以手機實現無線心電圖儀之設計

研究生：蔡維盈　　　　　　　　　　　　　　　　　　指導教授：張文輝、溫宏斌 博士

國立交通大學

電信工程研究所

摘要

因應高齡化社會的來臨，遠距醫療及健康照護已成為先進國家的新興服務產業。心電圖紀錄心臟搏動相關的電位變化資訊，常用於心血管疾病的診斷及治療。本論文開發一基於分散式訊源編碼架構的無線心電圖儀。分散式訊源編碼器的壓縮率及重建品質取決於訊源相關模型及校驗子生成機制，我們首先針對 MIT-BIH 心電圖資料褲，使用向量量化來建立與其匹配的訊源相關模型，並提出一基於索引層級的軟性輸出解碼演算法。系統模擬結果證實新的演算法大幅降低了資料量，並保留了心電圖於臨床上的重要資訊。在即時實作方面，我們結合心電圖感測器及執行訊號壓縮的嵌入式系統開發板，並藉由現有的行動手機網路將壓縮資料傳至遠端的伺服器以供訊號重建及監測。實作結果亦證實新的方法適用於無線健康照護系統的未來應用。

關鍵字：心電圖、心電圖壓縮、分散式訊源編碼、手機。

Design of a Mobile ECG System
Using a Mobile Telephone

Student: Wei-Ying Tsai Advisor: Dr. Wen-Whei Chang and Hung-Pin Wen

Institute of Communications Engineering,
National Chiao Tung University
Hsinchu, Taiwan

Abstract

In recent years, mobile telemedicine has been one of the emerging research topics. This thesis proposes using a mobile telephone to realize an electrocardiogram (ECG) signal monitoring system. In this system, we use a sensor to measure ECG signals and use an embedded system development board as the front-end device to compress the ECG signals. The proposed ECG signal compression technique is based on distributed source coding (DSC). The compressed ECG data are transmitted from the board to the back-end server via mobile cellular networks. At the server, we use the modified BCJR algorithm which integrates the symbol-level *a priori* information into the soft-output decoding algorithm. Through off-line simulation and real-time implementation, we test and verify the functions of each module of the ECG system. The results show that our proposed system is very feasible for future wireless health-care applications.

Keywords: Electrocardiogram (ECG), ECG compression, distributed source coding (DSC), mobile telephone.

Acknowledgements

I would like to acknowledge my advisors, Prof. Wen-Whei Chang and Prof. Hung-Pin Wen, for their valuable guidance throughout my research. I would also like to express my appreciations to Prof. Suh-Yin Lee for her kind helps and useful suggestions. Next, I wish to thank the members in Speech Communication Lab, especially Hung-Tsai Wu, for their useful discussions. Also, I am grateful to my bosom friends, Shun-Ting Chang and Chi Jeng, for being with me through the years. Last but not least, I would like to thank my parents for their encouragements and supports, especially financial support.

This thesis is dedicated to the above people whom I cherish forever and ever.

Contents

List of Tables

List of Figures

Chapter 1 Introduction

Wireless patient monitoring has been of recent interest to researchers aiming to develop *ubiquitous* health-care systems able to provide personalized medical treatment continuously and remotely []. To realize such systems, physiological signals such as ECG are measured and transmitted wirelessly to the remote server for monitoring or analysis. One of the emerging issues is how to exploit the mobile communication technologies. Due to the bandwidth limitations of mobile cellular networks, a compression method is used to reduce the large amount of data. Moreover, the method must be simple since the sensor is constrained by its low battery power. This thesis presents a mobile ECG system based on distributed source coding (DSC). In the system, measured ECG is transmitted from the sensor to a front-end device for data compression and then transmitted to a back-end server via mobile cellular network. Finally, the ECG signals are reconstructed and monitored on the screen.

1.1 Motivation

Like most countries in the world, Taiwan is aging with an increase of senior populations. With the increase of the elderly (65 years and over), the health-care issues are becoming more and more significant. Although traditional medical treatments such as face-to-face consultant cannot be replaced, some treatments can be done more efficiently with the biotelemetry. According to Table 1.1 [2], heart diseases account for 10.8% of all deaths in 2010. The ECG is a graphical representation of the electrical activity in the heart and is useful for cardiac disease diagnosis. Goal of the ECG system is designed for monitoring ECG signals and detecting abnormalities of the heart. Whenever a patient is under risk, the system could detect the abnormalities and notify the hospital or clinical center. Electrocardiogram (ECG) is measured by an appropriate sensor and fed into a front-end device. The most important task of the device is to compress and transmit ECG to a remote back-end server for further analysis. Wireless technologies can be drawn in to increase the mobility of the system. Among them, ZigBee and Bluetooth are insufficient since they only provide short-distance transmission. Although WLAN (IEEE 802.11) provides RF coverage up to several hundred meters, they are not well suited for developing biotelemetry applications [3]. Once the front-end device is out of the service range, it must be re-connected manually by users. In contrast, the mobile cellular network is a better choice [4, 5]. Thanks to the wide coverage provided by the mobile cellular network, a *ubiquitous* health-care system can be realized. In this work, ECG is delivered to a location (base station) through the mobile cellular network and then to the hospital through the Internet.

Table 1.1 Taiwan’s ten leading causes of death in 2010 [2]

Rank	Causes of Death	Deaths (person)	Crude mortality rate 0/0000	Percentage of total deaths(%)
1	Malignant neoplasms	41,046	177.4	28.4
2	Heart disease	15,675	67.7	10.8
3	Cerebrovascular disease	10,134	43.8	7.0
4	Pneumonia	8,909	38.5	6.2
5	Diabetes mellitus	8,211	35.5	5.7
6	Accidents and adverse effects	6,669	28.8	4.6
7	Chronic lower respiratory diseases	5,197	22.5	3.6
8	Chronic liver disease and cirrhosis	4,912	21.2	3.4
9	Hpertensive disease	4,174	18	2.9
10	Nephritis, nephrotic syndrome, and nephrosis	4,105	17.7	2.8

With the advanced development of mobile cellular network ranging from the global system for mobile communications (GSM) to general packet radio service (GPRS), to 3G, and then to 4G, more services can be designed for home health-care applications. Most GPRS devices cannot transmit voice and data simultaneously and hence they are not used here. Both 3G and 4G support simultaneous voice and data transmission. Since 4G is not available to all users, 3G is adequate and is chosen in this work.

Due to the limited transmission bandwidth, a coding technique that provides channel efficiency is needed. Constrained by the battery power of the front-end device, the technique must also be simple. This thesis introduces the distributed source coding (DSC) scheme, in which two or more correlated sources are encoded separately and decoded jointly. DSC exploits the inter-user correlation without the high cost of inter-user communication. By viewing the heartbeat segments as correlated sources, it is feasible to use DSC to compress the ECG signals.

1.2 Related Works on ECG Compression

Most of the ECG compression methods adopt one-dimensional (1-D) representations for ECG signals [6], including direct waveform coding, transform coding, and parameter extraction methods. However, since the ECG signals have both sample-to-sample (intra-beat) and beat-to-beat (inter-beat) correlation, some 2-D compression techniques have been proposed for higher compression ratios [7, 8]. These methods start with a preprocess procedure that converts 1-D to 2-D representations through the combined use of QRS detection and period normalization. Afterwards, the conventional image codec such as the JPEG standard [9] can be used to compress these resulting 2-D arrays. However, JPEG codec suffers from a complex encoder and hence may not be suited for wireless sensor network. Previous work in [10] proposed a beat-based ECG compression method based on gain-shape vector quantization (GSVQ). It achieved high compression ratio at the expense of relatively large codebook size, which induced high encoding complexity. In [1], a joint source-channel coding scheme was presented to exploit the inter-beat correlations. However, the bit-level iterative decoder leads to sup-optimal performance with high decoding complexity.

The performance of DSC highly depends on the source correlation model. In this work, a simple compression method based on GSVQ is used to construct the ECG correlation model and the resulting quantizer indexes are fed to DSC. In addition, recursive systematic convolutional code (RSC) is used in the implementation of DSC. For the convolutional code, there are two different decoding algorithm, including Viterbi algorithm [11] and BCJR algorithm [12]. While Viterbi algorithm is a maximum likelihood (ML) decoding that minimizes the probability of codeword error, BCJR algorithm is a maximum *a posteriori* probability (MAP) decoding that minimizes the probability of information bit error. The DSC decoder used in this work is a modified BCJR algorithm that performs better than Viterbi algorithm and classical BCJR algorithm by integrating the symbol-level *a priori* information into the MAP process.

1.3 Objective

Goal of this work is to design and implement a mobile ECG system using a coding technique able to provide errors robustness, channel efficiency and low encoding complexity. Also proposed is the compression method based on DSC to reduce the large amount of ECG data. Besides, an ECG correlation model based on GSVQ is presented to match the DSC application. To better exploit the ECG correlations, a combined use of QRS detection and period normalization is applied to obtain 2-D ECG arrays prior to GSVQ.

To develop a *ubiquitous* health-care system, the 3G mobile cellular network is involved. ECG signals are measured by a wireless sensor and transmitted to an embedded system development board. The board serves as the front-end device which receives, compresses incoming ECG signals and then transmits them to a back-end server. Compressed ECG data are transmitted to the telephone via Bluetooth and to the server via mobile cellular network and Internet. A commercially available Bluetooth-enabled 3G mobile telephone is used in this work. Finally, the ECG signals can be reconstructed and displayed on the monitor.

1.4 Thesis Organization

The rest of this thesis is organized as follows. In Chapter 2, the DSC framework will be reviewed, including the DSC theory and its practical implementation. Chapter 3 introduces the ECG measurement and correlation modeling. Details of the hardware architecture and the software implementation of the system implementation are given in Chapter 4. Chapter 5 presents the results of offline simulation and real-time implementation. Finally, Chapter 6 gives our conclusions.

Chapter 2 Distributed Source Coding Framework

This chapter presents a comprehensive introduction to distributed source coding (DSC), including the DSC theorem and its practical implementation. Distributed source coding, also known as Slepian-Wolf source coding (SWC), was originated from Slepian-Wolf theorem [13] in 1970s. In 2003, S.S. Pradhan *et al.* first proposed a practical way to design SWC codes using syndromes [14]. As we will see later, channel coding plays an important role in DSC. In this thesis, recursive systematic convolutional code (RSC) is applied in conjunction with a modified BCJR algorithm for symbol decoding of binary convolutional code. Unlike conventional BCJR algorithm that decodes one bit at a time, our previous work [15] proposed an index-based BCJR algorithm which outperforms bit-based BCJR algorithm. To be consistent, the syndrome former (SF) should also be modified to operate at the index level.

2.1 Distributed Source Coding Theorem

The Slepian-Wolf theorem states that, given two or more correlated information sources, the lossless compression of these sources which do not communicate with one another can reach the same theoretical entropy as they communicates. The above statement is true only if the correlated-sources are well-modeled and the compressed data are jointly decoded. Fig. 2.1 [16] shows the compression system with two correlated sources which do not communicate with each other. The most distinctive feature of DSC is that the complexity can be shifted from the encoder to the decoder. With this property, DSC is suitable to the application of wireless sensor networks (WSN), in which sensors require low computational complexity.

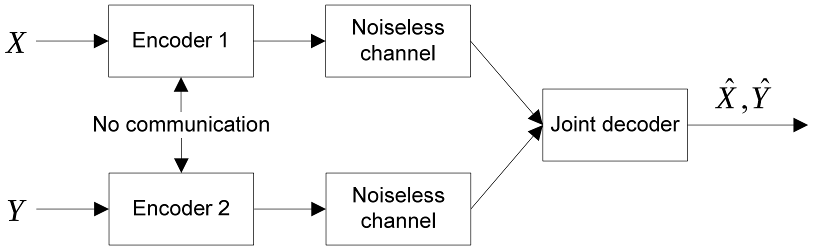


Fig. 2.1 The flow of lossless compression of two correlation source [16].

Assume that X and Y are two independent and identically distributed (i.i.d.) binary random variables. Let $H(X)$ and $H(Y)$ be the entropies of sources X and Y respectively, and let $H(X,Y)$ be their joint entropy. The Shannon's coding theorem declares that $H(X,Y) \le H(X)+H(Y)$, which implies that a rate given by $H(X,Y)$ is sufficient if X and Y are jointly encoded. The DSC theorem further proves that, given that the correlation between X and Y is known and Y is available at the decoder side, the lossless coding rate $H(X,Y)$ can still be achieved if X and Y are separately encoded and jointly decoded. To be more precisely, Y is encoded into $H(Y)$ bits per sample (bps), and the rate of X can be reduced to $H(X\,|\,Y)=H(X,Y)-H(Y)$. The corner point of A, in Fig. 2.2, shows the case where SWC codes are designed to approach the rate $H(X,Y)=H(Y)+H(X\,|\,Y)$. Fig. 2.2 also shows the achievable rate region of DSC which is given by $R_x \geq H(X\,|\,Y)$, $R_y \geq H(Y\,|\,X)$, and $R_x + R_Y \geq H(X,Y)$.

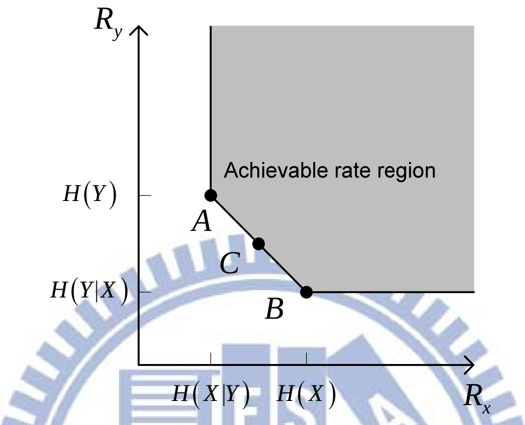


Fig. 2.2 Rate region for two sources.

Constructive methods to designing SWC codes can be classified into two categories: asymmetric and symmetric. Designing code to approach the corner of points A or B refers to as asymmetric, while to approach the mid-point C is regarded as symmetric. Our work falls into the asymmetric DSC methods.

2.2 DSC System Implementation

The basic concept in DSC theorem is binning, which refers to partitioning all possible source outputs into cosets (or bins). Consider an i.i.d. n-bit binary source and a rate k/n linear channel code. Then we have 2^n virtual codewords, whereas there are only 2^k valid codewords. The concept of binning is to partition 2^n virtual codewords into 2^{n-k} disjoint cosets. For each coset, there are 2^k codewords and an $(n-k)$-bit coset index (syndrome) is assigned. Proceeding with the binning, it should be ensured that minimum Hamming distances between any two codewords within the same coset are equal.

In DSC theorem a compression ratio $n:n-k$ is achieved by transmitting the $(n-k)$ -bit syndrome rather than the n-bit input itself. Fig. 2.3 illustrates the structure of DSC using syndromes [17]. The source encoder at the transmitter can be viewed as a syndrome former

(SF), which maps an n-bit input sequence X^n to an $(n-k)$-bit syndrome S^{n-k}. The source decoder at the receiver is composed of an inverse syndrome former (ISF) and a virtual channel decoder. The output from the ISF is added (in modulo 2 addition) to the side information Y^n, and the result is fed to the virtual channel decoder to perform channel decoding. If the channel code is well-designed, the reconstructed sequence $\hat{X}^n$, retrieved by adding (in modulo 2 addition) the output from the virtual channel decoder to the output from the ISF, can be very close to the original sequence X^n. The correlation between Y^n and X^n is usually modeled by a virtual binary symmetric channel (BSC) as $Y^n = X^n \oplus Z^n$, where Z^n can be taken as virtual bit error sequences. In other words, Y^n is a noisy version of X^n.

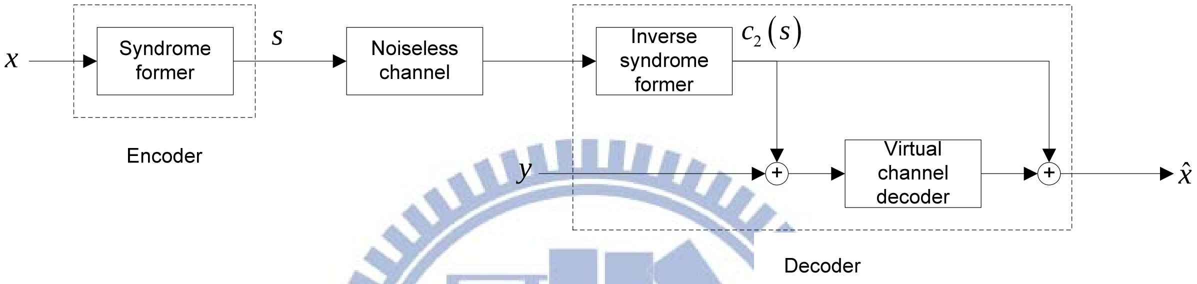


Fig. 2.3 The structure of the DSC encoder and decoder [17].

A more detailed process of DSC is described as below. An n-bit input source x is passed through the SF and a syndrome s is generated in a way that x belongs to the coset indexed by syndrome s, or $x = c_1(s)$. Assume that the syndrome s is transmitted to the receiver through a noiseless channel. The ISF uses the received syndromes to choose any codeword $c_2(s)$ from the coset indexed by the syndrome s. Let z be the virtual binary bit error sequence which relates input x and side information y through $y = x \oplus z$. The input of the virtual channel decoder is given by

$$y \oplus c_2(s) = x \oplus z \oplus c_2(s) = c_1(s) \oplus c_2(s) \oplus z. \qquad (1)$$

According to channel coding theorem, applying modulo 2 addition to any two codewords in the same coset leads to a valid codeword in the coset indexed by the syndrome 0, that is, $c_1(s) \oplus c_2(s) = c_3(0)$. Therefore, we have

$$y \oplus c_2(s) = c_3(0) \oplus z. \qquad (2)$$

It should be noticed that $c_3(0) \oplus z$ represents a noisy codeword (a valid codeword plus some bit errors), implying that a powerful channel code can recover $c_3(0)$. Afterwards, the original input source x can be recovered by adding $c_2(s)$ to $c_3(0)$ as follows:

$$c_2(s) \oplus c_3(0) = c_2(s) \oplus y \oplus c_2(s) \oplus z = y \oplus z = x. \tag{3}$$

From (3), we know that it is critically essential to build a good correlation model and use it to choose a powerful channel code.

The kernel of SF is to design a channel code according to the source correlation model. Consider a (n,k) convolutional code with $k \times n$ generator matrix G and $(n-k) \times n$ parity check matrix H. According to channel coding theorem, we have $GH^T = 0$ (null vector), where H^T denotes the transpose of H. With regard to SF construction, the syndrome s can be computed by using H^T in the form of

$$s = xH^T. \tag{4}$$

The operation of the ISF can be described by a matrix $(H^{-1})^T$ which satisfies

$$(H^{-1})^T H^T = I, \tag{5}$$

where I is an identity matrix. The output from ISF can be any codeword $c_2(s)$ in the coset indexed by the syndrome s and is given by

$$c_2(s) = s(H^{-1})^T = xH^T(H^{-1})^T. \tag{6}$$

2.3 Convolutional-Based Syndrome Former

When applying the DSC, both the compression ratio and the quality of reconstructed signal depend on the choice of channel code used for SF. In practical DSC systems, the binning is often constructed using a (n,k,m) convolutional code, where n is the number of output bits, k is the number of input bits and m is the memory order (e.g., the number of shift registers). Basically, the convolutional encoder can be modeled as a finite-state machine (FSM) with its states defined by the contents of the shift registers. The dynamic behavior of the encoder in time can be represented by a trellis diagram. Fig. 2.4 shows the state diagram and trellis diagram of a $(2,1,2)$ encoder with generator matrix $G(D) = \left[1+D^2 \quad 1+D+D^2\right]$. Through the trellis diagram, we observe that the encoding process of convolutional code utilizes the information of the past codewords. This makes the convolution code more efficient from the viewpoint of error correction. In addition, the construction of trellis diagram allows the decoder to decode the sequences in linear time.

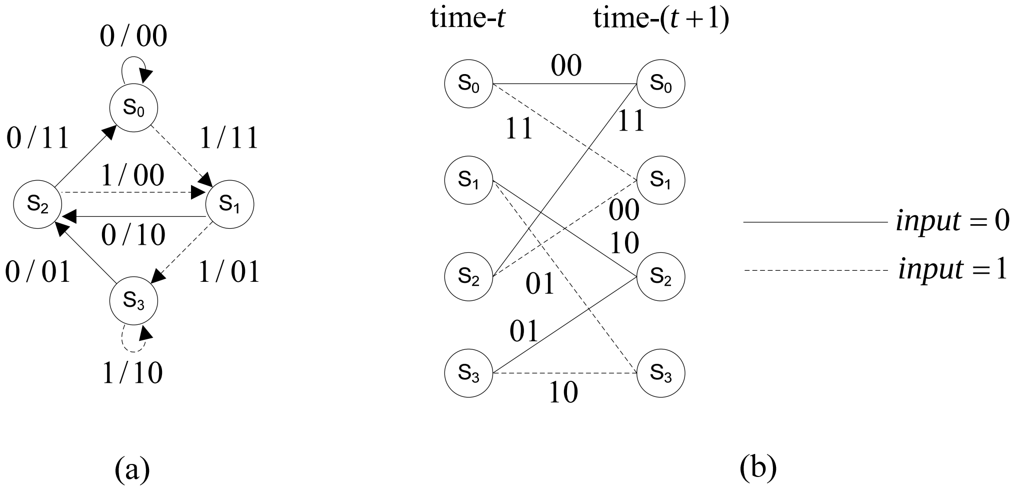


Fig. 2.4 (a) State diagram and (b) trellis diagram.

2.4 Virtual Channel Decoder

In 1974, Bahl, Cocke, Jelinek, and Raviv introduced a maximum *a posteriori* probability (MAP) decoder that can be applied to any linear block code or convolutional code [12]. In the MAP decoder the probability of information bit error is minimized. However, the usefulness of conventional BCJR algorithm may be restricted since it is derived based on a bit-level trellis and only exploits the bit-level source *a priori* probability (APP). A better solution is to develop a decoding algorithm operating in the index-level.

Consider a (2,1,2) convolutional code with generator polynomial $G(D) = \left[1+D^2 \quad 1+D+D^2\right]$. The bit-level trellis used in classical BCJR algorithm is shown in Fig. 2.5. We observe that the channel codeword and the next state are determined by the input bit and the current state. For applying an index-based BCJR algorithm, we should sectionalize the bit-level trellis according to the index length. To achieve this goal, the bit-level trellis is merged by viewing M consecutive bits as a unit. Fig. 2.6 shows the sectionalized trellis diagram where two stages ($M = 2$) of the bit-level trellis are merged.

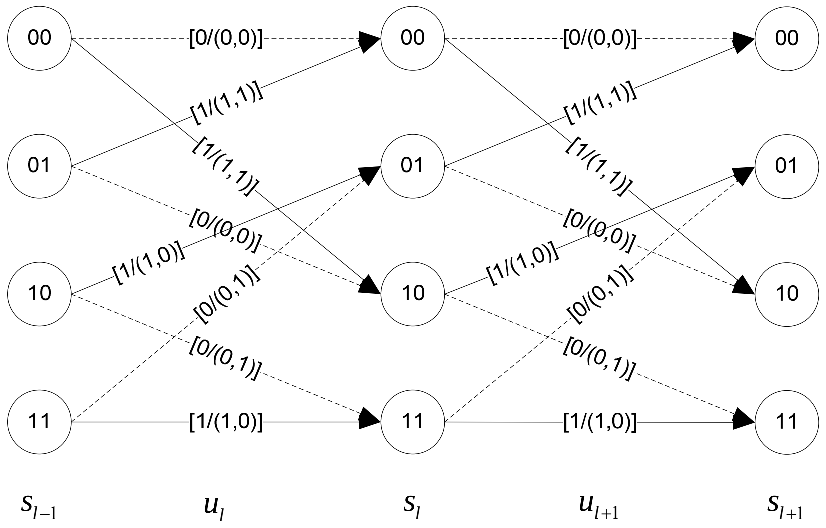


Fig. 2.5 Bit-level trellis diagram.

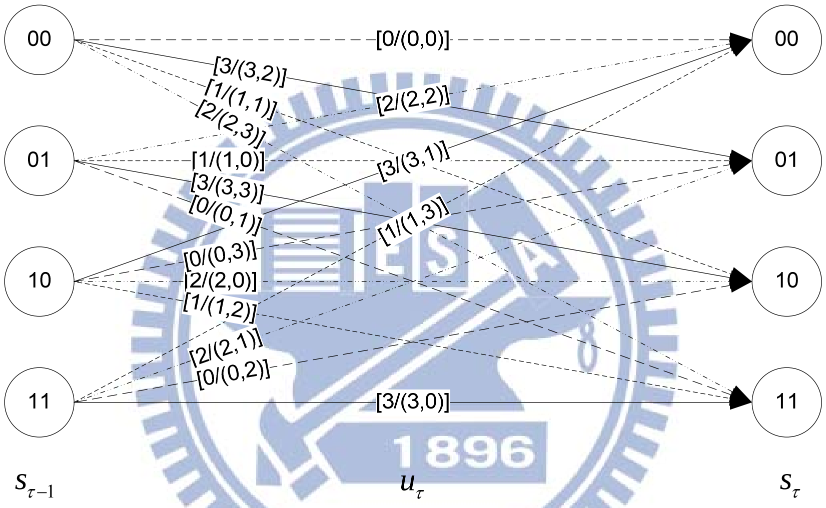


Fig. 2.6 Sectionalized trellis diagram.

Through the sectionalized trellis diagram, the *a priori* information on index-level can be fully exploited in the decoding process. At time instant τ, the input symbol is u_τ, the state of the encoder is σ_τ, and the channel codeword is r_τ. Taking the sectionalized trellis state σ_τ into consideration, we rewrite the *a posteriori* probability of a symbol $u_\tau = i$ given the received sequence $\tilde{R}_1^T = (\tilde{r}_1, \tilde{r}_2, ..., \tilde{r}_T)$ as follows:

$$\begin{aligned} P(u_\tau = i \mid \tilde{R}_1^T) &= \sum_{\sigma_\tau} \frac{P(u_\tau = i, \sigma_\tau, \tilde{R}_1^T)}{P(\tilde{R}_1^T)} \\ &= C \sum_{\sigma_\tau} P(u_\tau = i, \sigma_\tau, \tilde{R}_1^T), \end{aligned} \tag{7}$$

where C is a normalizing factor. The joint probability term can be further decomposed using the Bayes theorem as

$$\begin{aligned} P(u_\tau = i, \sigma_\tau, \tilde{R}_1^T) &= P(u_\tau = i, \sigma_\tau, \tilde{R}_1^\tau) \cdot P(\tilde{R}_{\tau+1}^T \mid u_\tau = i, \sigma_\tau, \tilde{R}_1^T) \\ &\triangleq \alpha_\tau^i(\sigma_\tau) \cdot \beta_\tau^i(\sigma_\tau), \end{aligned} \tag{8}$$

where forward metric $\alpha_\tau^i(\sigma_\tau)$ and backward metric $\beta_\tau^i(\sigma_\tau)$ can be computed in a recursive form as follows:

$$\begin{aligned} \alpha_\tau^i(\sigma_\tau) &= P\left(u_\tau = i, \sigma_\tau, \tilde{R}_1^\tau\right) \\ &= \sum_{\sigma_{\tau-1}} \sum_j P\left(u_\tau = i, \sigma_\tau, u_{\tau-1} = j, \sigma_{\tau-1}, \tilde{R}_1^{\tau-1}, \tilde{r}_\tau\right) \\ &= \sum_{\sigma_{\tau-1}} \sum_j P\left(u_{\tau-1} = j, \sigma_{\tau-1}, \tilde{R}_1^{\tau-1}\right) \cdot P\left(u_\tau = i, \sigma_\tau, \tilde{r}_\tau \mid u_{\tau-1} = j, \sigma_{\tau-1}, \tilde{R}_1^{\tau-1}\right) \\ &\triangleq \sum_{\sigma_{\tau-1}} \sum_j \alpha_{\tau-1}^j(\sigma_{\tau-1}) \cdot \gamma_{i,j}(\tilde{r}_\tau, \sigma_\tau, \sigma_{\tau-1}), \end{aligned} \tag{9}$$

$$\begin{aligned} \beta_{\tau-1}^i\left(\sigma_{\tau-1}\right) &= P\left(\tilde{R}_\tau^T \mid u_{\tau-1} = i, \sigma_{\tau-1,} \tilde{R}_1^{\tau-1}\right) \\ &\triangleq \sum_{\sigma_\tau} \sum_j \beta_\tau^j\left(\sigma_\tau\right) \cdot \gamma_{j,i}\left(\tilde{r}_\tau, \sigma_\tau, \sigma_{\tau-1}\right). \end{aligned} \tag{10}$$

An encoder starts at $\sigma_0 = 0$ and terminates at $\sigma_T = 0$, the recursion will start with the following initial conditions:

$$\alpha_0(\sigma_0) = \begin{cases} 1, & \sigma_0 = 0 \\ 0, & \sigma_0 \neq 0 \end{cases} \tag{11}$$

and

$$\beta_T(\sigma_T) = \begin{cases} 1, & \sigma_T = 0 \\ 0, & \sigma_T \neq 0 \end{cases}. \tag{12}$$

Using the Markov property of the symbols and the memoryless assumption of the virtual binary symmetric channel, we have

$$\begin{aligned} \gamma_{i,j}\left(\tilde{r}_\tau, \sigma_\tau, \sigma_{\tau-1}\right) &= P\left(u_\tau = i, \sigma_\tau, \tilde{r}_\tau \mid u_{\tau-1} = j, \sigma_{\tau-1}, \tilde{R}_1^{\tau-1}\right) \\ &= P\left(u_\tau = i, \sigma_\tau \mid u_{\tau-1} = j, \sigma_{\tau-1}, \tilde{R}_1^{\tau-1}\right) \\ &\quad \cdot P\left(\tilde{r}_\tau \mid u_\tau = i, \sigma_\tau, u_{\tau-1} = j, \sigma_{\tau-1}, \tilde{R}_1^{\tau-1}\right) \\ &\approx P\left(u_\tau = i, \sigma_\tau \mid \sigma_{\tau-1}\right) \cdot P\left(\tilde{r}_\tau \mid u_\tau = i, \sigma_\tau\right). \end{aligned} \tag{13}$$

In (13), the term $P\left(u_\tau = i, \sigma_\tau \mid \sigma_{\tau-1}\right)$ can be viewed as *a priori* information and the term $P\left(\tilde{r}_\tau \mid u_\tau = i, \sigma_\tau\right)$ accounts for the channel-related information.

As described in (2), the virtual channel decoder of the DSC system is designed to decode a noisy codeword $c_3(0) \oplus z$. To apply the BCJR algorithm, we may view the received codeword at time instant τ as $\tilde{r}_\tau \triangleq r_\tau \oplus z_\tau$, where $r_\tau \triangleq c_{3,\tau}(0)$. By observing the sectionalized trellis diagram, we find that the channel codeword r_τ can be uniquely determined by u_τ and σ_τ, that is, $r_\tau \triangleq f(u_\tau, \sigma_\tau)$. Let the input of SF be x_τ and the output of SF be $S^{(\tau)}$ at time instant τ. We have $c_{1,\tau}(S^{(\tau)}) \oplus c_{2,\tau}(S^{(\tau)}) = c_{3,\tau}(0)$. The term $P(u_\tau = i, \sigma_\tau \mid \sigma_{\tau-1})$ can be further rewritten as (14).

$$\begin{aligned} P(u_\tau = i, \sigma_\tau \mid \sigma_{\tau-1}) &= P\left(r_\tau = f(u_\tau = i, \sigma_\tau) \mid \sigma_{\tau-1}\right) \\ &= P\left(c_{3,\tau}(0) = f(u_\tau = i, \sigma_\tau) \mid \sigma_{\tau-1}\right) \\ &= P\left(c_{1,\tau}(S^{(\tau)}) = f(u_\tau = i, \sigma_\tau) \oplus c_{2,\tau}(S^{(\tau)}) \mid \sigma_{\tau-1}\right) \\ &= P\left(x_\tau = f(u_\tau = i, \sigma_\tau) \oplus c_{2,\tau}(S^{(\tau)}) \mid \sigma_{\tau-1}\right). \end{aligned} \tag{14}$$

From (14), it is observed that the *a priori* information can be exploited by taking into consideration the current state σ_τ, the input index x_τ, and the output of ISF $c_{2,\tau}(S^{(\tau)})$. The channel-related term $P(\tilde{r}_\tau \mid u_\tau = i, \sigma_\tau)$ can be reduced to

$$P\left(\tilde{r}_\tau \mid \sigma_{\tau-1}, u_\tau = i\right) = P(\tilde{r}_\tau \mid r_\tau = f(u_\tau = i, \sigma_{\tau-1})) = (1-\varepsilon)^{n-d} \cdot \varepsilon^{d}, \tag{15}$$

where ε is the bit error rate in the virtual channel and $d = d_H\left(\tilde{r}_\tau, r_\tau = f(x_\tau = i, \sigma_\tau)\right)$ is the Hamming distance between the received index $\tilde{r}_\tau$ and corresponding codeword r_τ. In summary, $\alpha_\tau^i(\sigma_\tau)$ and $\beta_\tau^i(\sigma_\tau)$ can be rewritten as

$$\alpha_\tau^i(\sigma_\tau) \approx P\left(\tilde{r}_\tau \mid u_\tau = i, \sigma_\tau\right) \sum_{\sigma_{\tau-1}} \sum_j \alpha_{\tau-1}^j(\sigma_{\tau-1}) \cdot P\left(u_\tau = i, \sigma_\tau \mid \sigma_{\tau-1}\right), \tag{16}$$

$$\beta_{\tau-1}^i\left(\sigma_{\tau-1}\right) \approx \sum_{\sigma_\tau} \sum_j \beta_\tau^j\left(\sigma_\tau\right) \cdot P\left(u_\tau = j, \sigma_\tau \mid \sigma_{\tau-1}\right) \cdot P\left(\tilde{r}_\tau \mid u_\tau = j, \sigma_\tau\right). \tag{17}$$

To avoid the numerical representation problem, both $\alpha_\tau^i(\sigma_\tau)$ and $\beta_\tau^i(\sigma_\tau)$ are normalized as follows:

$$\alpha_\tau^i(\sigma_\tau) = \frac{\sum_{\sigma_{\tau-1}} \gamma_{i,j}\left(\tilde{r}_\tau, \sigma_\tau, \sigma_{\tau-1}\right) \cdot \alpha_{\tau-1}^j(\sigma_{\tau-1})}{\sum_{\sigma_\tau} \sum_{\sigma_{\tau-1}} \gamma_{i,j}\left(\tilde{r}_\tau, \sigma_\tau, \sigma_{\tau-1}\right) \cdot \alpha_{\tau-1}^j(\sigma_{\tau-1})}, \tag{18}$$

$$\beta_{\tau-1}^i(\sigma_{\tau-1}) = \frac{\sum_{\sigma_\tau} \gamma_{j,i}\left(\tilde{r}_\tau, \sigma_\tau, \sigma_{\tau-1}\right) \cdot \beta_\tau^j(\sigma_\tau)}{\sum_{\sigma_{\tau-1}} \sum_{\sigma_\tau} \gamma_{j,i}\left(\tilde{r}_\tau, \sigma_\tau, \sigma_{\tau-1}\right) \cdot \beta_\tau^j(\sigma_\tau)}. \tag{19}$$

The procedure of BCJR algorithm can be summarized as follow:

1. Initialize forward metric $\alpha_\tau^i(\sigma_\tau)$ and backward metric $\beta_\tau^i(\sigma_\tau)$ by (11) and (12).
2. Once the received codeword $\tilde{r}_\tau$ is available, we calculate branch metric by (13), (14), and (15), and update the forward metric by (18).
3. After the entire sequence $\tilde{R}_1^T$ is received, update the backward metric $\beta_\tau^i(\sigma_\tau)$ by (19), and the *a posteriori* probability of each symbol $P(u_\tau = i \mid \tilde{R}_1^T)$ is obtained.
4. A symbol estimate $\hat{u}_\tau$ that has the maximum *a posteriori* probability is obtained and is used to determine the codeword $r_\tau \triangleq c_{3,\tau}(0)$. Finally, the signal estimate $\hat{x}_\tau$ is computed by (3).

2.5 Construction of Index-Based Syndrome Former

To be consistent with the index-based BCJR algorithm, the SF should also be designed to operate at the index level. J. Li *et al*. proposed a scheme exploiting rate-compatible punctured convolutional (RCPC) code [18] and is described as follows. Consider a non-recursive polynomial:

$$A(D) = a_0 + a_1 D + a_2 D^2 + a_3 D^3 + \ldots + a_n D^n, \tag{20}$$

where $(a_0, a_1, a_2, a_3, \ldots, a_n)$ is a binary sequence and *D* is the delay element. The binary sequence can be divided into t sub-sequences by polyphase decomposition. In other words, each $a_i, 0 \le i \le n$, is divided by t and we get the remainder. Therefore, we have the following subsequences:

$$(a_0, a_t, a_{2t,\ldots})$$
$$(a_1, a_{t+1}, a_{2t+1,\ldots})$$
$$\ldots$$
$$(a_{t-1}, a_{2t-1}, a_{3t-1,\ldots})$$

and then we have

$$A_j^{(t)}(D) = \sum_{i=0}^{n} a_{it+j} D^i,\ j = 0,1,\ldots,t-1. \tag{21}$$

By (21), (20) can be rewritten as:

$$A(D) = \sum_{j=0}^{t-1} D^j A_j^t(D^t). \tag{22}$$

The *t*-cyclic elementary matrix of $A(D)$ can be defined as:

$$\widetilde{A}^{(t)}(D)=\begin{bmatrix} A_0(D) & A_1(D) & \dots & A_{t-2}(D) & A_{t-1}(D) \\ DA_{t-1}(D) & A_0(D) & \dots & A_{t-3}(D) & A_{t-2}(D) \\ \dots & \dots & \dots & \dots & \dots \\ DA_1(D) & DA_2(D) & \dots & DA_{t-1}(D) & A_0(D) \end{bmatrix}. \qquad (23)$$

The *t*-cyclic elementary matrix is an important tool for deriving the generator matrix for sectionalized trellis diagram. Assume the generator matrix G of a rate $2/3$ RSC is

$$G=\begin{bmatrix} 1 & 0 & \dfrac{1+D}{1+D^2} \\ 0 & 1 & \dfrac{1+D+D^2}{1+D^2} \end{bmatrix}. \qquad (24)$$

Since two bits are merged to construct a rate $4/6$ RSC, we now define 2-cyclic elementary matrix for each polynomial in G. First, define G as

$$G=\begin{bmatrix} U_1(D) & U_2(D) & \dfrac{U_3(D)}{V(D)} \\ U_2(D) & U_1(D) & \dfrac{U_4(D)}{V(D)} \end{bmatrix}, \qquad (25)$$

where

$$\begin{aligned} U_1(D) &\triangleq 1 \\ U_2(D) &\triangleq 0 \\ U_3(D) &\triangleq 1+D \\ U_4(D) &\triangleq 1+D+D^2 \\ V(D) &\triangleq 1+D^2. \end{aligned} \qquad (26)$$

For polynomials in (26), the corresponding 2-cyclic elementary matrices are

$$\begin{aligned}
\tilde{U}_1^{(2)}(D) &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\
\tilde{U}_2^{(2)}(D) &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \\
\tilde{U}_3^{(2)}(D) &= \begin{bmatrix} 1 & 1 \\ \mathrm{D} & 1 \end{bmatrix} \\
\tilde{U}_4^{(2)}(D) &= \begin{bmatrix} 1+\mathrm{D} & 1 \\ \mathrm{D} & 1+\mathrm{D} \end{bmatrix} \\
\tilde{V}^{(2)}(D) &= \begin{bmatrix} 1+\mathrm{D} & 0 \\ 0 & 1+\mathrm{D} \end{bmatrix}.
\end{aligned} \tag{27}$$

From (27), the 2-cyclic elementary matrix for $\frac{U_3(D)}{V(D)}$ and $\frac{U_4(D)}{V(D)}$ in G can be defined as

$$\begin{aligned}
\tilde{U}_3^{(2)}(D)\cdot\tilde{V}^{(2)}(D)^{-1} &= \begin{bmatrix} 1 & 1 \\ \mathrm{D} & 1 \end{bmatrix} \bullet \begin{bmatrix} \frac{1}{1+\mathrm{D}} & 0 \\ 0 & \frac{1}{1+\mathrm{D}} \end{bmatrix} = \begin{bmatrix} \frac{1}{1+\mathrm{D}} & \frac{1}{1+\mathrm{D}} \\ \frac{\mathrm{D}}{1+\mathrm{D}} & \frac{1}{1+\mathrm{D}} \end{bmatrix} \\
\tilde{U}_4^{(2)}(D)\cdot\tilde{V}^{(2)}(D)^{-1} &= \begin{bmatrix} 1+\mathrm{D} & 1 \\ \mathrm{D} & 1+\mathrm{D} \end{bmatrix} \bullet \begin{bmatrix} \frac{1}{1+\mathrm{D}} & 0 \\ 0 & \frac{1}{1+\mathrm{D}} \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{1+\mathrm{D}} \\ \frac{\mathrm{D}}{1+\mathrm{D}} & 1 \end{bmatrix}.
\end{aligned} \tag{28}$$

Finally we derive a rate $4/6$ generator matrix G_t from the rate $2/3$ generator matrix as follows:

$$\begin{aligned}
G_t &= \begin{bmatrix} \tilde{U}_1^{(2)}(D) & \tilde{U}_2^{(2)}(D) & \tilde{U}_3^{(2)}(D)\cdot\tilde{V}^{(2)}(D)^{-1} \\ \tilde{U}_2^{(2)}(D) & \tilde{U}_1^{(2)}(D) & \tilde{U}_4^{(2)}(D)\cdot\tilde{V}^{(2)}(D)^{-1} \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 & \frac{1}{1+D} & \frac{1}{1+D} \\ 0 & 1 & 0 & 0 & \frac{D}{1+D} & \frac{1}{1+D} \\ 0 & 0 & 1 & 0 & 1 & \frac{1}{1+D} \\ 0 & 0 & 0 & 1 & \frac{D}{1+D} & 1 \end{bmatrix}.
\end{aligned} \tag{29}$$

Any SF-ISF pair is valid if $GH^T = 0$ and $(H^{-1})^T H^T = I$. So the SF and ISF can also be designed as

$$SF \triangleq H_t^T = \begin{bmatrix} \frac{1}{1+D} & \frac{1}{1+D} \\ \frac{D}{1+D} & \frac{1}{1+D} \\ 1 & \frac{1}{1+D} \\ \frac{D}{1+D} & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{30}$$

$$ISF \triangleq (H_t^{-1})^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{31}$$

By (30) and (31), SF and ISF are modified to operate at index level, making the DSC better match the index-based BCJR algorithm.

Chapter 3 ECG Measurement and Correlation Modeling

Electrocardiography (ECG) is commonly used to measure the regularity of one's heartbeat. ECG, which refers to the record from electrocardiography, can be measured by attaching electrodes to the surface of skin and by recording the electrical activity in the heart. The ECG sources used in this research are acquired by using an ECG sensor or using an open ECG database. As described in the previous chapter, ECG correlation modeling is essential for DSC applications. In this chapter, basic characteristics of ECG, the ECG sources used in our work, and ECG correlation modeling are discussed.

3.1 Basic Characteristics of ECG

Goal of an ECG device is to detect, amplify, and record the electrical changes caused by depolarization and repolarization of heart muscle. Each heart muscle cell, at rest condition, has a negative charge across its cell membrane and is called polarized. The negative charge can be increased to zero, and the phenomenon of depolarization causes the heart to contract. Afterwards, the heart muscle cell will be recharged, called repolarization, which makes the heart to expand. A cardiac cycle begins when the sinoatrial node (SA) generates the impulse, which will run through the heart. The conducting system of the heart can be summarized as follows:

1. The impulse generated from SA node will signal the muscle in the atria to beat, resulting in contraction, during which the blood is pushed from atria into ventricles.
2. The impulse propagates to atrioventricular (AV) node and then delays for about 1 millisecond to allow the blood to fill the ventricles.
3. The impulse propagates to the ventricles through the right bundle branch (RBB), the left bundle branch (LBB), and other nerves. This results in the ventricular contraction, during which the blood is pushed to the body.
4. The muscle cells are recharged (repolarization) and it expands the atrial so that the blood is allowed to return to the heart. Then, the next heartbeat repeats.

A typical cardiac cycle (ECG cycle) is composed of a P wave, a QRS complex, and a T wave. In addition, there are U wave and J wave within an ECG cycle, but their amplitude is so low that they are often ignored. Different types of wave reflect the different stages of the heartbeats. A detailed description of each wave and its duration are shown in Fig. 3.1 and Table 3.1. ECG is the most important tool to diagnose any damage to the heart. Symptoms like arrhythmia and myocardial infarction can be easily detected through ECG. For example, hyperacute T waves indicate that acute myocardial infarction may occur.

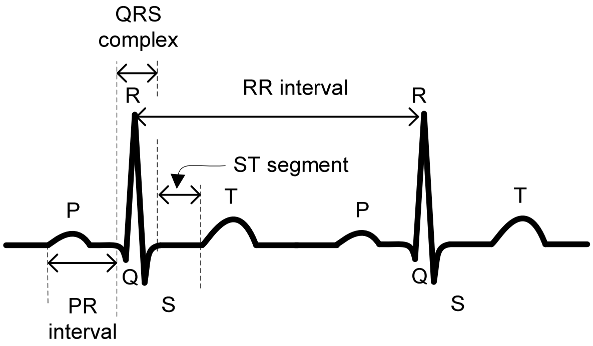


Fig. 3.1 Typical cardiac cycle.

Table 3.1 Descriptions of waves and durations.

Name	Descriptions	Duration
P wave	Represents the atrial depolarization.	Shorter than 0.12 seconds
PR interval	The interval between the points of the P wave to Q wave.	Longer than 0.12 seconds and shorter than 0.2 seconds.
QRS complex	Represents the ventricular depolarization and atrial repolarization.	Longer than 0.08 seconds and shorter than 0.12 seconds.
ST segment	The period during which the ventricles are depolarized.	Longer than 0.08 seconds and shorter than 0.12 seconds
T wave	The last wave of a normal cardiac cycle, representing the ventricular repolarization.	Shorter than 0.16 seconds
RR interval	The interval between two R waves.	Longer than 0.6 seconds and shorter than 1.2 seconds

3.2 ECG Measurements

While the heartbeats are caused by a series of electrical activities in the heart, we can collect and record these signals by attaching electrodes on the surface of the skin. When measuring ECG, usually more than two electrodes are used and they can be combined into a pair whose output is called a lead. The most common clinically-used one is 12-lead ECG where ten electrodes are used. Each electrode has a specific label (name), including RA, LA, RL, LL, V1, V2, V3, V4, V5, and V6. The placements and labels of electrodes are illustrated in Fig. 3.2 [19] and Table 3.2, respectively.

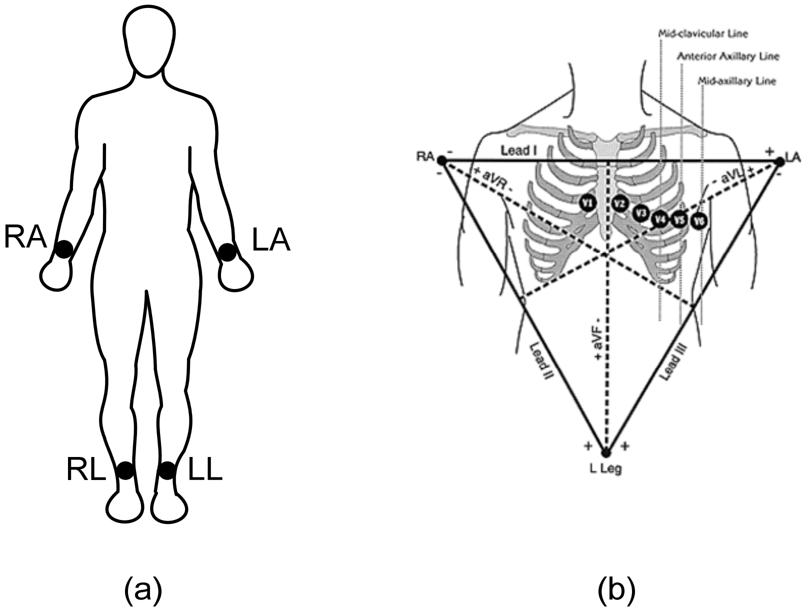


Fig. 3.2 Placements of electrodes [19].

Table 3.2 Placements of electrodes

Electrodes' label	Placement
RA	On the right arm.
LA	On the left arm.
RL	On the right leg.
LL	On the left leg.
V1	In the space, between rib 4 and rib 5, to the right side of the breastbone.
V2	In the space, between rib 4 and rib 5, to the left side of the breastbone.
V3	In the place between lead V2 and lead V4.
V4	In the space between rib 5 and rib 6, and on an imaginary line extended from the collarbone's midpoint.
V5	In the place between lead V4 and lead V6.
V6	In the space horizontally even with lead V4 and V5, and on an imaginary line extended from the middle of the armpit.

The twelve leads contain six precordial leads (V1~V6) in the horizontal plane, three standard limb leads (I, II, III), and three augmented limb leads (aVR, aVL, aVF) in the frontal plane. The twelve leads can also be divided into two types: bipolar and unipolar. While the former has one positive and one negative pole, the latter has two poles with the negative one made of signals from many other electrodes. For example, leads I, II, and III are bipolar leads, while others are unipolar leads. The definitions of twelve leads are given as below.

1. Lead aVR: the positive electrode is on the right arm and the negative one is a combination of two electrodes on the left arm and left leg.

$$Lead\ aVR = V_{RA} - (V_{LA} + V_{LL}) / 2$$

2. Lead aVL: the positive electrode is on the left arm and the negative one is a combination of two electrodes on the right arm and left leg.

   $$Lead\ aVL = V_{LA} - (V_{RA} + V_{LL}) / 2$$

3. Lead aVF: the positive electrode is on the left leg and the negative one is a combination of two electrodes on the left arm and right arm.

   $$Lead\ aVF = V_{LL} - (V_{LA} + V_{RA}) / 2$$

4. Lead I: the positive electrode is on the left arm and the negative one is on the right arm.

   $$Lead\ I = (V_{LA} - V_{RA})$$

5. Lead II: the positive electrode is on the left leg and the negative one is on the right arm.

   $$Lead\ II = (V_{LL} - V_{RA})$$

6. Lead III: the positive electrode is on the left leg and the negative one is on the left arm.

   $$Lead\ III = (V_{LL} - V_{LA})$$

3.3 ECG Sources

Our work proceeds in a two-step procedure. First, offline simulations are used to test and verify the proposed compression algorithm using MIT-BIH Arrhythmia Database [20] as the ECG sources. In the second step, the algorithm is implemented, as described in Chapter 4, on an embedded system using a 1-lead sensor to measure real-time humans' ECG.

The MIT-BIH Arrhythmia Database has been widely used for evaluating automated arrhythmia analysis in research community. The ECG recordings came from the Beth Israel Deaconess Medical Center and were further digitized and annotated by a group at Massachusetts Institute of Technology. The database contains a total of 48 half-hour data, and each of which was recorded from a two-channel electrodes. One channel is a modified limb II (MLII), and the other is usually V1 but can be V2, V4, or V5. The sample rate is 360Hz and each sample point is scalar quantized into 11 bits. Annotations were made by cardiologists to indicate specific phenomenon. For example, "A", "·" and "V" stand for a heartbeat. These three annotations represent atrial premature beat, normal beat and ventricular premature beat, respectively.

In this thesis, two ECG recordings with number 110 and 122, are used in offline simulations. Typical segments of their waveforms are given in Fig. 3.3 and Fig. 3.4.

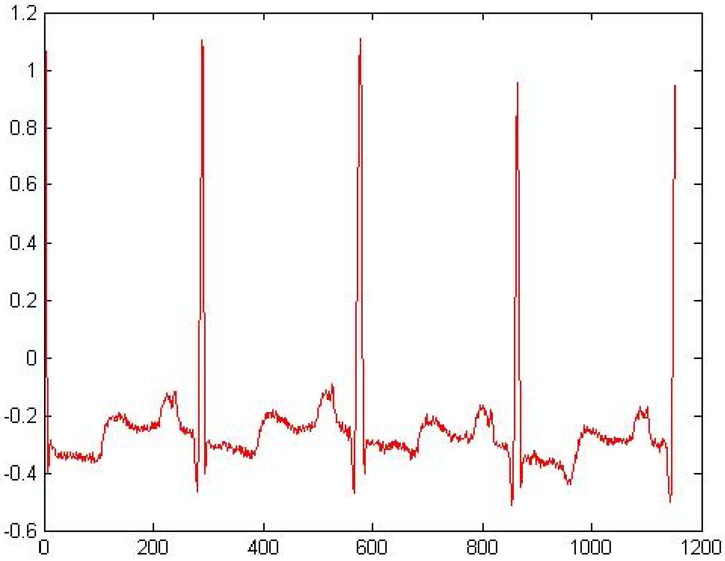


Fig. 3.3 A segment of waves in MIT-BIH 100.

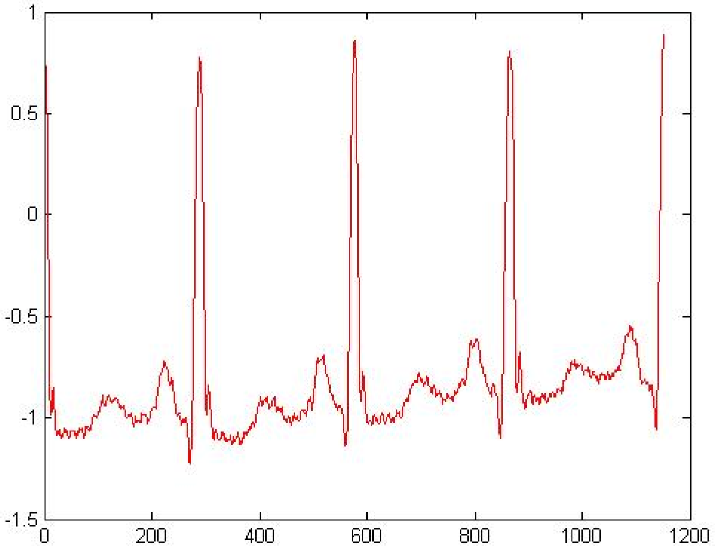


Fig. 3.4 A segment of waves in MIT-BIH 122.

3.4 ECG Preprocess and Correlation Modeling

Constructing an ECG correlation model is essential to match the DSC application. The proposed modeling methods are based on vector quantization (VQ). Fig. 3.5 shows the flow chart for constructing the ECG correlation model.

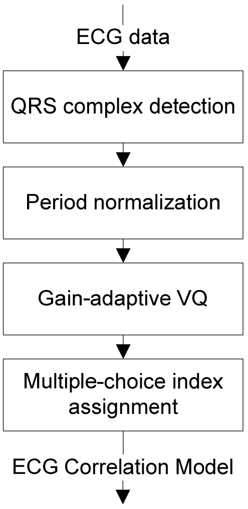


Fig. 3.5 The flow chart of ECG correlation modeling.

A (k, M) vector quantizer [21] can be viewed as a function that maps a vector in k dimensional space R^k to a set S containing 2^M representative codewords. The set S is often referred to a codebook, of which size is 2^M with k-dimensional codewords. A VQ-based compressing system is composed of the encoder and the decoder. The encoder calculates the distance between the input vector $V^k = (v_1, v_2, ..., v_k)$ and each codeword c_i in the codebook C, and determines an index $i \in I = \{1, 2, ..., 2^M\}$ associated with the codeword closet to the input vector. The index is then transmitted to the decoder. The decoder uses the received index to look up the same codebook and finds the corresponding codeword approximated to the input vector.

Although conventional VQ works well for its high compression ratio and high quality reconstruction, there exist many variants for different applications. The conventional VQ encoder requires computational complexity proportional to $k2^M$, implying that the complexity grows exponentially. A large VQ codebook is needed to achieve reasonable performance if the range of the input vector is large, since there should be more codewords to represent original input vectors. Therefore a good performance of VQ is reached at the cost of high encoding complexity due to the use of large codebook. To solve the problem, gain-adaptive VQ rather than conventional VQ is used in this thesis. Unlike conventional VQ, gain-adaptive VQ computes the gain of the input vector and sends the gain-normalized input vector to VQ. By normalizing the input vector, the range of input vector can be reduced, leading to better performance.

We first consider the use of two preprocesses, including QRS complex detection and period normalization, to maximize the inter-beat correlation prior to gain-adaptive VQ. Also proposed is a multiple-choice index assignment scheme to better match DSC application. Unlike conventional VQ that chooses the codeword closest to the input vector, the multiple-choice index assignment scheme chooses the top-five closest codewords as candidates, and then decides the best codeword by some heuristic rules.

3.4.1 QRS Complex Detection

The QRS complex is the most distinct part in the ECG waveform and there exists many QRS complex detection algorithm [22]. For our real-time application, off-line QRS complex detection methods are impracticable. R.G. Lee *et al.* proposed a modified So and Chan (MSC) algorithm [23], which is utilized in this work to implement a simple QRS complex detection algorithm. Let $x(n)$ be the amplitude of filtered ECG points at time instant n. As shown in Fig. 3.6, the steps of the detection algorithm are described in detail as follows:

1. Low-pass filter: The transfer function and the difference equation are given by

$$H_l(z) = \frac{(1-z^{-6})^2}{(1-z^{-1})^2}. \tag{32}$$

2. High-pass filter: The transfer function and the difference equation are described by

$$H_h(z) = \frac{(-1+32z^{-16}+z^{-32})}{(1-z^{-1})}. \tag{33}$$

3. Slope calculation: The slope of filtered ECG signals can be calculated by

$$slope(n) = -2x(n-2) - x(n-1) + x(n+1) + 2x(n+2). \tag{34}$$

4. QRS wave onset detection: The QRS wave onset can be detected if the condition $|slope(n)| > |slope_{th}|$ is satisfied for two consecutive points, where a predefined value $slope_{th}$ is defined as the ratio to the maximal slope of first 500 ECG points.

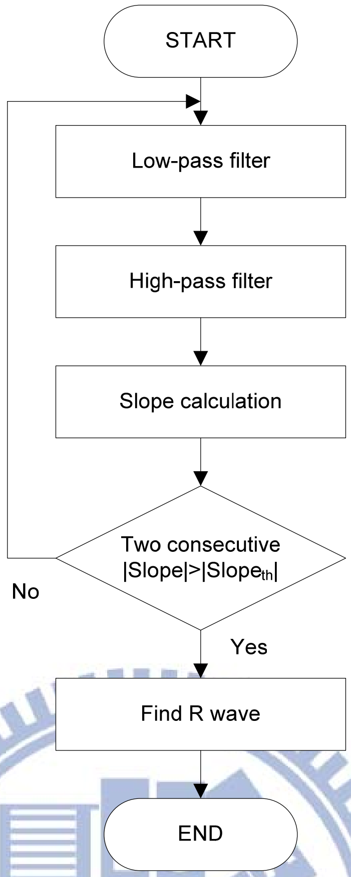


Fig. 3.6 Flow chart of QRS complex detection.

3.4.2 Period Normalization

ECG itself is one-dimensional in time-domain, but can be viewed as a two-dimensional signal in terms of its periodicity. Since the lengths of heart segment are different, each ECG cycle has to be normalized to a fixed length for further processing. A predefined length is obtained by calculating the average length of a large training set of ECG cycles. Higher order interpolations make little difference but require more computation complexity. Therefore, linear interpolation is a better choice for its simplicity.

3.4.3 Gain-Adaptive Vector Quantization

In gain-adaptive vector quantization, the dynamic range of input can be reduced if an encoder quantizes the gain-normalized vector rather than the original one. Two gain-adaptive approaches are implemented in our application, including backward gain-adaptive VQ [24] and gain-shape VQ [10]. Two main differences exist between backward gain-adaptive VQ and gain-shape VQ: 1) In gain-shape VQ, there is no VQ decoder at the transmitter; and 2) In backward gain-adaptive VQ, there is no side information to be sent. Gain predictors at both the transmitter and the receiver use the same past quantized vectors to estimate a gain. But in gain-shape VQ, the VQ index of gain codebook must be transmitted to the receiver.

[A] Backward Gain-Adaptive VQ:

Fig. 3.7 shows the structure of backward gain-adaptive VQ [24]. The gain predictor estimates the gain by using previous quantized vectors, which are the only information available at the receiver. At the transmitter, there are both VQ encoder and decoder. At sample time n, the k-dimensional input vector $X_n = (x_{n1}, x_{n2}, ..., x_{nk})$ is divided by a gain g_n coming from the gain predictor. The normalized vector is quantized by the VQ encoder and the specific index I_n corresponding to its closest codeword $\hat{Y}_n$ will be transmitted. In addition, $\hat{Y}_n$ is multiplied by a gain g_n to obtain the reconstructed vector $\hat{X}_n = (\hat{x}_{n1}, \hat{x}_{n2}, ..., \hat{x}_{nk})$, which will be used by gain predictor to estimate the gain g_{n+1} at sample time $n+1$. At the receiver, a similar operation is performed to generate the reconstructed signal $\hat{X}_n$ and the next gain g_{n+1}.

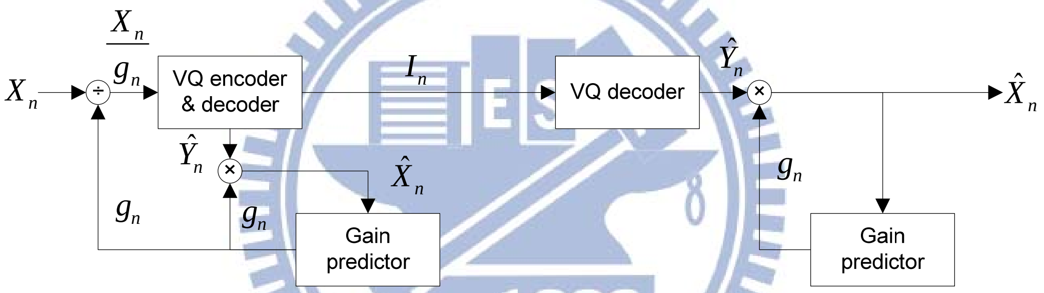


Fig. 3.7 Structure of backward gain-adaptive VQ [24].

Four backward adaptive algorithms have been proposed in previous work [24]. Among them, the exponential-average gain predictor is used here due to its low-delay estimation. More specifically, the gain g_n is predicted by

$$g_n = \frac{1-\alpha}{\alpha} \sum_{i=1}^{\infty} \alpha^i \left\| \hat{X}_{n-i} \right\|, \tag{35}$$

where $0 < \alpha < 1$. $\left\| \hat{X}_n \right\|$ denotes the norm of $\hat{X}_n$ and is given by

$$\left\| \hat{X}_n \right\| = \left(\sum_{i=1}^{k} \hat{x}_{ni}{}^2 \right)^{1/2}, \tag{36}$$

where k is the dimension of the vector $\hat{X}_n$. For a recursive implementation, we have

$$g_n = \alpha g_{n-1} + (1-\alpha) \left\| \hat{X}_{n-1} \right\|. \tag{37}$$

[B] Gain-Shape VQ:

In gain-shape VQ, the “gain” g_n is the norm of an input vector X_n, while “shape” S_n refers to the normalized input vector, that is, $S_n = X_n / g_n$ [10]. Forward adaptation of the gain is given by

$$g_n = \|X_n\| = \left(\sum_{i=1}^{k} x_{ni}^{\ 2} \right)^{1/2} . \tag{38}$$

The basic structure of gain-shape VQ is shown in Fig. 3.8. Each input vector is normalized prior to the shape VQ and both the shape and gain VQ encoder are applied to choose the best index. It is impractical to transmit uncompressed gains to the receiver, since the overhead is huge. In our work, both the shape and gain are vector quantized, and their codebooks minimizing the distortion are generated using the LBG algorithm [25]. The receiver performs codebook look-up to determine the closest codeword $\hat{Y}_n$, which is then multiplied by a gain g_n to obtain the reconstructed signal $\hat{X}_n$.

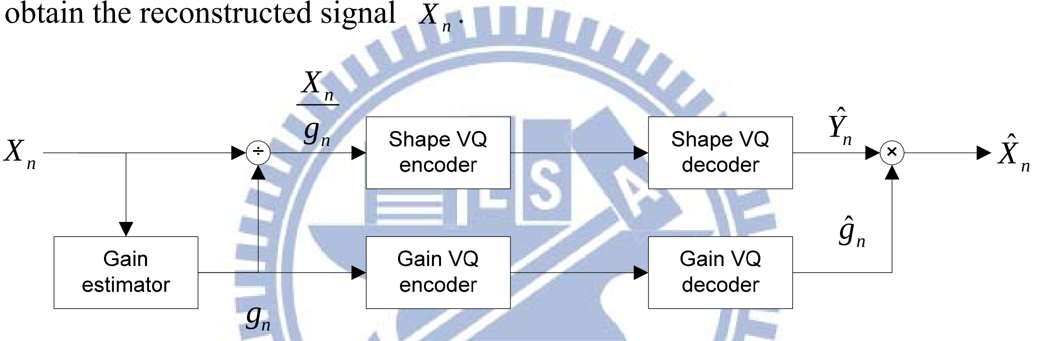


Fig. 3.8 Structure of gain-shape VQ.

3.4.4 Multiple-Choice Index Assignment

Goal of the ECG preprocess is to ensure that the BER of two correlated sources is low enough to be compensated by the channel decoder. To better match DSC, multiple-choice index assignment scheme is proposed to lower the BER while keeping reasonable reconstruction quality. In the so-called multiple-index VQ, the process of codebook training is the same as in gain-adaptive VQ. Besides training a codebook, there is an additional step to construct the reference model. Consider the k-dimensional VQ of an ECG signal which consists of n cycles and m points per cycle. The detailed steps of building a reference model are described as follows:

1. Quantize all training sequences (the total number of points is $n \times m$), and arrange the resulting $n \times (m/k)$ indexes as a 2-dimensional array of n rows and m/k columns.

2. For each of m/k columns, find the index which occurs the most among the rows. This results in a total of m/k indexes, which form the reference model.

When looking up codebook, the index is chosen by the following rules:

1. Compare the i-th input vector to every codewords of the codebook and find five indexes $I = (i_1, i_2, i_3, i_4, i_5)$ that represent top-five closest codewords.
2. Choose among them the index with minimum Hamming distance to the i_{ref} as the output.

Chapter 4 System Architecture and Software Implementation

There are three major components in the ECG system, including 1) a sensor for ECG measurement; 2) a front-end device for ECG compression; and 3) a back-end server for signal reconstruction. To increase the mobility, wireless communication technologies such as Bluetooth and mobile cellular network are involved in the real-time implementation. In the previous chapter we introduce the DSC theorem and practical methods to construct ECG correlation model. The encoder is implemented on an embedded system development board, which serves as the front-end device, while the decoder is designed on the back-end server.

4.1 Hardware Architecture

The hardware architecture is illustrated in Fig. 4.1. The patient unit consists of the BtECG sensor and the embedded system development board WinFast PXA310 mini-board. ECG signals are first measured by BtECG sensor and transmitted to the board, where the ECG signals are segmented and compressed. Afterwards the compressed ECG data are transmitted via a wireless link to the back-end server. The wireless transmission from the board to the server includes 1) a Bluetooth link between the board and a mobile telephone; 2) WCDMA to a base station; and 3) Internet to the server. To proceed with this, the WinFast PXA310 board is connected to a Bluetooth module via a USB interface. In addition, a commercially available Bluetooth-enabled 3G mobile telephone SAMSUNG S3370 is used.

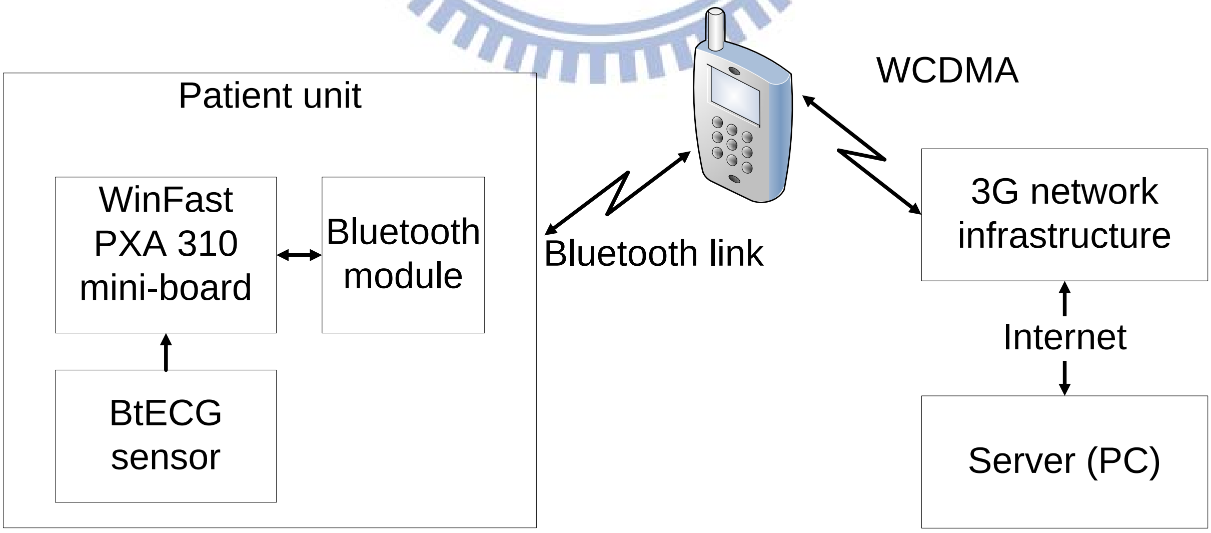


Fig. 4.1 The hardware architecture of the mobile ECG system.

The specification of the patient unit is summarized in Table 4.1. A detailed description of the BtECG sensor and WinFast PXA310 mini-board are given in 4.1.1 and 4.1.2 respectively.

Table 4.1 The specification of the patient unit.

BtECG sensor	Size: 5.5 cm * 3.5 cm * 1.6 cm Duration of power supply: 20 hours
ECG signals	Channel: one ECG channel ADC resolution: 12 bits Sampling rate: 250 Hz Band-pass filter: 0.1 Hz-40 Hz
Bluetooth module	Version: v2.1 + EDR Frequencies: 2.402-2.48 Hz RF power coverage: 10 meters (Class II) Theoretical data rate: 3 Mbps
WinFast PXA310 mini-board	Size: 11 cm * 9.5 cm * 3.5 cm Marvell PXA 310 processor: -Intel XScale core with 624 MHz -2*256 kB SRAM -USB 1.1 and other peripheral

4.1.1 BtECG Sensor

The BtECG sensor is developed by K&Y Lab. and LeadTek Research Inc. Its small size sensor and Bluetooth communication capability are extremely suitable for use in developing wireless medical telemetry applications. BtECG sensor comprises an analog-to-digital converter (ADC), a DSP processor KY202, and a Bluetooth module. The processor incorporates into one single chip the novel technologies of heart rate variability (HRV), including R wave detection, noise rejection, interpolation in time domain, and fast Fourier transform in frequency domain. All the analysis results, together with measured ECG signals, are transmitted via Bluetooth. The appearance of BtECG sensor is shown in Fig. 4.2. Two physiological signals can be collected by using three electrodes. Two of the three electrodes are used for measuring ECG and the other for body temperature. Since the ADC resolution is 12 bits, the value of an ECG sample ranges from 0 to 4095. A segment of ECG waveforms measured by BtECG sensor is given in Fig. 4.3.

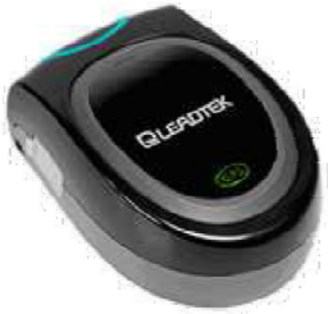


Fig. 4.2 The appearance of BtECG sensor.

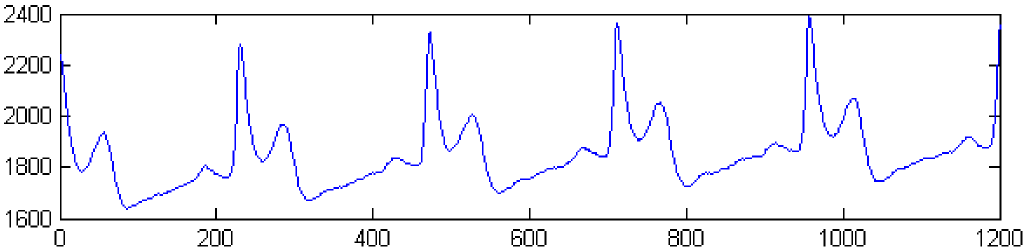


Fig. 4.3 A segment of ECG waves collected by BtECG sensor.

4.1.2 **WinFast PXA310 Mini-Board**

WinFast PXA 310 mini (mini-board) featuring the Marvell PXA310 processor is chosen in our application. The board which serves as the front-end device is shown in Fig. 4.4. Marvell PXA310 processor uses an optimized XScale architecture with a top clock speed of 624 MHz. Many peripheral components are supported by WinFast PXA 310 mini, as shown in Fig. 4.5. Among them, USB 1.1 and 10/100 Ethernet are used. A Bluetooth USB module can be incorporated to allow the board to communicate with BtECG sensor and the mobile telephone.

Fig. 4.4 The exterior of WinFast PXA310 mini-board.

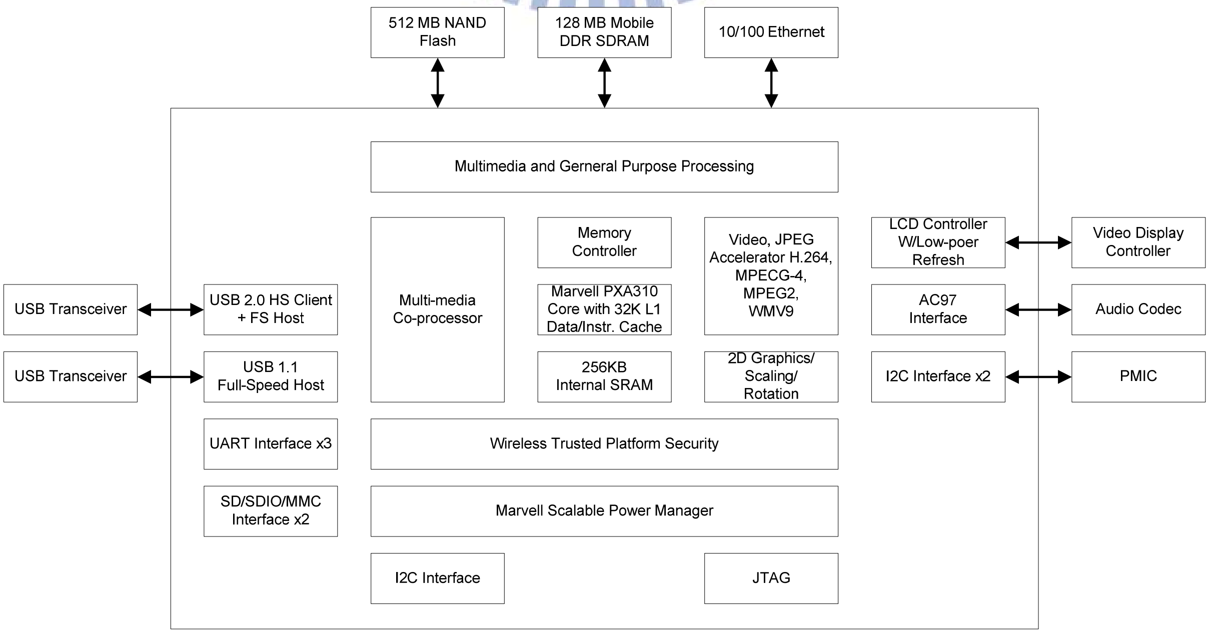

512 MB NAND Flash
128 MB Mobile DDR SDRAM
10/100 Ethernet
Multimedia and Gerneral Purpose Processing
Multi-media Co-processor
Memory Controller
Marvell PXA310 Core with 32K L1 Data/Instr. Cache
256KB Internal SRAM
Video, JPEG Accelerator H.264, MPECG-4, MPEG2, WMV9
2D Graphics/ Scaling/ Rotation
LCD Controller W/Low-poer Refresh
AC97 Interface
I2C Interface x2
Video Display Controller
Audio Codec
PMIC
USB Transceiver
USB Transceiver
USB 2.0 HS Client + FS Host
USB 1.1 Full-Speed Host
UART Interface x3
SD/SDIO/MMC Interface x2
Wireless Trusted Platform Security
Marvell Scalable Power Manager
I2C Interface
JTAG

Fig. 4.5 Block diagram of WinFast PXA310 mini.

4.2 Software Implementation

The software design in our implementation can be divided into two parts, the one is on the development board and the other is on the server.

4.2.1 Development Language and Tools

To match the WinCE 6.0 operating system installed on the PXA310 mini, we have chosen Microsoft Visual Studio 2005 for our software development. In addition, Windows Embedded CE 6.0 platform builder is installed on a PC to develop applications operating on WinCE 6.0. Windows Embedded CE 6.0 platform builder is a plug-in of Microsoft Visual Studio 2005, where the application on the board is developed by using Microsoft Foundation Class (MFC). The software on the server is developed with MATLAB 7.6.

4.2.2 Software design

Goal of the software development on the server is to receive and reconstruct the compressed ECG signals. A socket is created to listen and accept the connection request, and then receive data coming from the board. Once receiving a complete heartbeat segment, they are decoded. The proposed modified DSC decoder and GSVQ decoder are implemented. The resulting decoded signals are then period normalized to its original heartbeat length and the reconstructed signals can be monitored on the screen.

As shown in Fig. 4.6, the software implemented on the board performs the following three tasks:

1. System initialization: In the phase of system initialization, the connection between the board and the server is established and the resources are allocated. The *recvBuf* with a buffer size of 512 bytes is allocated to store the data coming from BtECG. The *sendBuf* is allocated to store the compressed ECG signals prior to transmission. The size of *sendBuf* is not fixed but depends on the parameters chosen in the compression method. The link between the board and the server is set up as described in 4.2. A COM port and a socket are created for communicating with the mobile telephone and the server, respectively. To transmit data to the server, the IP address and the port of the server are specified. In addition, many parameters used for ECG compression such as the size of shape codebook are set.
2. ECG signal compression: The ECG signals received from BtECG sensor are stored in *recvBuf*. The PXA 310 processor is notified to read the data in *recvBuf* when the buffer is full. Once the QRS of each complete ECG cycle has been detected, a 1-D

to 2-D process proceeds with period normalization and compression. The remaining ECG signals, not forming a complete cycle, will be QRS detected until the next time when the processor reads the *recvBuf*. Afterwards, a series of compression processes is applied. The PXA 310 processor stores the following information to the *sendBuf* prior to transmission: heartbeat length from period normalization, gain index from GSVQ and syndrome from the DSC encoder.

3. Wireless communication: The board will retrieve the contents in the *sendBuf* and then transmit to the server based on TCP protocol. To realize the TCP/IP transmission, the Windows Sockets (Winsock) application programming interface provided by Microsoft is used.

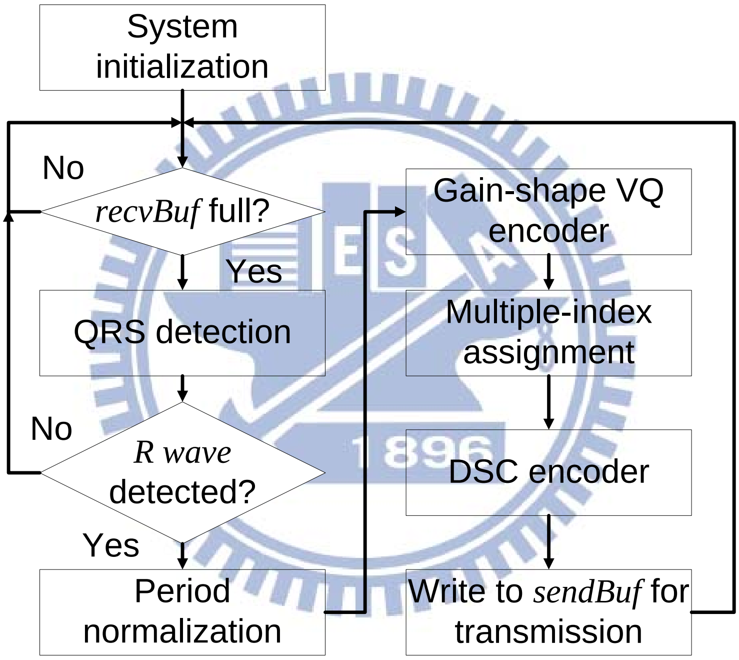


Fig. 4.6 The software architecture on the transmitter side.

We digress here for a moment to discuss an issue. As described in 4.1.1, BtECG sensor performs R wave detection. Why should we implement the QRS complex detection on the board? It is because BtECG sensor only uses R wave detection to calculate HRV, but it does not transmit any information about R wave to the board. The board does not know the R wave location within the received ECG signals, and the QRS complex detection is needed on the board.

The wireless transmissions in our work can be divided into two parts. A built-in Bluetooth module is responsible for transmitting signals from BtECG sensor to the board. The second part takes charge of transmitting compressed data from the board to server (PC) and there are many available wireless technologies. Mobile cellular network is a better choice.

With a *Bluetooth-enabled* cell phone, data can be transmitted to the phone and then to the PC whenever the cell phone is in the working range of mobile cellular network.

A Bluetooth profile is a specification with respect to Bluetooth wireless communication between two devices. There are many profiles in Bluetooth technology. Among them, Serial Port Profile (SPP) and Dial-up Network Profile (DUN) are used in our work. SPP emulates a serial cable to provide a substitute for RS-232. DUN is based on SPP and is a wireless technology standing for connecting to the Internet with a cell phone. In our work, the development board can access the Internet by dialing up on the cell phone wirelessly. From this perspective, the cell phone can be simply viewed as a modem.

It is the Bluetooth stack that allows the transmission between the board and the server to work. Fig. 4.7 shows the implementation of Bluetooth stack in the WinCE. Among various layer, only Radio Frequency Communication (RFCOMM), Service Discovery Protocol (SDP), and COM Port Emulation layer are concerned. The SPP is based on the RFCOMM which refers to serial port emulation. SDP is used so that a device can discover the nearby devices and their services. The COM Port Emulation allows virtual COM ports to be created over RFCOMM layer, and it hosts SPP and DUN. Depending on the Bluetooth device, a profile is given in companion with a specific RFCOMM channel number. RFCOMM channel number can be acquired by using SDP.

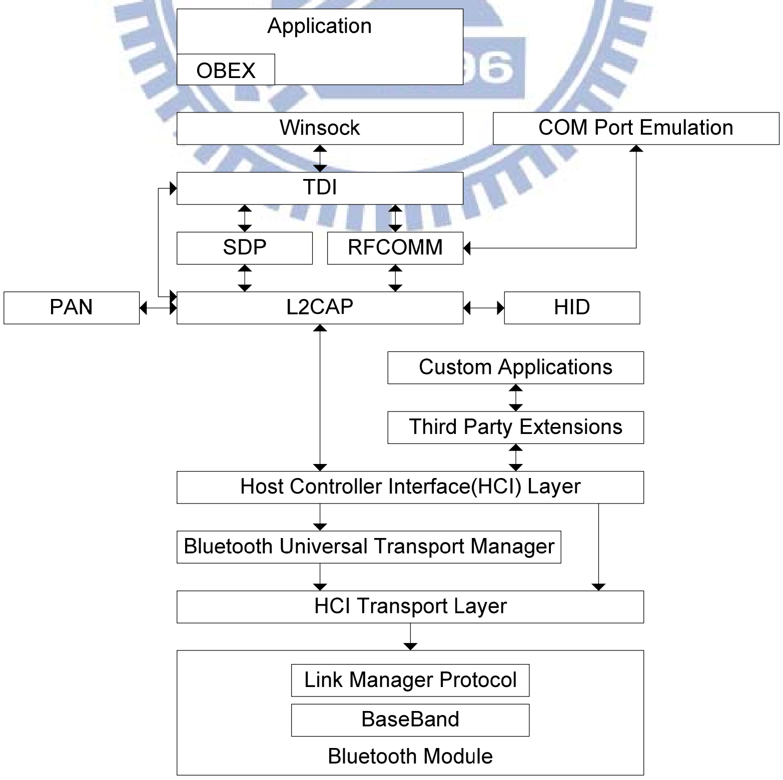


Fig. 4.7 The Bluetooth implementation in WinCE.

In our work, the development board first uses SDP to discover nearby devices and their services. Therefore, the addresses and services (RFCOMM channel numbers) of remote devices are recognized. Suppose that a Bluetooth-enabled cell phone is discovered. The board creates a virtual COM port together with the COM port number and the RFCOMM channel number to connect to the phone. By assigning the RFCOMM channel number (for DUN), the board can use the cell phone as a modem. This is the way how the WinFast PXA310 mini-board accesses the Internet via mobile cellular networks. Once the board accesses the Internet, compressed ECG data can be transmitted through the TCP protocol for reliable data transmission. The board connects to the BtECG sensor in a similar way, but there are two differences in data transmission. First, after using SDP, the board uses the SPP to connect to the device, rather than DUN. Second, when receiving ECG data from BtECG sensor, the board simply reads data from the COM port. Notice that this COM port is different from the port for communicating with the cell phone. Details of Bluetooth communication implementation are described in Appendix A.

To make the application more responsive to I/O, multithreading programming technique is used in this work. There are three threads and each of which has its own missions and data. While one thread is used to perform the ECG compression, the other two threads are responsible for receiving ECG data from BtECG sensor and transmitting the compressed ECG data to the server wirelessly. A thread often needs to communicate with other threads and therefore, shared memory model is used for interprocess communication. Concurrent access to shared memory can be a potential problem since it may result in data inconsistency. In our work, *critical section object* provided by Win32 Application Programming Interface (API) is used to maintain data consistency.

4.2.3 **Data Format of Compressed ECG**

Three types of data (heartbeat length, gain indexes, and syndromes) are stored in packets prior to the transmission from the board to the PC. There are five fields within a packet, as described as follows.

[A] Number of Cycles

This field indicates how many ECG cycles are in the packets. It occupies 8 bits, so a packet can contain up to 256 ECG cycles.

[B] Header

This field is 8 bits long and contains some attributes of the packet. Recall that the board needs to transmit side information to DSC decoder. There is one bit in this field to indicate whether the packet contains side information. Other 7 bits are not used, but reserved for further usages.

[C] Gain Indexes

The length of this field is variable depending on how many gain indexes are transmitted. For each index, it is 8 bits long. As we will see in Chapter 5, we only use 6 bits to represent an index, and keep the space for future expansion to 8 bits.

[D] Heartbeat Length

The length of this field is variable depending on how many ECG cycles are transmitted. This field contains the original length of each ECG cycle (e.g., the number of points in an ECG cycle before period normalization). The sample rate of BtECG sensor is 250 Hz and human record 78 beats per minute in average. We can conclude that the length of an ECG cycle will be more than 256. In our work, we use 2 bytes to record the length of a cycle.

[E] Compressed Data

The length of the field is also variable. Data can be divided into two categories, syndrome and side information. For side information, we allocate 8 bits to record the index of normalized ECG vector. For the syndrome, we pack four syndromes (2 bits per syndrome) into a byte.

Chapter 5 Simulation and Implementation Results

In the previous chapters we have described some new ECG compression algorithm. In the first step, offline simulations are conducted to test and verify the proposed ECG compression algorithm. QRS complex detection and period normalization are applied for preprocessing. Algorithm I refers to the non-DSC compression algorithm using the backward gain-adaptive VQ. Algorithm II represents the combined use of gain-shape VQ and DSC. In both algorithms, two MIT-BIH recordings, MIT-100 and MIT-122, are used as our ECG sources, and each is sampled at 360 sample/second and quantized with 11 bits. After simulation, we implemented the mobile DSC-based ECG system as described in Chapter 4. In this chapter, we first illustrate the experimental setup used in offline simulation. Simulation results and some discussions on Algorithm I and Algorithm II are given. Finally, the real-time implementation results of the mobile ECG system are presented.

5.1 Offline Simulation

5.1.1 Experimental Setup

Two preprocesses are applied to MIT-100 and MIT-122. We first have these ECG sources QRS detected, and period normalization is then applied to make each cycle have 288 points. These period-normalized signals are used as sources for Algorithm I and Algorithm II. In each source, we choose the first 1,500 ECG cycles as training sequences, and the latter 200 ECG cycles as testing sequences. We use 9 bits to record the original heartbeat length as side information.

Percent root mean square difference (PRD), compression ratio (CR), and symbol error rate (SER) are used to evaluate the performance. PRD is a measure of the fidelity of the compressed signal and is given by

$$PRD\left(\%\right)=\sqrt{\frac{\sum_{i=1}^{N}[x(i)-\hat{x}(i)]^2}{\sum_{i=1}^{N}x(i)^2}}\times 100, \tag{39}$$

where L is the number of ECG samples, x is the original signal, and $\hat{x}$ is the reconstructed signal. The CR is defined as

$$CR = \frac{n_x}{n_d + n_g + n_p}, \tag{40}$$

where n_x denotes the number of bits per sample in the original signal, n_d denotes the number of bits per sample in the encoded signal, n_g denotes the number of bits per sample used to code the index of gain, and n_p denotes the number of bits per sample used to code the length of an ECG cycle. SER is the number of VQ index errors divided by the total number of VQ indexes per cycle.

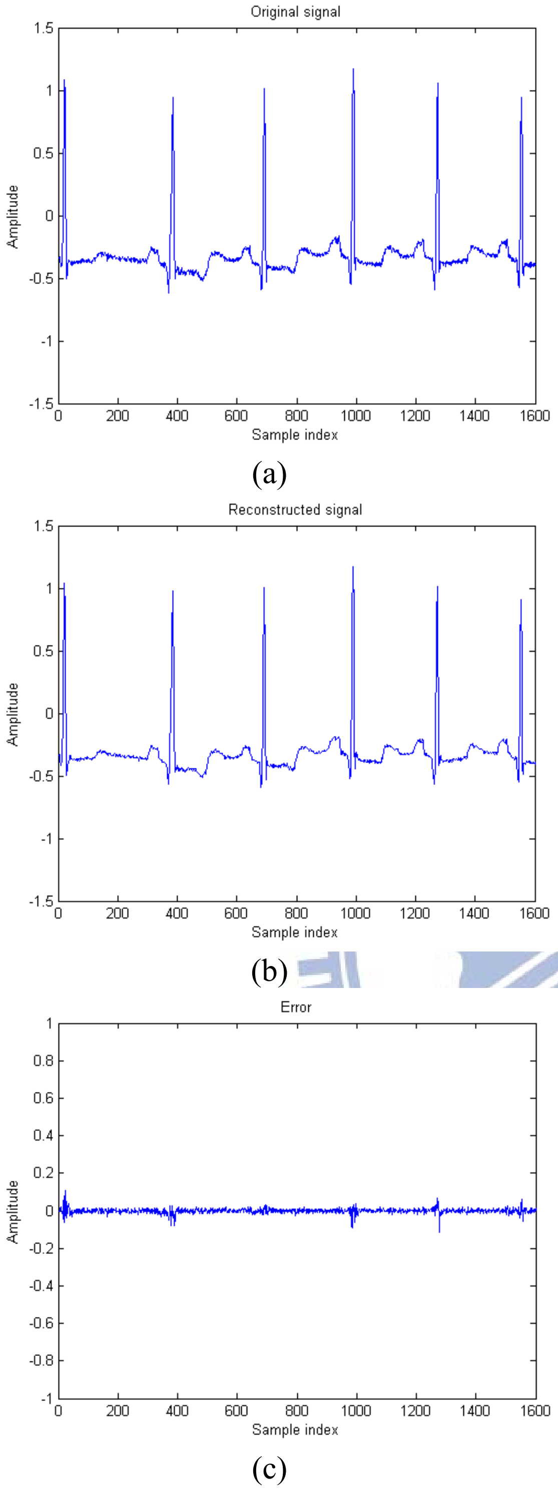
5.1.2 **Experimental Results of Algorithm I**

Following the preprocessing steps described in 5.1, backward gain-adaptive VQ is applied with $n_g = 0$ bps and vector dimension $k = 4$. We first calculate the gains of period-normalized signals by using (36) and (37), where $\alpha = 0.45$. After gain normalizing, a $(4,6)$ vector quantizer is applied to the gain-normalized vector. With this arrangement, we have $n_x = 11$ bps, $n_d = 1.5$ bps, $n_g = 0$ bps, and $n_p = 0.03125$ bps.

Table 5.1 summarizes the performance results of Algorithm I. Original and reconstructed waveforms of MIT-100 and MIT-122 are shown in Fig. 5.1 and Fig. 5.2, respectively.

Table 5.1 Performance results of Algorithm I.

ECG sources	PRD	CR
MIT-100	4.338	7.18
MIT-122	4.6073	7.18

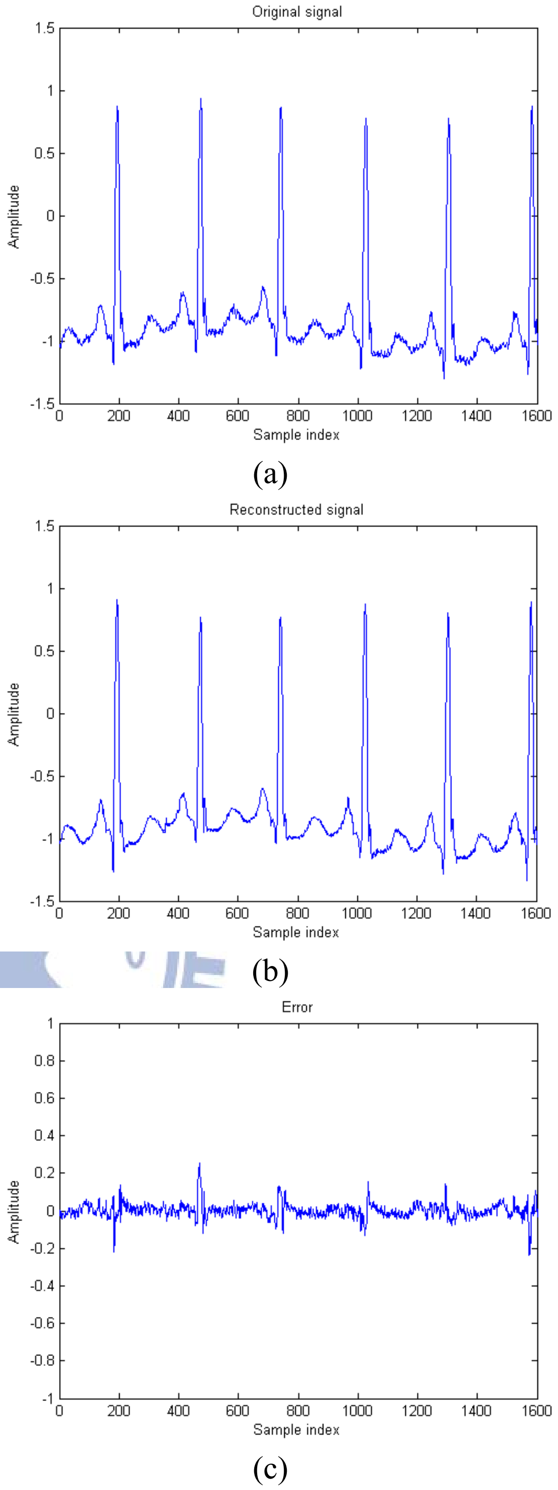


(a)

(b)

(c)

Fig. 5.1 Results of Algorithm I: (a) a typical segment of waveforms in MIT-BIH 100, (b) reconstructed ECG waveforms, and (c) error signals.



(a)

(b)

(c)

Fig. 5.2 Results of Algorithm I: (a) a typical segment of waveforms in MIT-BIH 122, (b) reconstructed ECG waveforms, and (c) error signals.

The results show that most reconstruction errors occur near the QRS complex. The reason is that most of the gain-normalized signals have low amplitude, implying that many low-amplitude vectors may exist in the codebook. However, the vectors representing the QRS complex are of high amplitude. When vector quantizing, these vectors are prone to being assigned indexes corresponding to low-amplitude signals. These errors still cannot be well corrected even if gains are multiplied back.

We found that PRD is high if DSC is applied. Moreover, at the transmitter and receiver sides, the gain estimators use their past reconstructed signals and gain to predict the current gain. The errors will be propagated if the past reconstructed signals have large distortion. This disadvantage results in poor reconstruction quality.

5.1.3 **Experimental Results of Algorithm II**

Following the preprocessing steps described in 5.1, gain-shape VQ is applied. Gains of period-normalized signals are calculated by using (38), where $k=4$. Through gain normalization, we have the shape signals. Both gains and shapes are vector quantized, where a $(4,6)$ vector quantizer for the shapes, and a $(36,6)$ vector quantizer for the gains. With this arrangement, we have $n_x = 11$ bps, $n_d = 0.5$ bps, $n_g = 0.041667$ bps, and $n_p = 0.03125$ bps. After gain-shape VQ and the multiple-choice index assignment scheme have been applied, the DSC encoder and decoder are used for further compression. A rate 4/6 RSC code is chosen for the syndrome former. The generator matrix G is given by

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & \frac{1}{1+D} & \frac{1}{1+D} \\ 0 & 1 & 0 & 0 & 1 & \frac{1}{1+D} \\ 0 & 0 & 1 & 0 & \frac{1}{1+D} & \frac{1}{1+D} \\ 0 & 0 & 0 & 1 & \frac{1}{1+D} & 1 \end{bmatrix}. \tag{41}$$

The SF-ISF pair is designed as

$$SF \triangleq H_t^T = \begin{bmatrix} \frac{1}{1+D} & \frac{1}{1+D} \\ 1 & \frac{1}{1+D} \\ \frac{D}{1+D} & \frac{1}{1+D} \\ \frac{D}{1+D} & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{42}$$

$$ISF \triangleq (H_t^{-1})^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{43}$$

The block diagram of the 4/6 RSC code is shown in Fig. 5.3. The 72 indexes corresponding to the 288 points of the first ECG cycle are viewed as side information Y at the decoder side. And a total of 14,328 indexes of the following 199 ECG cycles are taken as the ECG source X to be compressed.

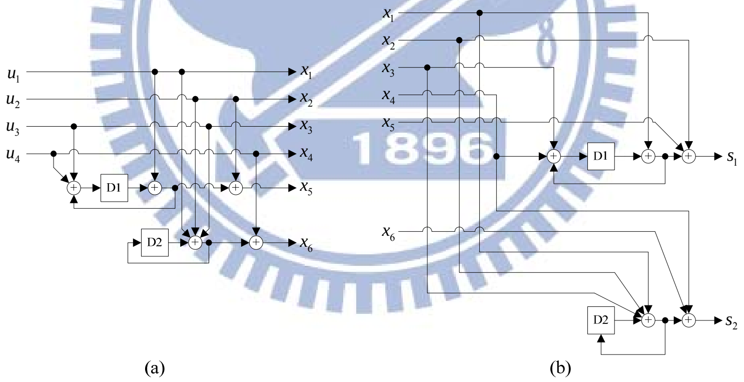
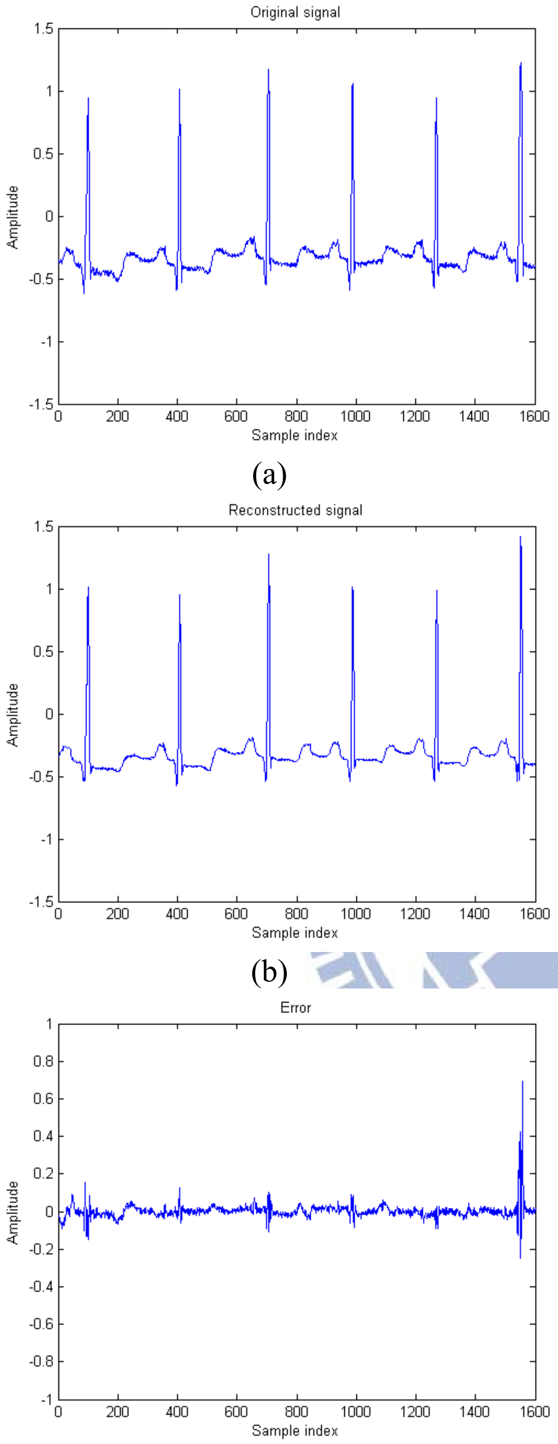


Fig. 5.3 The proposed rate 4/6 RSC code and its SF: (a) the encoder. (b) the implementation of SF.

Table 5.2 summarizes the performance of Algorithm II. Original and reconstructed waveforms of MIT-100 and MIT-122 are shown in Fig. 5.4 and Fig. 5.5, respectively.

Table 5.2 Performance results of Algorithm II.

ECG sources	SER	PRD	CR
MIT-100	1.15	10.5679	19.2
MIT-122	0.99	4.3272	19.2

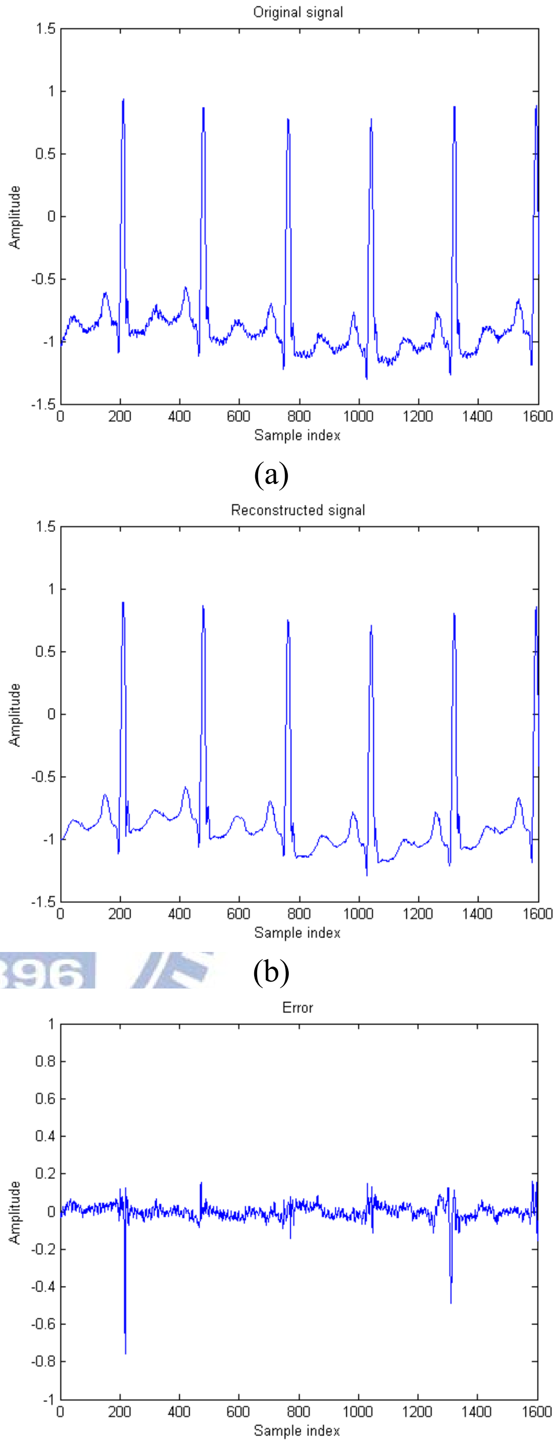


(a)

(b)

(c)

Fig. 5.4 Results of Algorithm II: (a) a typical segment of waveforms in MIT-BIH 100, (b) reconstructed ECG waveforms, and (c) error signals.



(a)

(b)

(c)

Fig. 5.5 Results of Algorithm II: (a) a typical segment of waveforms in MIT-BIH 122, (b) reconstructed ECG waveforms, and (c) error signals.

Prior to the DSC encoding, the ECG correlation model is constructed by using gain-shape VQ and multiple-choice index assignment. The result shows that high CR is achieved while keeping PRD in an acceptable range. Like Algorithm I, most reconstruction errors occur near the QRS complex. Compared to $n_g = 0$ bps of backward gain-adaptive VQ, gain-shape VQ adds overhead $n_g = 0.041667$ bps. This is because the gains should be transmitted as side information to the decoder, but it avoids the error propagation problem. If DSC is applied or the communication channel is noisy, gain-shape VQ performs much better than backward gain-adaptive VQ.

5.2 Real-Time Implementation

In addition to offline simulations, the mobile ECG monitoring system is implemented in real-time as described in Chapter 4. The ECG signals measured by BtECG sensor are sampled at 250 samples/s and quantized with 12 bits. We first measure the ECG signals for 30 minutes, and the first 1,500 ECG cycles are used to train the shape codebook and gain codebook. The system parameters are the same as those used in Algorithm II in offline simulations. Therefore, we have $CR = 12/(0.5 + 0.041667 + 0.03125) = 20.95$. Fig. 5.6 shows the experimental setup of the system. A person who wears the BtECG sensor has the electrodes attached. The function of the board is to store, compress, frame, and multiplex the incoming data. The compressed data are then fed to a SAMSUNG S3370 mobile telephone via a Bluetooth link. Afterwards the telephone and the server are connected via mobile cellular networks. Finally, the server reconstructs the ECG signals and displays them on the screen. Experimental results clearly demonstrate that clinical features are retrained.

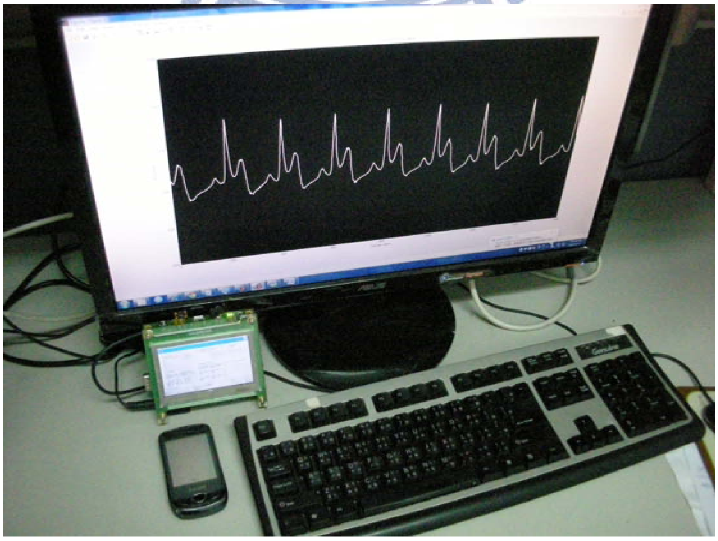


Fig. 5.6 Experimental setup: the WinFast PXA310 mini-board, the SAMSUNG S3370 mobile telephone, and the server (PC).

Chapter 6 Conclusions and Future Works

The main contribution of this thesis is to implement a mobile ECG system based on the DSC algorithm. We first introduce the DSC theorem and its practical implementation. For the DSC application, both the ECG source correlation modeling and the channel code used for SF are essential. VQ is used to construct the correlation model and two preprocess, including QRS complex detection and period normalization, are used to maximize the inter-beat correlation. Compared with conventional VQ, gain-adaptive VQ is shown to obtain better performance. Also proposed is a multiple-choice index assignment scheme (multiple-index VQ) which better matches the DSC application. In this work, RSC is used for SF and a modified BCJR algorithm is proposed for symbol decoding of binary convolutional code. Both the offline simulations and real-time implementation are conducted to verify the proposed ECG compression algorithm. The result shows that the proposed algorithm achieves good reconstruction quality with high compression ratio.

The ECG compression system is implemented on an embedded system and PC. Besides ECG compression, our implementation supports the remote monitoring of ECG signals by using wireless technologies. Our system is distinct from others in using the cell phone to transmit the ECG signals, making it more portable. Moreover, the system still works well when making a phone call. Due to the rapid development of the smart phone, the computing hardware in the phone becomes faster and faster. A future development will be carried out to implement the system on a smart phone.

Future directions in research are listed as follows.

1. Backward gain-adaptive VQ is still an attractive way to model the ECG correlation without extra need of gain-related side information. Other gain estimators may also be investigated to achieve better performance when DSC is applied.
2. We follow the DSC theorem that assumes the correlation between two sources is modeled by a virtual binary symmetric channel. Whether the correlation of the VQ indexes can be modeled by more elaborate channel modes, like Markov-chain model, still needs observing.
3. In this work, only normal ECG signals are investigated. Analyzing the abnormal heart beats is more challenging. Some useful information (heart rate variability) analyzed by BtECG sensor is sent to the server for further usage. Please refer to Appendix B for more information about heart rate variability.

Appendix A. Bluetooth Communication Implementation

In Chapter 4, we described the way how the WinFast PXA310 mini-board communicates with the back-end server. The board accesses the Internet by using a Bluetooth-enabled mobile telephone as a modem. Also, the server (PC) is connected to the Internet. First, the connection between the board and the server is setup and then data transmission begins. The transmission between the board and the telephone is implemented in a Bluetooth link while that between the telephone and the server is the mobile cellular network and the Internet. Data is first transmitted to the telephone and then to the server. To achieve this goal, the Windows Sockets (Winsock) application programming interface (API) provided by Microsoft is used. Details of WinSock are available in [26]. This appendix demonstrates the methods implemented on the board used to communicate with Bluetooth device (the telephone) and server.

[A] Discovering Bluetooth Devices

The main task of discovering a device is to retrieve the address of the device. Four WinSock programming elements are used when creating query for discovering Bluetooth devices: *WSAQUERYSET* structure, *WSALookupServiceBegin* function, *WSALookupNext* function, and *WSALookupServiceEnd* function.

1. Create and configure the WSAQUERYSET variable to restrict the query to Bluetooth devices.
2. Call *WSALookupServiceBegin* function to perform a device inquiry by passing the WSAQUERYSET variable created in the step 2. Also, a parameter is passed to enable Service Discovery Protocol (SDP) to search Bluetooth devices. A handle *h* is returned if the function is successful.
3. Call *WSALookupServiceNext* function to emulate devices scanned in step 2. First, initialize a WSAQUERYSET structure to store device data which will be returned later by *WSALookupServiceNext*. Second, call *WSALookupServiceNext* by passing the handle *h* returned by *WSALookupServiceBegin* and the names and the address of the Bluetooth devices are obtained. The address will be stored in a *bt_addr* variable which will be used when querying for services on the device.
4. Call *WSALookupServiceEnd* function by passing the handle *h* to terminate the process of device discovery.

[B] Querying Service on Remote Bluetooth Devices

Now a Bluetooth device is discovered and its address is stored in the *bt_addr* variable. In querying for service, four WinSock programming elements are also used: *WSAQUERYSET* structure, *WSALookupServiceBegin* function, *WSALookupNext* function, and *WSALookupServiceEnd* function. The procedure of querying service is similar to that of discovering devices, but there exist two differences.

1. When configuring the *WSAQUERYSET* structure, the device address *bt_addr* is specified.
2. When calling the *WSALookupServiceBegin* function, a different parameter is passed to perform a service query on the device.

Services are returned after calling *WSALookupServiceNext* function and each is stored as an integer *channel*, which is the so-called RFCOMM channel number. To connect to the server by using a Bluetooth-enabled mobile telephone, the phone must have the Dial-up Network profile (DUN) service.

[C] Connection Remote Bluetooth Devices

The connection between two Bluetooth devices can be created by using the WinCE 6.0 Com Port emulator facility, which provides access to the RFCOMM layer based on a virtual COM port. Before setting up a connection, a remote Bluetooth device address *bt_addr* and its service *channel* must be discovered. The following describes how the connection is setup and released.

1. Configure the *PORTEMUPortParams* structure by specifying two attributes *bt_addr* and *channel* for a virtual COM port.
2. Register the device by calling the *RegisterDevice* function, which specifies the port type as COM and the port number. Also passed to the function is the *PORTEMUPortParams* structure.
3. Call the *CreateFile* function to open the COM port.
4. If the existing virtual COM port is no longer in use, call the *CloseHandle* and *DeregisterDevice* function to remove the port.

With opening a virtual COM port by assigning the correct *bt_addr* and *channel*, the board now should be able to access the Internet by using the Bluetooth-enabled mobile phone as a modem.

[D] Communication between the Board and the Server

Both the board and the server are capable of accessing the Internet. Now the connection between the board and the server can be established and the transmission is based on TCP client-server model. To realize such transmission, the Winsock API provided by Microsoft is used. The server creates a socket to listen and accept the client’s connection request by calling *socket*, *bind*, *listen* and *accept* function. The board which serves as the client creates a socket and makes a connection request by calling *socket* and *connect* function. Once the request is accepted by the server, the connection is established and data transmission can be realized by calling *send* and *recv* function. Finally, the connection can be released by calling *close* function.

Appendix B. Heart Rate Variability

Heart beats are controlled not only by the sinoatrial node (SA) but also the autonomic nervous system (ANS). The ANS is a part of the nervous system and non-voluntarily controls all organs of the body. The ANS can be classified into two subsystems: sympathetic nervous system (SNS) and parasympathetic nervous system (PSNS). An increase in SNS stimulation causes increase in heart rate, while an increase in PSNS stimulation causes decrease in heart rate. Research has shown the significant relationship between the ANS and cardiovascular mortality. Heart rate variability (HRV) is a physiological phenomenon where the interval between two consecutive heart beats oscillates and is widely used as a tool in heart disease diagnosis. Among many methods [27], two most widely used methods are time-domain methods and frequency-domain methods.

[A] Time Domain Methods

Time-domain methods are based on inter-beat or NN intervals. The term NN is similar to RR, but it emphasizes that the beats are normal beats. The following variables can be calculated for both long-term and short-term ECG recordings:

1. SDNN: The standard deviation of NN interval reflects all the cyclic components responsible for variability and is calculated over a 24-hour period.
2. SDANN: The standard deviation of the average NN intervals is often calculated over 5 minutes, which is a measure of changes in heart rate due to cycles longer than 5 minutes.
3. Square root of the mean squared differences of successive NN intervals (RMSSD).
4. The number of interval differences of successive NN intervals more than 50 ms (NN50).
5. The proportion of NN50 divided by the total number of NN intervals (pNN50).

[B] Frequency Domain Methods

Power spectral density (PSD) analysis, which can be achieved by use of Fast Fourier Transformation (FFT), provides the information of power distribution as a function of frequency. Total power (TP, 0-0.625 Hz), very low frequency (VLF, $\leq$ 0.04 Hz), low frequency (LF, 0.04-0.15 Hz) and high frequency (HF, 0.15-0.4 Hz) are three main components obtained from short-term recordings of 2-5 minutes. Measurement of these components is made in absolute values of power (ms^2). It is found that HF activity reflects the parasympathetic tone and fluctuations caused by respiratory sinus arrhythmia. LF reflects both parasympathetic and sympathetic tone. However, the physiological meaning of VLF is unclear and disputable.

[C] Measured Data of BtECG Sensor

BtECG sensor is equipped with a KY202 DSP module for HRV. The technologies of HRV include R point detection, interpolation in time domain, and FFT in frequency domain. The analysis result is formatted in ASCII code such as “HR=085”, which refers to the mean heart rate is 85 beats per minutes. More results are summarized as follows.

Table C.1 HRV parameters measured from BtECG sensor

Command	Function	Remarks
HR=nnn	Mean heart rate	n=0~9 (Here, we have three consecutive n, which denotes that it is three bytes long.)
SD=nnn	Standard deviation of RR intervals	n=0~9
HF=nnnnn	High frequency power (0.15~0.4Hz) of HRV	n=0~9
LF=nnnnn	Low frequency power (0.04-0.15Hz) of HRV	n=0~9
VL=nnnnn	Very low frequency power (0-0.04Hz) of HRV	n=0~9
TP=nnnnn	Total power (0-0.625Hz) of HRV	n=0~9
WL=nnnnn	Windows length	n=0~9

Bibliography

[1] K. Watanabe, K. Takizawa, and T. Ikegami, "Novel Joint Source-Channel Coding of Periodic ECG Signals for Reliable Wireless Patient Monitoring," *IEICE Transactions on Commmunication*, vol. E93-B, no. 4, pp. 819-825, Apr. 2010.

[2] 行政院衛生署衛生統計資訊專區. *99 年度死因統計*. [Online]. Available: http://www.doh.gov.tw/CHT2006/DM/DM2_p01.aspx?class_no=117&now_fod_list_no=4146&level_no=1&doc_no=38031. [Accessed: Dec. 2011]

[3] S.N. Yu and J.C. Cheng, "A Wireless Physiological Signal Monitoring System with Integrated Bluetooth and WiFi Technologies," in *27th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Shanghai, China, Sept. 1-4, 2005, pp. 2203-2206.

[4] M. F. A. Rasid and B. Woodward, "Bluetooth Telemedicine Processor for Multichannel Biomedical Signal Transmission via Mobile Cellular Networks," *IEEE Transactions on Information Technology in Biomedicine*, vol. 9, no. 1, pp. 35-43, 2005.

[5] Y. Chu and A. Ganz, "A Mobile Teletrauma System Using 3G Networks," *IEEE Transactions on Information Technology in Biomedicine*, vol. 8, no. 4, pp. 456-462, 2004.

[6] S. M. S. Jalaleddine, C. G. Hutchens, R. D. Strattan, and W. A. Coberly, "ECG Data Compression Techniques-A Unified Approach," *IEEE Transactions on Biomedical Engineering*, vol. 37, no. 4, pp. 329-343, 1990.

[7] A. Bilgin, M. W. Marcellin, and M. I. Altbach, "Compression of Electrocardiogram Signals Using JPEG2000," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 4, pp. 833-840, 2003.

[8] H.H. Chou, Y.J. Chen, Y.C. Shiau, and T.S. Kuo, "An Effective and Efficient Compression Algorithm for ECG Signals with Irregular Periods," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 6, pp. 1198-1205, 2006.

[9] *JPEG2000 Part I Final Draft International Standard*, ISO/IEC JTC1/SC29/WG1, Doc. No. N1855 2000.

[10] C.C. Sun and S.C. Tai, "Beat-based ECG Compression Using Gain-Shape Vector Quantization," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 11, pp. 1882-1888, 2005.

[11] A. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260-269, Apr. 1967.

[12] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rrate," *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284-287, Mar. 1974.

[13] D. Slepian and J. Wolf, "Noiseless Coding of Correlated Information Sources," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471-480, 1973.

[14] S. S. Pradhan and K. Ramchandran, "Distributed Source Coding Using Syndromes (DISCUS): Design and Construction," *IEEE Transactions on Information Theory*, vol. 49, no. 3, pp. 626-643, Mar. 2003.

[15] Y.C. Pan, "Index-Based Iterative Source-Channel Decoding," M.S. thesis, Department of Communication Engineering, National Chiao Tung University, Hsinchu, Taiwan, Republic of China, 2008.

[16] P. Tan and J. Li (Tiffany), "A Practical and Optimal Symmetric Slepian-Wolf Compression Strategy Using Syndrome Formers and Inverse Syndrome Formers," in

Proceedings of 43rd Allerton Conference on Communication, Control and Computing, Monticello, Illinois, USA. , Sept. 28-30, 2005.

[17] Z.Y. Tu, J. Li (Tiffany), and Rick S. Blum, "An Efficient SF-ISF Approach for the Slepian-Wolf Source Coding Problem," *Eurasip Journal on Applied Signal Processing-Special Issue on Turbo Processing*, vol. 2005, no. 6, pp. 961-971, May 2005.

[18] J. Li (Tiffany) and H. Alqamzi, "An Optimal Distributed and Adaptive Source Coding Strategy Using Rate-Compatible Punctured Convolutional Codes," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005 (ICASSP '05)*. Philadelphia, Pennsylvania, USA, Mar. 18-23, 2005, pp. iii/685-iii/688 Vol. 3.

[19] Frank G. Yanowitz. *Lesson 1: The Standard 12 Lead ECG*. [Online]. Available: http://library.med.utah.edu/kw/ecg/index.html. [Accessed: Apr. 2012]

[20] *MIT-BIH Arrhythmia Database*. [Online]. Available: http://www.physionet.org/physiobank/database/mitdb. [Accessed: Aug. 2011]

[21] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer Academic, 1992.

[22] G. M. Friesen, T. C. Jannett, M. A. Jadallah, S. L. Yates, S. R. Quint, and H. T. Nagle, "A Comparison of the Noise Sensitivity of Nine QRS Detection Algorithms," *IEEE Transactions on Biomedical Engineering*, vol. 37, no. 1, pp. 85-98, 1990.

[23] R.G. Lee, I.C. Chou, C.C. Lai, M.H. Liu, and M.J. Chiu, "A Novel QRS Detection Algorithm Applied to the Analysis for Heart Rate Variability of Patients with Sleep Apnea," *Journal of Biomedical Engineering—Applications, Basis and Communications (EI)*, vol. 17, no. 5, pp. 258-262, Oct. 2005.

[24] J.H. Chen and A. Gersho, "Gain-Adaptive Vector Quantization with Application to Speech Coding," *IEEE Transactions on Communications*, vol. 35, no. 9, pp. 918-930, Sept. 1987.

[25] Y. Linde, A. Buzo, and R. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84-95, 1980.

[26] *Microsoft Developer Netowrk (MSDN)*. [Online]. Available: http://msdn.microsoft.com/. [Accessed: Apr. 2011]

[27] Task Force of the European Society of Cardiology and the North American Society of Pacing Electrophysiology, "Heart Rate Variability: Standards of Measurement, Physiological Interpretation, and Clinical Use," *Circulation*, vol. 93, no. 5, pp. 1043-1065, 1996.