TIME-FREQUENCY ANALYSIS OF INTRACARDIAC ELECTROGRAM

A Thesis Presented to the Faculty of
California Polytechnic State University,
San Luis Obispo, California

In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Biomedical Engineering

By
Erik Brockman
June 2009

COMMITTEE MEMBERSHIP

TITLE:	Time-Frequency Analysis of Intracardiac Electrogram
AUTHOR:	Erik Brockman
DATE SUBMITTED:	June 2009
COMMITTEE CHAIR:	Dr. Lily Laiho, Assistant Professor Biomedical Engineering
COMMITTEE MEMBER:	Dr. Lynne Slivovsky, Associate Professor Electrical Engineering
COMMITTEE MEMBER:	Dr. Kristen Cardinal, Assistant Professor Biomedical Engineering

ABSTRACT

Time-Frequency Analysis of Intracardiac Electrogram

Erik Brockman

The Cardiac Rhythm Management Division of St. Jude Medical specializes in the development of implantable cardioverter defibrillators that improve the quality of life for patients diagnosed with a variety of cardiac arrhythmias, especially for patients prone to sudden cardiac death. With the goal to improve detection of cardiac arrhythmias, this study explored the value in time-frequency analysis of intracardiac electrogram in four steps. The first two steps characterized, in the frequency domain, the waveforms that construct the cardiac cycle. The third step developed a new algorithm that putatively provides the least computationally expensive way to identifying cardiac waveforms in the frequency domain. Lastly, this novel approach to analyzing intracardiac electrogram was compared to a threshold crossing algorithm that strictly operates in the time domain and that is currently utilized by St. Jude Medical. The new algorithm demonstrated an equally effective method in identifying the QRS complex on the ventricular channel. The next steps in pursing time-frequency analysis of intracardiac electrogram include implementing the new algorithm on a testing platform that emulates the latest implantable cardioverter defibrillator manufactured by St. Jude Medical and pursuing a similar algorithm that can be employed on the atrial channel.

Keywords: time-frequency, intracardiac electrogram, QRS complex,
cardiac arrhythmia, frequency domain, time domain.

ACKNOWLEDGMENTS

This thesis would not be possible without the generous support and advice from Dr. Henry Wang at St. Jude Medical. I thank him for his time and willingness to share the knowledge that has led to my success and has propelled my interest in cardiac rhythm management.

I thank Dr. Lily Laiho, Dr. Lynne Slivovsky, and Dr. Kristen Cardinal for their participation and recommendations as members of my thesis committee.

Lastly, I thank my parents for their encouragement and financial support during my time at Cal Poly.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

AADS	Automatic Arrhythmia Detection System
AV Node	Atrioventricular Node
CPLX	Complexity Measure Analysis
DFA	Dominant Frequency Analysis
DFT	Discrete Fourier Transform
ECG	Electocardiogram
FFT	Fast Fourier Transform
ICEG	Intracardiac Electrogram
ICD	Implantable Cardioverter Defibrillator
VF	Ventricular Filtering
PSR	Phase Space Reconstruction
SA Node	Sinoatrial Node
SCD	Sudden Cardiac Death
SNR	Signal-to-Noise Ratio
SPEC	Spectral Analysis
TCI	Threshold Crossing Intervals

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Motivation of Thesis

Sudden cardiac death (SCD) is a major public health problem and a leading cause of mortality in the western world, claiming 450,000 lives per year in the United States (Zheng, et al. 2001). SCD results from an abrupt loss in cardiac function, often preceded by cardiac arrhythmia. The Cardiac Rhythm Management Division of St. Jude Medical specializes in the development of implantable cardioverter defibrillators (ICDs) that improve the quality of life for patients diagnosed with a variety of cardiac arrhythmias, including patients prone to SCD. Current ICDs depend on rapid and accurate recognition of intracardiac electrogram (ICEG) accomplished by algorithms operating in the time domain. Because analysis in the time domain only extracts partial information from ICEG such information, in some cases, is insufficient for accurate arrhythmia recognition. In an effort to improve the accuracy of ICEG recognition and to expand the range of treatable arrhythmias, St. Jude Medical has proposed the adoption of time-frequency analysis to the next generation of ICDs. This thesis explores the feasibility of time-frequency analysis of ICEG and is supported by St. Jude Medical through the MEDITEC consortium at California Polytechnic State University.

Document Overview

This thesis is organized into four primary topics: a background supporting the pursuit of time-frequency analysis, a methodology for assessing the feasibility of time-frequency analysis, the results comparing a new algorithm that uses time-frequency analysis to a currently used algorithm, and lastly, conclusions and recommendations on future research in the area of time-frequency analysis of ICEG.

Software Platform

All software was written in the Matlab programming environment using Matlab Version 7.6.0, made available by The Mathworks and administrated under an academic license owned by Cal Poly. This platform selection was made for three primary reasons: its superior plotting capabilities, its integrated fast Fourier transform (FFT), and the fact that it is currently used as a research tool at St. Jude Medical. Underlying all of these advantages resides Matlab's ability to reduce the amount of time required for debugging and altering a top-level algorithm implemented on an ICD.

BACKGROUND

Sudden Cardiac Death

SCD is a major public health problem and a leading cause of mortality in the western world, claiming 450,000 lives per year in the United States (Zheng, et al. 2001). SCD results from an abrupt loss in cardiac function, often preceded by abnormal electrical activity of the heart known as cardiac arrhythmia. The most common cardiac arrhythmia leading to SCD occurs when ventricular tachycardia, defined as abnormal acceleration of ventricular contraction, deteriorates into ventricular fibrillation, during which asynchronous contractions of cardiac muscle located in the ventricles fail to effectively eject blood (Rubart and Zipes 2005). When cardiac output suddenly drops, tissues throughout the body are deprived of adequate oxygen supply causing irreversible damage to the tissue and often patient death. Methods for preventing damage induced by reoccurring ventricular fibrillation include implantation of an ICD and prescription of antiarrhythmic drugs (AADs). A number of clinical trials have demonstrated the superiority of ICDs over AADs in the prevention of SCD. And most recently, a study at the University of Ottawa Heart Institute demonstrated that ICDs are underused in both the United States and Canada, suggesting that ICDs will continue to have an important place in preventing SCD (Birnie, et al. 2007). In order to further understand how ICDs prevent SCD, one must first understand cardiac anatomy and physiology of a normally functioning heart.

Cardiac Anatomy and Physiology

The normal heart is a strong, muscular pump designed to provide the body with necessary metabolic nutrients and waste removal by pumping blood throughout the circulatory system. As identified in Figure 1, the heart consists of four chambers, four valves and several key electrical pathways. Beginning at the right atrium, blood is forced through the tricuspid valve into the right ventricle. Upon ventricular contraction, the blood is pumped out the pulmonary semilunar valve and travels through the pulmonary arteries to the lungs. From there, oxygenated blood returns through the pulmonary veins and fills the left atrium. Blood is then forced through the mitral valve, filling the left ventricle. During a second ventricular contraction, blood is pumped through the aortic semilunar valve to the rest of the body. Lastly deoxygenated blood returns to the right atrium via the superior and inferior vena cava (Widmaier, Raff and Strang 2008).

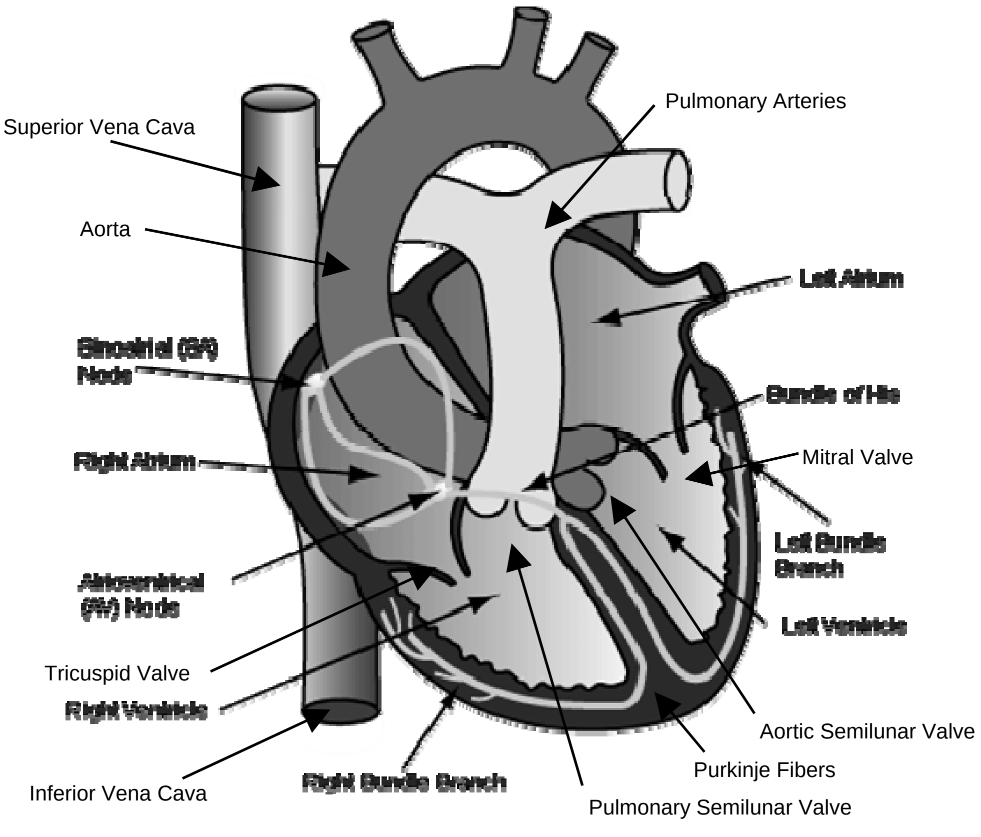

Figure 1: Cardiac Anatomy (St. Jude Medical, 2009)

Throughout the cardiac cycle, electrical signals play a key role in maintaining heart function. These electrical signals appear during cell membrane depolarization and repolarization, signaling muscle contraction and relaxation; the ions primarily responsible for the depolarization current are Ca^+ and Na^+. K^+ is primarily responsible for cell repolarization. A region of the heart called the sinoatrial (SA) node, also illustrated in Figure 1, sets the rate and timing at which all cardiac muscle cells, called myocytes, contract by emitting electrical signals (Widmaier, et al. 2008). Because myocytes are electrically coupled by gap junctions between adjacent cells, electrical signals from the SA node spread rapidly through the walls of the artria, causing simultaneous contraction of the myocytes in the atria. The signals continue propagating to another region of the heart called the atrioventricular (AV) node, located between the right atrium and right ventricle. After a short delay, the electrical signals continue to the apex and ventricles of the heart via the Bundle of his and Purkinje fibers (Widmaier, Raff and Strang 2008). Similar to the myocytes located in the atria, myocytes located in the ventricles contract at the same time in order to pump blood. Without synchronous myocyte contraction, the heart is unable to pump blood. Furthermore, because heart function depends on these electrical signals, any arrhythmia can significantly reduce heart function and requires treatment in order to restore cardiac output.

Implantable Cardioverter Defibrillators

An ICD monitors the electrical signals produced by the heart and based on those signals determines when therapy is needed. When an arrhythmia is detected, treatment is administered by delivering an electric shock. This shock reinstitutes synchronous contractions of the myocytes, restoring normal cardiac function.

An ICD consists of a small battery-powered electrical impulse generator connected to a set of pacing leads implanted in the patient. Figure 2 depicts the typical orientation for a dual-chamber ICD. This device is comprised of pacing leads that pass through the superior vena cava into the right chambers of the heart and a device body that is implanted into the right shoulder. ICDs are similar to pacemakers, however, pacemakers are designed to consistently set cardiac rhythm, while ICDs serve as relatively-permanent safeguards (5-10 years) against abnormal changes in cardiac rhythm. The pacing leads serve two purposes: one, they conduct the electrical signals generated by the heart to the device body where they are interpreted, and two, they are used to deliver an electrical shock from the device body to the heart.

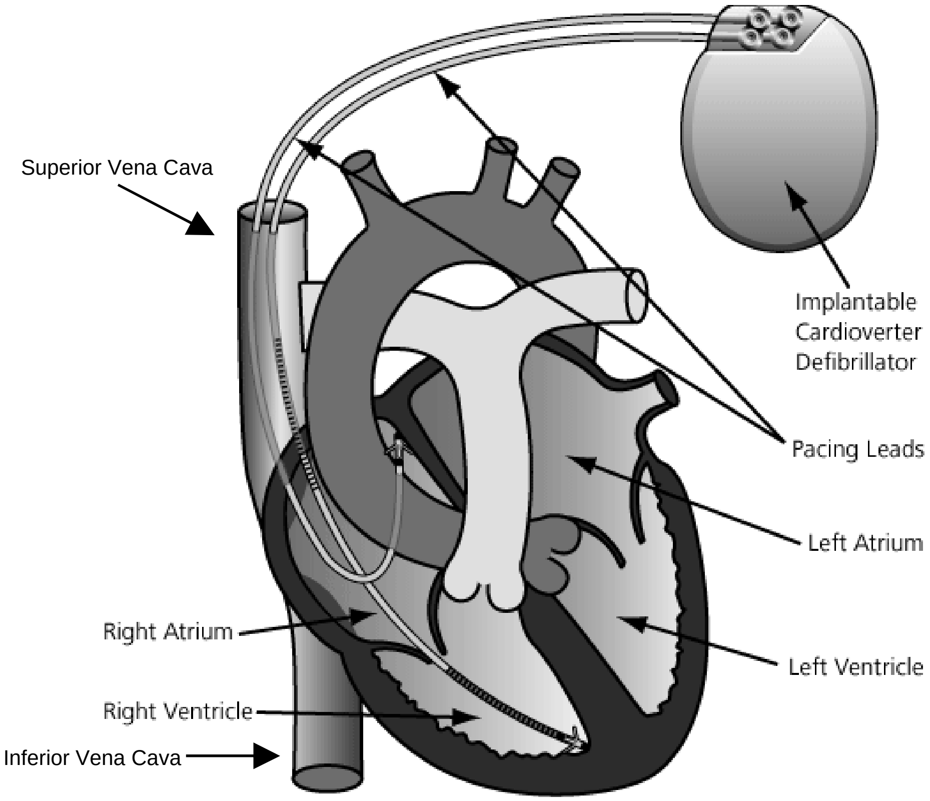

Figure 2: Cardiac Anatomy with Dual-Chamber ICD (St. Jude Medical, 2009)

Administration of the treatment predominantly depends on the rapid and accurate interpretation of the electrical signals generated by the heart. Most individuals recognize these electrical signals as electrocardiogram (ECG), but as recorded internally on an ICD they compose intracardiac electrogram (ICEG).

Intracardiac Electrogram

ICEG consists of four waveforms: the P wave, the QRS complex, the far-field R wave, and the T wave. The P wave and the QRS complex correspond with atrial and ventricular contraction respectively. The far-field R wave is associated with over-sensing of electrical signals propagated to the atrial sensing lead during ventricular contraction, and the T wave is associated with ventricular relaxation. In contrast to ECG that is comprised of the same waveforms on a single channel, ICEG is divided into two channels: the atrial channel and the ventricular channel. Depicted in Figure 3 (a), the atrial channel contains both P waves and far-field R waves, while the ventricular channel, depicted in Figure 3 (b), contains both QRS complexes and T waves. This difference in ICEG compared ECG stems from the fact that while ECG is recorded via surface electrodes placed on the skin, ICEG is recorded via ICD pacing leads implanted into the heart at two most common locations: the right atrium and the right ventricle. This difference in location marshals differences in the signals analyzed by algorithms operating on an ICD (Theres, W and al. 2006). Examining Figure 3 (a) more closely, the P wave is identified by consistently larger amplitude, while the far-field R wave is identified by consistently smaller amplitude. Figure 3 (b) illustrates both the high-frequency, larger-amplitude QRS complex and the slow-frequency, smaller-amplitude T wave. In order to utilize a common scale on Figures 3 (a) and (b), a portion of the QRS complex is not shown. The QRS complex extends to positive 2000 and drops to negative 1500 sampled units. Detection and differentiation of the P wave, the far-field R wave, the QRS complex, and the T wave on their respective channels is an essential component of interpreting ICEG and a known challenge among dual chamber ICDs (Collins, Casavant et al. 2003).

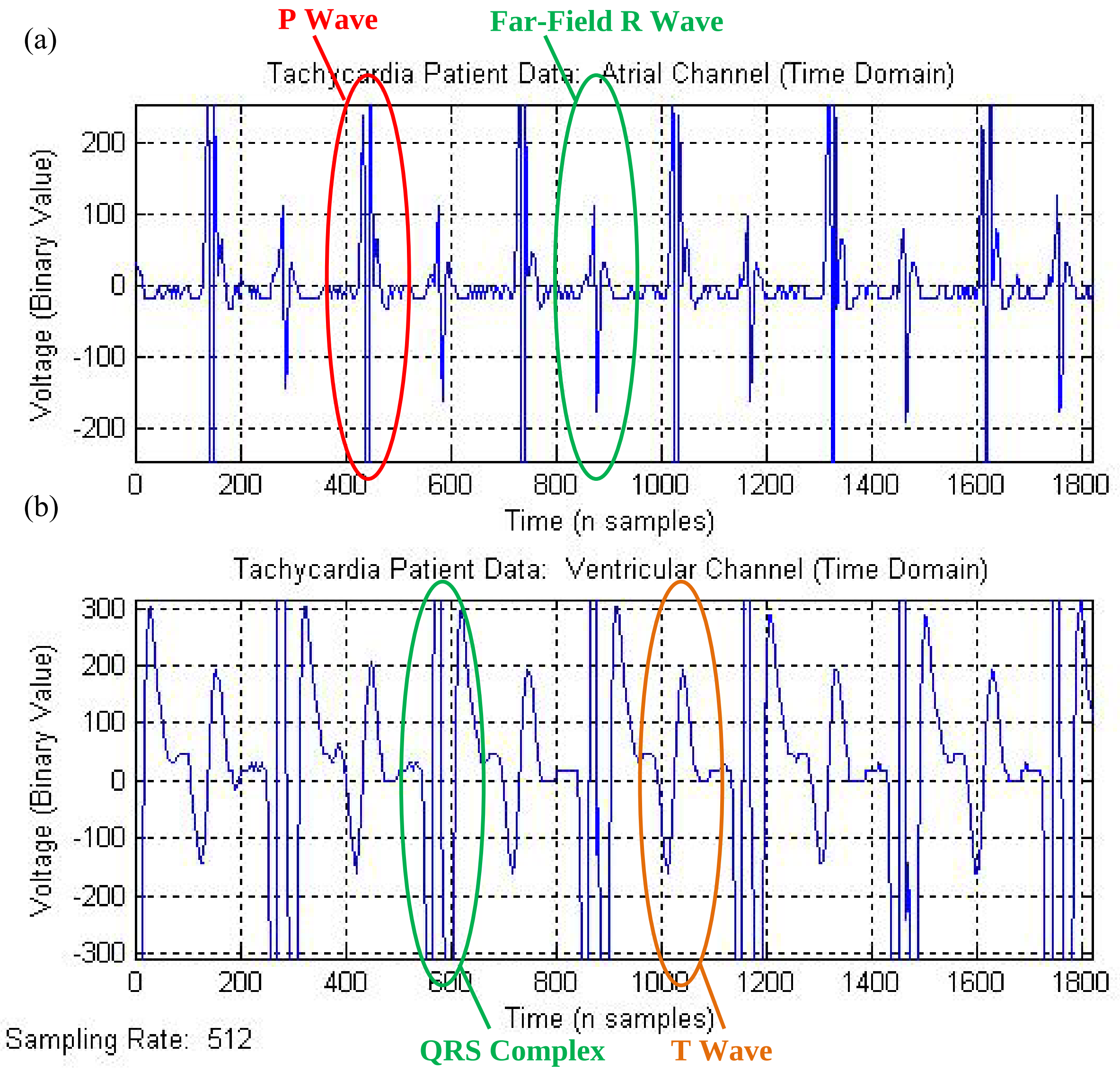

Figure 3: ICEG on the (a) Atrial Channel and on the (b) Ventricular Channel.

Another important quality of ICEG is its tendency to have a higher signal-to-noise ratio (SNR) as compared to ECG.

$$SNR = \left(\frac{Signal_{RMS}}{Noise_{RMS}}\right)^2$$

Factors including power line interference, muscle noise and motion artifact, all contribute to noise and subsequently a decrease in SNR (Sornmo, et al. 1993). This fact presents ICEG as a superior preference over ECG for the development of algorithms that analyze heart rhythm.

Computational Limitations

ICDs are designed to provide reliable heart arrhythmia detection and therapy for a long period of time. The first ICD batteries lasted about two years before needing a replacement, however advances in the battery technology have made it possible for an ICD battery to have a life expectancy between five and ten years, depending on how often they have to deliver power and at what magnitude. When a battery runs low on power, the whole unit has to be replaced surgically, and this replacement procedure is stressful for patients and carries the risk of post operative infection. While new developments continue to improve the capacity and reliability of ICD batteries, Li-ion batteries remain the battery of choice (Crespi, et al. 2001). Rechargeable batteries for ICDs have also been pursued; however they have yet to enter the market because of economic barriers.

In order to maximize battery life, designers strive to limit the number of computational cycles that are exploited for a given cardiac cycle. The percentage of time for a given cardiac cycle utilized by an algorithm operating on an ICD is called duty cycle; more advanced algorithms correlate with higher duty cycles (Afonso, et al. 1996). This computational limitation has hindered the implementation of advanced algorithms requiring higher duty cycles. However, one can justify more computationally expensive algorithms by improving the quality of heart arrhythmia detection and therapy. Time-frequency analysis has the opportunity to fit that profile.

Current Time Domain Analysis

The current generation ICD solely relies on time domain analysis of ICEG (Wang 2009). Although a variety of fibrillation detection algorithms have developed over the past two decades, both atrial and ventricular, annotation of ICEG is still completed in the time domain primarily due to its compliance with computational limitations. Examples of algorithms used to detect ventricular fibrillation range from threshold crossing intervals (TCI), developed in the late 1980s, to more recently in 2007, phase space reconstruction (PSR). The results of a recent study listed in Table 1 demonstrate a need for an improvement in the way in which cardiac rhythm is analyzed.

Table 1: Performance of Fibrillation Detection Algorithms (Unterkofler, Amann and Tratnig 2007)

	Overall Results (95% Specificity)
	Accurate Fibrillation Detection (%)
Threshold Crossing Intervals (TCI)	25.3
Ventricular Filtering (VF)	73.4
Spectral Analysis (SPEC)	69.8
Complexity Measure Analysis (CPLX)	38.8
Phase Space Reconstruction (PSR)	83.8

Proposed Time-Frequency Analysis

Because time domain analysis only extracts partial information from ICEG, such information, in some cases, is not sufficient for accurate arrhythmia recognition. It is predicted that the addition of frequency domain analysis, coupled with time domain analysis (time-frequency analysis) will not only enhance ICD detection and diagnosis of heart arrhythmia, but also enable brand-new algorithms that were not feasible using time domain analysis exclusively.

Previous Research in Time-Frequency Analysis

Other individuals have also predicted the value in pursuing time-frequency analysis of heart rhythm. In 2003, an automatic arrhythmia detection system (AADS) making use of time-frequency analysis was developed; however, the algorithm still relied on heart rate features that were first extracted using time domain analysis (Tsipouras and Fotiadis 2003). Also limiting the applicability of their algorithm, AADS did not interpret ICEG, but rather ECG, and was not intended to be implemented on an ICD. More recently in 2005, a group of researchers explored time-frequency analysis specific to its ability to monitor antiarrhythmic drug effects on atrial fibrillation, again using ECG and again not intended for an ICD (Husser, et al. 2005). One might suggest that an algorithm shown to effectively analyze ECG could simply be modified to operate on an ICD, however as previously discussed, the computational limitations of an ICD and the fact that ICDs interpret different waveforms prevent this approach. The feasibility of an algorithm that fully utilizes time-frequency analysis to annotate ICEG recorded on an ICD has until now yet to be assessed.

METHODOLOGY

Methodology Overview

The methodology assessing the feasibility of time-frequency analysis in this thesis consisted of four steps: (1) ascertaining a difference between the P wave and the far-field R wave on the atrial channel in the frequency domain; (2) establishing a difference between the QRS complex and the T wave on the ventricular channel in the frequency domain; (3) testing an algorithm called domain frequency analysis (DFA) that utilized time-frequency indicators to annotate ICEG recorded by St. Jude Medical ICDs; and (4) comparing the results of the DFA to a currently employed algorithm.

Before leaping to the methods used to characterize cardiac waveforms in the frequency domain, it is important to first describe the datasets, relevant specifics of hardware filtering, and a description of the fast Fourier transform (FFT).

Patient Data

ICEG was recorded and annotated on two St. Jude Medical ICDs implanted in two tachycardia patients; the exact platform will remain unnamed. Table 2 lists key characteristics of the datasets analyzed, including the duration, sampling rate, number of cardiac cycles and the heart rate of each dataset. Each cardiac cycle constitutes one sample that for which the annotation of the currently used algorithm can be compared to DFA. For example, because dataset one is made up of 55 cardiac cycles dataset one contains 55 comparable samples. The difference in heart rates between datasets provides variance in the data. Faster heart rates tend to be more difficult to analyze due to baseline wander, thus including dataset two will strengthen any conclusions that are drawn (Sornmo, et al. 1993). The datasets were obtained through Henry Wang at St. Jude Medical using a web-based tool called Large Electronic File Transfer, ensuring confidentiality and secure handling of the patient data. Two datasets provides a sufficient sample size for the initial steps that this thesis explores. Further investigation would benefit from a significantly larger number of datasets, providing increased representation of the population.

Table 2: Key Dataset Characteristics of ICEG Recorded on St. Jude Medical ICDs

	Duration (s)	**Sampling Rate f_s (samples/s)**	**Number of Cardiac Cycles**	**Heart Rate (bpm)**
Dataset 1	28.39	512	55	116.22
Dataset 2	50.06	512	125	149.81

Hardware Filtering

Another key aspect of any frequency domain analysis of ICEG is knowledge of any pre-filtering performed by hardware in the device body. Low pass filtering must coincide with the sampling rate in order to meet the lower bound requirements of the Nyquist rate, listed below (Geerts, Steyaert and Sansen 2002). One half of f_s, the sampling rate, must be much greater than f, the lowpass cut-off frequency used in hardware to ensure that data in the frequency domain will remain free of aliased signals. Lowpass cut-off frequencies currently utilized by SJM devices coincide with the Nyquist rate requirement.

$$f \ << \frac{1}{2} * f_s$$

Fast Fourier Transform

First published in 1965 by Cooley and Tukey, the FFT revolutionized digital signal analysis by greatly reducing the computing time required to calculate a discrete Fourier transform, an overall improvement from $\Theta(N^2)$ to $\Theta(N \log N)$ (Cooley and Tukey 1965). The FFT moves an input vector into the frequency domain by calculating the discrete Fourier transform of the input vector:

$$X_k = \sum_{n=1}^{N} x(n) e^{-i2\pi(k-1)\frac{(n-1)}{N}} K = 1, \dots, N$$

Matlab was used to calculate an N-point FFT; if the length of the input vector was less than N, the input vector was padded with trailing zeros to length N. If the length of the input vector was greater than N, the input vector was truncated. The function titled

“Positive FFT” located in Appendix A documents the source code utilized by this study to calculate the FFT of a particular vector. The basic steps are summarized in Table 3.

Table 3: Calculating an N-Point FFT for Frequency Domain Analysis

Step 1	Calculate the N-point FFT of an input vector using Matlab’s FFT.
Step 2	Eliminate any complex components by multiplying the output vector by its complex conjugate.
Step 3	Normalize the output vector by dividing each element by the output vector’s spectral sum.
Step 4	Return the new spectral sum, which should be equal to one.

Characterization of Waveforms in the Frequency Domain

Differentiating between waveforms on the atrial and ventricular channels was determined to be the first two steps in pursing an algorithm that utilizes time-frequency analysis. The atrial channel from each dataset was manually sliced into P waves and far-field R waves, and the ventricular channel from each dataset was manually sliced into QRS complexes and T waves. It is important to note that the slices acquired from each channel included the entire signal and represented the entire signal in the frequency domain. Each waveform was then transformed into the frequency domain using a 512-point FFT. Because this analysis was completed offline and because the importance of accuracy surpassed the need for efficiency in these steps, the increased computational expense associated with a 512-point FFT was not a concern. Each of the waveforms (P, far-field R, QRS and T) were “characterized” in the frequency domain by calculating the average spectral amplitude at each frequency. In order to infer whether or not a difference between waveforms on a given channel existed, a metric called total difference was calculated for each channel. This metric was calculated by summing the total area under the spectral curve derived by subtracting one characterized waveform from another on the same channel. For example on the atrial channel:

$$Total\ Difference = \sum \mid \overline{Characterized_{P\ wave}} - \overline{Characterized_{far-field\ R\ wave}} \mid$$

Obtaining a total difference greater than one indicated a significant difference between the two waveforms. Whereas, not achieving a total difference greater than one indicated no significant difference between the two waveforms. Further analysis, was only pursued if the total difference was greater than one.

Dominant Frequency Analysis

DFA exploits generalizations that can be made regarding the dominant frequency of a characterized waveform. As ICEG is sampled by an ICD, DFA windows the incoming signal and performs frequency domain analysis on each window. The dominant frequency is extracted and compared to the dominant frequency of a characterized waveform. If a predetermined threshold is met, that particular window is marked according to the respective waveform. Figure 4 illustrates the program flow for DFA. The key parameters are listed in Table 4. After testing a WINDOW_SIZE of 32, 64 and 128 samples, a WINDOW_SIZE of 64 samples was selected; this selection was based on a compromise between being able to pinpoint when a QRS complex occurs in time and being able to accurately identify a QRS complex. Subsequently, N was chosen to also be 64, maximizing efficiency. Based on the results of the QRS complex characterization, the QRS_F_THRESHOLD was chosen to be 12 Hz, and the STEP_SIZE was set to 24 samples, providing 38% overlap from one window to the next.

Table 4: Key Parameters of DFA Explained

Parameter	Significance
WINDOW_SIZE	The number of samples allocated to each FFT. Increasing this parameter decreases the algorithm's ability to pinpoint when a QRS complex has occurred. Decreasing this parameter reduces the algorithms efficacy in correctly identifying a QRS complex.
N	Number of points used to calculate the FFT. $\Theta(N \log N)$ Powers of 2 increase FFT efficiency and reduce duty cycle. Ideally, N will match WINDOW_SIZE.
QRS_F_THRESHOLD	Determines the spectral energy required to indicate that a QRS complex exists in a particular window.
STEP_SIZE	Defines the number of samples to slide the window forward with each iteration of the algorithm. A STEP_SIZE < WINDOW_SIZE indicates window overlap.

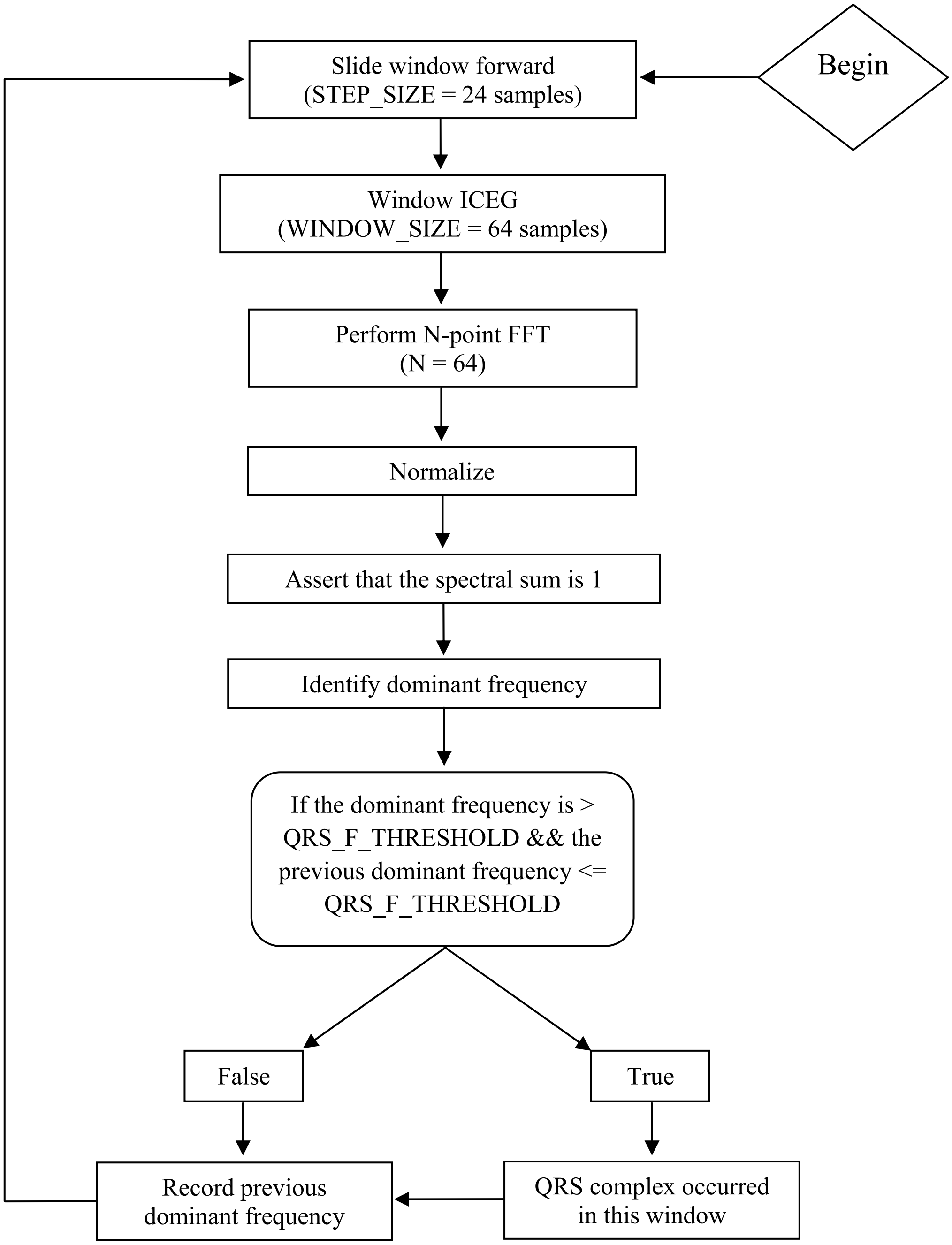
Begin
Slide window forward
(STEP_SIZE = 24 samples)
Window ICEG
(WINDOW_SIZE = 64 samples)
Perform N-point FFT
(N = 64)
Normalize
Assert that the spectral sum is 1
Identify dominant frequency
If the dominant frequency is > QRS_F_THRESHOLD && the previous dominant frequency <= QRS_F_THRESHOLD
False
True
Record previous dominant frequency
QRS complex occurred in this window

Figure 4: Dominant Frequency Analysis (DFA) Program Flow

Comparing Dominant Frequency Analysis

The last step in assessing the feasibility of time-frequency analysis was a comparison of DFA to an algorithm using threshold crossing analysis. Similar to algorithms used by St. Jude Medical, DFA annotates ICEG marking when in time critical events occur. This study specifically compared the accuracy of QRS marker annotation. Difference in QRS annotation was calculated for each sample by subtracting the sample number for a QRS marker identified by DFA from the sample number for a QRS marker identified by threshold crossing analysis. Figure 5 illustrates how the difference in QRS annotation is calculated. The QRS marker for St. Jude Medical's current threshold crossing analysis was determined to be located at sample number 272, while the QRS marker for DFA was determined to be located at sample number 275. In this example, the difference in annotation is 3 samples. This calculation was made for every cardiac cycle (every sample) in each dataset.

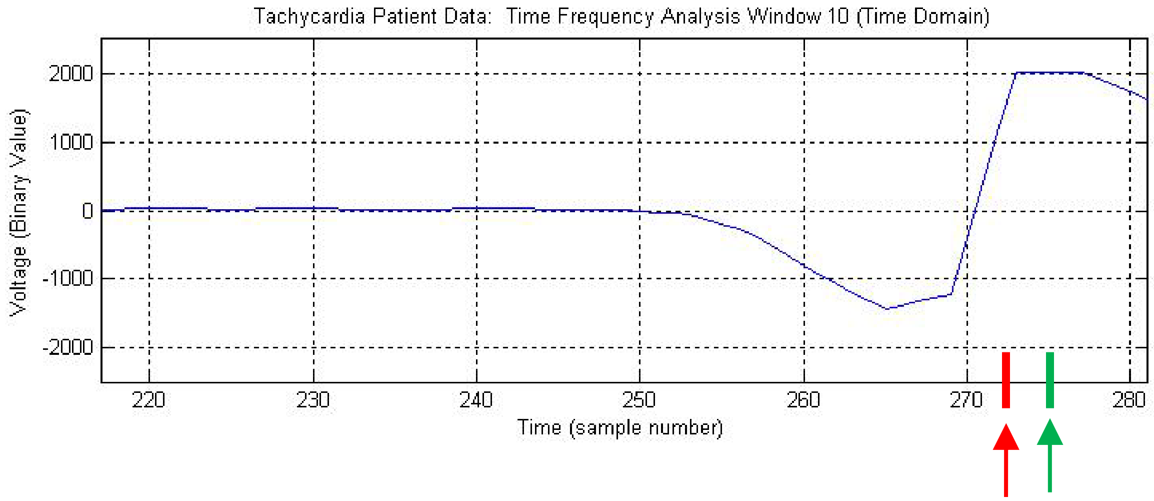

Figure 5: QRS Marker Comparison

DFA QRS Marker: 275

Threshold Crossing Analysis
QRS Marker: 272

RESULTS

P Wave and Far-Field R Wave

Figure 6 shows the characterization of the P wave and far-field R wave in the frequency domain. The spectral density of the P wave is evenly disturbed between 10 and 50 Hz, and the spectral density of the far-field R wave is more concentrated between 13 and 40 Hz. The total difference between the P wave and far-field R wave was 0.4947; because the total difference was less than one, further analysis differentiating the P wave and far-field R wave was not conducted.

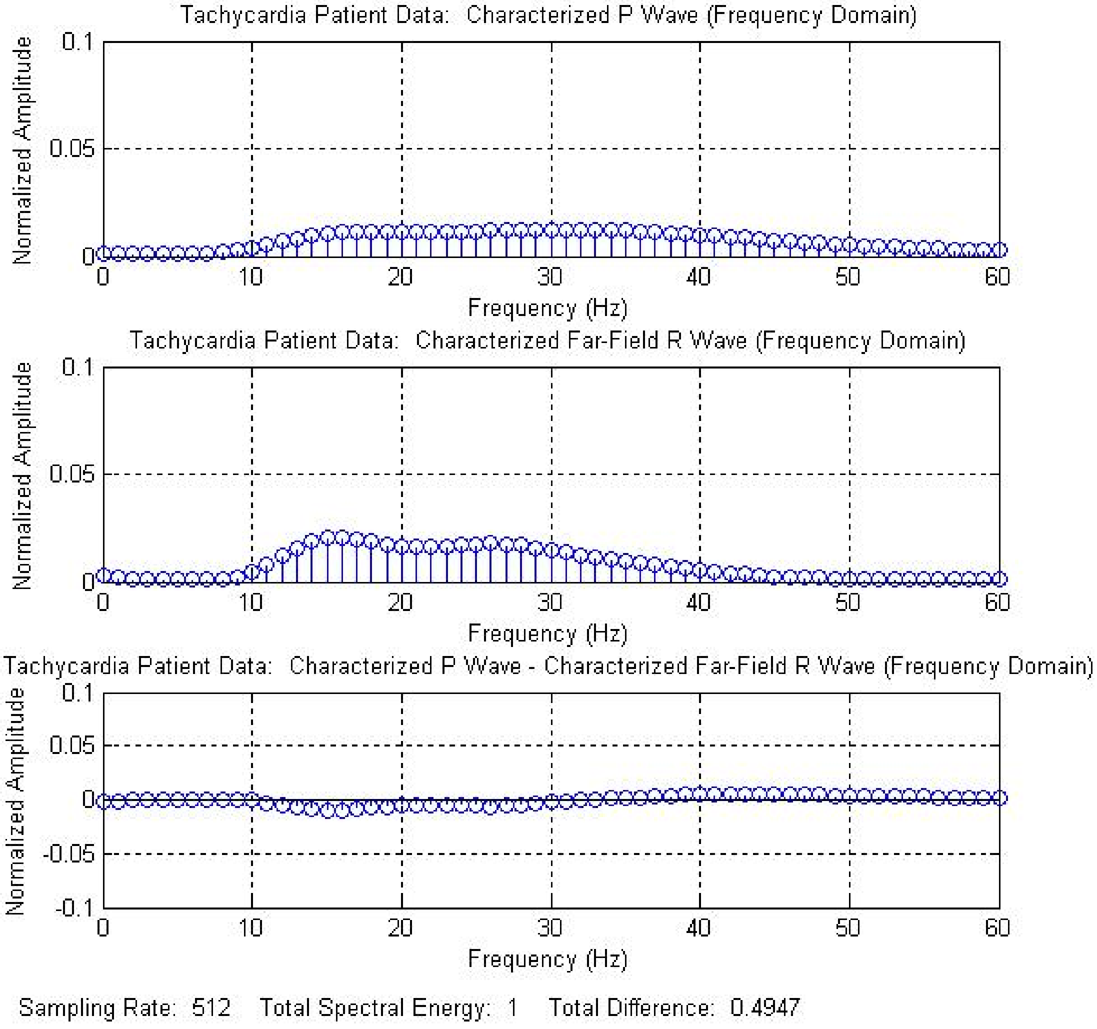

Figure 6: Characterization of the P Wave and the Far-Field R Wave in the Frequency Domain

QRS Complex and T Wave

Figure 7 shows the characterization of the QRS complex and T wave in the frequency domain. The QRS complex is centered around 13 Hz, while the T wave is centered around 7 Hz. The total difference between the QRS complex and T wave was 1.4339; this larger difference supports pursuing DFA.

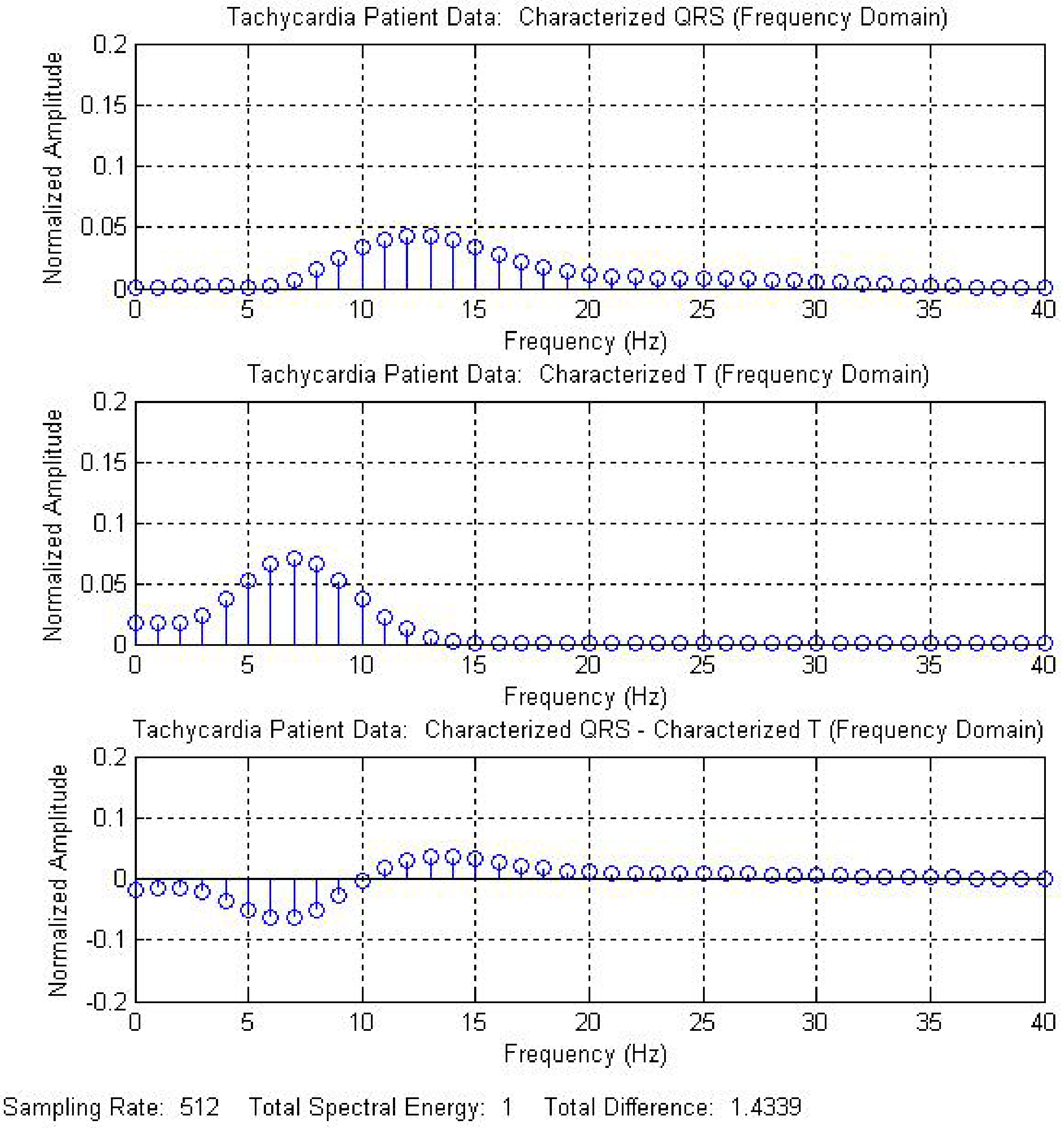

Figure 7: Characterization of the QRS Complex and the T Wave in the Frequency Domain

Qualitative Results for Dominant Frequency Analysis

Figure 8 qualitatively shows a correlation between the identification of the QRS complex and the magnitude of the dominant frequency, both plotted respect to time. The QRS_F_THRESHOLD previously identified by our characterization methods is plotted in red, and the T_F_THRESHOLD also previously identified is plotted in green. Although Figure 8 provides significant support for DFA, it is important to view the results of an empirical comparison.

Dominant Frequency Analysis Comparison

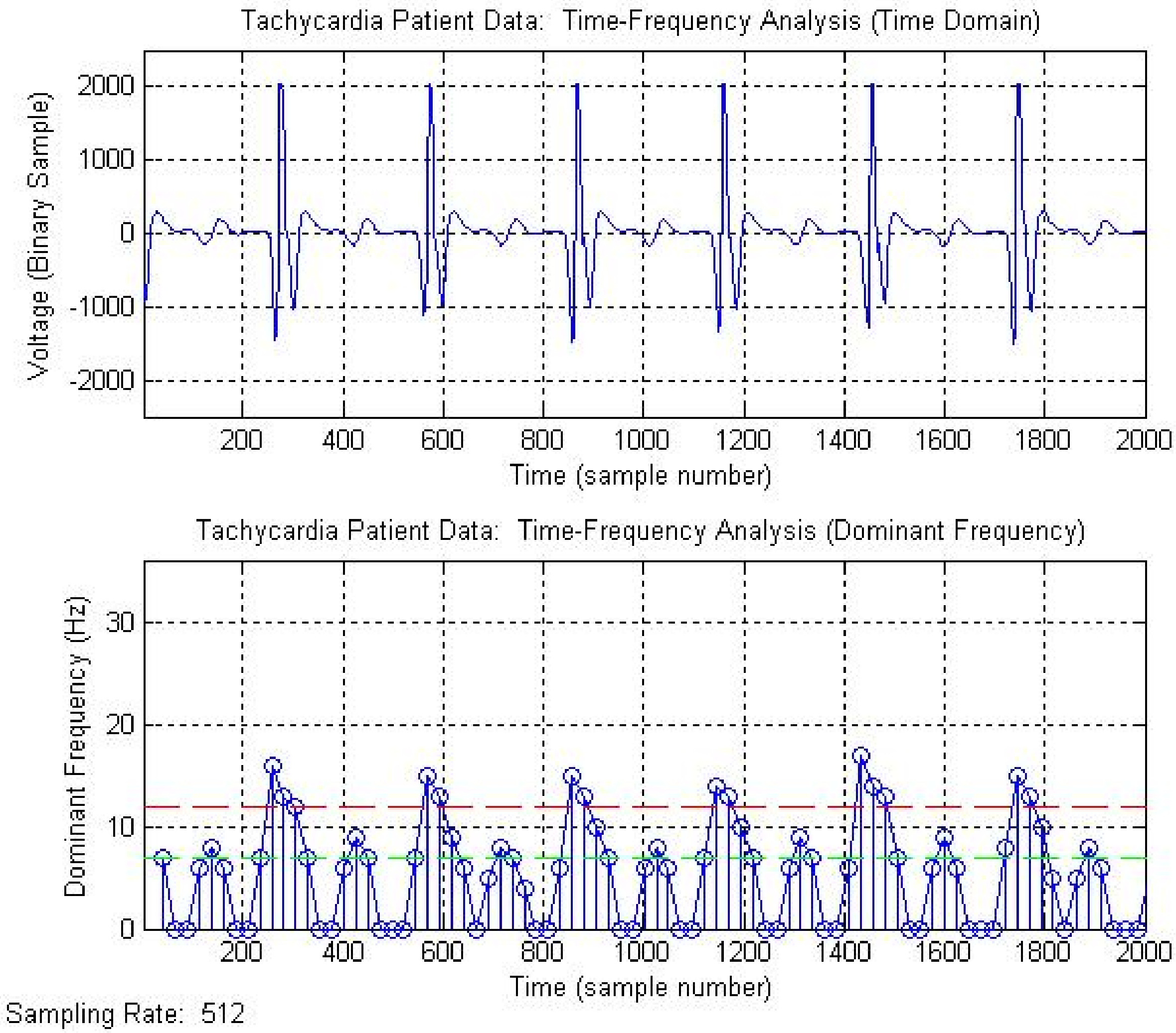

Figure 8: Qualitative Results for DFA

Figure 9 is a histogram compiling the differences in QRS markers for DFA and threshold crossing analysis. The distribution is centered around -6.036 samples with a standard deviation of 8.069. Negative differences between markers indicated that DFA identified a QRS marker at a later point in time compared to threshold crossing analysis; positive differences between markers indicated that DFA identified a QRS marker at an earlier point in time.

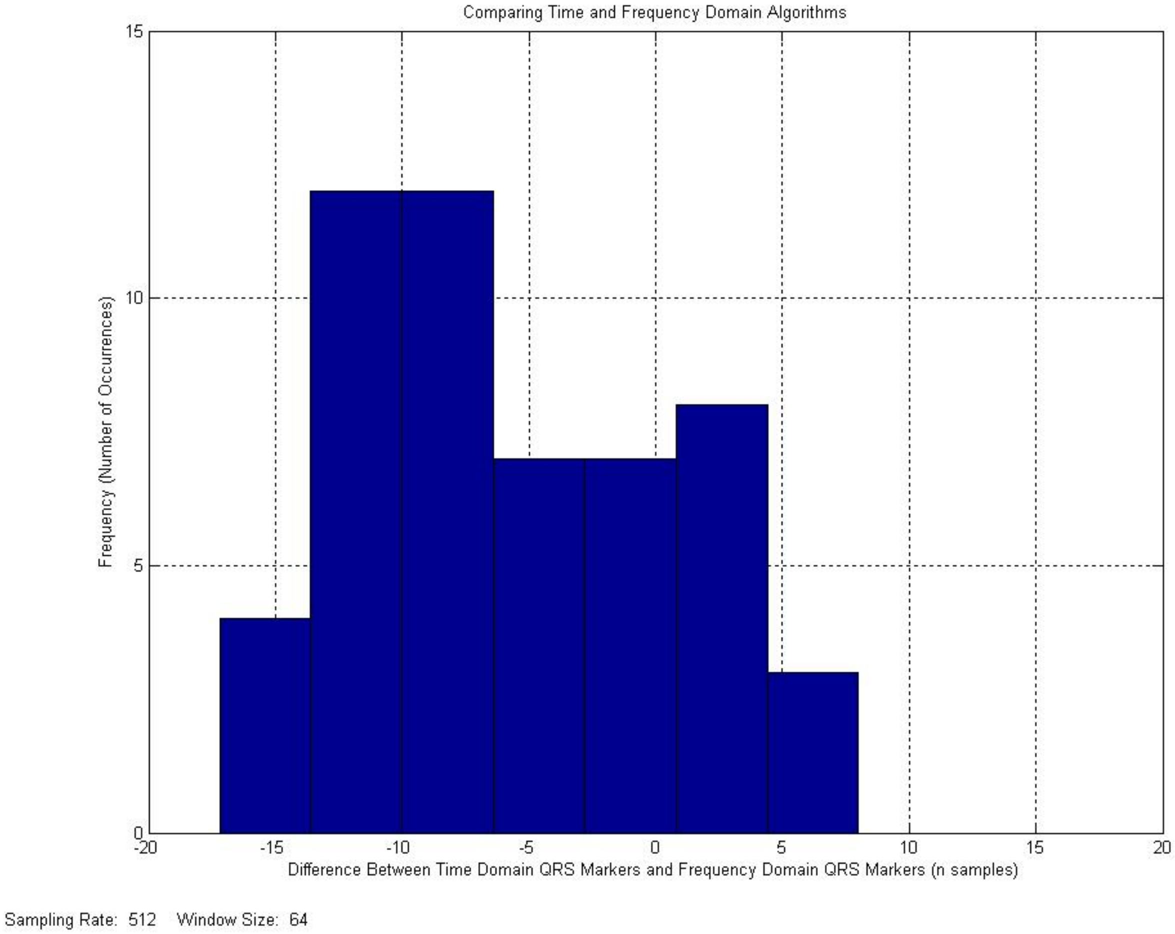

Figure 9: Histogram of Differences in QRS Markers for DFA

While Figure 9 provides an illustration of the results, Table 5 lists the average difference in QRS markers, standard deviation, sample size, and standard error for both datasets. In both datasets all of the QRS markers determined by DFA were within one window size (a parameter of the DFA itself) of the QRS markers determined by threshold crossing analysis.

Table 5: Dominant Frequency Analysis Error

	Average Difference in QRS Markers (number of samples)	Standard Deviation	Sample Size	Standard Error
Dataset 1	-6.036	8.069	55	1.088
Dataset 2	-1.288	11.50	118	1.059

CONCLUSIONS

Summary of Accomplishments

This thesis has explored the use of time-frequency analysis of ICEG by sharing the background supporting its pursuit, developing a method for analyzing ICEG: DFA, and comparing the results of DFA to St. Jude Medical's current threshold crossing analysis. Characterizing, in the frequency domain, the waveforms on the atrial and ventricular channels revealed little to no difference between the P wave and far-field R wave. While there appeared to be little difference on the atrial channel, a considerable difference between the QRS complex and the T wave was established on the ventricular channel. This difference enabled DFA to effectively identify the QRS complex on the ventricular channel for all cardiac cycles in both datasets. Furthermore, this study has shown that time-frequency analysis of ICEG is an equally effective method for identifying the QRS complex on the ventricular channel of an ICD.

Parameters of Dominant Frequency Analysis

In the development of DFA, several key parameters emerged: WINDOW_SIZE, N, QRS_F_THRESHOLD and STEP_SIZE. With respect to the WINDOW_SIZE, a compromise was established between being able to pinpoint when in time the QRS complex occurred and the efficacy in correctly identifying the QRS complex. N, the number of points used to calculate each FFT, greatly affected the computational expense associated with the algorithm. Larger values of N significantly increased the duty cycle required by DFA ($\Theta(N \log N)$), despite the advantages of the FFT. The QRS_F_THRESHOLD differentiates the higher dominant frequencies of the QRS complex from the lower dominant frequencies of the T wave. This study employed a threshold of 12 Hz based on the results of the characterization in the frequency domain. And lastly, the STEP_SIZE defined the degree overlap between windows. Decreasing the STEP_SIZE increased computational expense; however it also qualitatively improved DFA's ability to pinpoint when in time a QRS complex occurred.

Limitations of this Work

The most significant limitation of this work is the fact that the QRS markers determined by DFA were not compared to ground truth, but rather they were compared to QRS markers determined by the currently used threshold crossing analysis. Because of this fact there was no way to conclude that DFA is "better" than threshold crossing analysis, but rather that DFA is equally effective. Another significant limitation of this study was the fact all of the analysis was does offline. Implementing DFA on an actual ICD presents further challenges in itself: implementation of the algorithm in C, the calculation of an FFT on an ICD, and compliance with hardware limitations. Lastly, this study was limited by the number of datasets included. While the two datasets analyzed catered significant support for time-frequency analysis of ICEG, further investigation would benefit from a study including 20 to 50 datasets ranging in heart rates from 90 to 180 beats per minute.

Future Work

The next steps in pursing time-frequency analysis of intracardiac electrogram directly stem from the limitations of this study. The next best step in pursuing time-frequency analysis is the comparison of DFA to ground truth, hopefully including a wider range of patient data (between 20 and 50 datasets, an order of magnitude more than this study utilized). Secondly, the argument for DFA would be strengthened by a study demonstrating that DFA can accurately identify waveforms on an actual ICD platform. Furthermore, pursuing a similar time-frequency algorithm that can differentiate between P waves and far-field R waves will open doors for arrhythmia detection on both the ventricular and atrial channels.

LIST OF REFERENCES

Afonso, V., W. Tompkins, T. Nguyen, K. Michler, and S. Luo. "Comparing Stress ECG Enhancement Algorithms." *Engineering in Medicine and Biology Magazine, IEEE*, 1996: 37-44.

Birnie, D., C. Sambell, H. Johansen, et al. "Use of Implantable Cardioverter Defibrillators in Canadian and US Survivors of Out-of-Hospital Cardiac Arrest." *Canadian Medical Association Journal*, 2007: 41-46.

Collins, R., D. Casavant, et al. "Rate Dependent Far-Field R Wave Sensing in an Atrial Tachyarrhythmia Therapy Device." *Pacing and Clinical Electrophysiology*, 2003: 112-114.

Cooley, J., and J. Tukey. "An Algorithm for the Machine Computation of the Complex Fourier Series." *Mathematics of Computation*, 1965: 297-301.

Crespi, A., S. Somdahl, C. Schmidt, and P. Skarstad. "Evolution of Power Sources for Implantable Cardioverter." *Journal of Power Sources*, 2001: 33-38.

Geerts, Y., M. Steyaert, and W. Sansen. *Design of Mulit-bit Delta-sigma A/D Converters.* Springer, 2002.

Husser, D., M. Stridh, L. Sornmo, and C. Geller. "Time–Frequency Analysis of the Surface Electrocardiogram for Monitoring Antiarrhythmic Drug Effects in Atrial Fibrillation." *The American Journal of Cardiology*, 2005: 526-528.

Jurko, S., G. Rosina, et al. "High Resolution of the ECG Signal by Polynomial Approximation." *Radioengineering*, 2006: 571-579.

Kilpatrick, D., and P. Johnston. "Origin of the Electrocardiogram." *Engineering in Medicine and Biology Magazine, IEEE*, 1994: 479-486.

Mahmoud, S., Q. Fang, et al. "A Time-Frequency Approach for the Analysis of Normal and Arrhythmia Cardiac Signals." *Annual International Conference of the IEEE 2006*, 2002: 2491-2496.

Rubart, M., and D. Zipes. "Mechanisms of Sudden Cardiac Death." *Science in Medicine*, 2005: 2305-2315.

Sornmo, L., et al. "Time-Varying Digital Filtering of ECG Baseline Wanter." *Medical and Biological Engineering and Computing*, 1993: 503-508.

Thakor, N., Y. Zhu, et al. "Applications of Adaptive Filtering to ECG Analysis: Noise Cancellation and Arrhythmia Detection." *IEEE Transactions on Biomedical Engineering*, 1991: 785-794.

Theres, H., W. Sun W, et al. "P Wave and Far-Field R Wave Detection in Pacemaker Patient Atrial Electrograms." *Pacing and Clinical Electrophysiology*, 2006: 434-440.

Tsipouras, M., and D. Fotiadis. "Automatic Arrhythmia Detection based on Time and Time-Frequency Analysis of Heart Rate Variability." *Computer Methods and Programs in Biomedicine*, 2003: 95-108.

Unterkofler, K., A. Amann, and R. Tratnig. "Detecting Ventricular Fibrillation by Time-Delay Methods." *IEEE Transactions on Biomedical Engineering*, 2007: 174-177.

Wang, H., interview by E. Brockman. *Senior Software Engineer, Cardiac Rhythm Management Division, St. Jude Medical* (2009).

Widmaier, E., H. Raff, and K. Strang. *Vander's Human Physiology: The Mechanisms of Body Function.* McGraw-Hill, 2008.

Zheng, Z., J. Croft, W. Giles, and G. Mensah. "Sudden Cardiac Death in the United States, 1989 to 1998." *Circulation*, 2001: 2158-2163.

APPENDIX A: MATLAB CODE

The following Matlab code was utilized throughout the completion of this thesis:

```
%% SJM Meditec: Debug Function
%-----------------------------
% Author:        Erik Brockman
% Last Updated:  9/8/08
%-----------------------------

% arguments: string debugMessage
% returns:   void

function debug (debugMessage)

    global DEBUG_GLOBAL;

    if (DEBUG_GLOBAL == 1)
        disp (sprintf ('%s%s', 'debug: ', debugMessage));
    end
end

% END
```

```
%% SJM Meditec: Positive FFT
%----------------------------
% Author:       Erik Brockman
% Last Updated: 2/23/09
%----------------------------

% arguments: vector xTimeDomain
% returns:   vector XFrequencyDomain (y-axis)
% returns:   variable totalSpectralEnergy (should be equal to 1)

function [XFrequencyDomain, totalSpectralEnergy] = positiveFFT (xTimeDomain)

global N; % Calculate N-point FFT

debug('begin positive fft');

XFrequencyDomain = fft(xTimeDomain, N); % NOT normalized

XFrequencyDomain = XFrequencyDomain.*conj(XFrequencyDomain); % Eliminate complex nubmers
XFrequencyDomain = XFrequencyDomain / sum(XFrequencyDomain); % Normalize

totalSpectralEnergy = sum(XFrequencyDomain); % Sum should return 1

debug('end positive fft');

end

% END
```

```
%% SJM Meditec: Milestone 2
%----------------------------
% Author:       Erik Brockman
%----------------------------

% This meditec project was broken into milestones.  This particular
%  file served as the executable and is the milestone that includes the
%  time-frequency analysis described throughout this thesis.

% Global Variables
global DEBUG_GLOBAL; % 1 = Debug mode
DEBUG_GLOBAL = 1;
global PLOTS_ON; % 1 = Plot all data points
PLOTS_ON = 0;
global LAST_UPDATED;
LAST_UPDATED = '5/10/09';
global PATIENT_DATA_FILENAME;
PATIENT_DATA_FILENAME = 'data/sample1.csv';
global SLICES_FILENAME;
SLICES_FILENAME = 'data/sample1Slices.csv';
global SAMPLING_RATE; % Depends on data
SAMPLING_RATE = 512;

% FFT Global Variables
global N; % Calculate an N-point FFT
N = 64;
global CUT_OFF; % Cut off frequency for plotting
CUT_OFF = 40;
global TOTAL_CARDIAC_CYCLES;
TOTAL_CARDIAC_CYCLES; % Initialized at later point

% Time Frequency Global Variables
global WINDOW_SIZE; % Time frequency window size
WINDOW_SIZE = 64;
global STEP; % Time frequency window shifts forward by this number of samples
STEP = 24;
global QRS_F_THRESHOLD; % QRS frequency threshold for time frequency algorithm
QRS_F_THRESHOLD = 12;
global T_F_THRESHOLD; % T frequency threshold for time frequency algorithm
T_F_THRESHOLD = 7;

disp(sprintf('%s',   'SJM Meditec: Milestone 2'));
disp(sprintf('%s',   'SJM Meditec: QRS VS T'));
disp(sprintf('%s',   '------------------------------------'));
disp(sprintf('%s',   'Author:       Erik Brockman'));
disp(sprintf('%s%s', 'Last Updated: ', LAST_UPDATED));
disp(sprintf('%s%s', 'Data File:    ', PATIENT_DATA_FILENAME));
disp(sprintf('%s%s', 'Data File:    ', SLICES_FILENAME));
disp(sprintf('%s',   '------------------------------------'));
disp(sprintf('%s',   '--------------- BEGIN --------------'));
disp(sprintf('\n'));

debug(['using file ' PATIENT_DATA_FILENAME]);

% Read in data from file located at PATIENT_DATA_FILENAME
% Note that the header and the columns other than asense and vsense are
%  removed
debug('begin reading input');
inputVector = csvread(PATIENT_DATA_FILENAME, 0, 0);
debug('end reading input');

% Extract aSense and vSense vectors
sampleNumber = inputVector(:, 1); % Capture the sample numbers
aSense = inputVector(:, 2);       % aSense is a vector in the time domain
vSense = inputVector(:, 3);       % vSense is a vector in the time domain

% Extract the time domain QRS markers
QRSMarkers = [inputVector(:, 1), inputVector(:, 5)]; % Time domain algorithm marks these
                                                     %  events as R waves

% Read in time slices from file located at SLICES_FILENAME
% Note that this file is created by hand
debug('begin reading slices');
slices = csvread(SLICES_FILENAME);
debug('end reading slices');

TOTAL_CARDIAC_CYCLES = length(slices); % Initialize global variable

QRSBeginSlice = slices(:, 1); % Beginning index of the QRS vector to parse
QRSEndSlice = slices(:, 2);   % Ending index of the QRS vector to parse

TBeginSlice = slices(:, 3); % Beginning index of the T vector to parse
TEndSlice = slices(:, 4);   % Ending index of the T vector to parse

% Plot the data in the time domain
debug('begin plotting data in the time domain');

screenSize = get(0, 'ScreenSize'); % Set figure sizes
left = 40;
bottom = 200;
width = screenSize(3) / 1.5;
height = screenSize(4) / 1.5;

timeFigure = figure;
set(timeFigure, 'NumberTitle', 'off');
set(timeFigure, 'Name', ['Plotting ECG In The Time Domain:  ' PATIENT_DATA_FILENAME]);
set(timeFigure, 'OuterPosition', [left, bottom, screenSize(3) / 1.25, screenSize(4) /
 1.25;]);
set(timeFigure, 'Color', [1, 1, 1]);

% Display the sampling rate
legend = uicontrol('style', 'text');
set(legend, 'String', ['Sampling Rate:  ' num2str(SAMPLING_RATE)]);
set(legend, 'FontSize', 10, 'Position', [8 8 130 18]);
set(legend, 'BackgroundColor', 'white');

subplot(2, 1, 1);
plot(aSense);
axis([0, length(aSense), -2000, 2000]); % +/- 2000 mV on the y-axis
title('Tachycardia Patient Data:  Atrial Channel (Time Domain)');
xlabel('Time (n samples)');
ylabel('Voltage (Binary Value)');
grid;

subplot(2, 1, 2);
plot(vSense);
axis([0, length(vSense), -2500, 2500]); % +/- 2500 mV on the y-axis
title('Tachycardia Patient Data:  Ventricular Channel (Time Domain)');
xlabel('Time (n samples)');
ylabel('Voltage (Binary Value)');
grid;

debug('end plotting data in the time domain');

debug('begin calculating ffts');

% Declare the QRS and T vectors to increase efficiency
QRS = zeros(TOTAL_CARDIAC_CYCLES, N);
QRSSpectralSum = zeros(TOTAL_CARDIAC_CYCLES, 1);
T = zeros(TOTAL_CARDIAC_CYCLES, N);
TSpectralSum = zeros(TOTAL_CARDIAC_CYCLES, 1);

% The following loop builds two matrices in which each row is a vector in
%  the frequency domain.  There will be TOTAL_CARDIAC_CYCLES number of rows.
%  For QRS and T vectors use the vSense to parse out the vectors
for index = 1 : TOTAL_CARDIAC_CYCLES

    % Use a positive fast fourier transform to build index number of QRS plots
    [QRSOutput, QRSSpectralSumOutput] =
     positiveFFT(vSense(QRSBeginSlice(index):QRSEndSlice(index)));
    QRS(index, :) = QRSOutput;
    QRSSpectralSum(index) = QRSSpectralSumOutput;
    % Use a positive fast Fourier transform to build index number of T plots
    [TOutput, TSpectralSumOutput] =
     positiveFFT(vSense(TBeginSlice(index):TEndSlice(index)));
    T(index, :) = TOutput;
    TSpectralSum(index) = TSpectralSumOutput;

end

debug('end calculating ffts');

debug('begin calculating difference vectors');

% Declare the diffSense vector to increase efficiency
diffSense = zeros(TOTAL_CARDIAC_CYCLES, N);

for index = 1 : TOTAL_CARDIAC_CYCLES

    % Calculate index difference vectors
    diffSense(index, :) = abs(QRS(index, :) - T(index, :));

end

debug('end calculating difference vectors');

debug('begin plotting data in the frequency domain');

frequency = (0 : N - 1) / N * SAMPLING_RATE; % Frequency range

for index = 1 : TOTAL_CARDIAC_CYCLES
    if PLOTS_ON == 1
        frequencyFigure = figure;
        set(frequencyFigure, 'NumberTitle', 'off');
        set(frequencyFigure, 'Name', ['Plotting ECG In The Frequency Domain:  '
         PATIENT_DATA_FILENAME]);
        left = left + 20;
        bottom = bottom - 20;
        set(frequencyFigure, 'OuterPosition', [left, bottom, width, height]);

        % Display critical values
        legend = uicontrol('style', 'text');
        set(legend, 'String', ['Sampling Rate:  ' num2str(SAMPLING_RATE) '    Total
         Spectral Energy:  ' num2str(QRSSpectralSum(index)) '    Total Difference:  '
         num2str(sum(diffSense(index, :)))]);
        set(legend, 'FontSize', 10, 'Position', [8 8 440 18]);
        set(legend, 'BackgroundColor', 'white');

        subplot(3, 1, 1);
        stem(frequency, QRS(index, :));
        axis([0, CUT_OFF, 0, 0.2]);
        title(['Tachycardia Patient Data:  QRS' num2str(index) ' (Frequency Domain)']);
        xlabel('Frequency (Hz)');
        ylabel('Normalized Amplitude');
        grid;

        subplot(3, 1, 2);
        stem(frequency, T(index, :));
        axis([0, CUT_OFF, 0, 0.2]);
        title(['Tachycardia Patient Data:  T' num2str(index) ' (Frequency Domain)']);
        xlabel('Frequency (Hz)');
        ylabel('Normalized Amplitude');
        grid;

        subplot(3, 1, 3);
        stem(frequency, diffSense(index, :));
        axis([0, CUT_OFF, 0, 0.2]);
        title(['Tachycardia Patient Data:  |QRS' num2str(index) ' - T' num2str(index) '|
         (Frequency Domain)']);
        xlabel('Frequency (Hz)');
        ylabel('Normalized Amplitude');
        grid;
    end
end

debug('end plotting data in the frequency domain');

debug('begin characterizing the QRS and T vectors');

% To characterize the QRS vector, simply sum each column in QRS and
%  divide by TOTAL_CARDIAC_CYCLES
charQRS = zeros(1, N);

for index = 1 : N

    charQRS(index) = sum(QRS(:, index)) / TOTAL_CARDIAC_CYCLES;

end

% To characterize the T vector, simply sum each column in T and
%  divide by TOTAL_CARDIAC_CYCLES
charT = zeros(1, N);

for index = 1 : N

    charT(index) = sum(T(:, index)) / TOTAL_CARDIAC_CYCLES;

end

diffCharSense = charQRS - charT;

frequencyFigure = figure;
set(frequencyFigure, 'NumberTitle', 'off');
set(frequencyFigure, 'Name', ['QRS and T Characterization:  ' PATIENT_DATA_FILENAME]);
set(frequencyFigure, 'Color', [1, 1, 1]);
left = left + 20;
bottom = bottom - 20;
set(frequencyFigure, 'OuterPosition', [left, bottom, screenSize(3) / 1.25, screenSize(4)
 / 1.25]);

legend = uicontrol('style', 'text');
set(legend, 'String', ['Sampling Rate:  ' num2str(SAMPLING_RATE) '    Total Spectral
 Energy:  ' num2str(sum(charQRS)) '    Total Difference:  '
num2str(sum(abs(diffCharSense)))]);
set(legend, 'FontSize', 10, 'Position', [8 8 440 18]);
set(legend, 'BackgroundColor', 'white');

subplot(3, 1, 1);
stem(frequency, charQRS);
axis([0, CUT_OFF, 0, 0.2]);
title('Tachycardia Patient Data:  Characterized QRS (Frequency Domain)');
xlabel('Frequency (Hz)');
ylabel('Normalized Amplitude');
grid;

subplot(3, 1, 2);
stem(frequency, charT);
axis([0, CUT_OFF, 0, 0.2]);
title('Tachycardia Patient Data:  Characterized T (Frequency Domain)');
xlabel('Frequency (Hz)');
ylabel('Normalized Amplitude');
grid;

subplot(3, 1, 3);
stem(frequency, diffCharSense);
axis([0, CUT_OFF, -0.2, 0.2]);
title('Tachycardia Patient Data:  Characterized QRS - Characterized T (Frequency
 Domain)');
xlabel('Frequency (Hz)');
ylabel('Normalized Amplitude');
grid;

debug('end characterizing the QRS and T vectors');
debug('begin time frequency analysis');

startSample = 1;
FFTHead = startSample;               % Start the head at the first sample
FFTTail = startSample + WINDOW_SIZE; % Start the tail at the first sample plus the window
                                     %  size
stopSample = length(vSense);         % Stop sample

numberOfPlots = ceil((stopSample - FFTTail) / STEP); % Calculate number of plots
dominantFrequency = zeros(numberOfPlots, 3);
previousDominantFrequency = 0; % Initialize the previous dominant frequency variable

index = 1; % Keep track of plot numbers

% The following loop steps through cardiac cycles (beginning and ending
%  according to FFTHead and stopSample, respectively); it calculates an
%  N-point FFT on the window defined by FFTHead and FFTTail.  In each
%  iteration the window is incremented by STEP.  This simulates the time
%  frequency algorithm for eight cardiac cycles.

while FFTTail < stopSample

    [FFTOutput, FFTSpectralSumOutput] = positiveFFT(vSense(FFTHead:FFTTail));

    [temp1, temp2] = max(FFTOutput); % Find the frequency (and normalized amplitude) of
                                     %  the maximal specral line in the FFT

    % Record the FFTTail (sample number that identifies a particular
    %  window), the frequency of the maximal specral line in the FFT, and
    %  the normalized amplitude of the maximal spectral line.
    dominantFrequency(index, :) = [(FFTHead + 0.75 * STEP), ((temp2 - 1) / N *
     SAMPLING_RATE), temp1];

    % If the following conditional is true then we have identified a QRS
    %  complex using time frequency analysis.
    if previousDominantFrequency <= QRS_F_THRESHOLD && dominantFrequency(index, 2) >
     QRS_F_THRESHOLD
        QRSMarkers(dominantFrequency(index, 1), 3) = 1;
    end

    previousDominantFrequency = dominantFrequency(index, 2);

    if PLOTS_ON == 1
        timeFrequencyFigure = figure;
        set(timeFrequencyFigure, 'NumberTitle', 'off');
        set(timeFrequencyFigure, 'Name', ['QRS and T Time Frequency Analysis Window '
         num2str(index) ':  ' PATIENT_DATA_FILENAME]);
        left = left + 20;
        bottom = bottom - 20;
        set(timeFrequencyFigure, 'OuterPosition', [left, bottom, screenSize(3) / 1.5,
         screenSize(4) / 1.5]);

        legend = uicontrol('style', 'text');
        set(legend, 'String', ['Sampling Rate:  ' num2str(SAMPLING_RATE) '    Dominant
         Frequency:  ' num2str(dominantFrequency(index, 2)) ' Hz    Max Amplitutde:  '
         num2str(dominantFrequency(index, 3))]);
        set(legend, 'FontSize', 10, 'Position', [8 8 460 18]);
        set(legend, 'BackgroundColor', 'white');

        subplot(2, 1, 1);
        plot(vSense);
        axis([FFTHead, FFTTail, -2500, 2500]); % +/- 2500 mV on the y-axis
        title(['Tachycardia Patient Data:  Time Frequency Analysis Window '
         num2str(index) ' (Time Domain)']);
        xlabel('Time (n samples)');
        ylabel('Voltage (Binary Sample)');
        grid;

        subplot(2, 1, 2);
        stem(frequency, FFTOutput);
        axis([0, CUT_OFF, 0, 0.2]);
        title(['Tachycardia Patient Data:  Time Frequency Analysis Window '
         num2str(index) ' (Frequency Domain)']);
        xlabel('Frequency (Hz)');
        ylabel('Normalized Amplitude');
        grid;
    end

    FFTHead = FFTHead + STEP; % Move window forward by STEP
    FFTTail = FFTTail + STEP;
    index = index + 1; % Increment plot number
end

% Plot the recorded maximal spectral line of each window versus time.  %  This
%  serves as simplified spectrogram.

timeFrequencyFigure = figure;
set(timeFrequencyFigure, 'NumberTitle', 'off');
set(timeFrequencyFigure, 'Name', ['QRS and T Time Frequency Analysis:  '
 PATIENT_DATA_FILENAME]);
set(timeFrequencyFigure, 'Color', [1, 1, 1]);
left = left + 20;
bottom = bottom - 20;
set(timeFrequencyFigure, 'OuterPosition', [left, bottom, screenSize(3) / 1.25,
 screenSize(4) / 1.25]);

legend = uicontrol('style', 'text');
set(legend, 'String', ['Sampling Rate:  ' num2str(SAMPLING_RATE)]);
set(legend, 'FontSize', 10, 'Position', [8 8 130 18]);
set(legend, 'BackgroundColor', 'white');

subplot(2, 1, 1);
plot(vSense);
axis([startSample, stopSample, -2500, 2500]); % +/- 2500 mV on the y-axis
title('Tachycardia Patient Data:  Time Frequency Analysis (Time Domain)');
xlabel('Time (n samples)');
ylabel('Voltage (Binary Sample)');
grid;

subplot(2, 1, 2);
stem(dominantFrequency(:, 1), dominantFrequency(:, 2));
axis([startSample, stopSample, 0, 36]);
title('Tachycardia Patient Data:  Time Frequency Analysis (Dominant Frequency)');
xlabel('Time (n samples)');
ylabel('Dominant Frequency (Hz)');
grid;

hold on
plot([startSample stopSample], [QRS_F_THRESHOLD, QRS_F_THRESHOLD], '--r');
plot([startSample stopSample], [T_F_THRESHOLD, T_F_THRESHOLD], '--g');
hold off

debug('end time frequency analysis');

debug('begin comparing QRS markers');

timeDomainQRSMarkers = zeros(TOTAL_CARDIAC_CYCLES, 1);
frequencyDomainQRSMarkers = zeros(TOTAL_CARDIAC_CYCLES, 1);
diffQRSMarkers = zeros(TOTAL_CARDIAC_CYCLES, 3);

index2 = 1; % We need another index to track respective QRS markers

for index = 1 : length(QRSMarkers)

    if QRSMarkers(index, 2) == 1
        timeDomainQRSMarkers(index2) = index;
        index2 = index2 + 1;
    end
end

index2 = 1; % Reset second index
```

```
for index = 1 : length(QRSMarkers)
    if QRSMarkers(index, 3) == 1
        frequencyDomainQRSMarkers(index2) = index;
        index2 = index2 + 1;
    end
end

% Calculate the difference in the number of samples between the
%   timeDomainQRSMarkers and the frequencyDomainQRSMarkers for each cardiac
%   cycle.
for index = 1 : TOTAL_CARDIAC_CYCLES

    diffQRSMarkers(index, :) = [timeDomainQRSMarkers(index),
     frequencyDomainQRSMarkers(index), (timeDomainQRSMarkers(index) –
     frequencyDomainQRSMarkers(index))];
end

histogramFigure = figure;
set(histogramFigure, 'NumberTitle', 'off');
set(histogramFigure, 'Name', ['QRS and T Histogram:  ' PATIENT_DATA_FILENAME]);
set(histogramFigure, 'Color', [1, 1, 1]);
left = left + 20;
bottom = bottom - 20;
set(histogramFigure, 'OuterPosition', [left, bottom, screenSize(3) / 1.25, screenSize(4)
 / 1.25]);

legend = uicontrol('style', 'text');
set(legend, 'String', ['Sampling Rate:  ' num2str(SAMPLING_RATE) '    Window Size:  '
 num2str(WINDOW_SIZE)]);
set(legend, 'FontSize', 10, 'Position', [8 8 260 18]);
set(legend, 'BackgroundColor', 'white');

subplot(1, 1, 1);
hist(diffQRSMarkers(:, 3));
axis([-20, 40, 0, 30]);
title('Comparing Time and Frequency Domain Algorithms');
xlabel('Difference Between Time Domain QRS Markers and Frequency Domain QRS Markers (n
 samples)');
ylabel('Frequency');

debug('end comparing QRS markers');

disp(sprintf('\n'));
disp(sprintf('%s',   '---------------- END ---------------'));
disp(sprintf('%s',   '------------------------------------'));

% END
```

APPENDIX B: SAMPLE RUNTIME OUTPUT

```
SJM Meditec: Milestone 2
SJM Meditec: QRS VS T
------------------------------------
Author:       Erik Brockman
Last Updated: 5/10/09
Data File:    data/sample1.csv
Data File:    data/sample1Slices.csv
------------------------------------
--------------- BEGIN --------------


debug: using file data/sample1.csv
debug: begin reading input
debug: end reading input
debug: begin reading slices
debug: end reading slices
debug: begin plotting data in the time domain
debug: end plotting data in the time domain
debug: begin calculating ffts
debug: begin positive fft
debug: end positive fft
...
debug: begin positive fft
debug: end positive fft
debug: end calculating ffts
debug: begin calculating difference vectors
debug: end calculating difference vectors
debug: begin plotting data in the frequency domain
debug: end plotting data in the frequency domain
debug: begin characterizing the QRS and T vectors
debug: end characterizing the QRS and T vectors
debug: begin time frequency analysis
debug: begin positive fft
debug: end positive fft
...
debug: begin positive fft
debug: end positive fft
debug: end time frequency analysis
debug: begin comparing QRS markers
debug: end comparing QRS markers


---------------- END ---------------
------------------------------------
```