfilt(:));

for j=1:1:length(signalfilt);
    if signalfilt(j) > (m+stdev)
        signalfilt(j)=m+stdev;
    end
    if signalfilt(j) < (m-stdev)
        signalfilt(j)=m-stdev;
    end
end

% signalfilt=diff(signalfilt);
disp(i);
ecg=ECG(i,:);
ecg=ecg/(max(abs(ecg)));
signalfilt=signalfilt/(max(abs(signalfilt)));

subplot(2,1,2)
plot(t(1:1:end),signalfilt);

% legend('subq','subqfilt');
xlim([0 30]);
pause
xlim([0 2.5]);
itwave=input('Is the T wave inverted? (0 no
1 yes 2 indiscernable:');
ipwave=input('Is the P wave inverted? (0 no
1 yes 2 indiscernable:');
ithresh=input('First 3 R wave detection
threshold:');
iblack=input('Initial blanking period in
sec:');
iblack=iblack*400;
RRT1=0;
RRT2=0;
RRT3=0;
RRV1=0;
RRV2=0;
RRV3=0;
blankpercent=.80;
initialR=1;
ppercent=.30;
tpercent=.45;
rpercentforward=.15;
rpercentbackward=.10;
initialrsec=.050;
ramppercent=.80;
twave=zeros(1,length(signalfilt));
twave(:)=NaN;
pwave=zeros(1,length(signalfilt));
pwave(:)=NaN;
rwave=zeros(1,length(signalfilt));
rwave(:)=NaN;
trigger=zeros(1,length(signalfilt));
lastR=0;
RRTvariabilityfactor=.3;
RRVvariabilityfactor=.3;
clear pstats
clear rstats
clear tstats
pz=1;
rz=1;
tz=1;

for k=1:1:length(signalfilt);
    if initialR==3
        if signalfilt(k)>=ithresh && (k-
RRT1)>iblack
            trigger(k)=1;
            RRT3=RRT2;
            RRT2=RRT1;
            RRT1=k;
            lastR=k;
            RRV3=RRV2;
            RRV2=RRV1;

```

```

RRV1=max(signalfilt(k:(k+initialrsec*400))
);
    initialR=initialR+1;
    RRT3=RRT2-RRT3;
    RRT2=RRT1-RRT2;
    RRT1=RRT2;
end
elseif initialR==2
    if signalfilt(k)>=ithresh && (k-
RRT1)>iblank
        trigger(k)=1;
        RRT3=RRT2;
        RRT2=RRT1;
        RRT1=k;
        RRV3=RRV2;
        RRV2=RRV1;

RRV1=max(signalfilt(k:(k+initialrsec*400))
);
    initialR=initialR+1;
end
elseif initialR==1
    if signalfilt(k)>=ithresh
        trigger(k)=1;
        RRT3=RRT2;
        RRT2=RRT1;
        RRT1=k;
        RRV3=RRV2;
        RRV2=RRV1;

RRV1=max(signalfilt(k:(k+initialrsec*400))
);
    initialR=initialR+1;
end
else
    if
signalfilt(k)>=((RRV1+RRV2+RRV3)/3*ra
mppercent) && (k-
lastR)>(((RRT1+RRT2+RRT3)/3)*blankper
cent)
        trigger(k)=1;
        RRT3=RRT2;
        RRT2=RRT1;
        if (k-
lastR)>((RRT1+RRT2+RRT3)/3)*(1+RRTv
ariabilityfactor) || (k-
lastR)<((RRT1+RRT2+RRT3)/3)*(1-
RRTvariabilityfactor)
            else
                RRT1=k-lastR;
                %p,t,r wave capture and voltage
indexing
                pwave(round(k-(k-
lastR)*(ppercent+rpercentbackward)):round(
k-(k-

```

```

lastR)*rpercentbackward))=ECG(i,round(k-
(k-
lastR)*(ppercent+rpercentbackward)):round(
k-(k-lastR)*rpercentbackward));

pstats(1,pz)=abs(max(pwave(round(k-(k-
lastR)*(ppercent+rpercentbackward)):round(
k-(k-lastR)*rpercentbackward)))-
min(pwave(round(k-(k-
lastR)*(ppercent+rpercentbackward)):round(
k-(k-lastR)*rpercentbackward)))));
        twave(round(k-(k-lastR)*(1-
rpercentforward)):round(k-(k-lastR)*(1-
rpercentforward-tpercent)))=ECG(i,round(k-
(k-lastR)*(1-rpercentforward)):round(k-(k-
lastR)*(1-rpercentforward-tpercent)));

tstats(1,tz)=abs(max(twave(round(k-(k-
lastR)*(1-rpercentforward)):round(k-(k-
lastR)*(1-rpercentforward-tpercent))))-
min(twave(round(k-(k-lastR)*(1-
rpercentforward)):round(k-(k-lastR)*(1-
rpercentforward-tpercent)))));
        rwave(round(k-(k-
lastR)*(rpercentbackward)):round(k+(k-
lastR)*(rpercentforward)))=ECG(i,round(k-
(k-lastR)*(rpercentbackward)):round(k+(k-
lastR)*(rpercentforward)));

rstats(1,rz)=abs(max(rwave(round(k-(k-
lastR)*(rpercentbackward)):round(k+(k-
lastR)*(rpercentforward))))-
min(rwave(round(k-(k-
lastR)*(rpercentbackward)):round(k+(k-
lastR)*(rpercentforward)))));
        %p,t wave / r wave ratio indexing

pstats(2,pz)=pstats(1,pz)/rstats(1,rz);
tstats(2,tz)=tstats(1,tz)/rstats(1,rz);
[maxv,prt1]=max(rwave(round(k-
(k-lastR)*(rpercentbackward)):round(k+(k-
lastR)*(rpercentforward)))));
        %qrs length routine
        rmstart=round(k-(k-
lastR)*(rpercentbackward));
        rmend=round(k+(k-
lastR)*(rpercentforward));
        qrsseg=ECG(i,rmstart:rmend);
        qrsseg=qrsseg+(0-min(qrsseg));
        qrsseg=qrsseg/max(qrsseg);
        [qrsmax
ignoreindex]=max(qrsseg);
        averaging=6;
        clear averagingm
        qrswidthstart=0;
        iii=0;

```

```

    qrsdiffpercent=.8;
    for jkl=averaging:-1:1
    clear averagingm
    iii=0;
    if qrswidthstart==0
    for ii=(ignoreindex):-1:1;
    if qrswidthstart==0
    if iii<jkl
    averagingm(iii+1)=qrsseg(ii);
    iii=iii+1;
    else
    for
ll=1:1:(length(averagingm)-1)

averagingm(ll)=averagingm(ll+1);
end
    averagingm(jkl)=qrsseg(ii);
    if (qrsmax-
mean(averagingm))>=qrsdiffpercent*(qrsma
x-min(qrsseg(1:ignoreindex)))
    qrswidthstart=ii;
    end
    end
end
end
averaging=averaging-1;
end

    averaging=6;
    qrswidthend=0;
    for jkl=averaging:-1:1
    clear averagingm
    iii=0;
    if qrswidthend==0
    for
ii=(ignoreindex):1:length(qrsseg);
    if qrswidthend==0
    if iii<jkl
    averagingm(iii+1)=qrsseg(ii);
    iii=iii+1;
    else
    for
ll=1:1:(length(averagingm)-1)

averagingm(ll)=averagingm(ll+1);
end
    averagingm(jkl)=qrsseg(ii);
    if (qrsmax-
mean(averagingm))>=qrsdiffpercent*(qrsma
x-min(qrsseg(ignoreindex:end)))
    qrswidthend=ii;
    end
    end
end
end

    end
    averaging=averaging-1;
    end

    %
    qrsindices=(1:1:length(qrsseg));
    %
    plot(qrsindices,qrsseg)
    %
    hold;
    %
    plot(qrsindices(qrswidthstart),qrsseg(qrswid
hstart),'-rs','LineWidth',2,...
    %
    'MarkerEdgeColor','k',...
    %
    'MarkerFaceColor','g',...
    %
    'MarkerSize',10)
    %
    plot(qrsindices(qrswidthend),qrsseg(qrswid
hend),'-rs','LineWidth',2,...
    %
    'MarkerEdgeColor','k',...
    %
    'MarkerFaceColor','g',...
    %
    'MarkerSize',10)
    %
    hold;
    %
    pause

    if ipwave==1

[maxv,prt2]=min((pwave(round(k-(k-
lastR)*(ppercent+rpercentbackward)):round(
k-(k-lastR)*rpercentbackward))));
    else

[maxv,prt2]=max((pwave(round(k-(k-
lastR)*(ppercent+rpercentbackward)):round(
k-(k-lastR)*rpercentbackward))));
    end
    %
    [maxv,prt2]=min(abs(diff(pwave(round(k-
(k-
lastR)*(ppercent+rpercentbackward)):round(
k-(k-lastR)*rpercentbackward))));
    pstats(3,pz)=(prt1+round(k-(k-
lastR)*(rpercentbackward))-prt2-round(k-(k-
lastR)*(ppercent+rpercentbackward)))/400;
    if itwave==1

[maxv,trt2]=min((twave(round(k-(k-
lastR)*(1-rpercentforward)):round(k-(k-
lastR)*(1-rpercentforward-tpercent))));
    else

[maxv,trt2]=max((twave(round(k-(k-
lastR)*(1-rpercentforward)):round(k-(k-
lastR)*(1-rpercentforward-tpercent))));
    end
    %
    [maxv,trt2]=max(abs(diff(twave(round(k-(k-

```

```

lastR)*(1-rpercentforward)):round(k-(k-
lastR)*(1-rpercentforward-tpercent)))));
    plot(t,signalfilt);
    hold;
    mark=(prt2+round(k-(k-
lastR)*(ppercent+rpercentbackward)));
    mark2=(trt2+round(k-(k-
lastR)*(1-rpercentforward)));
    plot(t(mark),signalfilt(mark),'--
rs','LineWidth',2,...
'MarkerEdgeColor','k',...
'MarkerFaceColor','g',...
'MarkerSize',10)
    plot(t(mark2),signalfilt(mark2),'--
rs','LineWidth',2,...
'MarkerEdgeColor','k',...
'MarkerFaceColor','g',...
'MarkerSize',10)
    xlim([(prt2+round(k-(k-
lastR)*(ppercent+rpercentbackward)))/400-1
((prt2+round(k-(k-
lastR)*(ppercent+rpercentbackward)))+400*
1)/400])
    hold
    pause
    tstats(3,tz)=(trt2+round(k-(k-
lastR)*(1-rpercentforward))-lastR)/400;
    rz=rz+1;
    pz=pz+1;
    tz=tz+1;
end
RRV3=RRV2;
RRV2=RRV1;
if
max(signalfilt(k:(k+initialrsec*400)))>((RR
V1+RRV2+RRV3)/3)*(1+RRVvariabilityfa
ctor) ||
max(signalfilt(k:(k+initialrsec*400)))<((RR
V1+RRV2+RRV3)/3)*(1-
RRVvariabilityfactor)
else
RRV1=max(signalfilt(k:(k+initialrsec*400))
);
end
lastR=k;
end
end

stem(t,trigger,'linewidth',1);
pause
hold;
plot(t,signalfilt);
plot(t,trigger,'linewidth',2);
hold;
for l=0:3:27
    xlim([l (l+3)])
    pause
end

plot(t,ECG(i,:),'color','black','linewidth',1);
hold;
plot(t,pwave,'color','red','linewidth',2)
plot(t,twave,'color','blue','linewidth',2)
plot(t,rwave,'color','green','linewidth',2)
for l=0:3:27
    xlim([l (l+3)])
    pause
end
hold;

end

```