

1-1-2009

# Ventricular fibrillation detection algorithm for automated external defibrillators

Marjan M. Kusha  
*Ryerson University*

Follow this and additional works at: <http://digitalcommons.ryerson.ca/dissertations>



Part of the [Electrical and Computer Engineering Commons](#)

---

## Recommended Citation

Kusha, Marjan M., "Ventricular fibrillation detection algorithm for automated external defibrillators" (2009). *Theses and dissertations*. Paper 892.

81930 6313

RC  
683.5  
K87  
2009

# VENTRICULAR FIBRILLATION DETECTION ALGORITHM FOR AUTOMATED EXTERNAL DEFIBRILLATORS

by

Marjan M.Kusha  
B.Eng, Electrical Engineering  
Ryerson University, Toronto, 2007

A thesis  
presented to Ryerson University  
in partial fulfillment of the  
requirement for the degree of  
Master of Engineering  
in the Program of  
Electrical and Computer Engineering.

Toronto, Ontario, Canada, 2009

© Marjan M.Kusha, 2009

## Author's Declaration

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Signature



I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Signature



# **Abstract**

## **VENTRICULAR FIBRILLATION DETECTION ALGORITHM FOR AUTOMATED EXTERNAL DEFIBRILLATORS**

**Marjan M.Kusha**

**Master of Engineering**

**Department of Electrical and Computer Engineering**

**Ryerson University, 2009**

The automatic external defibrillator (AED) is a lifesaving device, which processes and analyses the electrocardiogram (ECG) and prompts a defibrillation shock if ventricular fibrillation is determined. This project investigates the possibility of developing a ventricular Fibrillation (VF) detection algorithm based on Autoregressive Modeling (AR Modeling) and dominant poles for the use in AEDs. In particular, the ECG segment is modeled using AR modeling and the dominant poles are extracted from the model transfer function. The dominant pole frequencies were then used in classification based on the distance measure. The potential use of this method to distinguish between VF and Normal sinus rhythm (NSR) is discussed. The method was tested with ECG records from the widely recognized databases of American Heart Association (AHA) and the Creighton University (CU). Sensitivity and specificity for the new VF detection method were calculated to be 66% and 94% respectively. The proposed method has some advantages over other existing VF detection algorithms; it has a high detection accuracy, it is computationally inexpensive and can be easily implemented in hardware.



## Acknowledgments

I would like to acknowledge my family and Mehdi Rafie for their continuous support and encouragement, which gave me the ability and courage to pursue my goals with confidence. In addition, I would like to thank my supervisor, Dr. Sri Krishnan, for his support and mentoring throughout my undergraduate and graduate experiences.

Also, thanks to my fellow colleagues and friends from the Signal Analysis Research (SAR) group for creating a good spirit in the lab and making this experience unforgettable.

Last but not least, I would like to thank everyone at Ryerson University who provided me with an enhanced educational experience, in which I will never forget.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Characteristics of ECG Signal . . . . .	2
1.2	AED . . . . .	4
1.2.1	External influences on AEDs . . . . .	6
1.3	Measure of Diagnostic Accuracy . . . . .	6
1.3.1	Sensitivity . . . . .	7
1.3.2	Specificity . . . . .	7
1.3.3	Receiver Operating Characteristic . . . . .	7
1.4	AED Hardware Implication . . . . .	8
1.5	Organization of the report . . . . .	10
<b>2</b>	<b>Literature Review</b>	<b>12</b>
2.1	Ventricular Fibrillation Detection By Adaptive Algorithm . . . . .	13
2.2	Ventricular Fibrillation Detection Using Pseudo Wigner-Ville Distribution .	14
2.3	Hilbert Transform . . . . .	16
2.4	Generalized Linear Method . . . . .	17
<b>3</b>	<b>Methods</b>	<b>19</b>
3.1	ECG Recordings . . . . .	20
3.1.1	Data Libraries . . . . .	20
3.1.2	Database Analysis . . . . .	21
3.1.3	Preprocessing . . . . .	21

3.2	Autoregressive Modeling . . . . .	24
3.3	Dominant Poles . . . . .	24
3.3.1	Classification . . . . .	26
3.3.2	Implementation and Result . . . . .	26
3.3.3	Discussion and Future work . . . . .	31
<b>4</b>	<b>MATLAB based FPGA design</b>	<b>32</b>
4.1	Hardware Description Language . . . . .	34
4.1.1	Levels of representation and abstraction . . . . .	34
4.2	Converting MATLAB floating point computations to fixed point . . . . .	36
4.2.1	Numeric representations . . . . .	36
4.2.2	Fixed point modeling in MATLAB . . . . .	37
4.3	Techniques for translating MATLAB/Simulink to RTL . . . . .	38
4.3.1	Language translation: Hand translation . . . . .	38
4.3.2	Language translation: Auto interactive translation . . . . .	39
4.3.3	Language translation: MATLAB/Simulink $\rightarrow$ C/C++ $\rightarrow$ RTL . . . . .	40
4.3.4	IP Instantiation . . . . .	40
<b>5</b>	<b>Conclusions</b>	<b>42</b>
	<b>Bibliography</b>	<b>43</b>

# List of Tables

3.1	Signals in the dataset . . . . .	21
3.2	Classification result . . . . .	30

# List of Figures

1.1	Annual Deaths in USA . . . . .	2
1.2	Early defibrillator . . . . .	3
1.3	Automated External Defibrillator . . . . .	3
1.4	ECG tracing of normal cardiac cycle . . . . .	4
1.5	ECG segment during VF . . . . .	5
1.6	Survival rate as a function of time . . . . .	6
1.7	Receiver Operating Characteristic . . . . .	8
1.8	Overall block diagram of AED . . . . .	9
1.9	Organization of the project report . . . . .	11
2.1	Schematic of adaptive filter . . . . .	13
2.2	Distribution of EE . . . . .	14
2.3	Box-Plot results for RatioSTD parameter . . . . .	16
2.4	ECG signals (left) and their corresponding phase-space plots (right) . . . . .	17
3.1	Proposed method . . . . .	20
3.2	Red-Spectrum of ECG segment before filtering, . . . . .	22
3.3	Each segment consists of at least one cardiac cycle . . . . .	23
3.4	a) Spectrum of normal ECG segment b) Corresponding model spectrum (p=8)	25
3.5	a)Spectrum of VF ECG segment b) Corresponding model spectrum (p=8) .	25
3.6	Distribution of the Poles in Normal ECG Segment . . . . .	27
3.7	Distribution of the Poles during VF . . . . .	28
3.8	Classification result for NSR . . . . .	29

3.9	Classification result for VF	29
3.10	Receiver operator characteristic of the proposed method	30
4.1	FPGA design flow for MATLAB based DSP algorithms	33
4.2	Levels of abstraction: Behavioral, Structural and Physical	35
4.3	Fixed point and floating point representation	37
4.4	Hand translation of MATLAB/Simulink to RTL	39
4.5	IP instantiation and netlisting	41

# List of Acronyms

AED - Automated External Defibrillator  
AHA - American Heart Association  
AR - Auto Regressive  
CPR - Cardiopulmonary Resuscitation  
DSP - Digital Signal Processing  
ECG - Electrocardiogram  
EE - Energy of Error  
FPGA - Field Programmable Gate Arrays  
GLM - Generalized Linear Method  
HDL - Hardware Description Language  
ICD - Implantable Cardioverter Defibrillators  
IP - Intellectual Property  
LMS - Least Mean Square  
PAD - Public Access Defibrillation  
ROC - Receiver Operating Characteristic  
RTL - Register Transfer Level  
SCA - Sudden Cardiac Arrest  
SCD - Sudden Cardiac Death  
SNR - Signal to Noise Ratio  
SR - Sine Rhythm  
VF - Ventricular Fibrillation  
VHDL - VHSIC hardware description language  
VT - Ventricular Tachycardia

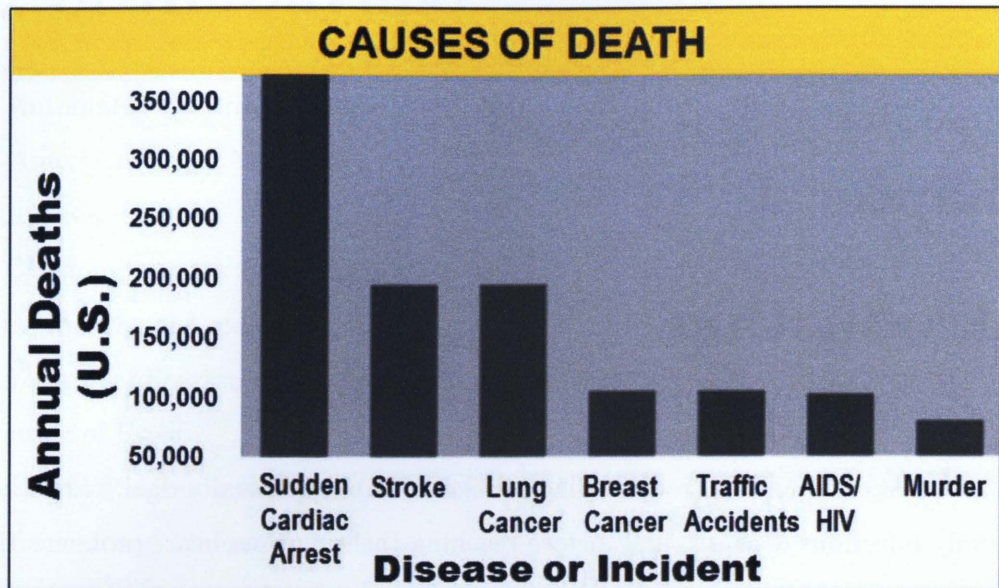
# Chapter 1

## Introduction

People had always suffered from cardiac problems, but they usually died from other causes, especially infectious diseases, long before reaching the age when heart problems threatened their health. As medicine advanced and people lived longer, heart disease became a serious health issue. In fact, according to the Heart and Stroke Foundation, every 7 minutes, someone dies from heart disease or stroke in USA. Sudden cardiac arrest (SCA) is a major public health concern, as it claims the lives of more than 350,000 Americans every year-more than breast cancer, accidents, Alzheimer's disease and AIDS and murder (Figure 1.1). The most common cause of cardiac arrest is a heart rhythm disorder or arrhythmia called ventricular fibrillation (VF) [1].

Fortunately, SCA is reversible in most victims if it's treated within few minutes with an electric shock to the heart to restore the normal heartbeat. This process is called defibrillation. In fact, defibrillation is the definitive treatment for this life threatening arrhythmia. The first defibrillator (Figure 1.2) was developed by Claude Beck in 1947 and used spoons as electrodes. Major advances in defibrillation have been made and have resulted in the development of Automated External Defibrillators (AED) and Implantable Cardioverter defibrillators (ICD). AEDs (Figure 1.3) are automated devices which analyze the electrocardiogram (ECG) signal and advise a shock in case of ventricular fibrillation. ICDs are small devices which can be implanted beneath the skin and function automatically for several years. They





**Figure 1.1:** Annual Deaths in USA  
[Courtesy: American Heart and Stroke Foundation]

detect and treat life threatening disorders of cardiac rhythm. Both ICDs and AEDs are becoming very popular these days because of their life saving potential.

## 1.1 Characteristics of ECG Signal

Heart is a four chambered mechanical pump with two atria for collection of blood and two ventricles for pumping out the blood. The pumping mechanism of the heart is coordinated by electrical signals. In a healthy heart electric impulses travel along specialized fibers and then dart from cell to cell, causing the muscle fibers to contract and relax. Contraction of the ventricles generates a high pressure region which opens up the valves and the blood pours from ventricle into atria. Contraction of the atria in turn causes the blood to flow to the body. A healthy heart shows a typical ECG form, called sinus rhythm and it basically consists of P, QRS, and T waves. Figure 1.4 shows a typical normal ECG signal.

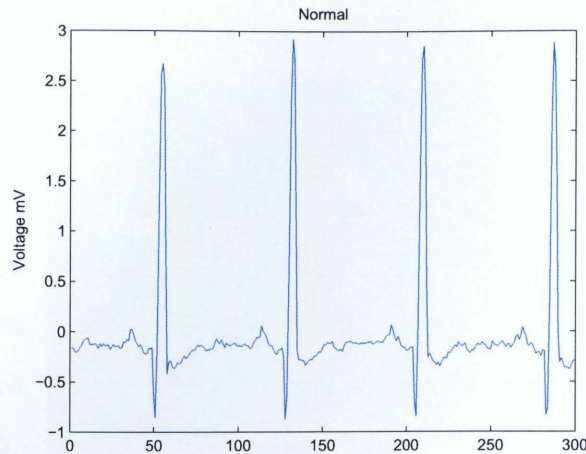


**Figure 1.2:** Early defibrillator  
[Courtesy: Medical History Center/Case Western Reserve University]



**Figure 1.3:** Automated External Defibrillator  
[Courtesy: Philips Medical Systems]



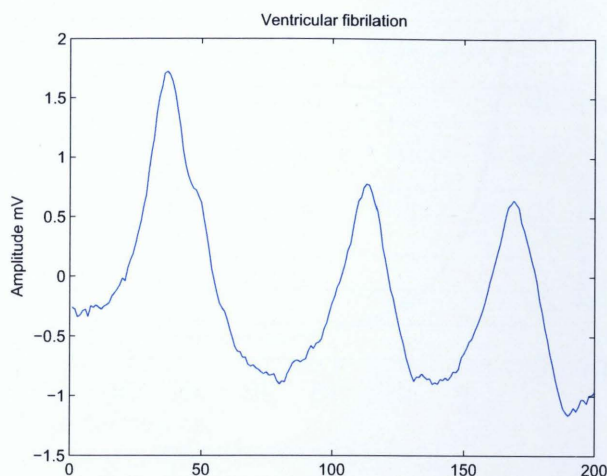


**Figure 1.4:** ECG tracing of normal cardiac cycle

But when things go wrong during VF, the electric impulses do not travel along the specialized fibers; instead, the auto rhythmic cells throughout the heart generate signals instructing cardiac contractile cells to contract. This results in many unsynchronized contractions, so that the heart cannot contract forcefully. Without synchronized contraction blood cannot flow through the body and organs rapidly begin to fail; within 10 minutes, the victim will almost certainly die. ECG signal of a heart suffering from VF shows an irregular cosine like structure as shown in Figure 1.5.

## 1.2 AED

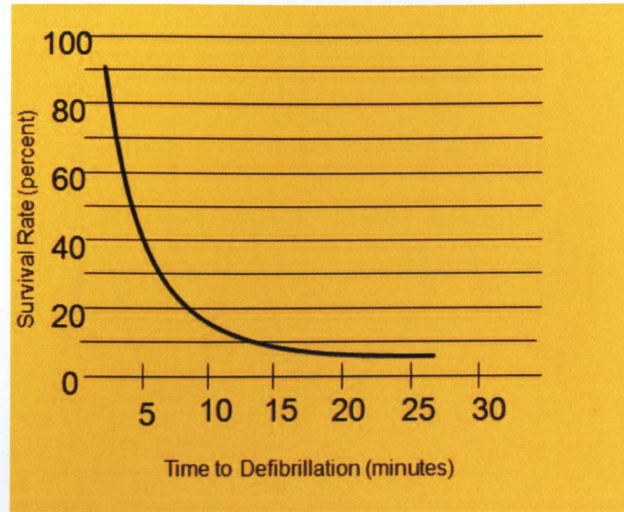
As was mentioned in previous section, when sudden cardiac arrest strikes, time is critical; survival is reduced by 10% for each minute that defibrillation is delayed (Figure 1.6). Brain death and permanent death start to occur in just four to six minutes after someone experiences cardiac arrest. Total response time from the time a 911 call is made until arrival of units and application of defibrillator often reaches 8-9 minutes. As a result, 95% of victims die before emergency personnel arrive. SCA is reversible in most victims if it's treated within a few minutes with an electric shock to the heart to restore a normal heartbeat. Use



**Figure 1.5:** ECG segment during VF

of manual defibrillators requires considerable amount of training especially in interpreting the ECG and since cardiac incidents occur most often out of hospital automated external defibrillators (AEDs) were introduced to increase the survival rate. Use of AEDs requires no training and therefore they can be used in public places by untrained bystanders. In fact, having an AED on site is rapidly becoming a basic standard of emergency care, equivalent to having a smoke alarm. The simplicity of operation of the AED has greatly reduced training requirements and extended the range of people who are able to provide defibrillation. As a result, AEDs can be more widely deployed in public places. One of the greatest challenges in designing the AED is the design of its digital signal processing (DSP) algorithm. DSP is considered as the brain of AED as it has to decide on its own whether the rhythm is shockable or not; therefore, it is of vital importance that the ECG analysis algorithms used by AEDs differentiate well between shockable and non shockable rhythms. AED should not deliver a shock if the patient has collapsed due to a reason other than cardiac arrest; on the other hand, a successfully defibrillated patient should not be defibrillated again due to an analysis error, which would possibly bring the subject back into cardiac arrest [2].





**Figure 1.6:** Survival rate as a function of time  
[Courtesy: American Heart and Stroke Foundation]

### 1.2.1 External influences on AEDs

An AED has to make its decision on the basis of the ECG signal obtained from 2 electrodes that are positioned on the patient's chest. The positioning of the electrode pads and quality of the electrode contacts are very important in acquiring the ECG signal since it might result in false ECG interpretation by the algorithm. The external disturbances on AED that might affect its diagnostic accuracy are: (1) Artifacts: Cardiopulmonary resuscitation (CPR), transport, respiration, motion of AED user. (2) Electric disturbances: Electrical fields (power line), radio communication, noise, implanted pace makers.

Therefore, a good VF detection algorithm should be robust to the above mentioned artifacts and noise and should work well under those conditions.

## 1.3 Measure of Diagnostic Accuracy

As mentioned in the previous section, the accuracy of the DSP algorithm used to differentiate between VF and no VF is very important and can be obtained by comparing the VF/no VF

decisions suggested by the algorithm with the annotated decisions suggested by cardiologists. The cardiologists' decisions are considered to be gold. The main quality parameters used in the literature to investigate the diagnostic accuracy of the DSP algorithms are Sensitivity, specificity, and Receiver operating characteristic (ROC) [2, 4, 5]. The closer the values of sensitivity and specificity are to 100%, the better is the diagnostic accuracy of the algorithm.

### 1.3.1 Sensitivity

Sensitivity is the percentage of the cases labeled as ventricular fibrillation by the algorithm that are actually VF. It is defined as:

$$\frac{DetectedCasesOfVF}{AllCasesOfVF} = \frac{TP}{TP + FN},$$

where  $TP$  is the number of true positive decisions, and  $FN$  the number of false negative decisions. In other words, sensitivity is the measure for how accurate the algorithm detects VF episodes in an ECG.

### 1.3.2 Specificity

Specificity is the probability to correctly identify no VF. In other words, it is the percentage of cases labeled as negative by the test that are actually negative. It is defined as:

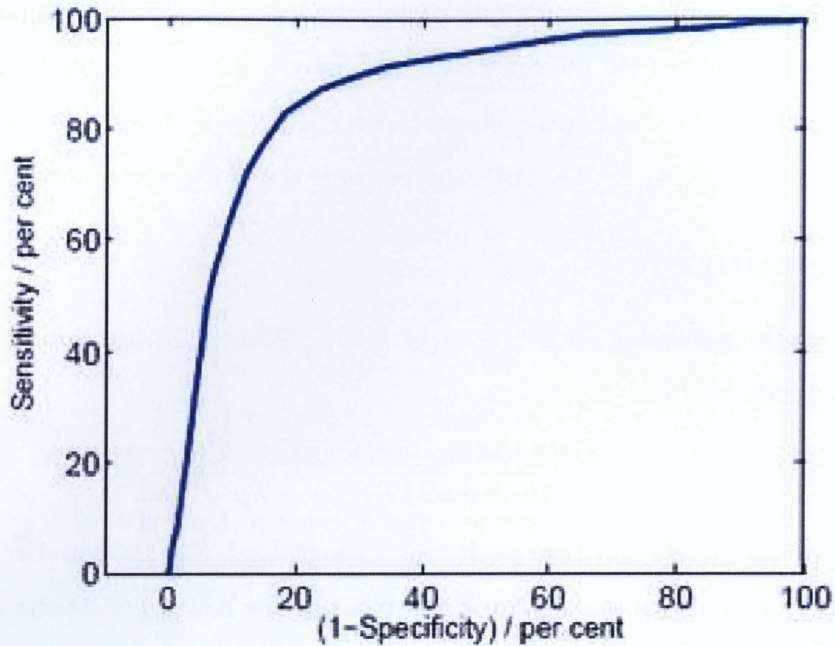
$$\frac{DetectedCasesOf"noVF"}{AllCasesof"noVF"} = \frac{TN}{TN + FP},$$

where  $TN$  is the number of true negative decisions, and  $FP$  is the number of false positive decisions. In real applications of AEDs, the specificity is more important than the sensitivity, since no patient should be defibrillated due to an analysis error since this would possibly damage the myocardial and bring the patient back into cardiac arrest [2, 3, 6].

### 1.3.3 Receiver Operating Characteristic

Measures of overall correct classification of patterns as percentages such as sensitivity and specificity provide limited indication of the accuracy of a diagnostic method. A special algorithm can have a high sensitivity, but a low specificity, or the opposite. In order to come





**Figure 1.7:** Receiver Operating Characteristic

to a single point quality parameter Receiver Operating Characteristic curve (ROC) is used. In other words, ROC curve describes the inherent tradeoff between sensitivity and specificity of a diagnostic test by plotting the sensitivity vs. specificity points obtained for a decision threshold or cut points of the decision stage of the algorithm. Measure of effectiveness of an algorithm is then given by the area under the ROC curve. A method that gives a larger area under the ROC curve indicates a better method than one with a smaller area [2, 4, 5]. Figure 3.10 shows a typical example of a ROC curve.

## 1.4 AED Hardware Implication

Figure 1.8 shows the overall block diagram of AED. The core subsystems of AED include: DSP processor for processing the chest-pad input, microcontroller for managing the user inputs, A/D converter for converting the analog ECG signals to digital information, D/A

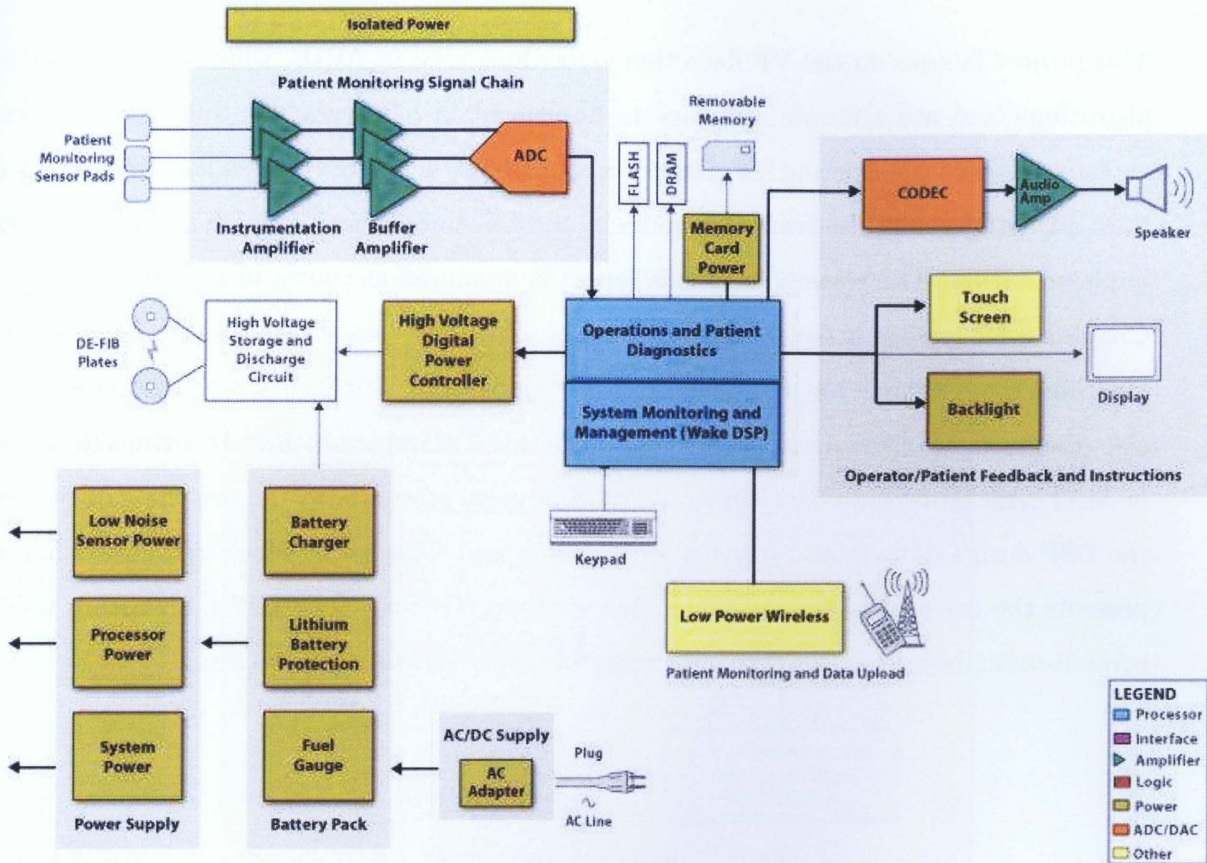


Figure 1.8: Overall block diagram of AED

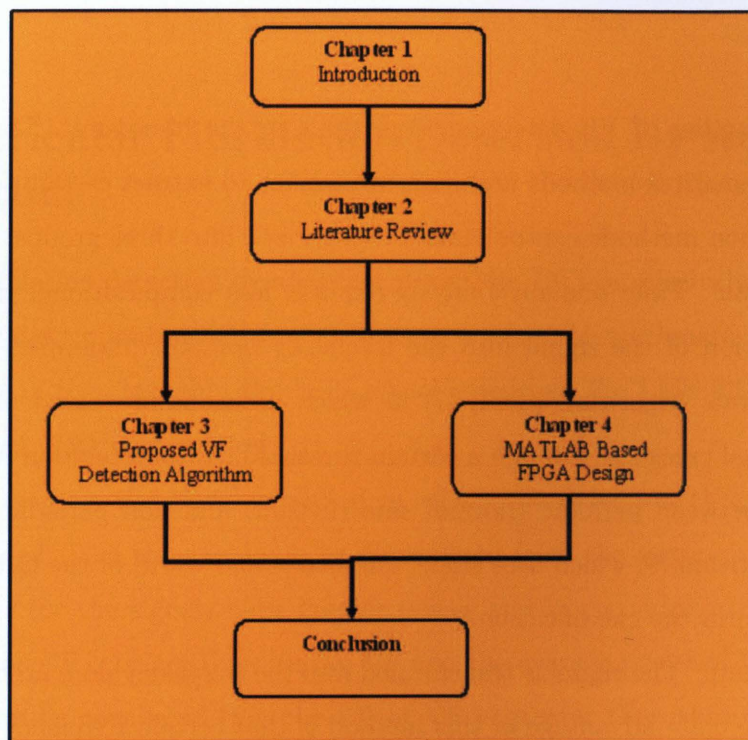
converter for converting the digital signals from the DSP to sound allowing the user to hear instructional commands, very low noise amplifiers for amplifying the pad input and power management system for controlling the battery power. As explained in Section 1.2 DSP is considered as the brain of the AED system. There exists a wide variety of DSP methods and ideas for detection of VF in the literature, but not all these algorithms are suitable for embedding in an AED microprocessor system. Other than having a high diagnostic accuracy, these algorithms should be easily implementable and should work in real-time in an AED.



## 1.5 Organization of the report

This project focuses on the VF detection algorithms used in AEDs, which therefore implies algorithms that are accurate and easy to implement in hardware. To this end, Chapter 2 explains some of the existing VF detection algorithms, and gives the reader some details on their advantages and disadvantages and why most of these algorithms fall short for physical implementation. The remainder of this report is organized as shown in Figure 1.9.

Chapter 3 focuses on a novel method based on AR modeling for detection of VF. Details regarding the AR modeling, dominant pole, decision criteria, and the experimental results are also discussed. Chapter 4 discusses different methods of implementing algorithms developed in MATLAB on FPGA. Chapter 4 also reviews past attempts to remove the gap between the DSP design domain and physical implementation. Chapter 5, which is the last chapter, presents the conclusion. In summary, this work proposes a VF detection algorithm that is computationally inexpensive and can easily be implemented in hardware.



**Figure 1.9:** Organization of the project report

## Chapter 2

# Literature Review

There are a number of VF detection algorithms in the literature. These algorithms use different mathematical methods and transformations to extract certain features of a signal. The VF detection methods can be mainly categorized into three groups:

**Time domain:** Time domain analysis requires less computational time, since there is no transformation of the signal into the frequency domain. Examples include Threshold Crossing Intervals Algorithm (TCI) [7] in which decisions are based on the number and position of signal crossings through a certain threshold, Autocorrelation Algorithm [8] which distinguishes between periodic (normal sine rhythm) and non periodic (VF) signals, and Tompkins algorithm [9] which uses slope, amplitude and width of the QRS information as a feature to perform the classification.

**Frequency domain:** The signal is transformed into the frequency domain by means of Fourier transform. The main advantage of this method is that certain frequency noise such as high frequency noise can be easily removed. In addition, frequency dependent features of SR or VF can be easily treated in this domain [2]. Examples include Spectral Filters [10, 11] and Spectral Algorithms which analyze the energy content of different frequency bands.

**Combined methods:** Example of this method which makes use of both time and frequency domain analysis is wavelet based algorithms [12]. The computational complexity of these algorithms might be higher than the techniques mentioned above. In the following sections some of the VF detection algorithms are explained in details.



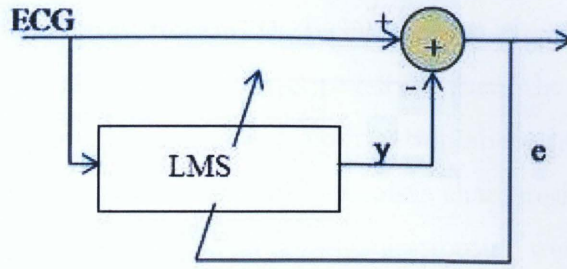


Figure 2.1: Schematic of adaptive filter

## 2.1 Ventricular Fibrillation Detection By Adaptive Algorithm

Ventricular Fibrillation detection by adaptive algorithm [13] is a relatively simple and accurate method for discriminating VF from VT using a Least Mean Square (LMS) algorithm. In the first stage of the algorithm the ECG signal is fed into the LMS filter and the energy of the error signal (EE) is calculated.

The second stage is the learning period in which the detection thresholds of VF and VT are determined. The threshold can be determined from the energy of the error signal. The EE is larger for the signals with greater power and worse autocorrelation. Since the autocorrelation of VT is better than that of VF and the average power of VF is greater than that of VT it can be concluded that the EE of VF is greater than that of VT. Figure 2.2 shows the distribution of VF and VT energy error. Thus, in learning phase thresholds  $T_1$  and  $T_2$  are obtained and the ECG signals with EE smaller than  $T_1$  can be considered VT and that larger than  $T_2$  VF. No decision will be made if EE is in the shadowing area. In this case sequential test is applied in which another segment of ECG signal is inputted to the LMS algorithm and EE is calculated until a decision is made.

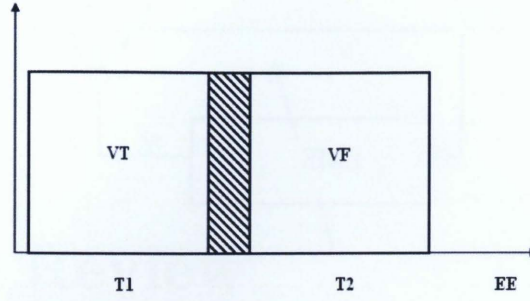


Figure 2.2: Distribution of EE

## 2.2 Ventricular Fibrillation Detection Using Pseudo Wigner-Ville Distribution

Rosado et al. [14] presented a real time VF detection algorithm that combines both time domain and time-frequency domain parameters which results in lower computational calculations. Time domain parameters are used as an initial stage of the VF detection algorithm due to its good behavior in non-VF rhythms rejection. Discarding segments that are clearly different from VF results in increase in specificity and avoids the calculation of time-frequency distribution in such cases. In VF episodes prominent QRS peaks are inexistent; therefore, squaring the ECG time series  $x(t)$  (obtaining  $x(t)'$ ) results in increase of peaks, and dividing  $x(t)'$  by its mean  $\mu$ , results in unit mean. The variance value  $\sigma$ , for  $x(t)'$  is closely related to QRS peak presence; a high value is considered as no VF.

$$VF = \frac{1}{N} \left[ \frac{x'}{\mu(x'(t))} - 1 \right]^2 = \sigma \left( \frac{x'}{\mu(x'(t))} \right) \quad (2.1)$$

The second time domain parameter is called RatioSTD which is the quotient between the standard deviation of the derivative and the standard deviation of the absolute value of the derivative for the ECG time series. This parameter gives an idea about the symmetry between positive and negative values. In the case of normal Sine rhythm no symmetry exists. On the other hand, in the case of VF, due to oscillating nature of the VF signal higher values



of RatioSTD are obtained. Figure 2.3 shows the percentile information about the VF like and non VF signals resulted from RatioSTD. Techniques for examining signals in the time and frequency domain have their limitations; they tell us where the signal component exists in the frequency domain, but they do not give information about how its frequency characteristics change over time. In order to obtain a complete characterization of ECG signals a set of time-frequency parameters using Wigner-Ville distribution were calculated. Regarding the time-frequency parameters, Rosado et.al defined two spectral bands: BALO (2-14 Hz) and BAH1 (14-28 Hz). For VF episodes, BAH1 does not contain any energy components, opposite to non-VF rhythms. The time-frequency parameters used in the algorithm are:

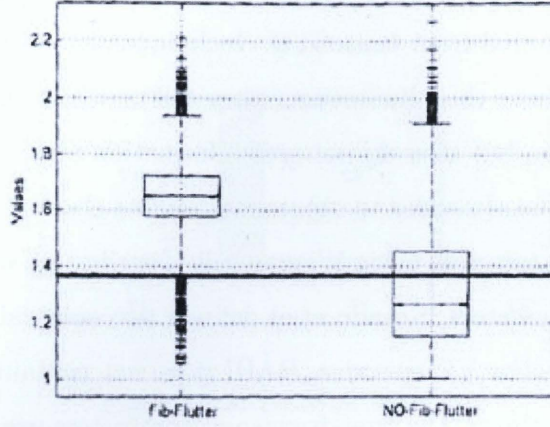
**QTH:** Percentage of the total number of nonzero terms existing in BAH1 band

**VDL8:** Standard deviation of the first-order derivative non-zero terms vector obtained by splitting the temporal segment into eight sub-segments. In case of normal sinus it takes a higher value due to the existence of bigger oscillations.

**LMTP:** Indicates the maximum temporal length of the area enclosing higher energy than 50% of the maximum. This value is higher in VF episodes than in normal sinus rhythms.

**Curve:** Analyzing the number of non-zero terms at every frequency bin of spectral resolution in the BALO band, and performing a parabolic approximation for the obtained points. The regression parabola is less during VF than Normal sine rhythm. TMY: Number of points in a TFR having energy between 50% and 100% of the maximum energy value existing in the TFR.

**CT8:** The temporal segment of the time frequency distribution is divided into 8 segments and its energy in BAH1 band is measured for every segment. TE: Total energy of the distribution in BAH1 band



**Figure 2.3:** Box-Plot results for RatioSTD parameter  
[Courtesy: Rosado et al.]

## 2.3 Hilbert Transform

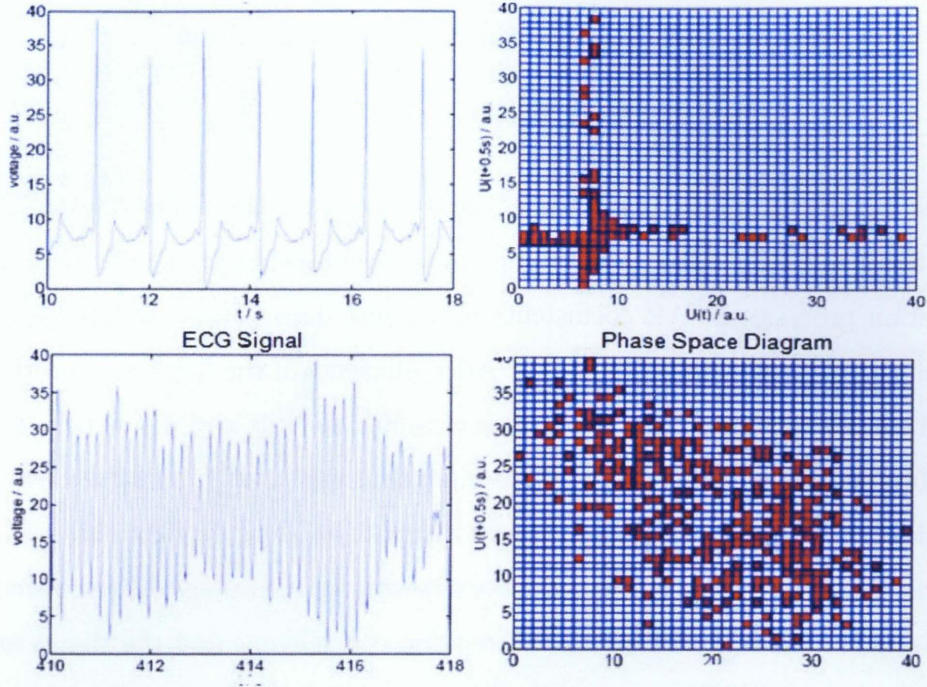
Aman et al. proposed a method that differentiates SR from VF based on phase-space plots. This method [15] is based on a method which is used in analyzing nonlinear signals. Hilbert transform is defined using the Cauchy principal value (denoted p.v.) and is given by:

$$X_H(t) = \frac{1}{\Pi} P.V \int \frac{x(\tau)}{t - \tau} d\tau \quad (2.2)$$

Hilbert transform can be considered as the convolution of the functions  $x(t)$  and  $1/\pi t$ . Due to the properties of convolution it can be concluded that the Hilbert transform can be realized by an ideal filter whose amplitude response is unity and phase response is a 90 degrees time shift at all frequencies  $w > 0$ . In the phase-space plot the signal is plotted on the x-axis, and the Hilbert transform of the signal,  $X_H(t)$  is plotted on the y-axis (Figure 2.4). A  $40 \times 40$  grid is produced and the number of boxes visited by the ECG signal is counted and the measure  $d$  is calculated:  $d = (\text{visited boxes}) / (\text{number of all boxes})$

If  $d$  is higher than a certain threshold, the corresponding ECG segment will be classified as VF. In [15], the threshold was chosen to be 0.15. Similar approach based on time-delay





**Figure 2.4:** ECG signals (left) and their corresponding phase-space plots (right)  
[Courtesy: Aman et al.]

approach is also given in [3].

## 2.4 Generalized Linear Method

Krishnan et al. [5] used Generalized Linear Method (GLM) based classification model to distinguish VF from SR. GLM is a statistical linear model and is given by:

$$\hat{Y} = A\beta + \epsilon, \quad (2.3)$$

where  $\hat{Y}$  is an  $N$  dimensional vector of observed responses,  $\beta$  is the unknown least square estimator, and  $A$  is a  $N(P+1)$  matrix of known AR coefficients. During the first phase of the classification, the classifier needs to be trained using the known classes of ECG signals. In this phase, the auto regressive coefficients are calculated for each signal in the training



set, and the observed signals ( $\hat{Y}$ ) is set to 1 for normal SR signals in the training set and -1 for VF. The least square estimator ( $\beta$ ) can then be computed from:

$$\beta = (A^T A)^{-1} A^T \hat{Y} \quad (2.4)$$

The least square estimator is calculated only once from the training samples (from the known classes of ECG). After the training phase, the classification can be performed. In the classification process, the AR coefficients of the test data (ECG segment to be analyzed) are obtained. If the multiplication of the AR coefficients of the test data and the previously obtained estimator is positive, the signal is classified as NSR and if it is negative the signal is classified as VF. In this project Burgs algorithm was used to compute AR coefficients. AR model becomes less sensitive to model order  $P$ , for  $P$  more than three. However, the AR model of order four was selected for extracting the features. The criteria used in [5] for selecting the model order are the correlation coefficient  $\rho$  and the signal to noise ratio ( $SNR$ ). The  $\rho$  and  $SNR$  are given by equations 2.5 and 2.6 respectively.

$$\rho = \frac{\sum_{i=1}^{i=N} (v(i) - m)(\tilde{v}(i) - \tilde{m})}{\sqrt{\sum_{i=1}^{i=N} (v(i) - m)^2 \sum_{i=1}^{i=N} (\tilde{v}(i) - \tilde{m})^2}}, \quad (2.5)$$

$$SNR = 10 \log \frac{\sum_{i=1}^{i=N} (v(i))^2}{\sum_{i=1}^{i=N} (v(i) - \tilde{v}(i))^2}, \quad (2.6)$$

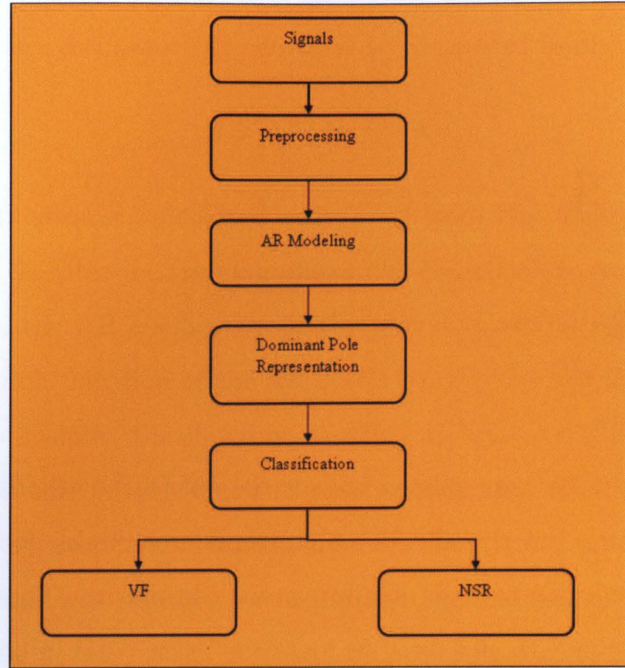
where  $v(i)$  and  $\tilde{v}(i)$  are the original and the simulated signals, and  $m$  and  $\tilde{m}$  are the mean of the original and simulated signals respectively.

# Chapter 3

## Methods

As explained in chapter 2 there exists a wide variety of DSP methods and ideas for detection of VF in the literature, but not all these algorithms are suitable for embedding in an AED microprocessor system due to high computational complexity. These algorithms should be easily implementable and should work in real-time in an AED in addition to having a high diagnostic accuracy. In this chapter a new VF detection method based on AR modeling is proposed. The use of AR modeling for feature extraction has been proved to have some advantages over other existing methods. It involves less amount of computation and processing time for extraction of features and classification which makes it suitable for real time classification in AEDs [5]. AR modeling has been used in various applications including classification of electroencephalograms [28]. In the proposed work, dominant poles are extracted from the model transfer function. These dominant poles represent the dominant features of the signal segments in the spectral domain and can be used for classification of the cardiac rhythm. The potential use of this method to distinguish between VF and Normal sinus rhythm (NSR) is discussed. Figure 3.1 shows the flow chart of the proposed method.





**Figure 3.1:** Proposed method

## 3.1 ECG Recordings

### 3.1.1 Data Libraries

Clinical storage formats that are generally used include the Extended European Data Format (EDF), which is commonly used for electroencephalograms; HL7 (an SML- based format for the exchange of data in hospitals); and Waveform DataBase (WFDB). Among these formats, WFDB is an excellent (if not best) current choice for storing ECG data [17]. WFDB records have three main components: an ASCII header file, a binary data file, and a binary annotation file. The header file contains information about the binary file format variety, the number and type of channels, the lengths, gains, and offsets of the signals, and any other clinical information that is available for the subject. The separate header file allows for rapid querying. Any number of annotation files can be associated with the main binary file just by using the same name (with a different extension). In addition, WFDB allows the virtual

Database	nsrdb	CU database
Number of Signals	18	35

**Table 3.1:** Signals in the dataset

concatenation of any number of separate files, without the need to actually merge them.

### 3.1.2 Database Analysis

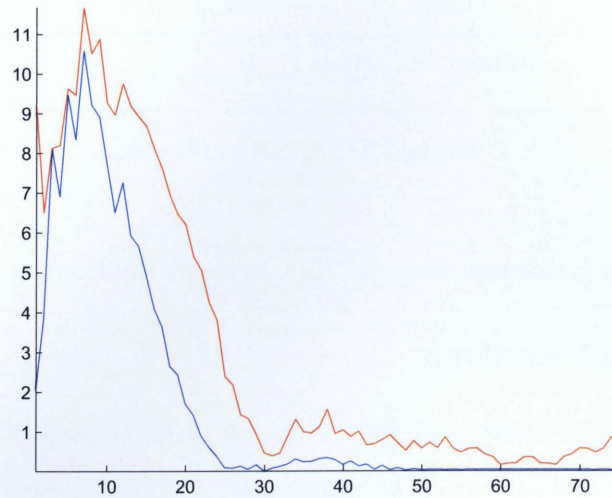
The most extensive and freely available collection of ECG (and related) waveforms can be found on PhysioNet [18] (the MIT Laboratory for Computation Physiology's Web site) or one of its many mirrors. This collection of databases is consisted of hundreds of multi lead ECG recorded from patients who suffer from various known heart conditions, as well as examples of healthy ECGs, for periods from 30 minutes to more than a day. These records have been annotated by expert clinicians. These ECG signals can be used to evaluate the diagnostic accuracy of VF detection algorithm.

Commonly used databases that can be found on PhysioNet are Boston's Beth Israel Hospital and MIT database (MIT-BIH), Creighton University ventricular tachyarrhythmia database (cudb), and the American Heart Association database (AHA). In this project the entire CU, and normal sine rhythm database (nsrdb) has been used. Table 3.1 shows the number of signals in each database.

### 3.1.3 Preprocessing

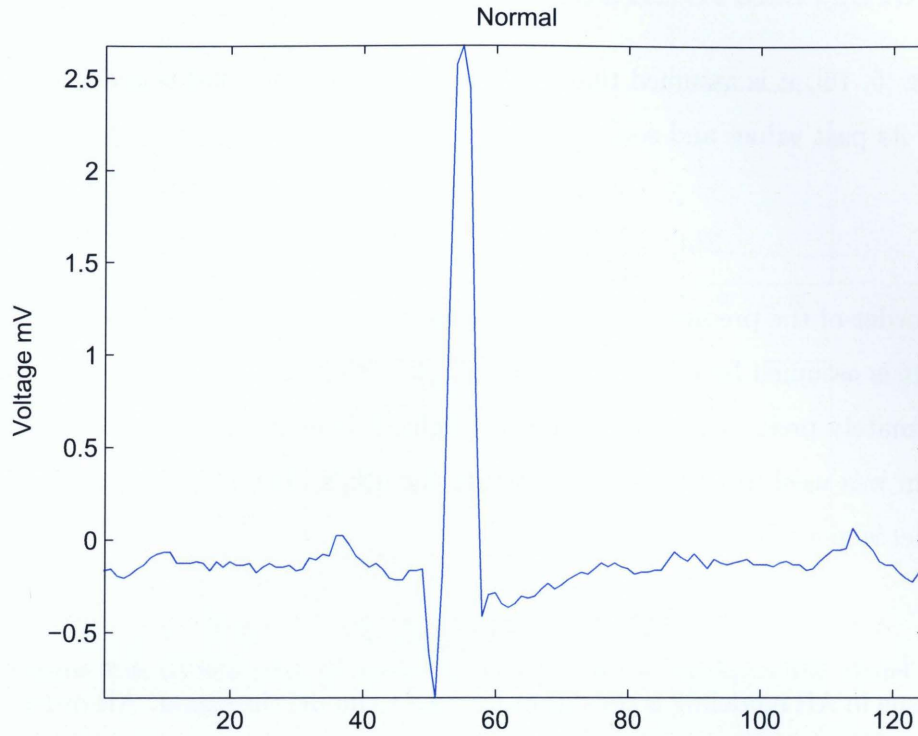
Signals from MIT-BIH Arrhythmia Database (mitdb) and CU database are sampled at 250 Hz and the those from MIT-BIH Normal Sinus Rhythm Database (nsrdb) are sampled at 125 Hz. In this analysis, signals from mitdb and CU database were down-sampled to the frequency of 125 Hz. The signals from the nsrdb data set do not have the desired resolution and in future work they will be interpolated so that all the data used in the analysis have a sampling frequency of 250 Hz. There are many different interpolation methods such as





**Figure 3.2:** Red-Spectrum of ECG segment before filtering,  
Blue-Corresponding segment after filtering

Nearest-neighbor interpolation, Linear interpolation, Polynomial interpolation, Spline interpolation, Rational interpolation and trigonometric interpolation [27]. In the future work the appropriate algorithm will be chosen based on the following criteria: How accurate is the method? How expensive is it? How smooth is the interpolant? How many data points are needed? The ECG signals in the database were filtered prior to the analysis in order to remove the noise including respiration, muscle noise, and base line drift. First, a moving average filter of order 5 was applied to the signal. Then, high pass filter with a cut off frequency of one Hz was used for drift suppression. Finally, a low pass Butterworth filter with a limiting frequency of 45 Hz was applied in order to get rid of high frequency noise like interspersions and muscle noise. Figure 3.2 shows the ECG spectrum before and after filtering.



**Figure 3.3:** Each segment consists of at least one cardiac cycle

### ECG Segmentation

Auto Regressive modeling given in equation 3.1 is a time invariant system and its obtained AR coefficients do not change with time. On the other hand, ECG signal is a nonstationary signal and its statistics change with time. So in order to be able to model the signal using equation 3.1, ECG signal should be segmented. The periodic nature of ECG signal makes the segmentation easy. In segmentation care must be taken to pick at least one cardiac cycle so that the signal can be accurately modeled and can be useful in diagnosis. Cardiac cycle length or RR intervals differ for normal sinus rhythm and VF. In VF, the RR intervals are much shorter than in a NSR. In this study a sample size of 125 (1 second) was used which is adequate to capture most of the information from a particular cardiac cycle (Figure 3.4)[5].

## 3.2 Autoregressive Modeling

In AR modeling [5, 16], it is assumed that the  $n$ th value of a signal  $s[n]$  is given by a linear combination of its past values and some input  $u[n]$ :

$$S[n] = \sum_{k=1}^{k=p} a[k]s[n-k] + Gu[n], \quad (3.1)$$

where  $p$  is the order of the predictor (model),  $a[k]$ 's are the AR model coefficients, and  $G$  is a gain factor. It is assumed here that the input  $u[n]$  is totally unknown and the signal  $s[n]$  can be approximately predicted from a linearly weighted summation of its past  $p$  samples. Burgs algorithm was used in this work to compute the  $a[k]$ 's. The transfer function,  $H(z)$ , of the AR model is given by:

$$H(z) = \frac{G}{1 + \sum_{k=1}^{k=p} a[k]z^{-k}} \quad (3.2)$$

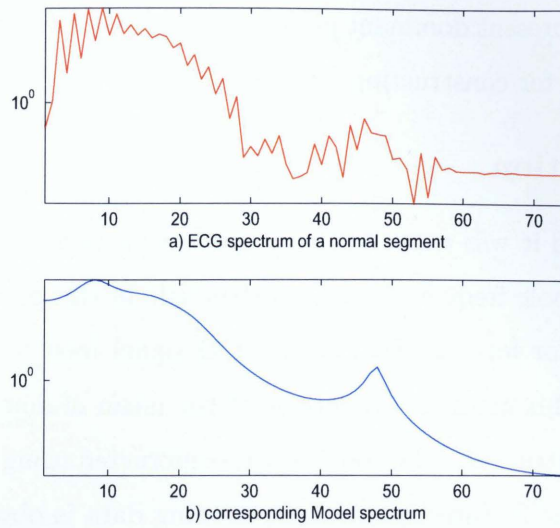
A critical issue in AR modeling is the AR order used to model the signal. AR order should be appropriately chosen to make sure that the signal is approximated with high accuracy. Various model orders were tested to obtain the best accuracy in the modeled signal. In this project Burgs algorithm was used to compute AR coefficients and the AR order was chosen to be 8. Figure 3.4 shows the spectrum of the normal ECG segment and its corresponding model spectrum for the selected model order.

## 3.3 Dominant Poles

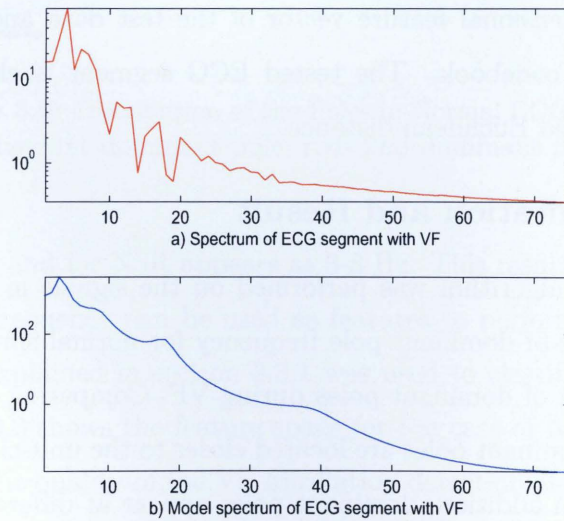
Poles of the AR model system could be extracted from the model transfer function by factorizing the denominator of equation 3.2. In order to reduce the dimensions of the feature vector, dominant poles were used as they represent the dominant features of the signal in the spectral domain. Equation 3.2 may be factorized into individual pole contributions as follows:

$$H(z) = \frac{G}{|(z - b_1)(z - b_2) \cdots (z - b_p)|}, \quad (3.3)$$





**Figure 3.4:** a) Spectrum of normal ECG segment b) Corresponding model spectrum ( $p=8$ )



**Figure 3.5:** a) Spectrum of VF ECG segment b) Corresponding model spectrum ( $p=8$ )



where  $b_1, b_2 \dots b_p$  correspond to the complex poles of the transfer function  $H(z)$ . Dominant poles are the ones which have the maximum distance from the origin in the  $z$ -plane. These dominant poles will represent dominant peaks of the signal in the spectral domain, and could be used as parameters for constructing feature vectors [16].

### 3.3.1 Classification

In the previous section it was explained how the feature vectors consisting of the first and the second dominant pole frequencies are constructed. In this section, these feature vectors are used as a descriptor for classification of ECG signal into NSR or VF by the distance measure method. In this method a codebook of the mean of dominant pole frequencies for each ECG class is constructed. The codebook is constructed using 8 signals per class. In the classification phase, the features vector of the testing data is obtained and then compared to the vectors of the codebook by the Euclidean distance measure.

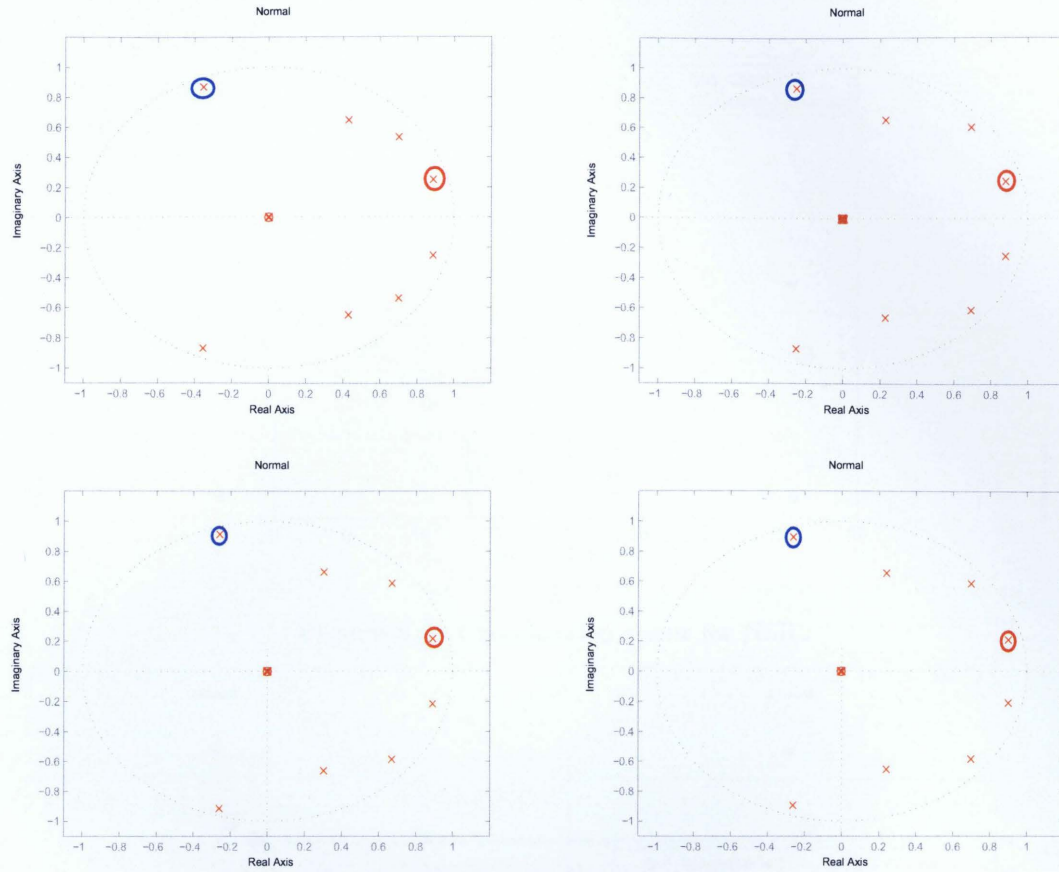
The Euclidean distance is defined as:

$$D = (P - Q)^T(P - Q), \quad (3.4)$$

where  $P$  is the 2 dimensional feature vector of the test data and  $Q$  is the 2 dimensional feature vector in the codebook. The tested ECG segment is classified to the class that minimizes the measured Euclidean distance.

### 3.3.2 Implementation and Result

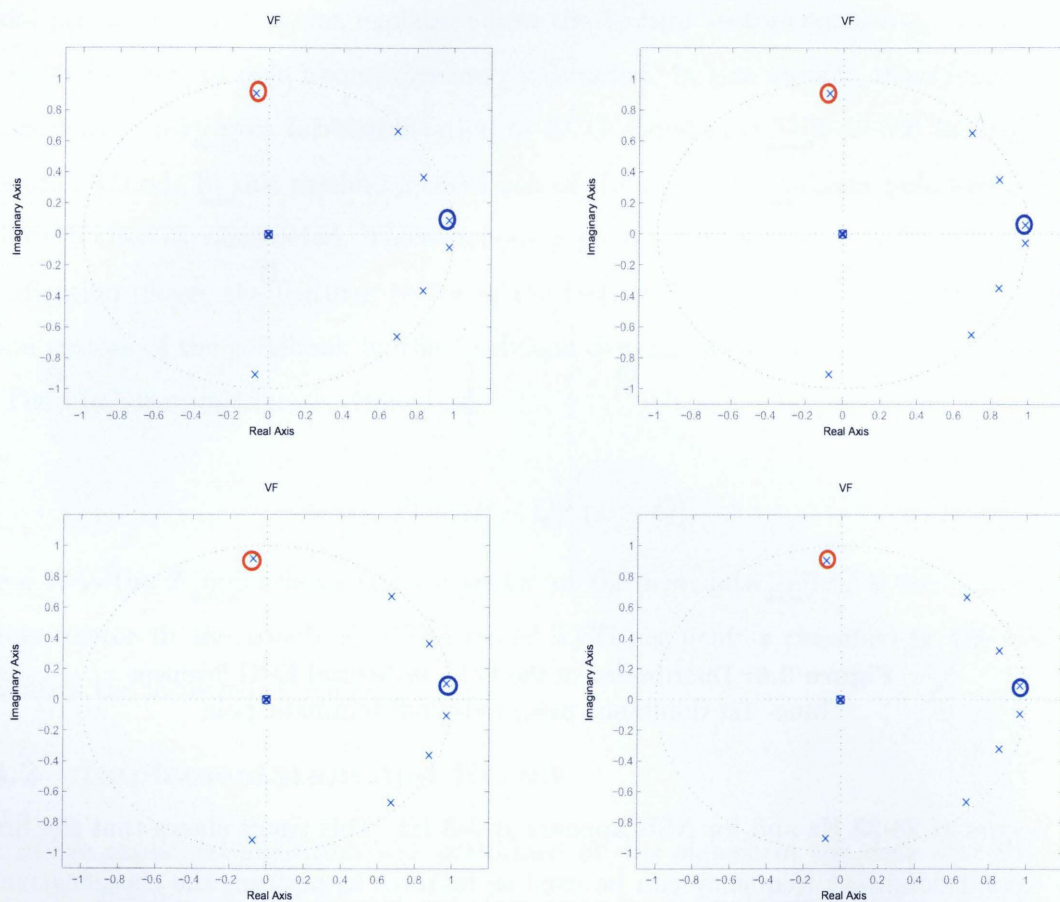
The above explained algorithm was performed on the signals in the data set. Figure 3.6 shows the distribution of dominant pole frequency for normal sinus rhythm and Figure 3.7 shows the distribution of dominant poles during VF. Comparing these two figures one can observe that in VF dominant poles are located closer to the unit circle whereas in NSR poles are located farther. In addition, dominant poles appear at different frequencies for normal and VF. The first dominant pole frequency for VF occurs at the region of 3-8 Hz; whereas for NSR it occurs in the region of 30-33 Hz. The second dominant frequency for the case of



**Figure 3.6:** Distribution of the Poles in Normal ECG Segment  
blue- 1st dominant pole, red- 2nd dominant pole

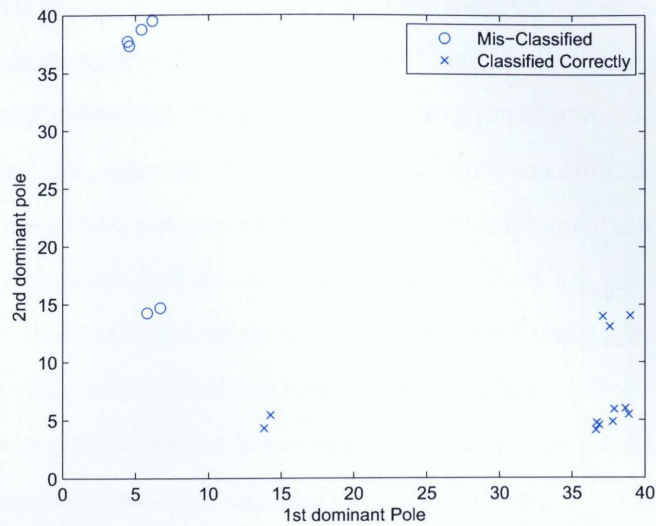
VF occurs at 28-33 Hz and for NSR appears at 3-8 Hz. This result shows that the first and the second dominant frequency can be used as features to perform the classification. The classification scheme explained in section 3.3.1 was used to classify the ECG segment into NSR and VF. Figure 3.8 shows the feature space for the case of NSR and the misclassified samples are labeled. The quality of the VF fibrillation detection algorithm was measured by calculating Sensitivity and specificity and plotting the ROC curve (Figure3.10). Assuming that the annotations are 100% correct, the perfect algorithm would have value of 100% for the sensitivity and specificity. The sensitivity and specificity of the proposed method is 94%

and 66% respectively. The summary of the classification results are given in table 3.2.

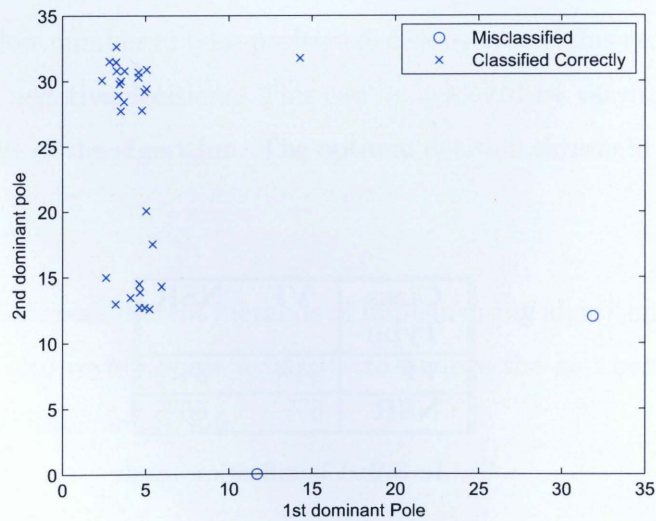


**Figure 3.7:** Distribution of the Poles during VF  
blue- 1st dominant pole, red- 2nd dominant pole

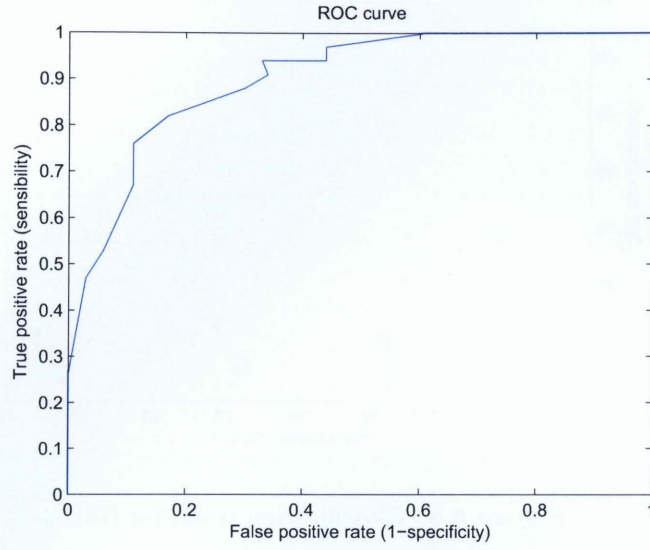




**Figure 3.8:** Classification result for NSR



**Figure 3.9:** Classification result for VF



**Figure 3.10:** Receiver operator characteristic of the proposed method

Class Type	VF	NSR
VF	94%	34%
NSR	6%	66%

**Table 3.2:** Classification result

### 3.3.3 Discussion and Future work

Different values of AR modeling orders were tested for the ECG signals and the result showed that AR order of eight is sufficient to model the ECG signal. The classification result showed that AR modeling and dominant poles can be used to discriminate between VF and NSR. The proposed method has some advantages over the other existing algorithms. Most of the VF detection algorithms are not suitable for hardware implementation but the AR coefficients and dominant pole frequencies in the proposed method are easy to compute; in addition, classification based on distance measure is computationally inexpensive which makes it suitable for embedding in an AED microprocessor system.

In this project, a fixed sample size of 125 samples has been used for AR modeling. As previously stated, each segment must be consisted of at least 1 cardiac cycle and the cardiac cycle length differs for NSR and VF; as a result, a variable sample size based on the estimation of the R-R interval might yield better results.

In the application of AED specificity is more important than the sensitivity, since AED should not deliver a shock to a patient that has collapsed due to a reason other than cardiac arrest. Defibrillation due to an analysis error might cause cardiac arrest. Therefore, it is desired to obtain a low number of false positive decisions even if this results in the increase in the number of false negative decisions. This can be achieved by varying the decision threshold the decision stage of the algorithm. The optimal decision threshold can be obtained from ROC curve.

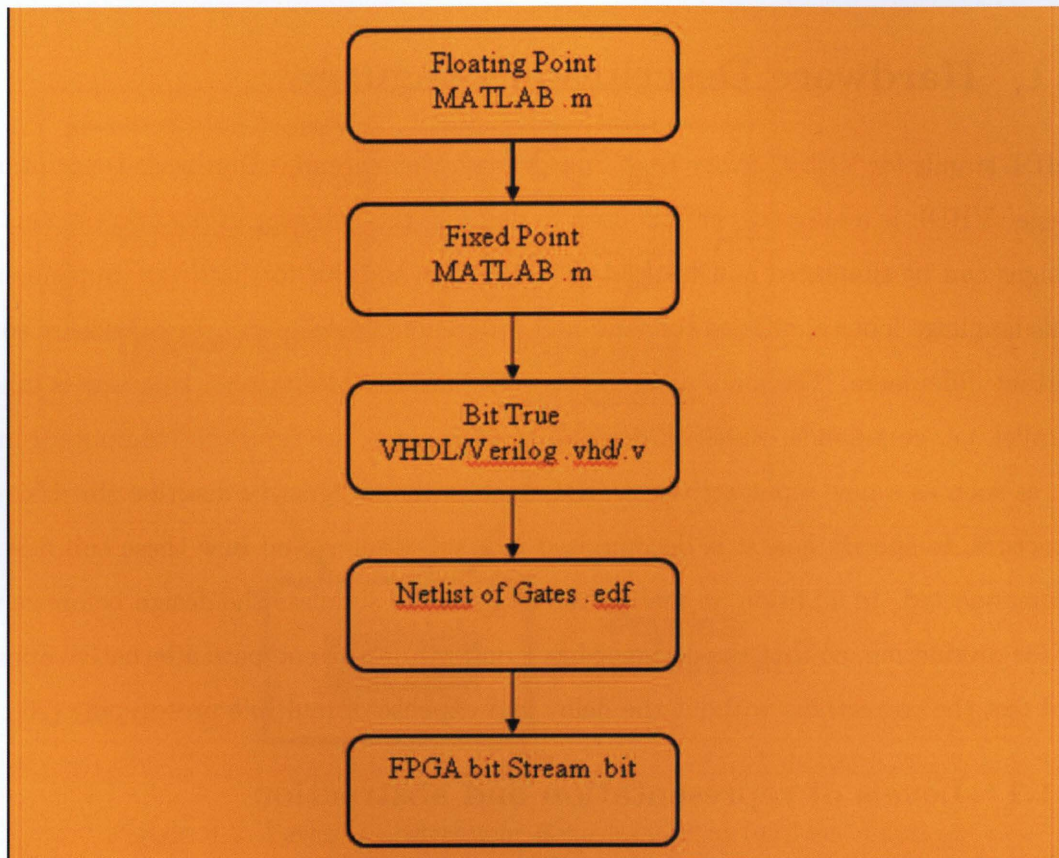
Next Chapter discusses different methods of implementing algorithms developed in MATLAB on FPGA. It also reviews past attempts to remove the gap between the DSP design domain and physical implementation.



## Chapter 4

# MATLAB based FPGA design

Many high-performance signal processing products are now being implemented in field programmable gate arrays (FPGAs). FPGAs can offer higher level of performance than the standard general purpose DSP processors; DSP processors are limited by fixed hardware architecture such as bus performance bottlenecks, a fixed number of multiply accumulate (MAC) blocks, fixed memory, fixed hardware accelerator blocks, and fixed data widths. Hence, the DSP processors are not suitable for certain applications that might require customization. On the other hand, FPGAs provide a reconfigurable solution for implementing DSP applications as well as higher data processing power than conventional DSP processors. Therefore, DSP systems implemented in FPGAs can have customized architecture, customized bus structure, customized memory, customized hardware accelerator blocks, and a variable number of MAC blocks [25]. Traditionally, DSP design has been divided into two types of activities: systems/ algorithm development and hardware/software implementation. For prototyping DSP algorithms DSP system designers and algorithm developers mostly use the MATLAB language. Hardware designers take the specifications created by the DSP engineers and create a physical implementation of the DSP design by creating a register transfer level (RTL) model in a hardware description language (HDL) such as VHDL and Verilog. Unfortunately there is a gap between algorithm and hardware design and this gap is even aggravated as algorithms continue to become more complex. Figure 4.1 shows the FPGA design flow for MATLAB based DSP algorithms.



**Figure 4.1:** FPGA design flow for MATLAB based DSP algorithms



In this section a brief introduction into VHDL language is given. In addition, past attempts to remove the gap between the DSP design domain and physical implementation is reviewed. It should be noted that there are many ways to implement a given DSP design on a given FPGA, but each implementation is different in terms of performance and the number of logic blocks used. Hence, the ability to quickly evaluate alternative implementations is critical to achieving high quality DSP realization in a timely manner [23].

## **4.1 Hardware Description Language**

VHDL stands for VHSIC (Very High Speed Integrated Circuits) Hardware Description Language. VHDL is a language that is used to describe the behavior of digital systems. VHDL designs can be simulated and translated into a form suitable for hardware implementation. This language looks similar to conventional programming languages, but there are some important differences. The main difference is that hardware description language is inherently parallel, i.e. commands, which correspond to logic gates, are executed (computed) in parallel, as soon as a new input arrives. VHDL enables the designer to describe the design in its structure, to specify how it is decomposed into sub designs and how these sub designs are interconnected. In addition, it enables the designers to simulate the design before sending it off for production, so that the designer has a opportunity to compare alternative approaches and test the correctness without the delay and expense of multiple prototyping [26].

### **4.1.1 Levels of representation and abstraction**

A digital system can be represented at different levels of abstraction: Behavioral, Structural and Physical. This keeps the description and design of complex systems manageable. Figure 4.2 shows different levels of abstraction.

The highest level of abstraction is the behavioral level that describes a system in terms of what it does (or how it behaves) rather than in terms of its components and interconnection between them. In addition, it specifies the relationship between the input and output signals. The structural level, on the other hand, describes a system as a collection of gates and

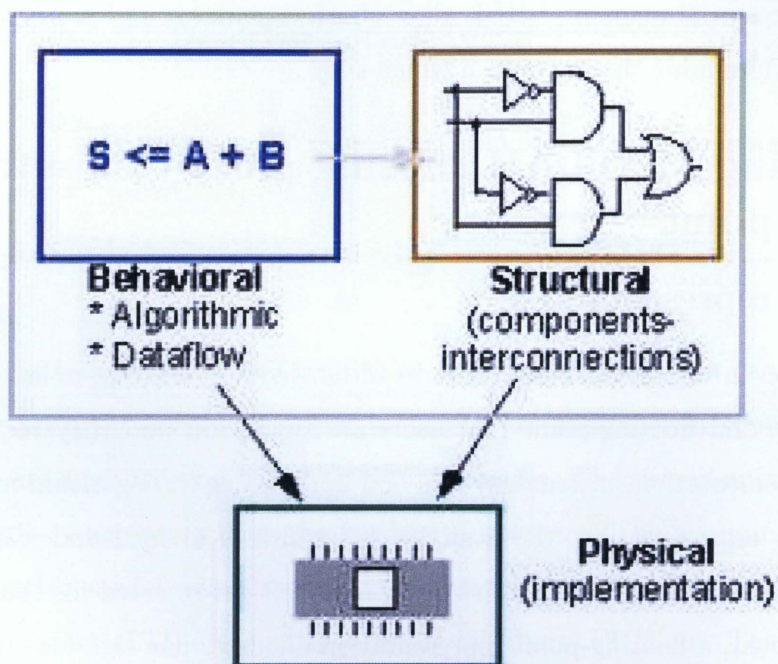


Figure 4.2: Levels of abstraction: Behavioral, Structural and Physical



components which is closer to the physical realization of a system.

VHDL describes a digital system at the structural or the behavioral level. The behavioral level can be further divided into two kinds of styles: Data flow and Algorithmic. The dataflow representation describes how data moves through the system (registers). The data flow model makes use of concurrent statements that are executed in parallel as soon as data arrives at the input. In other words, in data flow model the order in which statements are given does not matter. This is in contrast to conventional, software programs that execute the statements in a sequential manner. VHDL allows both concurrent and sequential signal assignments that will determine the manner in which they are executed.

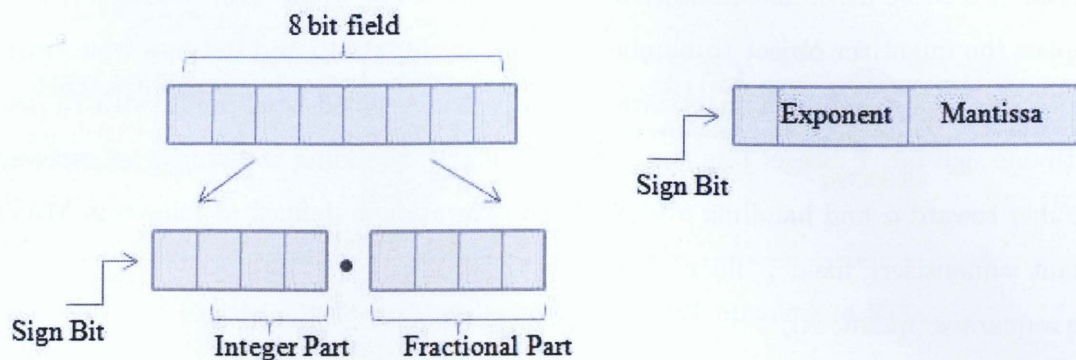
## **4.2 Converting MATLAB floating point computations to fixed point**

### **4.2.1 Numeric representations**

Numeric representation in digital hardware may be either fixed or floating-point. The differences between fixed and floating-point processors are significant and they require very different internal implementation, instruction sets and approaches for algorithm implementation. In fixed-point representation, the available bit width is divided and allocated to the integer part and the fractional part, with the extreme left bit reserved for the sign (2's complement). In contrast, a floating-point representation allocates one sign bit and a fixed number of bits to an exponent and a mantissa. Figure 4.3 shows the Fixed point and floating point representation. In fixed-point, relatively efficient implementations of arithmetic operations are possible in hardware. In contrast, the floating-point representation needs to normalize the exponents of the operands for addition and subtraction [22, 23, 24].

Most FPGA designs of digital signal processing applications are limited to fixed point arithmetic due to the cost and complexity of floating point hardware. On the other hand, MATLAB is optimized for floating-point mathematical operations and it slows down significantly with fixed-point representations, in which each operation checks for overflow, underflow, rounding, and so forth. Hence, the first step in the flow to map MATLAB algorithm





**Figure 4.3:** Fixed point and floating point representation

into hardware is the conversion of the floating point MATLAB algorithm into a fixed point version.

The strategies for floating-point to fixed-point conversion can be categorized into two groups [19, 20]. The first one is an analytical approach in which the finite word length effects are analyzed due to fixed-point arithmetic. The second approach is based on bit-true simulation originating from the hardware designers. The analytical approach started from attempts to model quantization error statistically; then it was expanded to specific linear time invariant (LTI) systems such as digital filters, FFT, etc. The potential benefits of the bit-true simulation is that it can handle non-LTI systems as well as LTI systems [19]. There has been some work in the recent literature on automated compiler techniques for conversion of floating point representations to fixed-point representations [20].

## 4.2.2 Fixed point modeling in MATLAB

Fixed-point representation is provided using the MATLAB quantization functionality namely "quantizer()" and "quantize()" that comes with the Filter Design and Analysis (FDA) Toolbox [19, 22].

The "quantizer" function is used to define the quantizer object, which allocates the

bit-widths to be used along with whether the number is signed or unsigned, what kind of rounding is to be used, and whether overflows saturate or wrap. The "quantize()" function applies the quantizer object to numbers, which are inputs to and outputs from arithmetic operations. For example, a quantization model of type signed fixed-point, with 16 total bits with one sign bit, 7 integer bits, and 8 fractional bits, rounding to the nearest representable number toward  $\alpha$  and handling overflow with saturation is defined as follows in MATLAB:

```
quant = quantizer("fixed"; "floor"; "saturate"; [16 8]);
Xq = quantize(quant; X);
```

This quantizer object is used to quantize an arbitrary numerical value "X". The resulting number "Xq" has a double floating-point representation in MATLAB, but can be exactly represented by a 16-bit fixed-point signed number with 7 integer and 8 fractional bits.

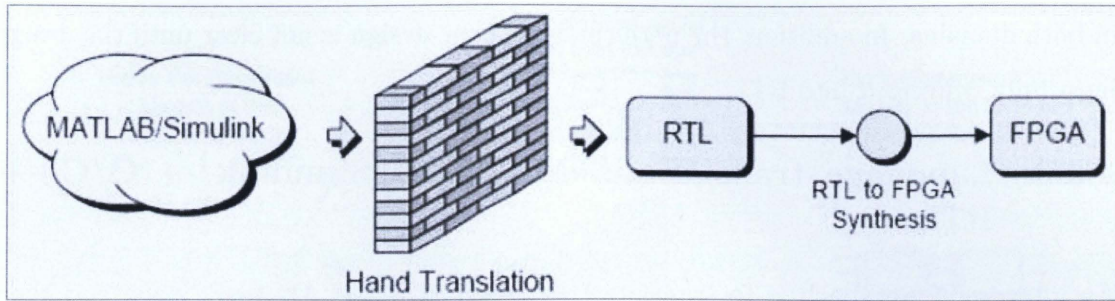
### 4.3 Techniques for translating MATLAB/Simulink to RTL

Once the DSP algorithm is proved in MATLAB or Simulink and the system architecture is defined at a high level of abstraction, the design has to be transformed into a physical implementation. Traditional techniques for removing the gap between the architectural and implementation domains typically fall into two main categories: language translation, and intellectual property (IP) instantiation and netlisting [23].

#### 4.3.1 Language translation: Hand translation

Hand translation involves no automation and yet, it is not uncommon to find design teams using this approach. There are numerous problems associated with this flow. The first key problem is that there is a gap between those working in the MATLAB/Simulink domain and hardware engineers working in the implementation domain and there few engineers who are experts in both domains. In addition, it is very time consuming to hand-code the RTL for a large, complex design. Another problem with this approach is that the implementation is device-specific. Even though the RTL synthesis (process by which an RTL is turned into a





**Figure 4.4:** Hand translation of MATLAB/Simulink to RTL

design implementation in terms of logic gates) tool is capable of targeting the RTL to any FPGA, achieving the best implementation requires that the RTL be coded with a specific device in mind.

### 4.3.2 Language translation: Auto interactive translation

There exists auto interactive translation that has the ability to go directly from MATLAB M-code representations into equivalent RTL. This process is called "algorithmic synthesis".

The way in which this works is as follows. Consider a mathematical function in MATLAB such as  $y = \text{fft}(x)$ . In MATLAB it is possible to present the input with an entire frame of data and immediately receive a corresponding frame of output without any time having elapsed, but there is no obvious corresponding implementation for such a construct. The solution is for the user to analyze the M-code associated with the DSP design and, for each abstract construct such as  $y = \text{fft}(x)$ , to assign the function a new name such as  $y = \text{myfft}(x)$  and to specify details as to a specific implementation to be associated with this function. For example, in the case of the "myfft", the user should select between different ways of implementing the algorithm; decide whether to perform buffering and storage using FIFOs, registers, or RAM blocks; and make micro-architecture decisions such as how many pipeline stages to use. Once all the constructs have been treated in this manner, the RTL representation can be generated using the library of forms-based functional specifications.

Same as hand translation, a major problem with this approach is the gap between the



algorithm developers and the hardware engineers and there few engineers who are experts in both domains. In addition, the performance of the design is not clear until the design has been fully converted into RTL.

### **4.3.3 Language translation: MATLAB/Simulink $\rightarrow$ C/C++ $\rightarrow$ RTL**

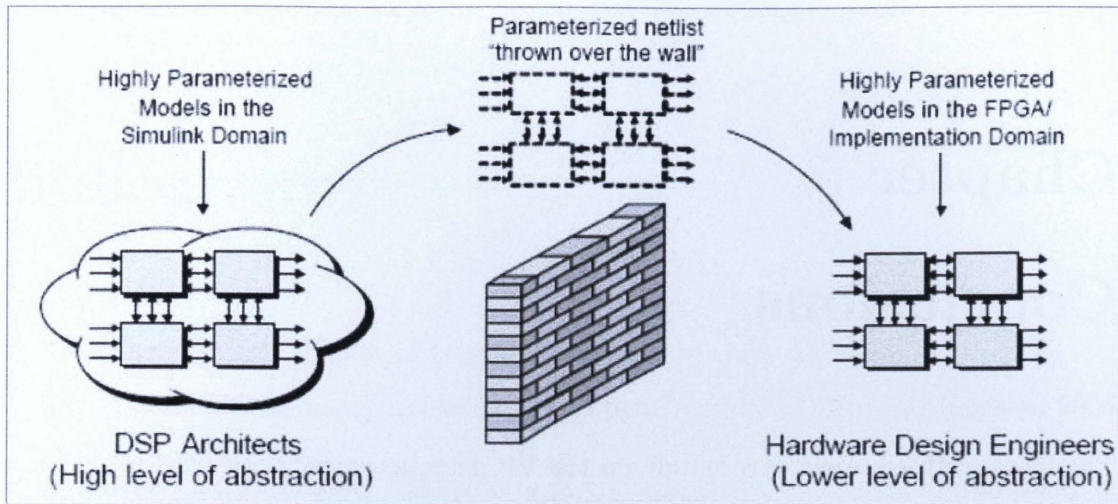
An alternative approach is to translate the design in MATLAB to C/C++ and then use a tool that translates a design from a C/C++ representation to either an equivalent RTL description or directly into an implementation-level netlist. Both MATLAB and Simulink are capable of generating C/C++ representations, but this is rarely used to create a hardware implementation of the DSP algorithms. Instead, it is common to hand translate the design from MATLAB/Simulink to C/C++ representation.

A concern with this approach is that C/C++ unlike HDL is inherently sequential in nature. In order to overcome this problem, representations have to be augmented with keywords called pragmas (pragmatic information) that specify concurrency, resource-sharing, and timing.

Some of these design flows take the C/C++ representation and synthesize it directly into an implementation- level and others use a synthesis/translation engine to generate RTL and then use the RTL synthesis to progress into the final implementation. These latter flows are preferred by some because RTL synthesis technology is extremely mature, whereas many C-based synthesis engines are not.

### **4.3.4 IP Instantiation**

A growing number of major FPGA companies have created libraries of intellectual property (IP) models, where each IP model represents some DSP operation such as FFT. Each model has an instantiation in Simulink which is called a blockset, and an identical instantiation in the FPGA domain. Once the design is proved in Simulink, it can be quickly and easily transferred into the implementation realm as a parameterized netlist. If the IP is RTL-based, it can be more easily used across all implementation technologies. Non-RTL-based IP on



**Figure 4.5:** IP instantiation and netlisting

the other hand is usually less portable. Similar to other techniques explained so far, a major problem with this technique is the gap between the algorithmic and implementation domains. Although the DSP engineers are supposed to be working at a high level of abstraction in the Simulink, in reality they have to make low-level implementation-specific decisions that are really in the scope of the hardware design engineers. For example, in the case of a FIR block, the DSP algorithm designer has to specify how to implement the delay line (RAM versus distributed registers), how much latency is to be involved, whether or not to use a shared MAC infrastructure, and so forth. It can be concluded that this method is the most automated approach, but one big issue with this solution is that it is vendor specific; that is, once a design is created in Simulink using a vendor-supplied library of IP blocks, the design cannot easily be used for FPGA of another vendor.



# Chapter 5

## Conclusions

The focus of this project was mainly on the VF detection algorithms used in AEDs, which therefore implies algorithms that are accurate and easy to implement in hardware. Chapter 1 gave a brief introduction on ECG signal characteristics, requirements of VF detection algorithms and AEDs. In Chapter 2 some of the existing VF detection algorithms was explained, and some details on their advantages and disadvantages was given. It was also discussed why most of these algorithms fall short for physical implementation. Chapter 3 proposed a novel method based on AR modeling and dominant pole frequencies for detection of VF. Details regarding AR modeling, dominant pole, decision criteria, and the experimental results were also discussed. The proposed AR modeling and dominant poles method have been shown to be effective for the classification of VF and NSR. This method has a high detection accuracy and is also suitable for real-time implementations into AEDs. Chapter 4 discussed different methods of implementing algorithms developed in MATLAB on FPGA. It also reviewed past attempts to remove the gap between the DSP design domain and physical implementation.



# Bibliography

- [1] "Statistics Canada, Causes of Death 2004."  
<http://cansim2.statcan.ca/cgi-win/cnsmcgi.pgm?regtk=&C2Sub=&> Released 2006.
- [2] A. Amann, K. Unterkofler. "Reliability of Old and New Ventricular Fibrillation Detection Algorithms for Automated External Defibrillators." *BioMedical Engineering Online* 4, 1-23 (2005)
- [3] A. Amann, R. Tratnig, K. Unterkofler. "Detecting Ventricular Fibrillation by Time-Delay Methods." *Biomedical Engineering, IEEE Transactions on*, Volume 54, Issue 1, Jan. 2007 Page(s):174 - 177
- [4] R. M. Rangayyan, *Biomedical signal analysis: A case-study approach*. Wiley-Interscience, New York (2002)
- [5] D. Ge, N. Srinivasan, and S. M. Krishnan, "Cardiac arrhythmia classification using autoregressive modeling." *BioMedical Engineering OnLine*, <http://www.biomedical-engineeringonline.com/content/1/1/5D>
- [6] D. Markenson, L. Pyles, S. Neish. "Ventricular Fibrillation and the Use of Automated External Defibrillators on Children." *PEDIATRICS* Volume 120, Number 5 (2007)
- [7] N. V. Thakor, Y. S. Zhu, and K. Y. Pan, Ventricular Tachycardia and Fibrillation Detection by a Sequential Hypothesis Testing Algorithm, *IEEE Trans. Biomed. Eng.*, 37, 837-843 (1990).

- [8] S. Chen, N. V. Thakor, and M. M. Mower, Ventricular fibrillation detection by a regression test on the autocorrelation function, *Medical & Biological Engineering & Computing*, 25, 241-249 (1987).
- [9] J. Pan and W. J. Tompkins, A Real-Time QRS Detection Algorithm, *IEEE Trans. Biomed. Eng.*, BME-32, 230-236 (1985).
- [10] I. Jekova, V. Krasteva. Real time detection of ventricular fibrillation and tachycardia. *PHYSIOL. Meas.* 25 (2004)
- [11] I. Jekova, Comparison of five algorithms for the detection of ventricular fibrillation from the surface ECG, *Physiol. Meas.* 21, 429-439 (2000).
- [12] H. Namvar, V. Shahidi. Cardiac Arrhythmias predictive Detection Methods with wavelet-SVD analysis and support vector machines. *Engineering in Medicine and Biology Society*, 2004. Volume 1, Issue , Page(s):365 - 368 (2004)
- [13] D. Li, Y. Zhu, W. Wu. "Ventricular Fibrillation Detection By Adaptive Algorithm" *Engineering in Medicine and Biology Society*, 1992. Vol.14. Proceedings of the Annual International Conference of the IEEE page(s): 665-666 (1992)
- [14] A. Rosado, J. Guerrero, M. Bataller, J. Chorro, Fast Non-invasive Ventricular Fibrillation Detection Method using Pseudo Wigner-Ville Distribution, *Computer in Cardiology*, p.237- 240 (2001)
- [15] A. Amann, R. Tratnig, K. Unterkofler. " A New Ventricular Detection Algorithm For Automated External Defibrillators", *Computers in Cardiology*, p.559-562 (2005)
- [16] S. Tavathia, R. Rangayyan, C. Frank, G. Douglas, Ladly, and Yuan-Ting Zhang. "Analysis of Knee Vibration Signals Using Linear Prediction" *IEEE Transactions On Biomedical Engineering*, Vol. 39, No. 9. September 1992
- [17] D. Clifford, Francisco Azuaje and Patrick E. McSharry. *Advanced Methods And Tools for ECG Data Analysis*, Artech House, Boston, (2006)

- [18] Physionet: <http://www.physionet.org>
- [19] S. Roy, O. Banerjee, An algorithm for Trading Off Quantization Error with Hardware Resources for MATLAB Based FPGA Design, IEEE Transactions On Computers, Vol. 54, pages: 886-896 (2005)
- [20] S. Roy, O. Banerjee, An algorithm for Converting Floating-Point Computations to Fixed-Point in MATLAB based FPGA design, Design Automation Conference, 2004. Page(s): 484 - 487( 2004 )
- [21] Mathworks Corp., MATLAB Technical Computing Environment, [www.mathworks.com](http://www.mathworks.com), (Jan. 2003)
- [22] Tejas M.Bhatt,Dennis McCain,MATLAB as an environment for FPGA Design, 42nd Design Automation Conference (DAC), (2005)
- [23] C. Eddington, Unite algorithm and hardware design flows. DSP DesignLine (2006)
- [24] R. Cofer, B. Harding, Fixed-point DSP and algorithm implementation, EE Times-India(October 2006)
- [25] Altera, DSP Design Flow User Guide, [http://www.altera.com/literature/ug/ug\\_dsp\\_design\\_flow.pdf](http://www.altera.com/literature/ug/ug_dsp_design_flow.pdf) (November 2006)
- [26] A. Krukowski, I. Kale, Simulink/MATLAB to VHDL Rout for Full Custom/FPGA Rapid Prototyping of DSP Algorithms, MATLAB DSP Conference (1999)
- [27] M. Rangayyan, *Numerical Analysis: A Mathematical Introduction*. Clarendon Press, Oxford. Chapters 4 and 6 (2002)
- [28] CW. Anderson, EA. Stolz,S. Shamssunder. " Multivariate autoregressive models for classification of spontaneous electroencephalographic signals during mental task. IEEE trans Biomed Eng 1998, 45:277-286



