Louisiana State University
LSU Digital Commons

LSU Doctoral Dissertations | Graduate School

2012

New Coding/Decoding Techniques for Wireless Communication Systems

Yonas Gebregziabher Debessu
Louisiana State University and Agricultural and Mechanical College

Follow this and additional works at: https://digitalcommons.lsu.edu/gradschool_dissertations

 Part of the Electrical and Computer Engineering Commons

Recommended Citation

Debessu, Yonas Gebregziabher, "New Coding/Decoding Techniques for Wireless Communication Systems" (2012). *LSU Doctoral Dissertations*. 1898.
https://digitalcommons.lsu.edu/gradschool_dissertations/1898

This Dissertation is brought to you for free and open access by the Graduate School at LSU Digital Commons. It has been accepted for inclusion in LSU Doctoral Dissertations by an authorized graduate school editor of LSU Digital Commons. For more information, please contactgradetd@lsu.edu.

NEW CODING/DECODING TECHNIQUES FOR WIRELESS COMMUNICATION SYSTEMS

A Dissertation
Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

The Department of
Electrical and Computer Engineering

by
Yonas Debessu
B.S. Jimma University
M.Tech. Indian Institute of Technology, Madras
August 2012

To Amlakawit and Robel

ACKNOWLEDGMENTS

First of all, I would like to express my utmost gratitude to my major professor and dissertation adviser, Dr. Hsiao-Chun Wu, for his supervision, advice, and guidance during my Ph.D. research. I am thankful that he was always encouraging me to work independently, and at the same time he was there to guide me when I falter and to inspire me when I get discouraged. Without his insightful comments and constructive criticisms, this dissertation wouldn't have been completed.

I am also very grateful to other faculty members in my Ph.D. dissertation committee, including Dr. Jerry Trahan, Dr. Xuebin Liang, Dr. Mark Davidson, and Dr. Juana Moreno for kindly spending their valuable time and providing me with outstanding suggestions and advice on my dissertation work.

I also want to acknowledge my laboratory mates Dr. Kun Yan, Dr. Lu Lu, Dr. Xiaoyu Feng, Ms. Charisma Edwards, Ms. Hongting Zhang and Mr. Tian Xia for their warm friendship, for creating a motivating atmosphere, and for the invaluable collaboration on research.

I am also thankful to my wife and my son, for their love, support, and patience during this Ph.D. research. I am greatly indebted to them.

Finally, I would like to express my appreciation to the Huel D. Perkins Fellowships that funded my tuition and part of my research.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Wireless communication encompasses cellular telephony systems (mobile communication), wireless sensor networks, satellite communication systems and many other applications. Studies relevant to wireless communication deal with maintaining *reliable* and *efficient* exchange of information between the *transmitter* and *receiver* over a wireless channel. The most practical approach to facilitate reliable communication is using *channel coding*. In this dissertation we propose novel coding and decoding approaches for practical wireless systems. These approaches include variable-rate convolutional encoder, modified turbo decoder for local content in Single-Frequency Networks, and blind encoder parameter estimation for turbo codes. On the other hand, energy efficiency is major performance issue in wireless sensor networks. In this dissertation, we propose a novel hexagonal-tessellation based clustering and cluster-head selection scheme to maximize the lifetime of a wireless sensor network.

For each proposed approach, the system performance evaluation is also provided. In this dissertation the reliability performance is expressed in terms of bit-error-rate (BER), and the energy efficiency is expressed in terms of network lifetime.

1. INTRODUCTION

Channel coding, in telecommunication applications, is a technique commonly used for *detecting* and *correcting* errors during data transmissions over *unreliable* or *noisy* channels. The on-going studies on theory of coding are to discover efficient coding and decoding techniques that facilitate reliable communications over any noisy channel. The fundamental theorem regarding the *achievable efficiency* of channel coding was established by C. Shannon in his 1948 paper [2]. For a channel with capacity C, Shannon stated his theory as follows:

> It is possible to send information at the rate C through the channel with as small a frequency of errors or equivocation as desired by proper encoding. This statement is not true for any rate greater than C.

Although this distinguished paper pronounces the achievable rate for a reliable communication, it does not facilitate any practical coding/decoding scheme that achieves this optimal rate. After Shannon's aforementioned contribution, the field of coding theory has experienced tremendous evolution starting from the discovery of *Hamming Codes* to the discovery of *Turbo Codes*. However, there still exists a need for development of new efficient codes that are suitable for various specific applications under given circumstances.

To address the urgent demands for new efficient coding/decoding methodologies, this dissertation is dedicated to exploration on advanced coding and decoding techniques that

enhance the reliability of the communications for different types of applications in wireless networks and systems. The *reliability* is measured in terms of *bit error rate* (BER) for certain channel conditions.

1.1 Existing Solutions and Limitations

Error-control coding, or channel coding, plays a crucial role in modern digital communication systems. A simplified block diagram of a digital communication system is depicted by Figure 1.1.

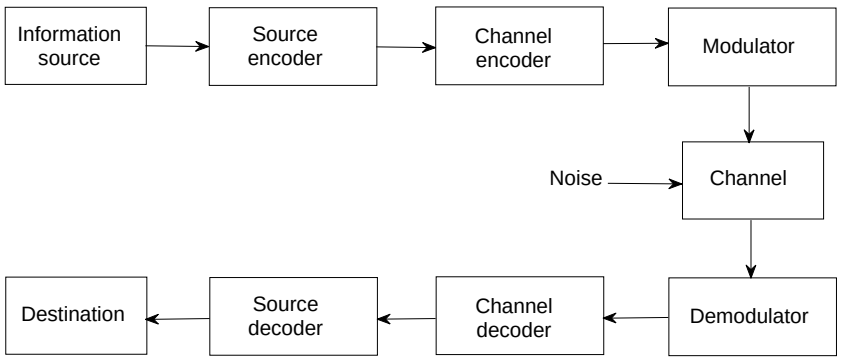

Figure 1.1: Block diagram of a typical digital communication system.

In many applications, such as telephony, the original information source generates an *analog signal*, which is digitalized by the *analog-to-digital convertor* (A/D) [3]. The resultant *digital signal* is then passed to the *source encoder*, which maps the input information bits into an efficient representation by eliminating redundancy so as to provide a small expected data length [4]. This mechanism is also called *compression*. After compression, a *channel encoder* transforms the information sequence into a discrete-time coded sequence [5]. Channel coding is performed on the information data in such a way to mitigate the negative effects of noise

and interference incurred in the communications channel by involving redundancy, where the extra bits are included for the future error correction or error detection at the receiver. The coded bits are *modulated* to generate an analog signal by the *modulator*. To undertake an efficient transmission of signals over the channel and to enable channel sharing, the modulated signal is up-converted to a higher radio frequency (RF). This RF signal passes through a channel and is received by the *receiver*.

At the receiver, the incoming signal is first demodulated by the demodulator. The output sequence from the demodulator is then decoded at the channel decoder. The output bit sequence from the channel decoder is then further decoded by the source decoder to recover the original information bits, which will be utilized at the destination.

The capability of the demodulator to restore the transmitted signals is hampered by different channel factors including *noise*, *interference*, *Doppler shift*, *multipath fading*, etc. These factors result in *demodulation errors* and hinder reliable communication. The purpose of a channel encoder is, therefore, to facilitate a way to combat these errors caused by the unfavourable channel conditions.

Today's error correction codes can be generally classified into two major categories: (i) *block codes* and (ii) *convolutional codes* [5]. In block coding, the information sequence is segmented into blocks, each of them consists of k symbols, and then each block of symbols is independently transformed into an n-tuple *codeword*. In addition to the parameters k and n, block codes may also be characterized by the *minimum distance*, $d_{\min}$, which is actually the minimum weight of a nonzero codeword in a given code [5]. Accordingly, block codes are represented as (n, k) or $(n, k, d_{\min})$. Owing to their beautiful structured nature, the block codes were explored first in history. Among all block codes, *Hamming codes* were developed

first [6], which are *linear block codes* with parameters given by $(2^m - 1, 2^m - m - 1, 3)$ for $m \geq 3$, and are capable of correcting a single error at maximum per every codeword. For $m = 2$, Hamming codes become trivial *repetition codes*. Golay codes (see [7]) were also developed early, which include *triple-error-correcting binary* $(23, 12, 7)$ *code* and *double-error-correcting ternary* $(11, 6, 5)$ *code*. Another early class of codes are the *Reed-Muller multiple-error-correcting codes*. The Reed-Muller codes were discovered by Muller in 1954 [8] and the decoding algorithm was first developed by Reed in 1954 [9]. Next, cyclic codes were invented by Prange at the Air Force Cambridge Research Center (AFCRC) [10]. A class of cyclic multiple-error-correcting codes known as *Bose-Chaudhuri-Hocquenghem (BCH) codes* were discovered in 1959 [11, 12]. Similar codes, known as *Reed-Solomon (RS) codes*, were based on the idea of employing the elements of a finite field as *alphabets*. Reed-Solomon (RS) codes are non-binary cyclic error-correcting codes [13]. After an efficient decoding algorithm for RS codes was developed by Berlekamp [14, 15], the RS codes have been widely used in industrial and consumer electronic devices. The RS codes are the prevalent channel coding schemes in modern telecommunications related industrial standards.

The second class of codes, namely the convolutional codes, were first introduced by Elias in 1955 [16]. In [16], it has been shown that there exist the exponential upper- and lower-bounds of the error probability in terms of a function of delay. Elias also indicated that it is possible to transmit at a rate equal to the channel capacity, error-free with probability of one. At a convolutional encoder, a stream of k incoming bits is applied to the shift registers. At the output of the convolutional encoder, the combinations of the intermediate bits of those shift registers, namely n code bits, will be delivered. Hence, the *code rate* is $\frac{k}{n}$. The error correcting capability of a convolutional encoder, in terms of error probability, is expressed

as a function of the corresponding *constraint length*, ν, which denotes the sum of all shift-registers' lengths. Large constraint-length codes tend to facilitate low error probabilities at the receiver. Unfortunately, the transceivers equipped with large constraint-length convolutional coding mechanisms are inflicted with high decoder complexity. There are several effective decoding algorithms for convolutional codes, but the most popular is the *Viterbi algorithm*, invented by Andrew Viterbi in 1967 [17].

In the various coding techniques as previously discussed, lowering bit-error-rate requires either *rate reduction* or *bandwidth expansion* to accommodate the incremented parity bits. A new coding technique was designed by Ungerboeck [18], which improves the error performance without sacrificing data rate or expanding bandwidth. This is achieved by using a novel technique called *trellis coded modulation (TCM)* which combines modulation with coding so as to increase the *free Euclidean distance*, or the minimum distance between any two symbols in the same coset.

In 1993, a new class of codes, namely Turbo codes, were first introduced by Berrou, Glavieux, and Thitimajshima [19, 20]. These codes are constructed using a parallel concatenation of two *recursive systematic convolutional-encoders* (RSC). The information bits get through the first RSC directly and pass the second RSC through the interleaver. At the receiver, two separate decoders will be used accordingly. Each decoder invokes an extrinsic information from the other decoder and utilizes it in the maximum *a posteriori* (MAP) algorithm. Berrou, Glavieux, and Thitimajshima have designed a coding scheme that achieves a bit-error-rate as low as 10^{-5} using a rate 1/2 code over an additive white Gaussian noise (AWGN) channel and binary phase shift keying (BPSK) modulation at the signal-to-noise ratio of 0.7 dB.

Nevertheless, advanced coding/decoding techniques are still in demand for the future telecommunication systems. The rationale is very clear. The protocols to be developed under way will be much more complicated than the basic transceiver model depicted in Figure 1.1. The quality-of-service requirements will be rather different from one application to another. The tradeoffs among hardware implementation cost, resource dependence, and quality-of-service can vary a lot among different telecommunication applications and industrial standards. Therefore, we try to address the needs of some applications by designing innovative coding/decoding techniques to advance telecommunication technologies in this dissertation work.

1.2 Research Motivation and Application

In the previous section, we have introduced the major developments in coding theory, an area of study drawing enormous research interest and great discoveries. In addition, we have mentioned that there is still need for further research and developments on application-oriented reliability and efficiency improvement. This need for more application-specific schemes is manifested by the demand for various coding/decoding techniques in different communication standards.

In [21], an overview was presented on techniques to select an appropriate code for a particular application. For example, when the application requires a decoding scheme that is executed using combinatorial circuits, as in computer memory and data-storage systems, block codes are the top choice. In computer memories, since computational efficiency is also a major requirement, single-error-correcting and double-error-detecting codes are preferred.

On the other hand, for storage devices such as compact discs, the primary requirement for the codes is that they should be robust against *burst errors*, and hence RS codes are widely used in this application. Another concern in practical error control coding is bandwidth limitation for communications over telephone channels. The best suited code for this application is trellis coded modulation, where the error protection is facilitated by expanding the signal constellation while keeping the bandwidth fixed [22]. Current standards of mobile communications and digital video broadcast adopt turbo codes and convolutional codes, mainly due to their superior bit-error-rate performance.

Motivated by the general reality that different applications work optimally with respect to different codes, in this dissertation, we will develop new coding techniques, new decoding algorithms, and new encoder estimation techniques for future wireless communication applications. Our general objective is to improve the bit-error-rate performance, which is a crucial measure to evaluate the *data transfer reliability*. Furthermore, to enhance the efficiency of data communications in battery-power limited wireless systems, we develop a novel algorithm which maximizes the lifetime of any network subject to the limited energy (battery) resource.

The solutions to different reliability and efficiency problems encountered in wireless communications presented in this dissertation include: variable-rate encoders for mobile communications, blind encoder estimation method for turbo codes, local-content extraction for single-frequency networks used in digital video broadcast (digital television), and sensor-clustering for increasing the lifetime of any wireless sensor network. We will investigate these solutions for three different wireless systems later on throughout this dissertation.

1.2.1 Mobile Communications

Mobile (cellular) communication technology is a prevalent wireless communication technology. In cellular communications, the primary source of performance degradation is not *thermal noise* generated in the receiver, which is modeled by the classic additive white Gaussian noise (AWGN) channel. Rather, it is the *multi-path fading* [23]. In cellular communications, small-scale fadings characterize the rapid variations in the received signal power, which occur due to a short-distance movement by the mobile device or any motion in its surroundings. Since the transmitted signal may arrive at the receiver through a direct *line of sight* (LOS) propagation or after being reflected from the ground or the surrounding objects, the signals experience a multipath channel. Hence, the impulse response of the channel is a train of impulses inferring the time dispersive nature of the channel [3]. If the mobile radio channel has a constant gain and a linear phase for the underlying frequency band (in use) greater than the bandwidth of the transmitted signal, then the signal dispersion behavior of the channel is categorized as *frequency-nonselective* (flat) fading. Otherwise, it is called *frequency-selective* fading. The dynamism of the environment around a mobile device and the motion of the device itself give rise to the multipath variations, or the time-variation property of the channel. If the channel impulse response changes at a rate higher than that of the transmitted signal, then the channel is categorized as *fast-fading*. Otherwise, it is called *slow-fading* [24]. In this dissertation, we will concentrate on the flat and slow fading channels.

Under fading channels, the changes in BER are linearly inversely proportional to SNR while in non-fading AWGN channels, BER decreases exponentially with the increment of

SNR [3]. Due to the limitation on the battery size and other technical concerns, one cannot simply increase the transmitting power arbitrarily to upgrade the system performance. This limitation is yet more important for mobile devices as their physical sizes stringently limit their battery capacity. Therefore, to improve the system performance, one needs to employ channel coding along with other techniques such as diversity subject to the limited transmitting signal power. In this dissertation, we will design a new coding scheme which is robust over different time-varying channels.

1.2.2 Digital Video Broadcasting-DVB

Digital video broadcasting (DVB) has been playing an important role in the current electronic technologies since its universal deployment in 2009. In this dissertation work, among the varieties of DVB standards, we will concentrate on the Digital Video Broadcasting-Satellite services to Handhelds (DVB-SH) standard. DVB-SH systems broadcast video signals to the target receivers over a hybrid satellite and terrestrial infrastructure. The services are thus provided to handheld terminals or devices characterized as light-weight and battery-powered apparatus, such as mobile phones, laptops, etc. This DVB-SH technology provides broadcasting services at the frequencies below 3 GHz through a single-frequency network. A *Single Frequency Network* (SFN) is a network of transmitters that use the same frequency to transmit the same information. Since DVB is a one-way communication technology, SFNs can help to extend the coverage area without the need of extra frequency bandwidth resource.

Besides, DVB-SH systems provide other services including radio programs as well as download services [25]. The DVB-SH system coverage is obtained by combining a *Satellite Component* (SC) and a *Complementary Ground Component* (CGC). The main purpose of

incorporating CGC is to ensure service continuity in the areas where the satellite alone cannot provide the required quality-of-service (QoS). The SC, transmitted via a satellite, can cover a wide area, whereas the CGC, transmitted via terrestrial transmitters, provides a cellular-type coverage. This localized nature of the CGC broadcasting allows the terrestrial broadcasters to insert *local content* into the CGC. This local-content insertion is performed using *hierarchical modulation*.

In this dissertation work, we will develop an innovative scheme to extract the local content from the received signal effectively. To decode the extracted local content, we design a new modified turbo decoder. In addition, we invent a new blind estimation technique for the tubo code identification in a wireless receiver.

1.2.3 Wireless Sensor Networks

Wireless sensor networks (WSNs) are a current scientific and engineering research focus. WSNs are composed of low-cost sensor nodes cooperating in gathering and reporting application-specific data. Broad WSN applications can be found thanks to the compactness of modern electronic devices as well as the inventions of the efficient and robust implementations of digital signal processing algorithms and wireless network protocols. Current WSN applications range from military tactics to weather monitoring tools, from deep-water sensing to land-surface navigation, from home-security surveillance to nuclear-radiation measurement, and so on. In most applications, the sensor nodes are scattered over the coverage area and have the capabilities of sensing and collecting data, processing data, and routing data to the *sink* or base station (BS) [26]. Each sensor is equipped with a battery whose size is constrained by the dimensions of the sensor node. Since energy density of any battery

is limited, it is clear that the size limitation imposes a significant constraint on the power capacity of sensor nodes. This constraint is related to the limitation on both transmission range and throughput in sensor communications.

As we have mentioned above, most applications of WSNs preclude the recharging or replacement of battery. Therefore, if the battery power of a WSN node is depleted, then this sensor node will cease sensing and processing information and will stop forwarding data during routing. Hence, the major objective of performance improvement in wireless sensor networks is obviously to increase the battery lifetime. Battery life time of a sensor node has a direct consequence on the lifetime of the entire network thereby. In this dissertation research, the lifetime of a sensor network is considered to be the time elapsed between the deployment of the network and the first death of any sensor node. In our research, we propose a new method of organizing the sensor nodes in groups, called *clusters*, in order to elongate the entire network lifetime.

2. VARIABLE-RATE CONVOLUTIONAL CODING

In this chapter we will discuss the coding scheme for a time-varying channels in mobile communication. Since the channel condition is time-varying, the constant-rate convolutional encoder cannot facilitate the best error protection that can be achieved in practice. To mitigate this drawback, we propose a new variable-rate convolutional encoder which can adapt to the dynamical channel conditions subject to the channel state information (CSI) available at the transmitter. Moreover, we also discuss the design of adaptive coded-modulation system built upon our proposed variable-rate convolutional encoder.

2.1 Literature Review

As mentioned in Chapter 1, the *channel gain* in mobile communications, for most urban and suburban areas, is usually characterized by the *Rayleigh-distributed fades* where the *coherence time* is determined by the random motion of the mobile receiver or the surrounding objects [24]. If the coding or modulation scheme does not adapt to the dynamical fading channel conditions, the communication sessions cannot be maintained at an acceptable performance level. The performance of any convolutional code depends on the *free distance* d_{free}, which is the minimum distance between any pair of finite-length codewords [5].

Thus, to achieve a lower bit error rate (BER), a communication system should rely on a convolutional channel encoding scheme with a large free distance [5]. In a fading channel,

this objective can be easily achieved by adopting low-rate convolutional codes having a free distance that is large enough in the presence of deep fades. This pessimistic design incurs the inefficient utilization of the available resources such as *bandwidth* and *transmitting power* [27]. Hence, different adaptive modulation and coding approaches have been proposed in order to establish the variable-rate coding schemes that provide robust error protection in different channel conditions.

Adaptive coding and modulation approaches were proposed for fading environments in the presence of additive white Gaussian noise (AWGN) in [27–33]. The transmission strategies were adapted to different channel conditions by varying the symbol rate or the constellation size, controlling the transmitting power, adjusting the coding scheme, etc. These strategies were performed to achieve a better *link spectral efficiency*, a lower BER, and less *transmitting power dissipation*, while different system requirements such as signal-to-interference ratio, transmission delay, data rate, and BER are still satisfied. For example, in [27], the separability of code design and modulation design for trellis and lattice codes, which were the special cases of the *coset codes* [34], was exploited for the benefit of the adaptive transmission strategy.

Similarly, in [35, 36], an adaptive transmission scheme known as *adaptive trellis-coded multiple-phase-shift-keying* was proposed. In this scheme, a $\frac{1}{2}$-rate convolutional encoder was employed to construct a family of *pragmatic trellis codes* and then a slightly modified Viterbi decoder was used at the receiver. To exploit the time-varying characteristics of the channel, more information is transmitted via high-rate pragmatic trellis codes when a good channel condition is attained. Nevertheless, when the channel condition becomes bad, the lower-rate trellis codes and the repetition codes are applied to secure reliable communications.

In all of the aforementioned methods, the adaptive coded-modulation strategy is generally facilitated by varying the number of uncoded bits that are used to select a particular constellation point from a given coset. This coset is determined by the outputs of the underlying convolutional encoder. Note that the channel encoders involved in all these schemes were just fixed-rate binary encoders. Different from the aforementioned adaptive transmission mechanisms, all of which were built upon fixed-rate convolutional encoders, there exist variable-rate channel encoders on the other hand. In [37], variable-rate turbo codes were proposed using the parallel concatenation of *tailbiting recursive systematic multi-binary (m-ary) convolutional codes*; the rate variability could range from $\frac{1}{2}$ to $\frac{7}{8}$ subject to a variable number of input bits to the encoder. In [38], the non-uniform error protection capabilities of the family of *rate-compatible punctured-convolutional codes* were studied; moreover, the interleaver was also used to mitigate the fading effect. In [39], rate-compatible punctured-convolutional codes were adopted to enable the variable-rate error protection in direct-sequence code-division multiple-access (DS-CDMA) systems. In [40], a variable-rate coding scheme was implemented with the help of punctured and repetition codes, where *punctured codes* were generated by periodically deleting coded bits prior to transmission.

Our proposed approach here is different from all of these existing methods relevant to variable-rate convolutional codes, since we do not need to rely on punctured codes and repetition codes. Our proposed scheme is also different from other adaptive coded-modulation schemes because the binary encoder we adopt is a variable-rate encoder. Therefore in this dissertation, we present a novel method for employing variable-rate convolutional codes. In our design, a single shift register is used to implement all possible convolutional codes with different rates. The coded bits will be picked by a *rate selector* for the ultimate modulation

in the transmitter. This new scheme can have a great and promising advantage over other existing methods based on punctured and repetition codes since our proposed method can exploit the *good convolutional codes* with a large free-distance. The *good convolutional codes* (see [41]) at different rates are used in our new scheme without any repetition or puncturing mechanism; therefore, the desirable *free distance property* will not be altered.

2.2 Our Proposed New Transmitter with Adaptive Convolutional Coding

In this section, we will present our proposed new transmitter built upon adaptive coding. This new transmitter can be widely adopted for all kinds of digital communication systems. The details are introduced in the subsequent subsections.

2.2.1 Adaptive Convolutional Coding

In our proposed adaptive coding system, we assume that the transmitter can acquire the complete knowledge about the channel state information. For slow fading channels, this assumption is valid with the help of an approximately error-free, low-capacity feedback channel. The instantaneous rate of our proposed variable-rate convolutional encoder is a function of the instantaneous channel state information. The proposed variable-rate encoder has three main components: a *fixed-rate convolutional encoder*, a *rate selector*, and a *queue*. The fixed-rate convolutional encoder serves as a core element with k input bits and η coded bits, each of which is an output of an adder (see Figure 2.1). These η output bits will be fed to the *rate selector*, which determines the rate of the transmitted codewords based on the current knowledge of the fading channel. Therefore, the rate selector is a function $\mathcal{R}$ from a

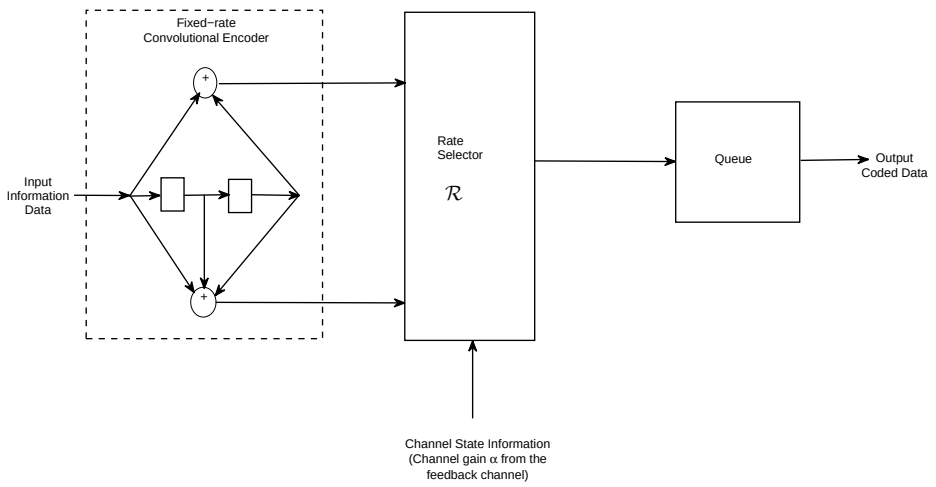

Figure 2.1: Our proposed new transmitter involving a variable-rate convolutional encoder.

vector of length η to a vector of length $n(\alpha)$ over the binary field F_2. It is given by

$$\mathcal{R} : F_2^{\eta} \rightarrow F_2^{n(\alpha)}, \tag{2.1}$$

where α is the *instantaneous channel gain* and $n(\alpha)$ is the number of coded bits given the instantaneous channel gain α.

Since the rate selector generates $n(\alpha)$ coded bits, the *instantaneous rate* of the variable-rate encoder given a channel gain α will therefore be $R(\alpha) = \frac{k}{n(\alpha)}$. Nevertheless, for a communication system where the transmission rate is considered to be fixed, we cannot directly modulate and transmit the coded bits arising from this rate selector in real time. To ensure that the transmitter conveys message at a fixed rate in practice, we propose to employ a *queue* that will buffer coded data prior to the actual modulation. Consequently, our proposed transmitter is depicted in Figure 2.1, where a simple example consisting of a four-state convolutional encoder with $k = 1$, $\eta = 2$, and a *constraint length* $\nu = 2$ (refer to [5] for the definition of constraint length) is illustrated. Later, in Subsection 2.2.3, we will discuss

in detail how the rate selector can be used to implement the variable-rate convolutional encoder.

2.2.2 Our Proposed New Adaptive Coded Modulation System

With our aforementioned variable-rate convolutional encoder stated in Subsection 2.2.1 as a basic building block, we can design a new adaptive codded-modulation strategy. In a generic coded-modulation scheme, the system contains a binary channel encoder, a coset selector, and a signal point selector [34]. In the conventional non-adaptive coset code design in [34], the number of coded bits and the number of uncoded bits are predetermined as some fixed values. In such a non-adaptive coded-modulation scheme, the binary encoder can be a block encoder or a convolutional encoder. When a block encoder is adopted, the corresponding coset codes will turn out to be lattice codes. On the other hand, when a convolutional encoder is used, the corresponding coset codes will be trellis codes instead.

In our proposed adaptive coded-modulation scheme, the binary encoder is a variable-rate convolutional encoder where the adopted constellation is an M-ary quadrature amplitude modulation (M-QAM). Thus, the constellation has M signal points and each signal point can represent $m = \log_2 M$ bits. In the coded modulation, these 2^m signal points are partitioned into $2^{k+r(\alpha)}$ subsets, where k is the number of input information bits to the binary encoder and $r(\alpha)$ is the redundancy of the encoder given the channel gain α. A popular partitioning method can be found in [34]. How to determine $r(\alpha)$ will be introduced in Subsection 2.2.3. After partitioning, each subset will be composed of $2^{m-k-r(\alpha)}$ signal points which can be represented by the $m - k - r(\alpha)$ uncoded bits.

Figure 2.2 illustrates the structure of our proposed adaptive coded-modulation scheme.

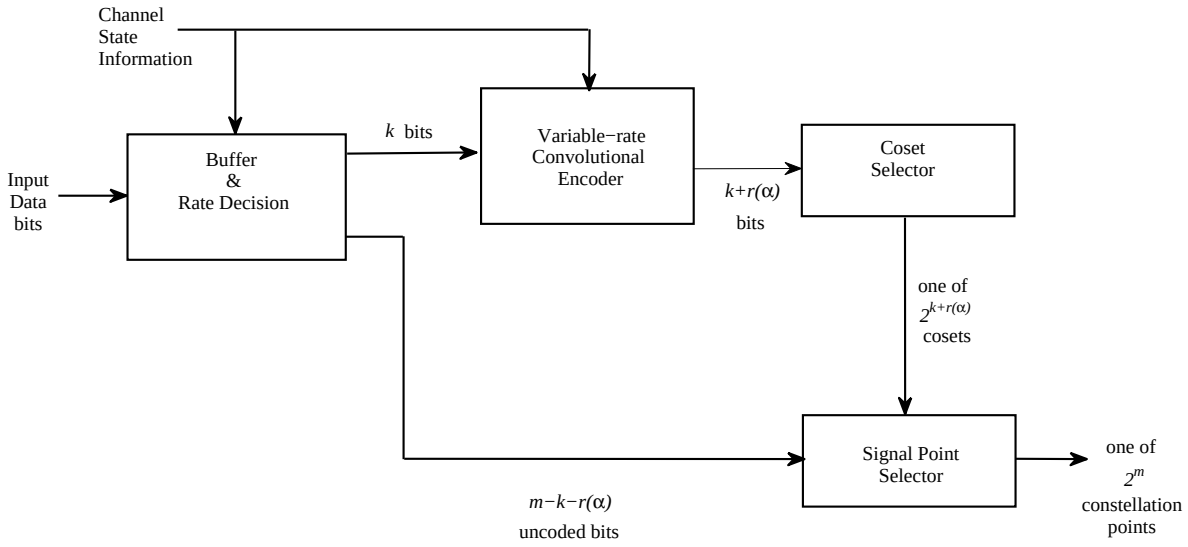

Figure 2.2: Our proposed new adaptive coded modulation scheme involving a variable-rate convolutional encoder.

Note that the general structure for coded-modulation involves every subsystem in Figure 2.2 except that our proposed variable-rate convolutional encoder is substituted with a simple fixed-rate channel encoder by other existing techniques. The first block in our proposed adaptive coded-modulation system determines the number of uncoded bits, namely $m - k - r(\alpha)$, subject to the channel state information (CSI) available at the transmitter. The CSI is also provided to the variable-rate convolutional encoder which takes k input bits to generate $k + r(\alpha)$ coded bits at the output. These coded bits are sent to the coset selector. Then, the coset selector will choose one of the $2^{k+r(\alpha)}$ subsets from a pre-partitioned signal constellation. In parallel, the uncoded bits are also sent to the *signal point selector* and they will be modulated by a signal point drawn from the $2^{m-k-r(\alpha)}$ constellation points in the particular subset just chosen by the coset selector.

In order to exploit the time-varying characteristics of the channel, the redundancy $r(\alpha)$

and the number of uncoded bits $m-k-r(\alpha)$ should both vary in our scheme. Accordingly, the total number of information bits per channel use, namely $m-r(\alpha)$, in other words, the *spectral efficiency*, is varied with respect to the dynamical channel gain. Higher spectral efficiency can be attained in good channel conditions by decreasing the redundancy so that the information bits per channel use can increase. On the other hand, when the channel condition becomes bad, the redundancy is increased in the trade-off of spectral efficiency for reliable communications.

2.2.3 Details of Variable-Rate Convolutional Encoder

We have provided the general description of our proposed adaptive coded-modulation system. The details of its core element, namely variable-rate convolutional encoder, will be introduced here. Convolutional codes that have identical coding rate and equal constraint length may have different bit-error-rate performances when they are decoded using the maximum likelihood decoding scheme (Viterbi algorithm). Some codes may lead to a better performance than others in a way that their performance will approach the lower-bound of the bit-error-rate more closely. Codes that attain such a superior performance are termed as *good codes* [5].

In this dissertation, we propose to generate variable-rate codes based on good codes only. We consider a variable-rate channel encoder which can switch the code rate among three different values at any time. These three candidate rates are set to be $\frac{1}{2}$, $\frac{1}{3}$, and $\frac{1}{4}$. The probability for selecting each candidate is set to be $\mathbb{P}_{\frac{1}{2}}$, $\mathbb{P}_{\frac{1}{3}}$ and $\mathbb{P}_{\frac{1}{4}}$, respectively. The *overall*

average transmission rate R_{ave} of our proposed system is thus given by

$$R_{ave} = \frac{1}{2\,\mathbb{P}_{\frac{1}{2}} + 3\,\mathbb{P}_{\frac{1}{3}} + 4\,\mathbb{P}_{\frac{1}{4}}}. \tag{2.2}$$

In this dissertation, we are interested in comparing our proposed method with the fixed-rate convolutional encoder of rate $\frac{1}{3}$, which is the medium value of the range of R_{ave}. Therefore, we will set $\mathbb{P}_{\frac{1}{2}} = \mathbb{P}_{\frac{1}{4}}$ and $\mathbb{P}_{\frac{1}{3}} = 1 - 2\mathbb{P}_{\frac{1}{2}}$ so as to achieve the same average rate $R_{ave} = \frac{1}{3}$ in our proposed scheme as the fixed-rate convolutional encoder.

In our proposed scheme depicted in Figure 2.1, the fixed-rate convolutional encoder at the first stage is to provide the rate selector with all the inputs needed to generate the variable-rate convolutional codes. The rate selector utilizes all or some of its inputs to build the corresponding codes based on the channel conditions. Note that the output of the fixed-rate convolutional encoder at the first stage in Figure 2.1 is independent of the channel state information. On the other hand, the rate selector operates based on the knowledge of the channel state information. It is responsible for selecting the appropriate rate for a given channel condition. Then, at this particular rate, the rate selector generates the coded bits in accordance with the good convolutional codes. For illustration, let's consider a convolutional encoder with a constraint length $\nu = 3$. According to [41], the corresponding *generating matrices* of the good codes at different rates are given as follows.

For $\frac{1}{2}$ rate,

$$G_{\frac{1}{2}}(D) = \left[\; 1 + D^2 + D^3,\;\; 1 + D + D^2 + D^3 \;\right]. \tag{2.3}$$

For $\frac{1}{3}$ rate,

$$G_{\frac{1}{3}}(D) = \left[1 + D^2 + D^3,\;\; 1 + D + D^3,\;\; 1 + D + D^2 + D^3\right]. \tag{2.4}$$

For $\frac{1}{4}$ rate,

$$G_{\frac{1}{4}}(D) = \left[1 + D^2 + D^3, \;\; 1 + D^2 + D^3, \;\; 1 + D + D^3, \;\; 1 + D + D^2 + D^3\right]. \tag{2.5}$$

To generate the convolutional codes with different rates, the rate selector relies on three unique generating polynomials:

$$\begin{gathered} 1 + D + D^3, \\ 1 + D^2 + D^3, \\ 1 + D + D^2 + D^3. \end{gathered} \tag{2.6}$$

Hence the convolutional encoder needed in our proposed system may include only three binary adders for each generating polynomial given by Eq. (2.6), which can provide the rate selector with the required information.

Table 2.1: Candidates for Good Convolutional Codes

ν	Rate	Generating Polynomials				d_{free}
2	$\frac{1}{4}$	5	5	7	7	10
	$\frac{1}{3}$	5	7	7	-	8
	$\frac{1}{2}$	5	7	-	-	5
3	$\frac{1}{4}$	13	13	15	17	13
	$\frac{1}{3}$	13	15	17	-	10
	$\frac{1}{2}$	15	17	-	-	6
4	$\frac{1}{4}$	25	27	33	37	16
	$\frac{1}{3}$	25	33	37	-	12
	$\frac{1}{2}$	23	35	-	-	8

Possible candidates for good convolutional codes of various constraint lengths and rates are listed in Table 2.1 (refer to [41]). In this table, the connections from the shift register to each adder are expressed in octal form (see [41]). For example, if a generating polynomial is $1 + D + D^3$ with a constraint length $\nu = 3$, the first, second, and fourth bits will thus be

connected with the adder and then its octal representation is 15. From Table 2.1, one can observe that the number of adders (number of unique generating polynomials) required to facilitate a variable-rate channel encoder is 2, 3, and 5 for the constraint lengths 2, 3, and 4, respectively. It is clear from Table 2.1 that the free distance for a given constraint length decreases with the increasing rate. Therefore, if we consider a constraint length $\nu = 2$, the free distance d_{free} is 10, 8, and 5 for rates $\frac{1}{4}$, $\frac{1}{3}$, and $\frac{1}{2}$, respectively. Consequently, for our variable-rate convolutional encoder that switches its rate among the aforementioned different rates, we may have codewords that have a Hamming distance of 5. This implies that in the variable-rate codes, the free distance assumes the smallest value among the the free distances corresponding to different possible rates. However, when this smallest free distance occurs due to a good channel condition, the bit transition probability resulting from the underlying symmetric memoryless channel should also be small. Otherwise, our rate selector would have chosen a smaller-rate coding scheme with a higher free distance. Hence, even though there is some chance that the resultant free distance from the variable-rate coding would be smaller than that from the fixed-rate coding, the average BER of the former scheme would still be better than that of the latter scheme. To explain this phenomenon more clearly, we will investigate the *input output weight enumerating function* (*IOWEF*) of our proposed variable-rate channel encoder for an example.

2.2.4 Input Output Weight Enumerating Function

Figure 2.3 depicts the state diagram for the variable-rate encoder shown in Figure 2.1. In the state diagram, each state transition is labeled with a branch gain, which is a function of p, q, W^w, X^x, Y^y, and Z^z where p is the probability of selecting a rate $\frac{1}{2}$ encoder (which is

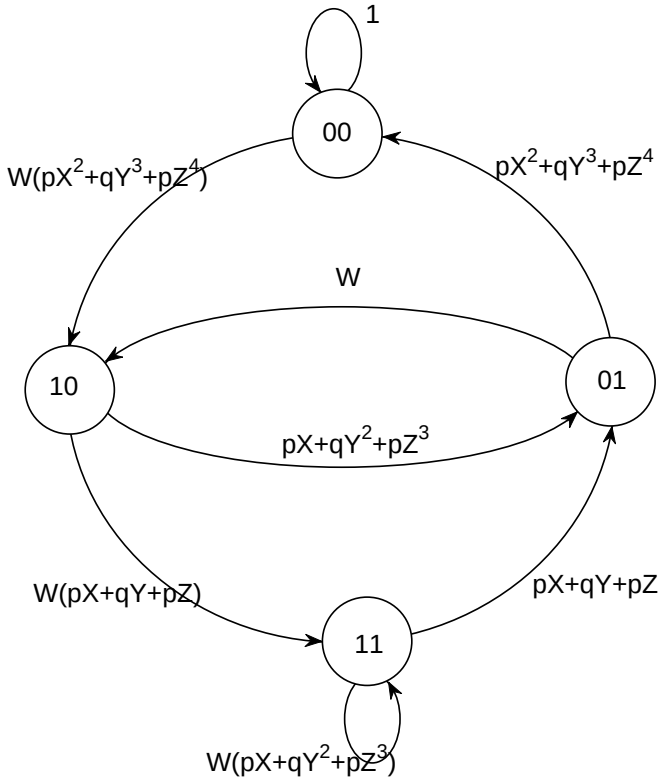

Figure 2.3: Finite state machine for a four-state variable-rate convolutional encoder.

equal to the probability of selecting a rate $\frac{1}{4}$ encoder), q is the probability of selecting a rate $\frac{1}{3}$ encoder, w is the *Hamming weight* of the input information bits, and x, y, and z are the Hamming weights of the encoded bits for the rates $\frac{1}{2}$, $\frac{1}{3}$, and $\frac{1}{4}$, respectively. Accordingly, if a branch gain is labeled as $W(pX^2+qY^3+pZ^4)$, it means that the input is a "1" and there is a chance of having the $\frac{1}{2}$-rate coded bits of weight 2 with probability p, or the $\frac{1}{3}$-rate coded bits of weight 3 with probability q, or the $\frac{1}{4}$-rate coded bits of weight 4 with probability p. Based on the state diagram in Figure 2.3, we can determine the IOWEF by measuring the gain of the paths leaving the state (00) and returning to it. These paths represent the possible *first event error* paths or the paths in the trellis that depart from the all zero event

and return to it for the first time. The IOWEF $A(W, X, Y, Z)$ is given by

$$A(W, X, Y, Z) = \sum_{z=0}^{\infty} \sum_{y=0}^{\infty} \sum_{x=0}^{\infty} a_{x,y,z} X^x Y^y Z^z, \tag{2.7}$$

where $a_{x,y,z}$ is a function of p, q, and W.

To evaluate our proposed variable-rate convolutional encoder in terms of BER, we need to consider the *bit weight enumerating function* $B(X, Y, Z)$ such that

$$\begin{aligned} B(X, Y, Z) &= \left. \frac{\partial A(W, X, Y, Z)}{\partial W} \right|_{W=1} \\ &= \sum_{z=0}^{\infty} \sum_{y=0}^{\infty} \sum_{x=0}^{\infty} \left. \frac{\partial a_{x,y,z}}{\partial W} \right|_{W=1} X^x Y^y Z^z \\ &= \sum_{z=0}^{\infty} \sum_{y=0}^{\infty} \sum_{x=0}^{\infty} \beta_{x,y,z} X^x Y^y Z^z, \end{aligned} \tag{2.8}$$

where

$$\beta_{x,y,z} \stackrel{\text{def}}{=} \left. \frac{\partial a_{x,y,z}}{\partial W} \right|_{W=1}.$$

The bit weight enumerating function for the finite state machine of a four-state variable-rate convolutional encoder depicted in Figure 2.3 is given by

$$\begin{aligned} B(X, Y, Z) &= \left. \frac{\partial A(W, X, Y, Z)}{\partial W} \right|_{W=1} \\ &= p^3 X^5 + 4p^4 X^6 + 4p^4 X^5 Z + 4p^3 q X^5 Y + 26p^4 q X^6 Y \\ &\quad + (2p^2 q^2 + p^2 q) X^4 Y^2 + (4p^3 q) X^4 Y Z + (26p^5) X^6 Z + 2p^4 X^4 Z^2 \\ &\quad + X^2 6p^4 q 5 Y Z + 13p^5 X^5 Z^2 + (13p^3 q^2 + 6p^3 q) X^5 Y^2 + 2p^2 q X^3 Y^3 \\ &\quad + (8p^2 q^2) X^3 Y^4 + (8p^3 q) X^3 Y^3 Z + (2p^3) X^3 Z^4 + p q^2 X Y^6 \\ &\quad + (4pq^3 + 2pq^2) X^2 Y^5 + (2p^2 q) X^2 Y^3 Z^3 + (4p^3 q) X^2 Y^3 Z^2 + \ldots \end{aligned}$$

For the fixed-rate convolutional encoder of rate $\frac{1}{3}$, constraint length $\nu = 2$, and generating

matrix $[1+D^2 \;\; 1+D^2 \;\; 1+D+D^2]$, its bit weight enumerating function is given by

$$B(Y) = 3Y^8 + 15Y^{10} + 58Y^{12} + 201Y^{14} + 655Y^{16} + 2052Y^{18} + \ldots.$$

Note that for fixed-rate convolutional codes the low-order terms of the bit weight enumerating function would dominate the system performance. This is a result from the fact that the exponent in each term represents the Hamming distance between codewords. For instance, the presence of X^5 implies that there are codewords with a Hamming distance 5 from our variable-rate convolutional coding scheme. Hence, an all-zero coded bit stream may be decoded incorrectly if there is a chance of erroneously converting three or more bits from "0" to "1" among the five given bit positions during transmission. According to [5], the error probability for decoding this bit stream incorrectly is given by

$$\mathcal{P}_{\text{decoded error, variable-rate}} \approx \sum_{i=3}^{5} \binom{5}{i} p_b^i (1-p_b)^{5-i}, \tag{2.9}$$

where p_b denotes the *bit transition probability* of the memoryless binary channel.

On the other hand, when we consider the case for a fixed-rate convolutional code of rate $\frac{1}{3}$, the smallest order among the terms of its bit weight enumerating function in Eq. (2.9) is 8, which is also the free distance. Hence, an all-zero coded bit stream may be decoded incorrectly if there is a chance of converting four or more bits from "0" to "1" among the eight given bit positions during transmission. The error probability for decoding this bit stream incorrectly is given by

$$\mathcal{P}_{\text{decoded error, fixed-rate}} \approx \frac{1}{2}\binom{8}{4} p_b^4 (1-p_b)^4 + \sum_{i=5}^{8} \binom{8}{i} p_b^i (1-p_b)^{8-i}. \tag{2.10}$$

Comparing Eq. (2.9) with Eq. (2.10), one may suggest that our proposed variable-rate encoder leads to a worse BER performance than the conventional fixed-rate encoder subject

to the same p_b. However, the values of p_b in Eqs. (2.9) and (2.10) are different. The overall advantage of our proposed scheme lies on the fact that if an X appears in Eq. (2.9), then it means that the rate selector is obliged to use rate $\frac{1}{2}$ and the bit transition probability p_b should be small accordingly.

Consequently, we have to determine the different bit transition probabilities p_b for different rates. Let p_X, p_Y, and p_Z represent the bit transition probabilities when a $\frac{1}{2}$-rate, $\frac{1}{3}$-rate and $\frac{1}{4}$-rate encoders are employed, respectively. These bit transition probabilities are characterized by both the channel property and the probabilities $\mathbb{P}_{\frac{1}{2}}$, $\mathbb{P}_{\frac{1}{3}}$, $\mathbb{P}_{\frac{1}{4}}$ for our proposed rate selector choosing rates $\frac{1}{2}$, $\frac{1}{3}$, and $\frac{1}{4}$, respectively. Consider $\mathbb{P}_{\frac{1}{2}} = \mathbb{P}_{\frac{1}{4}} = p$ and $\mathbb{P}_{\frac{1}{3}} = q$. The underlying channel is a wireless fading channel. Under such a channel, the received signal envelope α is assumed to have a Rayleigh distribution $f_A(\alpha)$ such that

$$f_A(\alpha) = \begin{cases} \left(\frac{\alpha}{\sigma^2}\right) e^{-\frac{\alpha^2}{2\sigma^2}}, & \text{if } 0 \leq \alpha \leq \infty \\ 0, & \text{if } \alpha \leq 0 \end{cases} \tag{2.11}$$

where $2\sigma^2$ is the time-average power of the received signal prior to the envelope detection [24]. For a wireless channel with instantaneous gain α, the instantaneous probability of error is given by $p_e(\alpha)$. This error probability is a function of channel gain α and varies with the modulation scheme in use. The average error probability incurred by the channel under the condition that a specific rate $R(\alpha) = \varrho$ is being used is given by

$$P_\varrho\{\text{error}\} = \frac{3\,\varrho}{\mathbb{P}_\varrho} \int_{A_\varrho} p_e(\alpha) \left(\frac{\alpha}{\sigma^2}\right) e^{-\frac{\alpha^2}{2\sigma^2}}\, d\alpha,$$

where $A_\varrho = \{\alpha \,|\, \text{rate is selected to be } \varrho\}$, and $\varrho = \frac{1}{2}$, $\frac{1}{3}$, or $\frac{1}{4}$. The region A_ϱ can be represented as a half-open half-closed interval $[\alpha_{\min}(\varrho), \alpha_{\max}(\varrho))$ where $\alpha_{\min}(\varrho)$, $\alpha_{\max}(\varrho)$ are the

corresponding lower- and upper-bounds. It yields

$$P_{\varrho}\{\text{error}\} = \frac{\int_{\alpha_{\min}(\varrho)}^{\alpha_{\max}(\varrho)} p_e(\alpha) \left(\frac{\alpha}{\sigma^2}\right) e^{-\frac{\alpha^2}{2\sigma^2}} d\alpha}{\int_{\alpha_{\min}(\varrho)}^{\alpha_{\max}(\varrho)} \left(\frac{\alpha}{\sigma^2}\right) e^{-\frac{\alpha^2}{2\sigma^2}} d\alpha}. \tag{2.12}$$

Specifically, for a *binary phase shift keying* (BPSK) modulation system over a flat fading channel, the instantaneous error probability is given by

$$p_e(\alpha) = Q\left(\sqrt{\frac{2\alpha^2 E_b}{N_o}}\right), \tag{2.13}$$

where E_b is the average bit energy, $\frac{N_o}{2}$ is the *noise power spectral density*, and $Q(\rho) = \frac{1}{\sqrt{2\pi}} \int_{\rho}^{\infty} e^{-\frac{\rho^2}{2}} d\rho$. Consequently, the average error probability subject to the channel gain α and the rate ϱ is given by

$$\begin{aligned} P_{\varrho}\{\text{error}\} &= \frac{3\,\varrho}{\mathbb{P}_{\varrho}} \int_{\alpha_{\min}(\varrho)}^{\alpha_{\max}(\varrho)} Q\left(\sqrt{\frac{2\alpha^2 E_b}{N_o}}\right) \left(\frac{\alpha}{\sigma^2}\right) e^{-\frac{\alpha^2}{2\sigma^2}} d\alpha \\ &= \frac{3\,\varrho}{\mathbb{P}_{\varrho}} \int_{\alpha_{\min}^2(\varrho)\frac{E_b}{N_o}}^{\alpha_{\max}^2(\varrho)\frac{E_b}{N_o}} Q\left(\sqrt{2\gamma}\right) \frac{1}{\Gamma} e^{-\frac{\gamma}{\Gamma}} d\gamma \end{aligned} \tag{2.14}$$

where $\gamma \stackrel{\text{def}}{=} \frac{\alpha^2 E_b}{N_o}$ and $\Gamma \stackrel{\text{def}}{=} \frac{2\,\sigma^2 E_b}{N_o}$. Furthermore, Eq (2.14) can be rewritten as

$$\begin{aligned} P_{\varrho}\{\text{error}\} &= \left[\frac{3\,\varrho\, Q\left(\sqrt{2\alpha_{\min}^2(\varrho)\frac{E_b}{N_o}}\right) e^{-\frac{\alpha_{\min}^2(\varrho)}{2\sigma^2}} - 3\,\varrho\sqrt{\frac{\Gamma}{1+\Gamma}} Q\left(\sqrt{2\alpha_{\min}^2(\varrho)\frac{E_b}{N_o}\frac{1+\Gamma}{\Gamma}}\right)}{\mathbb{P}_{\varrho}}\right] \\ &\quad - \left[\frac{3\,\varrho\, Q\left(\sqrt{2\alpha_{\max}^2(\varrho)\frac{E_b}{N_o}}\right) e^{-\frac{\alpha_{\max}^2(\varrho)}{2\sigma^2}} - 3\,\varrho\sqrt{\frac{\Gamma}{1+\Gamma}} Q\left(\sqrt{2\alpha_{\max}^2(\varrho)\frac{E_b}{N_o}\frac{1+\Gamma}{\Gamma}}\right)}{\mathbb{P}_{\varrho}}\right]. \end{aligned} \tag{2.15}$$

Observe Eq. (2.15) and one can find that $P_{\varrho}\{\text{error}\}$ is actually a function of $\alpha_{\min}(\varrho)$ and $\alpha_{\max}(\varrho)$. Therefore, we can express

$$P_{\varrho}\{\text{error}\} = P_{\varrho}\left(\alpha_{\min}(\varrho), \alpha_{\max}(\varrho)\right). \tag{2.16}$$

Based on the fact that the probability of selecting the rate $\frac{1}{2}$, $\frac{1}{3}$, and $\frac{1}{4}$ encoders are p,

$q = 1 - 2p$, and p respectively and according to the probability density function of channel gain given by Eq. (2.11), we can determine the corresponding subintervals $[\alpha_{\min}(\varrho), \alpha_{\max}(\varrho))$ of channel gain to these rates. It yields

$$A_{\frac{1}{2}} = \left[\sqrt{2\sigma^2 \ln\left(\frac{3}{2p}\right)}\ ,\ \infty \right), \tag{2.17}$$

$$A_{\frac{1}{3}} = \left[\sqrt{2\sigma^2 \ln\left(\frac{3}{3-4p}\right)}\ ,\ \sqrt{2\sigma^2 \ln\left(\frac{3}{2p}\right)} \right), \tag{2.18}$$

$$A_{\frac{1}{4}} = \left[0\ ,\ \sqrt{2\sigma^2 \ln\left(\frac{3}{3-4p}\right)} \right). \tag{2.19}$$

According to Eqs. (2.15)-(2.19), the bit transition probabilities p_X, p_Y, and p_Z can be evaluated as follows:

$$\begin{aligned}
p_X &\stackrel{\text{def}}{=} P_{\frac{1}{2}}\left(\sqrt{2\sigma^2 \ln\left(\frac{3}{2p}\right)},\quad \infty\right) \\
&= \frac{3}{2p}\left\{ Q\left(\sqrt{4\sigma^2 \frac{E_b}{N_o} \ln\left(\frac{3}{2p}\right)}\right)\ e^{-\ln\left(\frac{3}{2p}\right)} - \sqrt{\frac{\Gamma}{1+\Gamma}}\, Q\left(\sqrt{4\sigma^2 \frac{E_b}{N_o} \ln\left(\frac{3}{2p}\right)\ \frac{1+\Gamma}{\Gamma}}\right)\right\} \\
&= \frac{3}{2p}\left\{ Q\left(\sqrt{2\Gamma \ln\left(\frac{3}{2p}\right)}\right)\ e^{-\ln\left(\frac{3}{2p}\right)} - \sqrt{\frac{\Gamma}{1+\Gamma}}\, Q\left(\sqrt{2(1+\Gamma) \ln\left(\frac{3}{2p}\right)}\right)\right\}, \qquad (2.20) \\
p_Y &\stackrel{\text{def}}{=} P_{\frac{1}{3}}\left(\sqrt{2\sigma^2 \ln\left(\frac{3}{3-4p}\right)},\quad \sqrt{2\sigma^2 \ln\left(\frac{3}{2p}\right)}\right) \\
&= \frac{1}{q}\left\{ Q\left(\sqrt{2\Gamma \ln\left(\frac{3}{3-4p}\right)}\right)\ e^{-\ln\left(\frac{3}{3-4p}\right)} - \sqrt{\frac{\Gamma}{1+\Gamma}}\, Q\left(\sqrt{2(1+\Gamma) \ln\left(\frac{3}{3-4p}\right)}\right)\right. \\
&\quad \left. - Q\left(\sqrt{2\Gamma \ln\left(\frac{3}{2p}\right)}\right)\ e^{-\ln\left(\frac{3}{2p}\right)} + \sqrt{\frac{\Gamma}{1+\Gamma}}\, Q\left(\sqrt{2(1+\Gamma) \ln\left(\frac{3}{2p}\right)}\right)\right\}, \qquad (2.21) \\
p_Z &\stackrel{\text{def}}{=} P_{\frac{1}{4}}\left(0,\quad \sqrt{2\sigma^2 \ln\left(\frac{3}{3-4p}\right)}\right) \\
&= \frac{3}{4p}\left\{ \frac{1}{2}\left(1 - \sqrt{\frac{\Gamma}{1+\Gamma}}\right) - Q\left(\sqrt{2\Gamma \ln\left(\frac{3}{3-4p}\right)}\right)\ e^{-\ln\left(\frac{3}{3-4p}\right)}\right. \\
&\quad \left. + \sqrt{\frac{\Gamma}{1+\Gamma}}\, Q\left(\sqrt{2(1+\Gamma) \ln\left(\frac{3}{3-4p}\right)}\right)\right\}. \qquad (2.22)
\end{aligned}$$

The bit transition probabilities p_X, p_Y, p_Z versus average signal-to-noise ratio (Γ) for the three different code rates and the different rate-selection probabilities p ($=\frac{1}{5},\ \frac{1}{3},\ \frac{7}{15}$) are depicted in Figure 2.4 for illustration. Given $p = \frac{1}{5}, \frac{1}{3}, \frac{7}{15}$, according to Eqs. (2.17)-(2.19), we can have the following corresponding subintervals. Note that since Γ depends on σ^2, we can normalize σ^2 as $\sigma^2 = 1/2$ and let Γ be an independent variable accordingly.

For $p = \frac{1}{5}$,

$$A_{\frac{1}{2}} = [\, 1.4195\ ,\ \infty\,)\,, \tag{2.23}$$

$$A_{\frac{1}{3}} = [\, 0.5569\ ,\ 1.4195\,)\,, \tag{2.24}$$

$$A_{\frac{1}{4}} = [\, 0\ ,\ 0.5569\,)\,. \tag{2.25}$$

For $p = \frac{1}{3}$,

$$A_{\frac{1}{2}} = [\, 1.2264\ ,\ \infty\,)\,, \tag{2.26}$$

$$A_{\frac{1}{3}} = [\, 0.7667\ ,\ 1.2264\,)\,, \tag{2.27}$$

$$A_{\frac{1}{4}} = [\, 0\ ,\ 0.7667\,)\,. \tag{2.28}$$

For $p = \frac{7}{15}$,

$$A_{\frac{1}{2}} = [\, 1.0806\ ,\ \infty\,)\,, \tag{2.29}$$

$$A_{\frac{1}{3}} = [\, 0.9866\ ,\ 1.0806\,)\,, \tag{2.30}$$

$$A_{\frac{1}{4}} = [\, 0\ ,\ 0.9866\,)\,. \tag{2.31}$$

According to Figure 2.4, one can observe the fact that the bit transition probability for the signals coded with the $\frac{1}{2}$-rate encoder is less than that coded with the $\frac{1}{4}$-encoder at least by an order of magnitude. The margin gets even much larger when Γ increases.

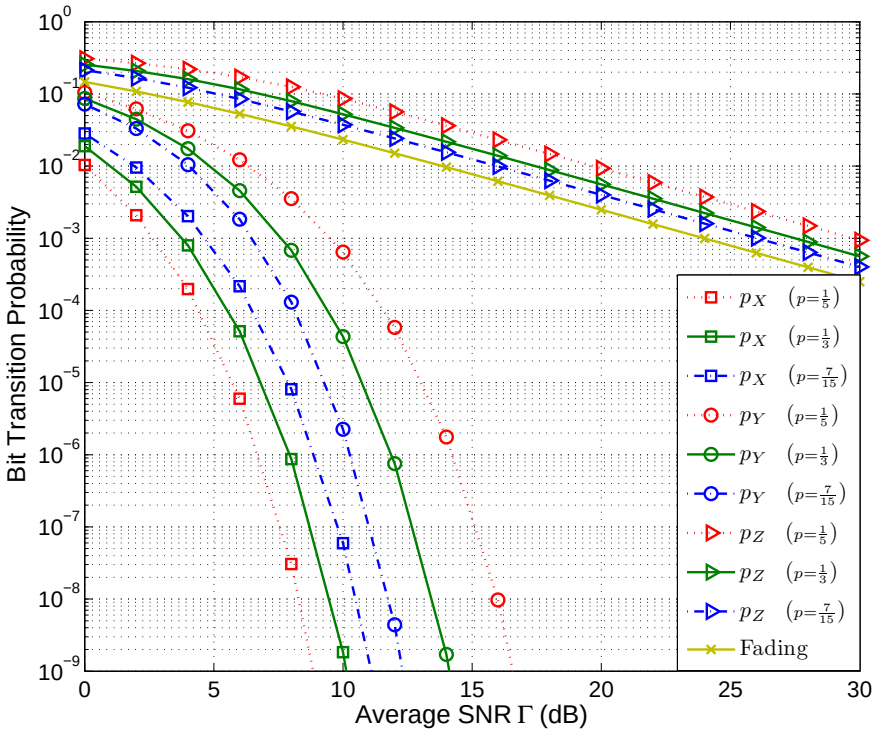

Figure 2.4: Bit transition probability versus average signal-to-noise ratio (Γ) for three different code rates and different rate-selection probabilities.

This fact supports the argument that even though the adoption of a $\frac{1}{2}$-rate encoder in our proposed variable-rate encoder would lead to a small Hamming distance, the corresponding bit transition probability is also very small on the other hand. Hence, in our proposed scheme, the ultimate symbol detection performance at the receiver would not be compromised when a high-rate encoder is selected for good channel conditions.

To study the decoder performance, we will analyze the bit error probability for the *maximum likelihood sequence detection*. The upper bound of the bit error probability for our proposed variable-rate convolutional encoder ($\mathcal{P}_{\text{decoded error, variable-rate}}$ as stated in Eq. (2.9))

can be computed from the aforementioned bit transition probabilities as

$$\mathcal{P}_{\text{decoded error, variable-rate}} \leq \sum_{z=0}^{\infty}\sum_{y=0}^{\infty}\sum_{x=0}^{\infty} \beta_{x,y,z}\, p_s(x,y,z), \tag{2.32}$$

where $p_s(x,y,z)$ is the *sequence error probability* of a variable-rate convolutional code associated with a *first event error path* which has a bit weight enumeration function given by $\beta_{x,y,z}X^xY^yZ^z$ [5]. A maximum likelihood detector picks this path over the all-zero path if the bit inversion has been incurred by the binary symmetric channel at a number greater than or equal to $\frac{x+y+z}{2}$ of bit positions. This sequence error probability is thus given by

$$\begin{aligned} p_s(x,y,z) &= \sum_{k=0}^{z}\sum_{j=0}^{y}\sum_{i=0}^{x} \mathcal{I}(i,j,k)\binom{x}{i}\binom{y}{j}\binom{z}{k} \\ &\qquad \times p_X^i(1-p_X)^{(x-i)}\, p_Y^j(1-p_Y)^{(y-j)}\, p_Z^k(1-p_Z)^{(z-k)}, \end{aligned} \tag{2.33}$$

where p_X, p_Y, and p_Z are the bit transition probabilities (see Eqs. (2.20), (2.21), (2.22)) in the memoryless channel when the codding rate is $\frac{1}{2}$, $\frac{1}{3}$, and $\frac{1}{4}$, respectively. The indicator function $\mathcal{I}(i,j,k)$ specifies the probability of an error event path being picked over the all-zero path for some particular combination of i, j, and k. When $(x+y+z)$ is odd, the indicator function $\mathcal{I}(i,j,k)$ is given by

$$\mathcal{I}(i,j,k) \stackrel{\text{def}}{=} \begin{cases} 1, & \text{if } i+j+k \geq \frac{x+y+z+1}{2} \\ 0, & \text{if } i+j+k < \frac{x+y+z+1}{2} \end{cases} \tag{2.34}$$

Otherwise, when $(x+y+z)$ is even, the indicator function $\mathcal{I}(i,j,k)$ is given by

$$\mathcal{I}(i,j,k) \stackrel{\text{def}}{=} \begin{cases} 1, & \text{if } i+j+k \geq \frac{x+y+z}{2}+1 \\ \frac{1}{2}, & \text{if } i+j+k = \frac{x+y+z}{2} \\ 0, & \text{if } i+j+k < \frac{x+y+z}{2} \end{cases} \tag{2.35}$$

In the next section, we will present an extensive numerical performance analysis to compare our proposed variable-rate convolutional encoder and the conventional fixed-rate convolutional encoder.

2.3 Experimental Results and Summary

To compare the fixed-rate and variable-rate convolutional encoders, we have carried out numerous Monte Carlo simulations over a slowly flat fading channel. The schemes in comparison include our proposed new method, the uncoded scheme (in the absence of any channel encoder), and the conventional fixed-rate channel coding method. The fading channel gain is generated from a quadrature uncorrelated zero-mean Gaussian random process with an identical variance σ^2 for both in-phase and quadrature components. The binary phase-shift keying (BPSK) scheme is adopted here to modulate the coded bits. The channel gain is considered to stay the same over a few consecutive symbols. We have used minimum mean square error (MMSE) channel estimation with various training sequence length (T_τ) and compared its BER performance with the performance under the assumption of perfect knowledge of CSI. At the receiver, the maximum likelihood decoder (Viterbi algorithm) is employed, and the associated *cost metric* is defined as the Hamming distance between codewords.

2.3.1 BER Performance Comparison for Variable-Rate and Fixed-Rate Coding Schemes

The BER performance is considered over the range of average signal-to-noise ratio (SNR) values from $\Gamma = 0$ to $\Gamma = 20$ dB with an increment of 2 dB. In each Monte Carlo trial for each average SNR condition, we randomly generate 300,000 uncoded bits. For both our proposed

variable-rate convolutional encoder and the conventional fixed-rate convolutional encoder, the constraint length is chosen to be 2 as indicated by Figure 2.1. For our proposed variable-rate convolutional coding scheme, we have undertaken simulations for different values of T_τ, which is the number of the training symbols for the channel gain estimation. The probability of selecting the $\frac{1}{2}$-rate convolutional encoder is set to be $p = \frac{1}{3}$. Note that the probability of selecting the $\frac{1}{3}$-rate and $\frac{1}{4}$-rate encoders are also $\frac{1}{3}$. The rate selector will select the good codes among the candidates with rates $\frac{1}{2}$, $\frac{1}{3}$, and $\frac{1}{4}$ given by Table 2.1. Then, the coded (or uncoded) bits are modulated by the BPSK scheme. In our transmission model, the modulated BPSK symbols are multiplied by the channel gain and added with the white Gaussian noise samples generated by the computer subject to the underlying SNR value. The underlying slow and flat fading channel is assumed to be constant over 16 or 18 symbols (i.e., the channel coherence time spans over 16 or 18 symbol periods), where the first 4 or 6 symbols are reserved for the minimum-mean-square-error (MMSE) channel estimation, respectively. The remaining 12 symbols are used to transmit the actual information symbols thereby. Twenty Monte Carlo trials (each involves 300,000 bits) are undertaken to evaluate the average performance.

In Figure 2.5, we plot the BER performances of the uncoded schemes under the perfect knowledge of CSI, the estimated CSI based on 6 training symbols, and the estimated CSI based on 4 training symbols (denoted by "Uncoded", "Uncoded MMSE ($T_\tau = 6$)", and "Uncoded MMSE ($T_\tau = 4$)"), respectively, together with the BER performances of the fixed-rate convolutional coding schemes under the perfect knowledge of CSI, the estimated CSI based on 6 training symbols, and the estimated CSI based on 4 training symbols (denoted by "Fixed Rate Coding", "Fixed Rate MMSE ($T_\tau = 6$)", and "Fixed Rate MMSE ($T_\tau = 4$)"),

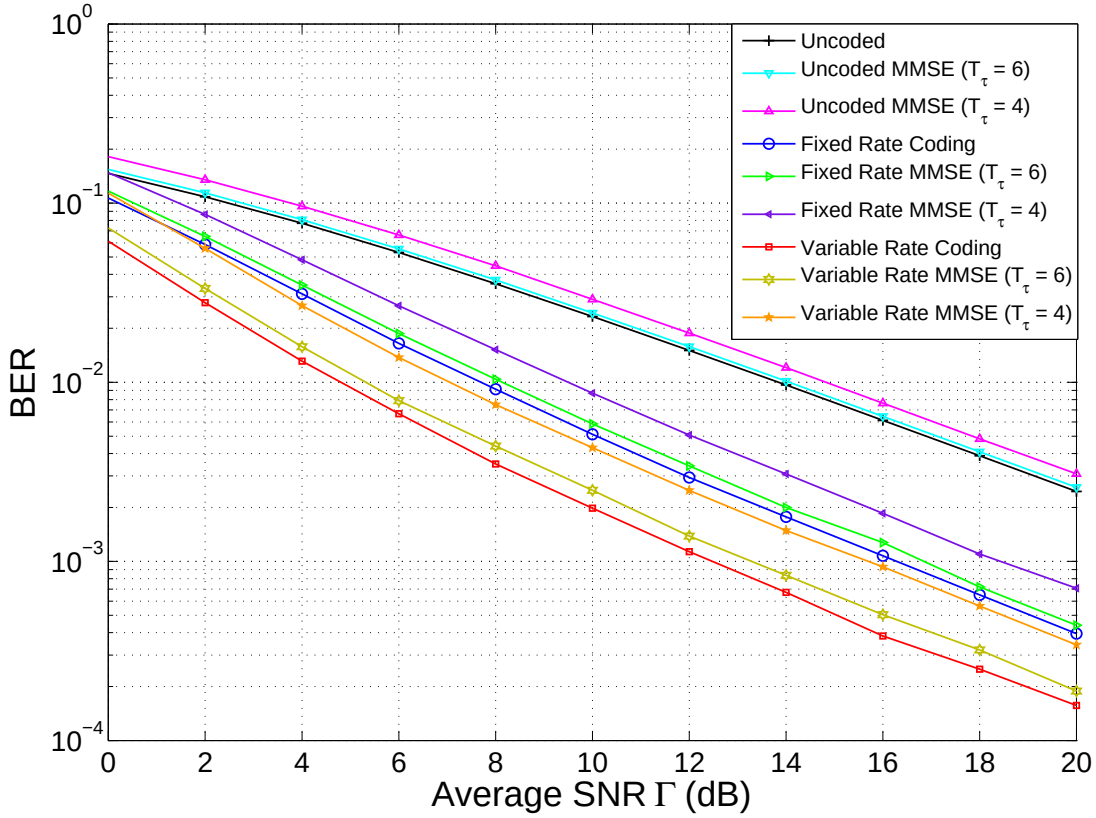

Figure 2.5: BER comparison versus average signal-to-noise ratio (Γ) for three different schemes.

respectively, as well as our proposed variable-rate convolutional coding scheme under the perfect knowledge of CSI, the estimated CSI based on 6 training symbols, and the estimated CSI based on 4 training symbols (denoted by "Variable Rate Coding", "Variable Rate MMSE ($T_\tau = 6$)", and "Variable Rate MMSE ($T_\tau = 4$)"), respectively. The rate selector will divide the channel conditions in terms of the estimated channel gain into three separate categories so as to facilitate the aforementioned rate-selection probabilities. Consequently, the average rate is equal to $\frac{1}{3}$. As one can see from Figure 2.5, our variable-rate convolutional encoder considerably outperforms the fixed-rate encoder with same average transmission rate across all SNR conditions. Our proposed new scheme can even register more than 3-dB margin

over the conventional fixed-rate encoding method.

The performance of our proposed decoder can be improved by incorporating the maximum likelihood detector with the estimated channel state information [42, 43]. For a given base-band received signal sequence $\zeta = \{\zeta_j\}$, the set of quantized channel state information $\mathcal{C} = \{c_j\}$, and the set of information sequence candidates $\mathcal{S} = \{S^{(l)}\}$ (as the output of the decoder) where $S^{(l)} \stackrel{\text{def}}{=} \left\{s_j^{(l)}\right\}$ and j is the symbol index, the maximum likelihood decision rule is given by

$$\max_l \sum_j \Lambda_j^{(l)}, \tag{2.36}$$

where

$$\Lambda_j^{(l)} \stackrel{\text{def}}{=} \log \mathcal{P}\left\{\zeta_j, c_j | s_j^{(l)}\right\}, \tag{2.37}$$

and $\mathcal{P}\left\{\zeta_j, c_j | s_j^{(l)}\right\}$ denotes the probability for the candidate symbol $s_j^{(l)}$ to cause ζ_j given the CSI c_j in the j^{th} symbol period. For our simulation, we adopt the BPSK modulation scheme and the channel state information is categorized into three regions (specified by Eqs. (2.17)-(2.19)) with bit transition probabilities p_X, p_Y and p_Z. On this circumstance, the metric $\Lambda_j^{(l)}$ in Eq. (2.36) can be redefined as

$$\Lambda_j^{(l)} \stackrel{\text{def}}{=} \begin{cases} \zeta_j \; s_j^{(l)} \log\left(\frac{1-p_X}{p_X}\right), & \text{if } \alpha \in A_{\frac{1}{2}} \\ \zeta_j \; s_j^{(l)} \log\left(\frac{1-p_Y}{p_Y}\right), & \text{if } \alpha \in A_{\frac{1}{3}} \\ \zeta_j \; s_j^{(l)} \log\left(\frac{1-p_Z}{p_Z}\right), & \text{if } \alpha \in A_{\frac{1}{4}} \end{cases} \tag{2.38}$$

In Figure 2.6, we plot the BER performances of the uncoded scheme (denoted by "Uncoded Theoretical BER"), the fixed-rate convolutional coding scheme based on the Hamming distance metric, and the fixed-rate convolutional coding scheme based on the metric given by

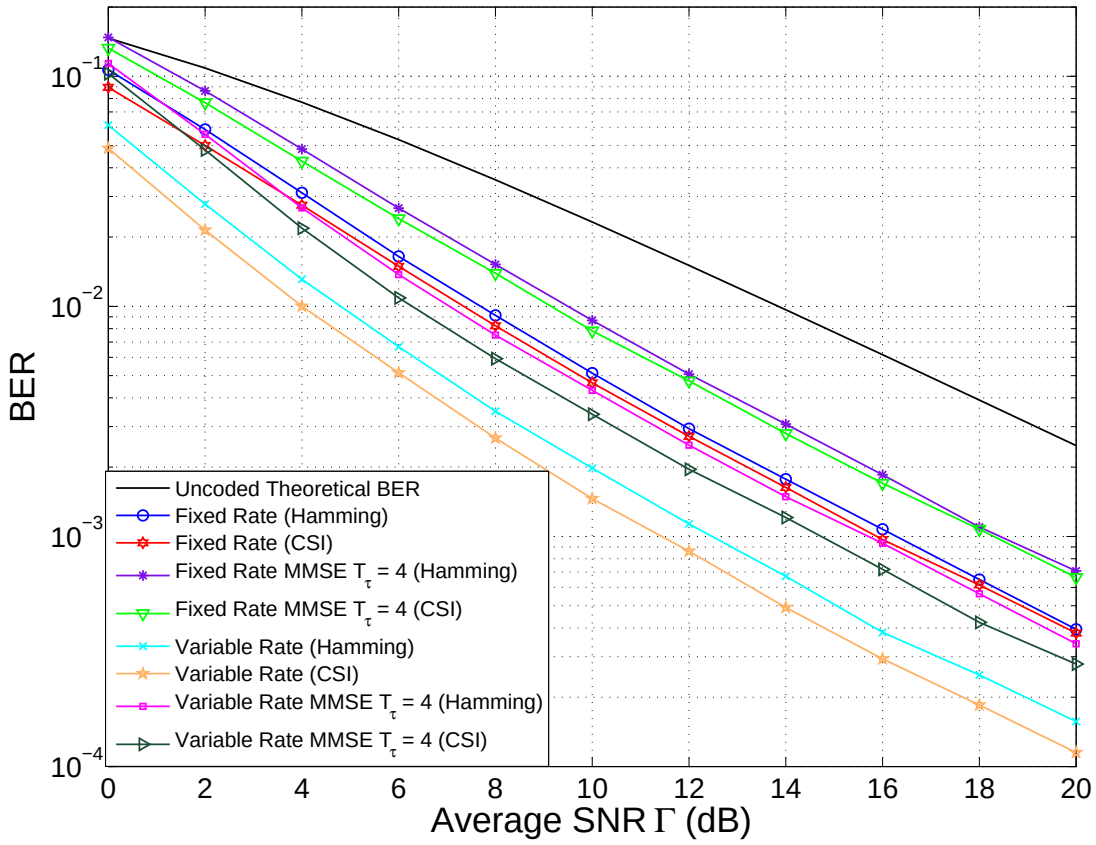

Figure 2.6: BER comparison versus average signal-to-noise ratio (Γ) for the fixed-rate convolutional coding scheme and the variable-rate convolutional coding scheme incorporated with two different decoding methods.

Eq. (2.38) with the perfect knowledge of CSI (denoted by "Fixed Rate (Hamming)" and "Fixed Rate (CSI)", respectively). In addition, we also plot the BER performances of the fixed-rate convolutional coding schemes based on the Hamming distance metric with the estimated CSI using four training symbols and the metric given by Eq. (2.38) with the estimated CSI using four training symbols (denoted by "Fixed Rate MMSE $T_\tau = 4$ (Hamming)" and "Fixed Rate MMSE $T_\tau = 4$ (CSI)", respectively). And our proposed variable-rate convolutional coding scheme when used with Hamming distance metric and the metric provided in Eq (2.38) with perfect knowledge of CSI (denoted by "Variable Rate (Hamming)" and

"Variable Rate (CSI)" respectively) and with approximated CSI with training length of four (denoted by "Variable Rate MMSE $T_\tau = 4$ (Hamming)" and "Variable Rate MMSE $T_\tau = 4$ (CSI)" respectively). For the proposed variable-rate convolutional coding scheme, we have run simulations for $p = \frac{1}{3}$. Since it is not practical to compute the values of p_X, p_Y, and p_Z in Eq. (2.38) for each Γ, we simplify the decoder by just picking a set of fixed numerical values. We suggest to set p_X, p_Y, and p_Z (according to Figure 2.4 for $p = \frac{1}{3}$ and $\Gamma = 4$ dB) to be 7.9640^{-4}, 0.0174, and 0.1601, respectively. From the simulation results, it is evident that our proposed method performs better than the conventional fixed-rate coding scheme subject to both metrics.

2.3.2 BER Performance Comparison for Different Coded-Modulation Schemes

We have also carried out Monte Carlo simulations to measure the performance of the *adaptive trellis coded modulation* (TCM) and the *non-adaptive TCM*. In the adaptive coded-modulation scheme, the binary encoder is a variable-rate convolutional encoder. We use the 16-QAM signal constellation, where $m = \log_2 16 = 4$. The input to the convolutional encoder is a single bit ($k = 1$). The encoder generates coded bits with redundancy $r(\alpha)$= 1, 2, or 3 subject to the channel condition. Then the number of uncoded bits, namely $m - k - r(\alpha)$, will be 2, 1 or 0, respectively. For the variable-rate convolutional encoder, the constraint length is chosen to be 3. The octal representations of the generating polynomials depend on the redundancy; when the redundancy is 1, they are 15 and 17; when the redundancy is 2, they are 13, 15, and 17; when the redundancy is 3; they are 11, 13, 15, and 17. It is clear that the total number of information bits per channel use is 2 for this adaptive coded-modulation scheme. Thus, we will compare this adaptive scheme to a non-adaptive TCM scheme with

$k = 1$, $r = 2$ where the number of uncoded bits is set to 1. The underlying slow and flat fading channel is assumed to be constant over 18 symbols, where the first 6 symbols are used for the MMSE channel estimation. The remaining 12 symbols are used to transmit the actual information data. Ten Monte Carlo trials are undertaken, and 300,000 16-QAM symbols are randomly generated by the computer for each trial.

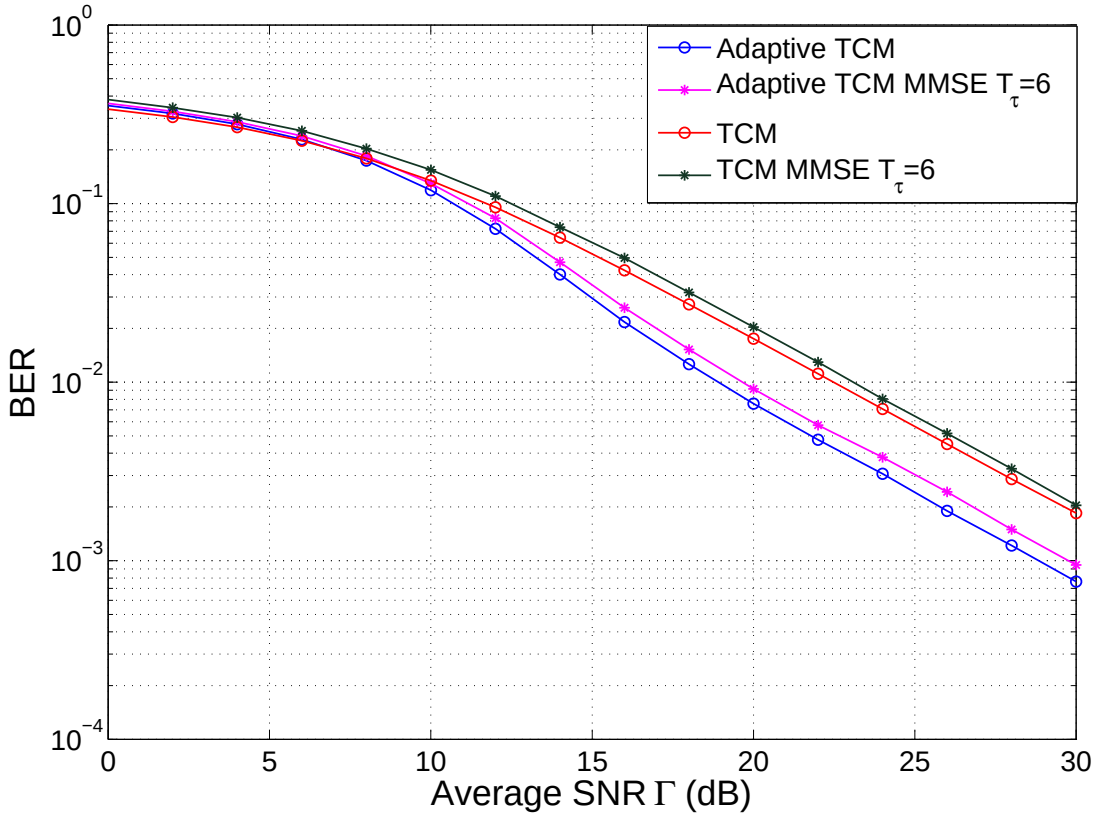

Figure 2.7: BER comparison versus average signal-to-noise ratio (Γ) for adaptive and non-adaptive trellis coded-modulation schemes.

In Figure 2.7, we plot the BER performances of our proposed adaptive trellis coded-modulation schemes with the perfect CSI knowledge and the estimated CSI (denoted by "Adaptive TCM" and "Adaptive TCM MMSE ($T_\tau = 6$)", respectively). Moreover, we also depict the BER performances of the non-adaptive trellis coded-modulation schemes with

the perfect CSI knowledge and the estimated CSI (denoted by "TCM" and "TCM MMSE ($T_\tau = 6$)", respectively), in the same figure. According to Figure 2.7, one can observe that our proposed adaptive TCM scheme demonstrates a remarkable advantage over the existing non-adaptive TCM scheme. Our proposed scheme can lead to a margin of more than 4-dB over the existing non-adaptive TCM method at high SNR values.

To sum up, wireless communications over fading channels still remain fairly challenging to researchers. To address this crucial problem, we proposed a novel variable-rate channel coding strategy that exploits the channel state information available at the transmitter via feedback. From Monte Carlo simulation results, our proposed new transceiver greatly outperforms the conventional system using the fixed-rate convolutional encoder at the same transmission data rate. The performance gain of our proposed scheme over the conventional method can reach up to 3 dB. Our proposed new variable-rate convolutional coding scheme can serve as the core part of the next generation adaptive coded-modulation transceivers.

3. MODIFIED TURBO DECODER

In this chapter, we propose and investigate a novel approach of removing the global content from the received signal. Moreover we will design a modified turbo decoder to decode the local content. Our new global content removal technique would relieve the receiver from performing coding and modulation on the detected global content so as to ensure less computational burden and a smaller latency than the current technology. To mitigate the bit-error-rate performance degradation on the local content, we propose to modify the turbo decoder so that the systematic bit streams and the parity bit streams will be decoded differently with respect to their corresponding signal-to-noise ratios.

3.1 Literature Review

Mobile television services, carried by cellular and satellite networks, are becoming more and more popular due to users' convenience. These services are expected to expand widely in the near future and dominate the mass-media markets. A fully-established standard in this area is the *Digital Video Broadcasting-Satellite Services to Handheld Devices (DVB-SH) Standard* [1]. DVB-SH systems broadcast video signals to the target receivers over a hybrid satellite and terrestrial single-frequency network [1]. The broadcasted service is thus provided to handheld terminals (devices). These systems operate at the central frequencies below 3 GHz. This frequency range is allocated to help deploy the TV service in

a less congested spectrum of the S-band. In a DVB-SH system, coverage improvement is achieved by combining a *satellite component* (SC) and a *complementary ground component* (CGC) [25]. The SC, transmitted via a satellite, covers a wide geographical area whereas the CGC, transmitted via terrestrial transmitters, leads to a cellular-type coverage. A coverage comparison between satellite TV service and hybrid satellite-terrestrial service can refer to [44]. This comparative study was based on field measurements and simulations. According to the results demonstrated in [44], for a particular combination of modulation, forward-error-correction, and interleaving schemes, the DVB-SH system operating via a sole satellite can provide up to a coverage of 90% of the entire area. On the other hand, the DVB-SH system operating in the *hybrid mode* using a satellite and a terrestrial network can provide a broader coverage of more than 99% of the entire area.

Henceforth, users would demand the DVB-SH systems operating in the hybrid mode due to its superior coverage in the future. The localized nature of the CGC broadcasting allows the terrestrial broadcasters to insert *local content* (LC) in addition to the *global content* (GC) which is transmitted by both the satellite and terrestrial transmitters. The LC insertion is performed using *hierarchical modulation* [45].

The hierarchical modulation for DVB systems has been investigated by many researchers. The tradeoff between the rate increment due to the addition of local component and the corresponding bit-error-rate (BER) degradation was investigated by [45], where a method was proposed to embed an LP (low-priority) data stream for OFDM-based wireless communication systems. In addition, a new receiver design was introduced where the two decoders for LC and GC could be incorporated [45]. A recent study on hierarchical modulation for DVB-SH systems in [46] analyzed the system performances for both GC and LC under the

assumption that the noise, experienced by the global content, appeared to be additive white Gaussian noise (AWGN). The feasibility of terrestrial digital video broadcast to mobile receivers employing hierarchical modulation and turbo coding was also studied in [47]. In [48], the use of Mobile WiMAX for the terrestrial local content transmission was suggested. The insertion of local content into the CGC has also been investigated in [49]. A concatenated encoding scheme mixed the two bit streams in order to make the encoding of the LP stream depend on the well-protected *high-priority* (HP) stream [49]. Besides, the system performance in terms of spectrum efficiency was analyzed when hierarchical modulation combined with error correction codes were employed in the DVB-SH systems [50].

In the aforementioned studies on DVB-SH systems, the received signal was treated as a noisy GC stream and hence demodulation and decoding were performed to detect the transmitted GC stream. Then, to detect the local content, the most widely used approach was to encode and modulate the estimated GC stream and to remove it from the received signal for acquiring the noisy LC. Ultimately, the transmitted LC stream can be estimated from the noisy LC. However, this existing method tends to have large latency since additional encoding and modulating operations on the estimated GC bit stream are required at the receiver.

In this dissertation, we propose a new scheme to remove the global content from the received signal. This is accomplished by generating a hybrid stream composed of two components. The first component is an output of a turbo decoder that estimates the global content and the second component is a stream consisting of directly detected parity bits of the global content. Our approach relieves the receiver from the tedious task of encoding and modulation, and hence the corresponding computational burden and latency are reduced.

Furthermore, to alleviate severe bit-error-rate performance degradation of the local content, we modify the conventional turbo decoder so that it can be used for local content decoding.

3.2 Hierarchical Modulation in DVB-SH

The basic DVB-SH transmitter includes turbo encoder, bit interleavers, symbol interleaver, and a modulator which employs hierarchical modulation and OFDM modulation. The global content, which is carried by an HP data stream, and the local content, which is carried by an LP data stream, are encoded separately (by the turbo encoder) as specified by [1]. To combat burst errors, the encoded bits are interleaved using a bit block interleaver and a convolutional time interleaver [1]. Then, at a terrestrial transmitter, these two streams are combined together and modulated. During the modulation process, the bit streams are mapped onto the non-uniform constellation points as shown in Figure 3.1. Then, the modulated symbols are interleaved and converted to the time domain by IFFT (inverse Fast Fourier transform) in the OFDM modulator.

Let the transmitted global content and local content streams be denoted by $\mathbf{u}_G$ and $\mathbf{u}_L$, respectively. These two bit streams are then encoded using turbo encoder specified in the DVB-SH standard [1]. Through a $\frac{1}{3}$-rate encoder, the global content data stream will engender the systematic bit stream $\mathbf{u}_G$, the parity bit stream from the first encoder, say $\mathbf{v}_{G_1}$, and the parity bit stream from the second encoder, say $\mathbf{v}_{G_2}$, and then these three resultant streams will be multiplexed together. Similarly for the local content, one can obtain a multiplexed stream from $\mathbf{u}_L$, $\mathbf{v}_{L_1}$, and $\mathbf{v}_{L_2}$. We will modulate the two aforementioned multiplexed streams independently, that is, we do not use Gray coding here. The modulated

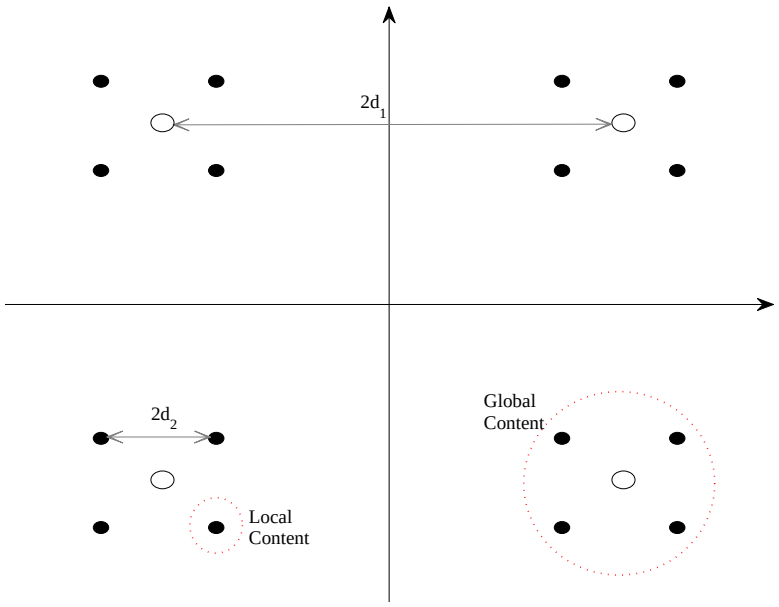

Figure 3.1: A typical constellation for hierarchical modulation ($2d_1$ denotes the minimum "intercluster" Euclidean distance for modulating HP symbols while $2d_2$ denotes the minimum "intracluster" Euclidean distance for modulating LP symbols).

QPSK (quadrature phase-shift keying) symbol sequences for global content and local content are denoted by $\mathbf{s}_G$ and $\mathbf{s}_L$, respectively. Accordingly, the complementary ground component from the terrestrial transmitter, say $\mathbf{s}_T$, and the global content from the satellite, say $\mathbf{s}_S$, are expressed by

$$\mathbf{s}_T = \mathbf{s}_G + \left(\frac{1}{1+\varrho}\right)\mathbf{s}_L, \tag{3.1}$$

$$\mathbf{s}_S = \mathbf{s}_G, \tag{3.2}$$

where the parameter ϱ is the minimum distance between two constellation points carrying different HP-bit values divided by the minimum distance between any two constellation points [1]. For the constellation given by Figure 3.1 we have $\varrho = \frac{d_1 - d_2}{d_2}$.

In the satellite and terrestrial transmitters, pilots are inserted into the information symbols and OFDM symbols are transmitted. The details of these procedures can be found

in [1, 25].

When the signal travels through the communication channel and arrives at the receiver, the receiver demodulates and decodes the received signal. The schematic of our proposed new DVB-SH receiver is depicted in Figure 3.2. The received signal is passed through FFT and the OFDM-demodulated signal, $\mathbf{y}$, can be written as

$$\begin{aligned} \mathbf{y} &= \mathbf{s}_T + \eta\,\mathbf{s}_S + \mathbf{n} && (3.3)\\ &= (1+\eta)\,\mathbf{s}_G + \left(\frac{1}{1+\varrho}\right)\mathbf{s}_L + \mathbf{n}, && (3.4) \end{aligned}$$

where η is the *channel gain* of the satellite signal and $\mathbf{n}$ is the AWGN. At the receiver, we execute the channel estimation which involves the estimation of ϱ. The channel estimation technique in [46] is adopted here. The *channel gain compensation* subsystem illustrated in Figure 3.2 produces two symbol streams as given by

$$\begin{aligned} \mathbf{y}_G &= \frac{\mathbf{y}}{(1+\eta)} && (3.5)\\ &= \mathbf{s}_G + \frac{1}{(1+\varrho)(1+\eta)}\mathbf{s}_L + \frac{\mathbf{n}}{(1+\eta)}, && (3.6)\\ \mathbf{y}_L &= (1+\varrho)\mathbf{y} && (3.7)\\ &= \mathbf{s}_L + (1+\varrho)(1+\eta)\mathbf{s}_G + (1+\varrho)\mathbf{n}. && (3.8) \end{aligned}$$

Next, the "demodulator", which actually does not carry out symbol detection, will convert the complex-valued received noisy symbol streams $\mathbf{y}_G$, $\mathbf{y}_L$ to the respective real-valued streams $\mathbf{x}_G$ and $\mathbf{x}_L$, each of which consists of the corresponding real components followed by the imaginary components. As shown by [46], the noisy stream $\mathbf{x}_G$ can be approximated by a standard AWGN channel model with the signal-to-noise ratio given by

$$\mathrm{SNR}_{\mathrm{global}} = \frac{(1+\eta)^2(1+\varrho)^2}{1+\mathrm{CNR}+(1+\eta)^2(1+\varrho)^2}\mathrm{CNR}, \qquad (3.9)$$

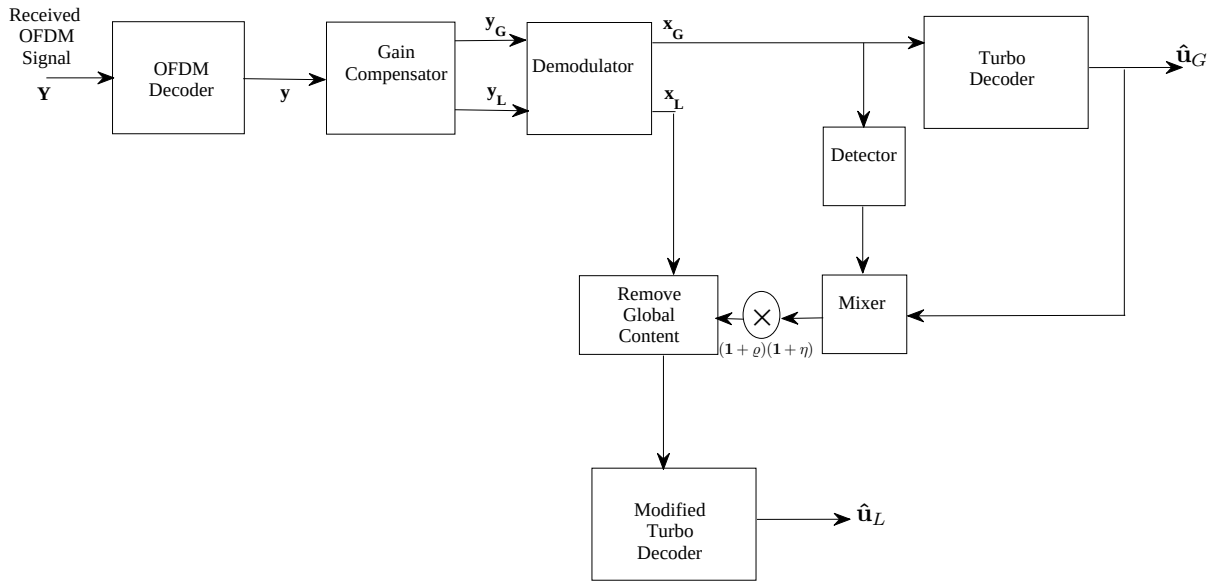

Figure 3.2: Our proposed new DVB-SH receiver consisting of the novel computationally-efficient global content removal scheme.

where CNR is the *carrier-to-noise ratio*, defined as the ratio of total signal power to noise power at the receiver.

For the local content, the signal-to-noise ratio can be given by

$$\mathrm{SNR}_{\mathrm{local}} = \frac{\mathrm{CNR}}{1 + (1+\eta)^2(1+\varrho)^2}, \tag{3.10}$$

when the global content is removed correctly. Feeding $\mathbf{x}_G$ to the standard turbo decoder provided by [19] one can estimate the global content as $\hat{\mathbf{u}}_G$. Then, we have to estimate the local content. This is where our main contribution lies. We propose a new approach to estimate the local content for achieving a better computational-efficiency and a shorter processing delay. Decreasing processing delay in decoding the local content is of crucial importance, since the use of the turbo decoder by itself has already incurred processing delay due to its need to buffer the received bit stream before decoding.

The straightforward method to decode the local content would be to encode and modulate

the decoded global content first, multiply this encoded signal by $(1+\varrho)(1+\eta)$, and then subtract it from $\mathbf{y}_L$. This approach would lead to a better performance for small values of ϱ and small SNR, but the associated encoding and modulation operations incur extra computational complexity and processing delay. Alternatively, one can perform the symbol detection on $\mathbf{y}_G$, which just determines the signs of its real and imaginary components, multiply the resultant symbol estimate sequence by $(1+\varrho)\,(1+\eta)$, and then subtract it from $\mathbf{y}_L$. The latter method can greatly mitigate the computational burden and thus significantly reduce the processing delay; however, this method has a drawback in terms of poor error performance for decoding the local content. To combat the aforementioned drawback, we propose a new method here to improve the BER performance while still maintaining the computational efficiency. For a $\frac{1}{3}$-rate turbo encoder in the DVB-SH standard, the encoded bit stream in the transmitter contains the systematic bits and two parity bit streams. To remove the interfering global content from $\mathbf{x}_L$, we first build a hybrid approximation of global content, namely $\mathbf{x}_H$. The stream $\mathbf{x}_H$ is generated by utilizing the decoder output and the detector output as depicted by Figure 3.2. This particular process is performed by the "*mixer*" and its operation is stated in Algorithm 1, where N is the number of bits in the output stream from the turbo decoder.

Algorithm 1 Hybrid stream generation

for $i = 1$ to N **do**
  $x_H(3(i-1)+1) = \hat{u}_G(i)$
  $x_H(3(i-1)+2) = \text{sign}(x_G(3(i-1)+2))$
  $x_H(3(i-1)+3) = \text{sign}(x_G(3(i-1)+3))$
end for

Then, we may remove the global content to generate $\mathbf{x}'_L$. It is

$$\mathbf{x}'_L = \mathbf{x}_L - (1+\varrho)(1+\eta)\mathbf{x}_H. \tag{3.11}$$

Decoding $\mathbf{x}'_L$ using the standard turbo decoder would not lead to the best BER performance. This is due to the fact that the two different components in the hybrid stream have different signal-to-noise ratios. The part of the hybrid stream coming from $\hat{\mathbf{u}}_G$ has better signal-to-noise ratio. To improve the ultimate BER performance, we design a modified turbo decoder in the next section.

3.3 The Modified Decoder

The principle of turbo coding is provided in [19]. The MAP (maximum *a posteriori* probability) decoder estimates the transmitted bits based on the logarithm of the *a posteriori* probability (LAPP) ratio. For our proposed modified turbo decoder, the noisy local content signal $\mathbf{x}'_L$ should be preprocessed, that is, we should categorize the systematic and parity bits as in Algorithm 2,where $\mathbf{x}^s_L$ is the systematic component, and $\mathbf{x}^{p1}_L$, $\mathbf{x}^{p2}_L$ are the parity components.

Algorithm 2 Separating systematic and parity bits

for $i = 1$ to N **do**
  $x^s_L(i) = x'_L(3(i-1)+1)$
  $x^{p1}_L(i) = x'_L(3(i-1)+2)$
  $x^{p2}_L(i) = x'_L(3(i-1)+3)$
end for

The decoded local content bit stream $\hat{\mathbf{u}}_L$ is detected using the MAP algorithm in [19]

such that

$$\hat{u}_L(k) = \begin{cases} +1, & \text{if } P\{u_L(k) = +1 | \mathbf{x}'_L\} > P\{u_L(k) = -1 | \mathbf{x}'_L\}, \\ -1, & \text{otherwise.} \end{cases} \tag{3.12}$$

Alternatively, one may use the LAPP ratio for detection. It yields

$$\hat{u}_L(k) = \text{sign}\{\text{L}\,(u_L(k))\}, \tag{3.13}$$

where $\text{L}\,(u_L(k)) \stackrel{\text{def}}{=} \log\left(\frac{P\{u_L(k)=+1|\mathbf{x}'_L\}}{P\{u_L(k)=-1|\mathbf{x}'_L\}}\right)$. For a given turbo encoder, $\text{L}\,(u_L(k))$ can be expressed as

$$\text{L}\,(u_L(k)) = \log\left(\frac{\sum_{S^+} P(s_{k-1}=p, s_k=q, \mathbf{x}'_L)}{\sum_{S^-} P(s_{k-1}=p, s_k=q, \mathbf{x}'_L)}\right), \tag{3.14}$$

where s_k specifies the state variable at time index k, $S^+ \stackrel{\text{def}}{=} \{(p,q) \mid s_{k-1}=p,\ s_k=q,\ u_k=+1\}$, and $S^- \stackrel{\text{def}}{=} \{(p,q) \mid s_{k-1}=p,\ s_k=q,\ u_k=-1\}$.

A well-known approach to solve Eq. (3.14) is the *BCJR algorithm* [51]. Since the hybrid signal $\mathbf{x}'_L$ which we use as the input to the turbo decoder is not the same as the classical received signal used as the input to the conventional turbo decoder, we have to modify the turbo decoder accordingly. To exhibit our modifications, we describe the *iterative BCJR decoding approach* (see [19]) as follows.

$$P(s_{k-1}=p,\ s_k=q,\ \mathbf{x}'_L) = \alpha_{k-1}(p)\ \gamma_k(p,q)\ \beta_k(q), \tag{3.15}$$

where

$$\alpha_{k-1}(p) \stackrel{\text{def}}{=} P(s_{k-1}=p,\ x'_L(1),\ \ldots,\ x'_L(k-1)), \tag{3.16}$$

$$\gamma_k(p,q) \stackrel{\text{def}}{=} P(s_k=q,\ x'_L(k) \mid s_{k-1}=p), \tag{3.17}$$

$$\beta_k(q) \stackrel{\text{def}}{=} P(x'_L(k),\ \ldots, x'_L(N) \mid s_k=q). \tag{3.18}$$

According to Eqs. (3.14) and (3.15), we have to calculate $\gamma_k(p,q)$, $\alpha_{k-1}(p)$, and $\beta_k(q)$ iteratively in the following sequential order (see [52]):

$$\begin{aligned}\gamma_k(p,q) &= P(x'_L(k)|u_k)\, P(u_k) \\ &= A_k e^{\left(\frac{1}{2}u_L(k)L^e(u_L(k))\right)} P(x'_L(k)|u_L(k)),\end{aligned} \tag{3.19}$$

where A_k is a constant which does not depend on $u_L(k)$ and $L^e(u_L(k)) \stackrel{\text{def}}{=} \log \frac{P(u_L(k)=+1)}{P(u_L(k)=-1)}$,

$$\alpha_k(q) = \sum_p \alpha_{k-1}(p)\, \gamma_k(p,q), \tag{3.20}$$

where $\alpha_0(s) = 1$ if $s = 0$ and $\alpha_0(s) = 0$ otherwise, and

$$\beta_{k-1}(p) = \sum_q \beta_k(q)\, \gamma_k(p,q), \tag{3.21}$$

where for a terminated encoder, $\beta_N(s) = 1$ if $s = 0$ and $\beta_N(s) = 0$ otherwise.

In our modified turbo decoder, the term $P(x'_L(k)|u_L(k))$ in Eq. (3.19) should be treated differently from the way it is in the conventional turbo decoder. This is due to the difference in the signal-to-noise ratios of the systematic stream and the parity streams. Hence, in our proposed method, we will estimate the individual signal-to-nose ratios for the systematic and parity streams in $\mathbf{x}'_L$. Therefore, for the modified decoder, $P(x'_L(k)|u_L(k))$ is formulated as

$$\begin{aligned}P(x'_L(k)|u_L(k)) &\propto \exp\left(-\frac{(x^s_L(k)-u_L(k))^2}{2\sigma_s^2} - \frac{(x^{pJ}_L(k)-v_{L_J}(k))^2}{2\sigma_p^2}\right) \\ &= B_k \exp\left(\frac{x^s_L(k)u_L(k)}{\sigma_s^2} + \frac{x^{pJ}_L(k)v_{L_J}(k)}{\sigma_p^2}\right),\end{aligned} \tag{3.22}$$

where $\frac{1}{\sigma_s^2}$ and $\frac{1}{\sigma_p^2}$ denote the estimated signal-to-noise ratios for the systematic and parity streams, respectively, B_k is a constant which does not depend on $u_L(k)$, and J takes values

1 or 2. According to Eqs. (3.19) and (3.22), we get

$$\gamma_k(p,q) \;=\; A_k B_k \exp\left(\frac{1}{2}u_L(k)L^e(u_L(k))\right)\exp\left(\frac{x_L^s(k)u_L(k)}{\sigma_s^2}+\frac{x_L^{pJ}(k)v_{L_J}(k)}{\sigma_p^2}\right). \quad (3.23)$$

Once the value of $\gamma_k(p,q)$ is calculated using our proposed new formula given by Eq. (3.23), it can be used later in the iterative decoding process.

3.4 Results and Summary

In order to measure the performance of our proposed scheme for the removal of the global content from the received signal and to observe the performance improvement due to our designed modified turbo decoder, we have carried out Monte Carlo simulations. We conducted the experiment for parameter values $\varrho = 1$, 1.25, 1.5, 1.75, and 2. The channel gain for the satellite component, $\eta = 1$, is considered. Besides, we assume that the CNR to be the ratio of the signal power (after the CGC and the SC are superimposed) to the noise power. The frame structure complies with the DVB-SH standard in [1]. Similarly, the encoder adopts the generating function $G(D)$ specified by [1], which is

$$G(D) = \left[1 \;\; \frac{1+D+D^3}{1+D^2+D^3} \;\; \frac{1+D+D^2+D^3}{1+D^2+D^3}\right]. \quad (3.24)$$

We implement the $\frac{1}{3}$-rate encoder with the *standard pattern*. At the turbo decoder, we have set the number of iterations to 5 for decoding the global content. For the local content, we vary iteration numbers for comparison. Figure 3.3 depicts the BER performances for decoding the local content only, as it is the only thing that is impacted by our proposed new scheme. The BER measure for decoding the global content is orders-of-magnitude smaller than that for decoding the local content especially at low CNRs. The notations in Figure 3.3

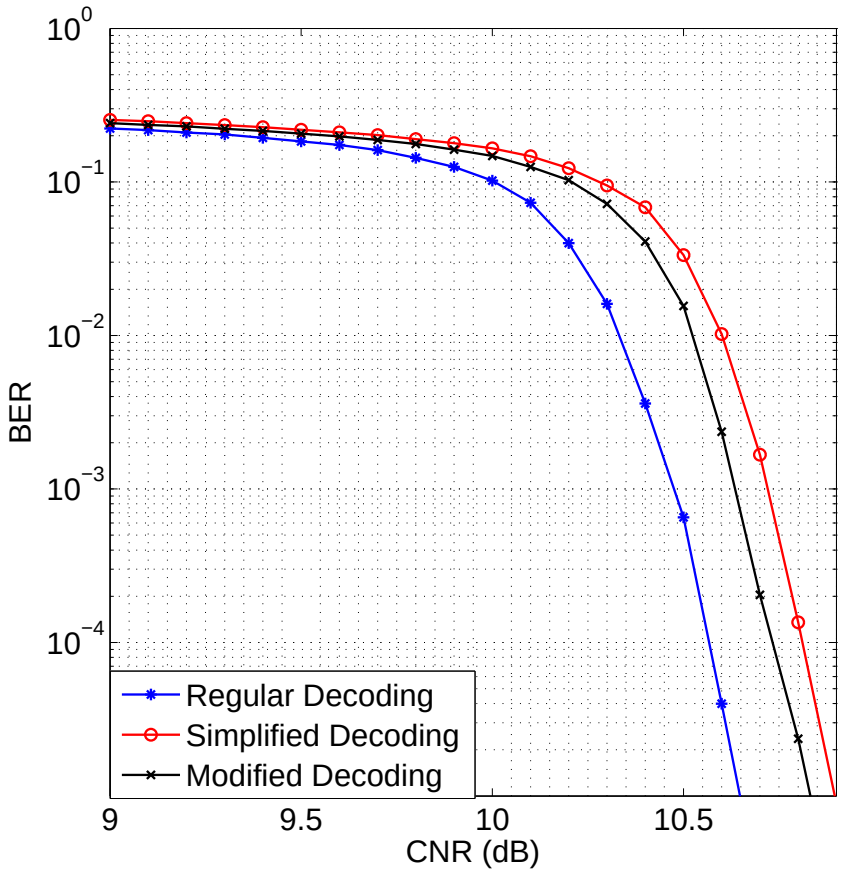

Figure 3.3: BER performances of the local content for different decoding schemes and $\varrho = 1$.

are clarified as follows: (i) "Regular Decoding": we encode and modulate the decoded global content prior to its removal from the received signal, (ii) "Simplified Decoding": we simply undertake the symbol detection process prior to the removal of the global content, (iii) "Modified Decoding": we generate a hybrid stream by mixing the decoded systematic bits and the symbol estimates of parity bits, remove the global content, and then send the resultant sequence to our proposed modified decoder.

According to Figure 3.3, for $\varrho = 1$, there is a performance degradation as the tradeoff of improvement of the computational efficiency and latency from "Regular Decoding" to

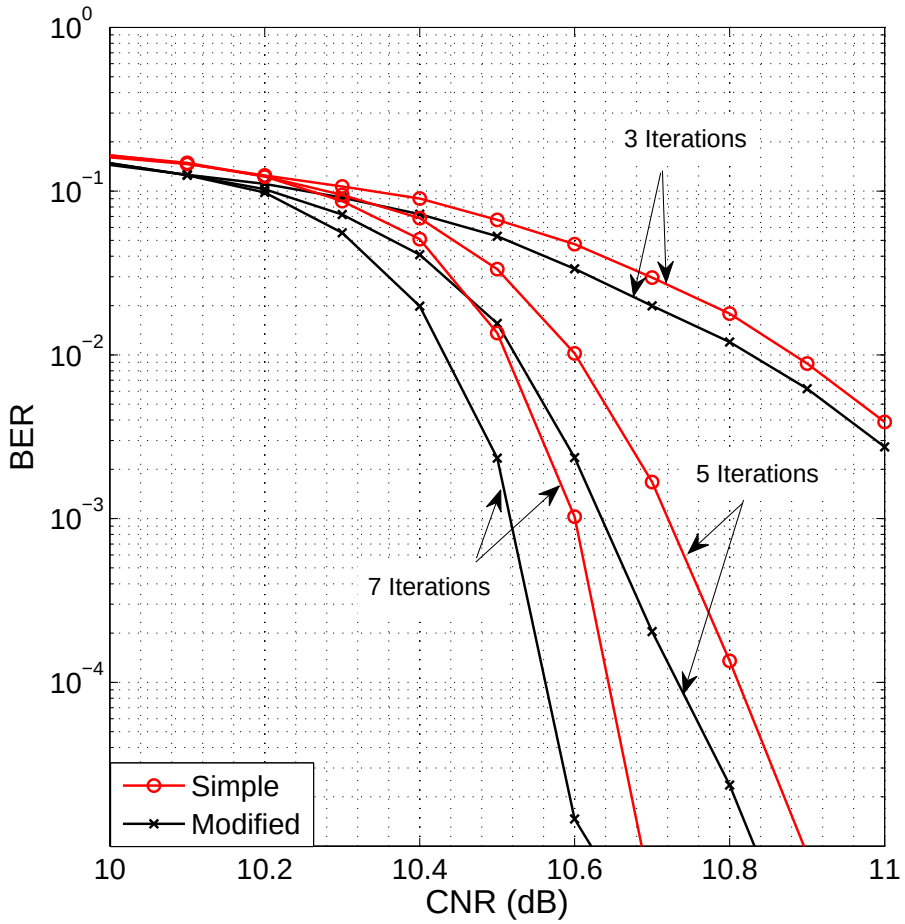

Figure 3.4: BER performances of the local content for different decoding schemes over various numbers of iterations and $\varrho = 1$.

"Simplified Decoding". However, our proposed modified decoding scheme can compensate some loss in BER performance. The BER improvement is related to the number of iterations taken in our proposed modified turbo decoder and this phenomenon is reflected by Figure 3.4. If the value of the parameter ϱ gets larger, the performance of our proposed scheme approaches that of the regular (conventional) decoding scheme. From the results demonstrated in Figure 3.5, it is evident that for $\varrho > 1.5$, our method performs as well as the regular decoding scheme. To understand this phenomenon, let's examine the required

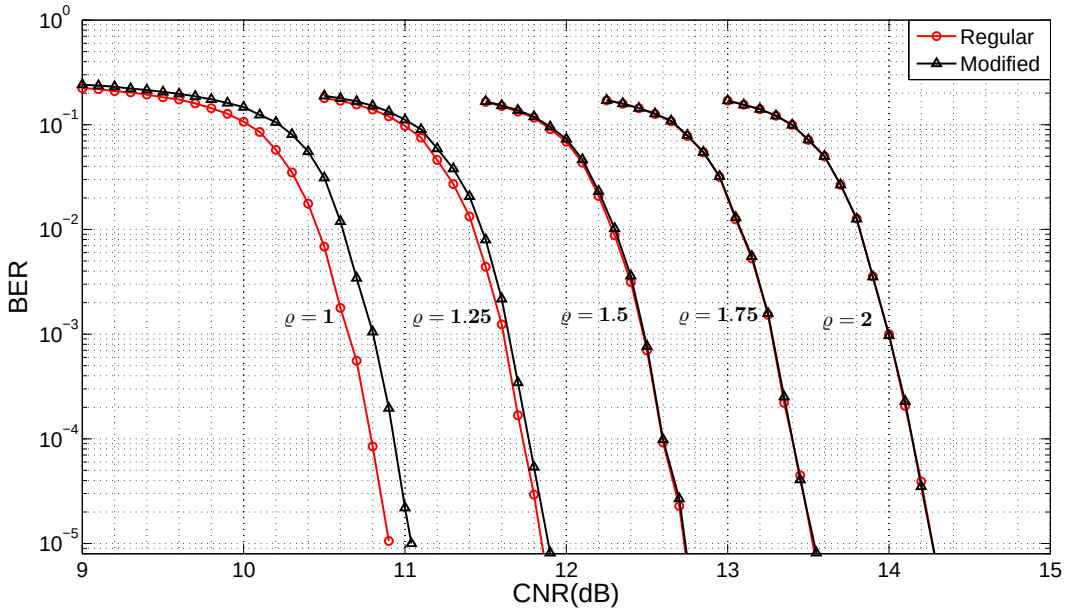

Figure 3.5: BER performances of the local content for different values of ϱ.

SNR for the local content to achieve the BER of 10^{-5}. According to Eq. (3.10), we get

$$\mathrm{SNR}_{\mathrm{local}} = \frac{\mathrm{CNR}}{1 + (1+\eta)^2(1+\varrho)^2}. \tag{3.25}$$

For $\eta = 1$ and $\varrho = 1$, assume that the BER of 10^{-5} is achieved when $\mathrm{CNR} = \mathrm{CNR}_1$. Therefore, the corresponding $\mathrm{SNR}_{\mathrm{local}}$ is $\frac{\mathrm{CNR}_1}{17}$. In order to achieve the same BER performance for $\varrho \neq 1$, $\mathrm{SNR}_{\mathrm{local}}$ should be kept constant at $\frac{\mathrm{CNR}_1}{17}$. Accordingly, the value of CNR should be increased by $10\log_{10}\frac{1+4(1+\varrho)^2}{17}$ dB. Therefore, the increments in CNR with respect to CNR_1 for $\varrho = 1.25$, 1.5, 1.75, and 2 are equal to 0.9691, 1.8452, 2.6440, and 3.3775 dB, respectively. This can be verified by the simulation results in Figure 3.5. In summary, as we increase the parameter ϱ, two effects can be perceived. First, the minimum distance between constellation points carrying different HP bits increases and leads to the better detection of the global content. Second, the higher requirement of CNR boosts $\mathrm{SNR}_{\mathrm{global}}$, which also leads to the better detection of the global content. These two consequences of higher ϱ

values manifest that our proposed new scheme approaches the same BER performance of the regular turbo decoding scheme for $\varrho > 1.5$.

To summarize, in this chapter we investigate a new approach to reduce the complexity for removing the global content from the received DVB-SH signal. In addition to that we have designed a new turbo decoder to improve error-rate performance of the local content. Based on the experimental results, our proposed new decoding technique can be a promising solution to the next generation DVB-SH receivers.

4. BLIND TURBO ENCODER ESTIMATION

In this chapter a novel blind parameter-estimation method which identifies a turbo encoder is discussed. The blind estimator is designed based on an iterative expectation-maximization algorithm. To implement this blind estimation scheme, we transform the recursive systematic convolutional (RSC) encoder into a non-systematic convolutional encoder preceded by a feedback encoder. The effect of the separate feedback encoder on the state sequence of the forward convolutional encoder will be studied. And, the effectiveness of our proposed new scheme will be evaluated using Monte Carlo simulation results.

4.1 Literature Review

Recently, blind signal processing has been used more and more widely for telecommunication applications [53–57]. Nevertheless, only few *blind encoder estimation* techniques have been presented up to now. Besides, there hardly exists any blind encoder estimation technique for the *turbo-coding* method. Therefore, we would like to propose a novel blind scheme to reconstruct the turbo-encoder at the receiver without the knowledge of the transmitted signal and in the absence of training symbols.

Blind parametric estimation of an encoder has been explored very lately by some researchers, though still at the primitive stage. In [58], the syndrome former of the convolutional encoder was estimated based on the syndrome former properties. This algorithm,

however, required an exhaustive search over all possible codes. If this algorithm is used for a fixed constraint length, its complexity will be exponential with respect to the constraint length. A different approach was proposed by [59], which presented the generalization of the key-equation and adopted the Euclidean Algorithm for the blind identification of the encoder parameters. In [60], an iterative method to identify the parameters and to reconstruct the generator matrix of a rate $\frac{k}{n}$ convolutional encoder was proposed. This iterative method could be the solution to the blind decoding problem for the encoders using dual codes. Generally speaking, all the aforementioned approaches were based on the algebraic properties of convolutional encoders and the associated dual code.

On the other hand, the *expectation-maximization* (EM) approach has been used for blind decoding lately. In [61], an iterative probabilistic method based on the EM algorithm to estimate the parameters of a convolutional encoder was proposed. This scheme was based on the *log-likelihood algebra* and the concept of *soft bits*. The pioneering idea of blind estimation related to encoders using expectation maximization can be retrieved back to the work in [62,63], which can estimate the coefficients of a linear-feedback binary shift-register.

Since the first category of blind decoding approaches (based on the encoders' algebraic properties) often encounter high-complexity and difficult-to-solve drawbacks, we would like to focus on the EM approach instead. In addition, because the turbo-coding schemes have been widely adopted in prevalent communication systems, in this chapter, we would like to propose a new blind decoding technique for turbo encoders. In our proposed method, the reconstruction of a turbo encoder is based on the EM algorithm. Our main contributions include the development of a new scheme to separate the feedback portion from the forward path in the recursive systematic convolutional encoder and the innovative model of the

decomposed encoder as a series concatenation of a "pure" feedback convolutional encoder and a forward non-systematic convolutional encoder. We then derive *emission probabilities* and *state probabilities* prior to the employment of the gradient ascent procedure for expectation maximization. Thus, the encoder parameters can be estimated for the further decoding of the received signal sequences.

4.2 Encoder Estimation Problem Formulation

In this section, the problem for blindly identifying the turbo encoder parameters from the noisy received signal will be introduced. Once these parameters are blindly estimated, one can easily reconstruct the encoder without the knowledge of the source information bits and in the absence of the training symbols. In this study, it is assumed that the constraint length of the underlying recursive convolutional encoder and the data rate are known at the receiver. Therefore, the sole objective of our work is to estimate the parameters associated with the connections in the feedback portion and those associated with the connections in the forward path. In addition, we cannot use the portion of the received signal that arises from the interleaved bits (see the lower group of the encoded bits in Figure 4.1). For illustration, let's consider a turbo encoder built upon a rate $\frac{1}{n}$ recursive convolutional encoder, which has the forward generating polynomials $G_1(D)$, $G_2(D)$, ..., $G_n(D)$, and the feedback polynomial $G_0(D)$ (See Figure 4.1). If we ignore the recursive convolutional encoder connected with the interleaver (the lower convolutional encoder in Figure 4.1), the coded bits can be represented by a vector $\mathbf{V}(D)$, each of whose elements is a polynomial over a finite field as

$$\mathbf{V}(D) = U(D)\left[1 \;\; \frac{G_1(D)}{G_0(D)} \;\; \frac{G_2(D)}{G_0(D)} \;\; \cdots \;\; \frac{G_n(D)}{G_0(D)}\right], \tag{4.1}$$

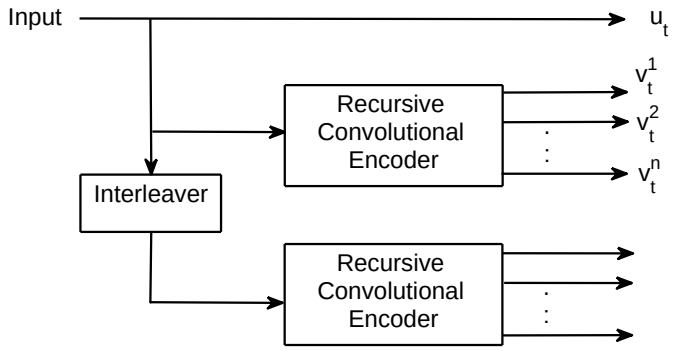

Figure 4.1: A typical example of a turbo encoder built upon a rate $\frac{1}{n}$ recursive convolutional encoder.

where $U(D)$ is a polynomial over a finite field and represents an input information sequence. To enable the blind encoder identification, we separate the feedback portion from the convolutional encoder. It yields

$$\begin{aligned} \mathbf{V}(D) &= \frac{U(D)}{G_0(D)} \left[G_0(D) \;\; G_1(D) \;\; G_2(D) \;\; \cdots \;\; G_n(D) \right] \\ &= W(D) \left[G_0(D) \;\; G_1(D) \;\; G_2(D) \;\; \ldots \;\; G_n(D) \right], \end{aligned} \tag{4.2}$$

where, $W(D) \stackrel{\text{def}}{=} \frac{U(D)}{G_0(D)}$. According to Eq. (4.2), we can treat the encoder parameter estimation problem in a different way. That is, we need to estimate the parameters associated with a forward convolutional encoder which receives the input bits from the output of a feedback convolutional encoder. Henceforth, our modified problem here is very different from the problem defined in [61], because the input information bits do not come directly from a source that generates independent and identically distributed (i.i.d.) random samples. Instead, the source information bits should first pass through a feedback convolutional encoder $\frac{1}{G_0(D)}$ in our problem. For illustration, the turbo encoder model and the details of the recursive convolutional encoder (adopted from the 3GPP2 specifications with omission of the termination switch) are presented in Figures 4.1 and 4.2. For the encoder depicted

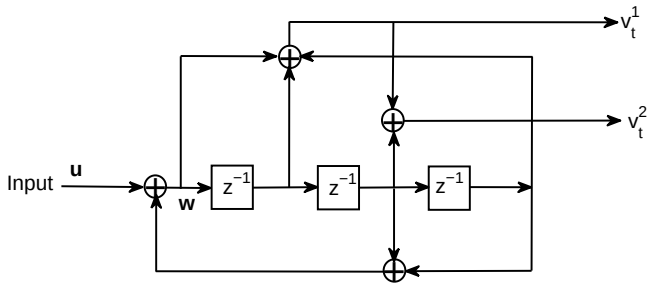

Figure 4.2: A recursive convolutional encoder adopted by the 3GPP2 specifications (see [1], page 15). Note that the *termination switch* is omitted here.

in Figure 4.2, we have $k = 1$, $n = 2$, $G_0(D) = 1 + D^2 + D^3$, $G_1(D) = 1 + D + D^3$, and $G_2(D) = 1 + D + D^2 + D^3$.

After the information bits are encoded, modulated, and transmitted, they will get through an additive white Gaussian noise (AWGN) channel. At the receiver, the received symbol vector at time t is denoted by $\mathbf{y}_t \stackrel{\text{def}}{=} [y_t^0 \; y_t^1 \; \cdots \; y_t^n]^T$, which is

$$\mathbf{y}_t = \mathbf{v}_t + \mathbf{n}_t \tag{4.3}$$

$$= \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_n \end{bmatrix} \begin{bmatrix} w_t \\ w_{t-1} \\ \vdots \\ w_{t-M} \end{bmatrix} + \mathbf{n}_t, \tag{4.4}$$

where, $\mathbf{g}_i \stackrel{\text{def}}{=} [g_i^0 \; g_i^1 \; \cdots \; g_i^M]$ specifies the connections of the i^{th} shift-register (M is the length of the shift-register) and $\mathbf{n}_t$ is a noise vector, each of whose components is a zero-mean i.i.d. Gaussian process with a variance of σ^2. Denote $\mathbf{g} \stackrel{\text{def}}{=} [\mathbf{g}_0^T \; \mathbf{g}_1^T \; \cdots \; \mathbf{g}_n^T]^T$. One may collect the instantaneous received signal vectors $\mathbf{y}_t$ for a period of time to construct the "long" received signal vector $\mathbf{y} \stackrel{\text{def}}{=} \left[\mathbf{y}_t^T\right]_{t=0,1,2,\ldots,\tau}^T$.

Our objective is to determine the optimal matrix $\mathbf{g}^*$ such that the probability of $\mathbf{y}$

conditioned on $\mathbf{g}$, namely $P(\mathbf{y}|\mathbf{g})$, is maximized. It yields

$$\mathbf{g}^* = \arg\max_{\mathbf{g}} P(\mathbf{y}|\mathbf{g}). \tag{4.5}$$

This is actually the maximum likelihood (ML) estimate of $\mathbf{g}$. Since the coefficients of the register connections, g_i^j, can only take values of either 1 or 0, gradient search cannot be used for this maximum-likelihood problem. Alternatively, we propose to employ the common practice of characterizing the binary values by a *sigmoid* function q_i^j, which can model the probability of the parameter g_i^j. Following [62], we define $q_i^j \stackrel{\text{def}}{=} P(g_i^j = 1)$ as

$$q_i^j \stackrel{\text{def}}{=} \frac{1}{1+e^{-\theta_i^j}}, \qquad -\infty < \theta_i^j < \infty. \tag{4.6}$$

For notational convenience, we write $\mathbf{q}_i \stackrel{\text{def}}{=} [q_i^0 \; q_i^1 \; \cdots \; q_i^M]$ and $\mathbf{q} \stackrel{\text{def}}{=} [\mathbf{q}_0^T \; \mathbf{q}_1^T \; \cdots \; \mathbf{q}_n^T]^T$.

4.3 Parameter Estimation Algorithm

In this section, we will present our new blind encoder parameter estimation method based on the EM algorithm. To use the gradient search at the maximization step, let's assume that the current estimate of $\mathbf{q}$ is given by $\mathbf{q}[k]$ (say at iteration k). We will update this estimate as $\mathbf{q}[k+1]$ at the next iteration (iteration $k+1$), such that $\log P(\mathbf{y}|\mathbf{q}[k+1]) > \log P(\mathbf{y}|\mathbf{q}[k])$, and the difference $\log P(\mathbf{y}|\mathbf{q}[k+1]) - \log P(\mathbf{y}|\mathbf{q}[k])$ is maximized. It is equivalent to the maximization of $\log P(\mathbf{y}|\mathbf{q}[k+1])$ with respect to $\mathbf{q}[k+1]$, since $\log P(\mathbf{y}|\mathbf{q}[k])$ is independent of $\mathbf{q}[k+1]$.

Now, let's include the unobserved data $\mathbf{w}$, where $\mathbf{w}$ is the output sequence of the separated feedback encoder (see Figure 4.2) and it is represented by the coefficients of the polynomial $W(D) = \frac{U(D)}{G_0(D)}$. Thus, we may express the log-likelihood function with respect to $\mathbf{q}$ based

on the current estimate $\mathbf{q}[k]$, which is given by

$$\log P(\mathbf{y}|\mathbf{q}) = \sum_{\mathbf{w}} P\left(\mathbf{w}|\mathbf{y}, \mathbf{q}\left[k\right]\right) \log \frac{P\left(\mathbf{y}, \mathbf{w}|\mathbf{q}\right)}{P\left(\mathbf{w}|\mathbf{y}, \mathbf{q}\left[k\right]\right)}. \tag{4.7}$$

According to (4.7), we can therefore determine the new ML estimate of $\mathbf{q}$ as

$$\begin{aligned} \mathbf{q}[k+1] &= \arg\max_{\mathbf{q}} \sum_{\mathbf{w}} P\left(\mathbf{w}|\mathbf{y}, \mathbf{q}\left[k\right]\right) \log \frac{P\left(\mathbf{y}, \mathbf{w}|\mathbf{q}\right)}{P\left(\mathbf{w}|\mathbf{y}, \mathbf{q}\left[k\right]\right)} \\ &= \arg\max_{\mathbf{q}} \sum_{\mathbf{w}} P\left(\mathbf{w}|\mathbf{y}, \mathbf{q}\left[k\right]\right) \log P\left(\mathbf{y}, \mathbf{w}|\mathbf{q}\right). \end{aligned} \tag{4.8}$$

Eq. (4.8) manifests the two major steps of the EM algorithm. In the expectation step, the expression $\mathcal{L}(\mathbf{q}) \stackrel{\text{def}}{=} \sum_{\mathbf{w}} P\left(\mathbf{w}|\mathbf{y}, \mathbf{g}\left[k\right]\right) \log P\left(\mathbf{y}, \mathbf{w}|\mathbf{q}\right)$ is evaluated using the current estimate $\mathbf{q}[k]$; in the maximization step, we update the estimation by $\mathbf{q}[k+1] = \arg\max_{\mathbf{q}} \mathcal{L}(\mathbf{q})$. Next, we will present the detailed derivation of $P\left(\mathbf{y}, \mathbf{w}|\mathbf{q}\right)$ and $P\left(\mathbf{w}|\mathbf{y}, \mathbf{g}[k]\right)$. Let's start from

$$P\left(\mathbf{y}, \mathbf{w}|\mathbf{q}\right) = P\left(\mathbf{y}|\mathbf{w}, \mathbf{q}\right) P\left(\mathbf{w}|\mathbf{q}\right). \tag{4.9}$$

One may observe that, from Figure 4.2, at a given time t, the term w_t is a sum of two binary bits where one comes from the feedback and the other from a random input. Given the knowledge of $\mathbf{q}$, let's assume p_f to be the probability that the bit from the feedback is "1" and p_u to be the probability that the input bit is "1". Then the probability that w_t equals 1 is given by $p_{w_t} = p_u(1-p_f) + (1-p_u)p_f$. Under the practical assumption of uniformly distributed input binary symbols, we have $p_u = 0.5$. Therefore $p_{w_t} = 0.5$ regardless of the bit from the feedback. Consequently, we may drop the term $P\left(\mathbf{w}|\mathbf{q}\right) = 0.5$, $\forall \mathbf{q}$ from Eq. (4.9), and only calculate $P\left(\mathbf{y}|\mathbf{w}, \mathbf{q}\right)$.

The distribution of $\mathbf{y}_t$ conditioned on $\mathbf{w}$ and $\mathbf{q}$ depends on the channel noise solely.

Hence, we get

$$P\big(\mathbf{y}\,\big|\,\mathbf{w},\mathbf{q}\big)=\prod_{t=0}^{\tau}P\big(\mathbf{y}_t\,\big|\,[w_t\ w_{t-1}\ \ldots\ w_{t-M}],\mathbf{q}\big), \tag{4.10}$$

where $w_t=0$, for $t<0$. Let's denote the states of the encoder as the contents of the memory elements and the current input bit. That is, $\mathbf{s}_t=[s_t^0\ s_t^1\ \cdots\ s_t^M]\stackrel{\text{def}}{=}[w_t\ w_{t-1}\ \ldots\ w_{t-M}]$. Thus,

$$P\big(\mathbf{y}\,\big|\,\mathbf{w},\mathbf{q}\big)=\prod_{t=0}^{\tau}P\left(\mathbf{y}_t|\mathbf{s}_t,\mathbf{q}\right). \tag{4.11}$$

Now, we can express the probability of $\mathbf{y}_t$ conditioned on the encoder parameters using the encoded bit sequence $\mathbf{v}_t\stackrel{\text{def}}{=}[v_t^0\ v_t^1\ \cdots\ v_t^n]^T$ (see Figure 4.2). Consequently,

$$P\left(\mathbf{y}_t|\mathbf{s}_t,\mathbf{q}\right)\ =\ \sum_{\mathbf{v}_t}P\left(\mathbf{y}_t|\mathbf{v}_t\right)P\left(\mathbf{v}_t|\mathbf{s}_t,\mathbf{q}\right). \tag{4.12}$$

Assume y_t^i, y_t^j are statistically independent of each other, $\forall i\neq j$. Hence we can calculate the marginal probability, one by one, as given by

$$P\left(y_t^i|\mathbf{s}_t,\mathbf{q}\right)\ =\ P\left(y_t^i|v_t^i=1\right)P\left(v_t^i=1|\mathbf{s}_t,\mathbf{q}_i\right)+P\left(y_t^i|v_t^i=0\right)P\left(v_t^i=0|\mathbf{s}_t,\mathbf{q}_i\right). \tag{4.13}$$

For the AWGN channel and the binary phase-shift keying (BPSK) modulation, this becomes

$$\begin{aligned}P\left(y_t^i|\mathbf{s}_t,\mathbf{q}\right)\ &=\ \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{1}{2\sigma^2}(y_t^i-1)^2}P\left(v_t^i=1|\mathbf{s}_t,\mathbf{q}_i\right)\\ &\quad+\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{1}{2\sigma^2}(y_t^i+1)^2}P\left(v_t^i=0|\mathbf{s}_t,\mathbf{q}_i\right).\end{aligned} \tag{4.14}$$

Given $\mathbf{s}_t$ and $\mathbf{g}_i$, v_t^i can be written as

$$\begin{aligned}v_t^i\ &=\ g_i^0s_t^0\oplus g_i^1s_t^1\oplus\ldots\oplus g_i^Ms_t^M &(4.15)\\ &=\ \sum_{j=0}^{M}\!\!\oplus\ g_i^js_t^j, &(4.16)\end{aligned}$$

where $\oplus$ denotes the binary addition operator and $\sum\!\!\oplus$ denotes the summation by binary additions. In [62], the values of $P\left(v_t^i=1|\mathbf{s}_t,\mathbf{q}_i\right)$ were calculated using a *forward-backward*

algorithm. We design a new approach namely a *forward accumulation approach*. These values can be calculated via

$$P\left(v_t^i = 1|\mathbf{s}_t, \mathbf{q}_i\right) = P\left(\sum_{j=0}^{M}\oplus\, g_i^j s_t^j = 1|\mathbf{s}_t, \mathbf{q}_i\right). \tag{4.17}$$

Let $q_i^m = P\{g_i^m = 1\}$. We further define P_i^m as

$$P_i^m \stackrel{\text{def}}{=} P\left(\sum_{j=0}^{m}\oplus\, g_i^j s_t^j = 1|\mathbf{s}_t, \mathbf{q}_i\right), \tag{4.18}$$

where

$$P_i^0 = q_i^0 s_t^0. \tag{4.19}$$

Based on Eq. (4.18)

$$\begin{aligned} P_i^m &= \begin{cases} \left(1 - P_i^{m-1}\right) q_i^m + P_i^{m-1}(1 - q_i^m), & \text{if } s_t^m = 1 \\ P_i^{m-1}, & \text{if } s_t^m = 0 \end{cases} \\ &= \left(1 - P_i^{m-1}\right) s_t^m q_i^m + P_i^{m-1}(1 - s_t^m q_i^m). \end{aligned} \tag{4.20}$$

Similarly, $P\left(\sum_{j=0}^{m}\oplus\, g_i^j s_t^j = 0|\mathbf{s}_t, \mathbf{q}_i\right)$ is equivalent to

$$\begin{aligned} 1 - P_i^m &= \begin{cases} \left(1 - P_i^{m-1}\right)(1 - q_i^m) + P_i^{m-1} q_i^m, & \text{if } s_t^m = 1 \\ 1 - P_i^{m-1}, & \text{if } s_t^m = 0 \end{cases} \\ &= \left(1 - P_i^{m-1}\right)(1 - s_t^m q_i^m) + P_i^{m-1} s_t^m q_i^m. \end{aligned} \tag{4.21}$$

Subtracting Eq. (4.21) from Eq. (4.20), we get

$$\begin{aligned} 2P_i^m - 1 &= (2P_i^{m-1} - 1)(1 - 2s_t^m q_i^m) \\ &= -\prod_{j=0}^{m}(1 - 2s_t^m q_i^m). \end{aligned} \tag{4.22}$$

It yields

$$P_i^m = \frac{1}{2} - \frac{1}{2}\prod_{j=0}^{m}(1 - 2s_t^j q_i^j) \tag{4.23}$$

Substituting Eq. (4.23) into Eq. (4.14) via Eqs. (4.17) and (4.18), we get

$$\begin{aligned} P\left(y_t^i|\mathbf{s}_t, \mathbf{q}\right) &= \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(y_t^i-1)^2} P_i^M + \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(y_t^i+1)^2}(1 - P_i^M) \\ &= \frac{1}{2\sqrt{2\pi}\sigma}\left(e^{-\frac{1}{2\sigma^2}(y_t^i+1)^2} + e^{-\frac{1}{2\sigma^2}(y_t^i-1)^2}\right) \\ &\quad + \frac{1}{2\sqrt{2\pi}\sigma}\left(e^{-\frac{1}{2\sigma^2}(y_t^i+1)^2} - e^{-\frac{1}{2\sigma^2}(y_t^i-1)^2}\right)\prod_{j=0}^{M}(1 - 2s_t^j q_i^j). \end{aligned} \tag{4.24}$$

Finally, we rewrite Eq. (4.9) as

$$P\left(\mathbf{y}, \mathbf{w}|\mathbf{q}\right) = \prod_{t=0}^{\tau}\prod_{i=0}^{n} P\left(y_t^i|\mathbf{s}_t, \mathbf{q}\right) \tag{4.25}$$

In addition, we need to compute $P\left(\mathbf{w}|\mathbf{y}, \mathbf{g}[k]\right)$ in (4.8). This can be done using the forward-backward algorithm introduced in [64] with the observation probability conditioned on the states given by Eq. (4.24).

According to Eq. (4.25) and the numerical results of $P\left(\mathbf{w}|\mathbf{y}, \mathbf{g}[k]\right)$ from the forward-backward algorithm, we adopt the gradient ascent method to optimize $\mathbf{q}$ as follows:

$$\theta_i^j[k+1] = \theta_i^j[k] + \mu\Delta\theta_i^j[k],$$

where

$$\Delta\theta_i^j[k] \stackrel{\text{def}}{=} \frac{\partial}{\partial\theta_i^j}\sum_t\sum_{\mathbf{s}_t} P(\mathbf{s}_t|\mathbf{y}, \mathbf{q}[k]) \log P\left(\mathbf{y}_t|\mathbf{s}_t, \mathbf{q}\right), \tag{4.26}$$

θ_i^j is related to q_i^j according to the definition by Eq. (4.6), and μ is a step size ($0 < \mu < 1$).

According to Eqs. (4.24) and (4.25), Eq. (4.26) can be rewritten as

$$\begin{aligned}\Delta\theta_i^j[k] &= \sum_t \sum_{\mathbf{s}_t} P(\mathbf{s}_t|\mathbf{y},\mathbf{q}[k]) \frac{1}{P\left(y_t^i|\mathbf{s}_t,\mathbf{q}\right)} \frac{1}{\sqrt{2\pi}\sigma} \left(e^{-\frac{1}{2\sigma^2}(y_t^i+1)^2} - e^{-\frac{1}{2\sigma^2}(y_t^i-1)^2}\right) \\ &\quad \times s_t^j\, q_i^j(q_i^j-1) \prod_{m=0,\, m\neq j}^{M} (1-2s_t^m q_i^m). \end{aligned} \tag{4.27}$$

4.4 Numerical Results and Summary

In this section, we will present the experiments to measure the performance of our proposed blind encoder parameter estimation scheme. Considered a rate $\frac{1}{3}$ turbo encoder with the recursive convolutional encoder having one input and one non-systematic output v_t^1 as depicted in Figure 4.2. The constraint length is set to be 4, and the transfer function (*generating matrix*) is given by $\left[1 \;\; \frac{1+D+D^3}{1+D^2+D^3}\right]$. We undertake our proposed blind encoder parameter estimator for 4,000 information bits. When the noisy signal arrives at the receiver, we do not use the portion of the sequence arising from the interleaver. The coded bits are modulated using BPSK, so that coded bit "1" is mapped to "+1" and coded bit "0" is mapped to "−1", respectively. At the beginning of the iterative EM procedure, θ_i^j is initialized to be zero so that $q_i^j = 0.5$, $\forall i$, $\forall j$. The severe AWGN channel is artificially made with the low signal-to-noise ratios (SNR) of 0, 2, and 4 dB, since turbo encoder is mostly adopted for low SNR conditions. One hundred Monte Carlo trials have been carried out for each SNR condition and the average blind estimation performances are evaluated.

After the EM procedure is completed, the encoder parameters g_i^j are estimated from q_i^j

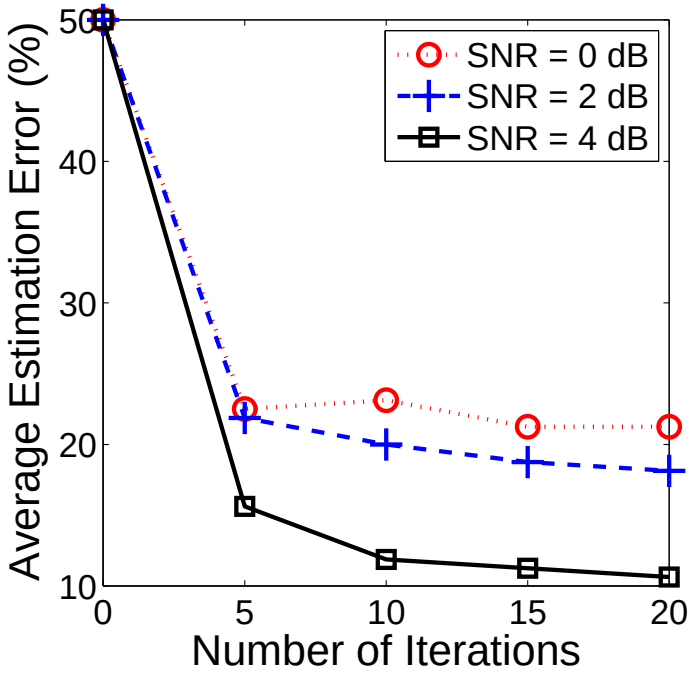

Figure 4.3: Average estimation error rate $\bar{\varepsilon}$ versus iteration numbers used in the EM algorithm for different signal-to-noise ratio (SNR) values.

such that

$$\hat{g}_i^j = \begin{cases} 1, & \text{if } q_i^j > 0.5 \\ 0, & \text{if } q_i^j < 0.5 \end{cases} \tag{4.28}$$

When $q_i^j = 0.5$, $\hat{g}_i^j$ can be randomly picked as either 0 or 1. For N Monte Carlo trials, we define the *average estimation error* $\bar{\varepsilon}$ as

$$\bar{\varepsilon} \stackrel{\text{def}}{=} \frac{1}{N} \sum_{k=1}^{N} \varepsilon_k, \tag{4.29}$$

where

$$\varepsilon_k \stackrel{\text{def}}{=} \frac{1}{(1+n)(1+M)} \sum_{i=0}^{n} \sum_{j=0}^{M} \hat{g}_i^j \oplus g_i^j. \tag{4.30}$$

The average estimation errors versus the iteration numbers used in the EM algorithm are plotted in Figure 4.3 for different SNR conditions. Note that to the best of our knowledge,

there exists no other blind encoder parameter estimator for turbo codes, and thus we cannot compare with other existing method(s).

To conclude, in this chapter, we make the first-ever attempt to design a novel blind encoder parameter estimator for turbo codes. The Monte Carlo simulations demonstrate the average estimation performance can lead to more than 90% accuracy for the severe channel condition with the signal-to-noise ratio at 4 dB. Once the encoder parameters are blindly estimated, the corresponding decoder can be facilitated immediately to retrieve the original information sequence. Our new blind decoding scheme could be very useful in the future communication technologies.

5. HEXAGONAL CLUSTERING

In this chapter we will discuss energy efficient clustering and cluster-head selection strategy for wireless sensor networks (WSNs). One of the constraints in designing a deployment strategy and a routing algorithm for WSNs is the limited battery capacity. To achieve longer network lifetime subject to this energy constraint, clustering and cluster-head selection have drawn a lot of research interest recently. In this chapter, we propose a new topology-dependent clustering and cluster-head selection scheme based on the hexagonal tessellation.

5.1 Literature Review

Wireless sensor networks WSNs are composed of low-cost sensor nodes cooperating in gathering and reporting the application-specific data. The sensor nodes are scattered over the coverage area and have the capabilities of sensing and collecting data, processing data, and routing data to the sink or base station (BS) [26]. Each sensor is equipped with a battery whose size is constrained by the dimensions of the sensor node. This imposes a significant constraint on the power capacity for sensor communications, thus limiting both transmission range and throughput. The aforementioned difficulty has prompted researchers to seek the ways to maximize the lifetime of a wireless sensor network subject to its limited energy source (battery). Therefore, various routing methods were proposed in order to minimize

the power dissipation by implementing the algorithms that utilize low-cost paths [65–70]. Energy-efficient scheduling schemes have also been proposed where sensor nodes stay in a low-power-consumption idle state, whenever they do not transmit or receive any data [65,71–73]. Another approach to maximize the lifetime of a sensor network is through clustering, where sensor nodes will transmit their data collectively to a specific node, known as *cluster-head*, and the cluster-head will transmit the aggregated data from all nodes to the BS [74–84]. The node clustering problem has been tackled by the *grouping* or *matching* approach as indicated in [79]. The deployment strategy designed in [80] suggested a clustering scheme for a network with multiple-sized fixed grids. In [81], the clustering problem was modeled as a *hypergraph partitioning* and its resolution was based on a *tabu search heuristic* using cliques of large size. In [82], a clustering algorithm based on a random contention model without *a priori* knowledge of the network was proposed. In [83], a protocol was presented, which could periodically select cluster-heads according to a hybrid of the *node residual energy* and a secondary parameter such as node proximity to its neighbors or number of neighbors.

In [75], it was suggested that the nodes could organize themselves into clusters with randomly selected cluster-heads responsible for communicating with the base station. The sensor nodes that are selected as the cluster-heads collect data from other sensor nodes, aggregate messages, and then transmit the aggregated data altogether to the base station. Therefore, if the same set of nodes are used as cluster-heads throughout the network lifetime, then the cluster-heads dissipate more energy than the other sensor nodes. The LEACH algorithm in [75] dealt with this problem as follows. First, a randomized rotation of cluster-heads was taken and therefore the energy dissipation could be evenly distributed across the sensors; second, data compression was performed at the cluster-heads prior to the data transmission

from the cluster-heads to the base station. The random cluster-head selection in [75] could perform luckily well only if cluster-head sensors are uniformly distributed in all geometric areas. Usually, the distribution of the cluster-heads over the coverage area is not uniform in practice. Hence, some sensor nodes would need to transmit their messages to a cluster-head over a long distance. As discussed in [77] and [78], it would happen that some cluster-heads are close-by nodes, some are placed near the network boundaries, and the LEACH algorithm could not work effectively thereby. On the other hand, static cluster-head assignments can manage to distribute the cluster-heads uniformly, but the sensor nodes will consume their energy and die out quickly under these schemes.

Motivated by the aforementioned problems, we attempt to design a new clustering and cluster-head selection approach to maximize the network lifetime. Our work has two main contributions. First, we present a dynamic clustering and cluster-head selection approach, which is different from other existing methods as it ensures the fairness in the cluster-head selection and can distribute the cluster-heads uniformly over the coverage area. Second, we will facilitate a detailed lifetime analysis for our proposed method and validate our theory with simulations.

In our proposed method, the entire coverage area in the Euclidean plane is first partitioned into regular hexagonal clusters. Another tessellation is further employed to break each hexagonal cluster into smaller hexagons called *cells*. Then, a cluster-head is selected through negotiation from a cell located at the center of a hexagonal cluster. The negotiation overhead in terms of energy dissipated in computation and communication during this period is negligible, since the number of nodes in a cell can be kept small by controlling the sizes of the clusters and the cells. After a cluster-head is selected it will transmit the data for all other

nodes within the same cluster. This is performed through different data-reporting schemes. We will analytically evaluate the lifetime of a WSN under different data-reporting schemes. These schemes include direct data reporting by each node to the sink or the base station, cluster-data reporting based on the hexagonal clustering with arbitrary cluster-heads, and cluster-data reporting based on the hexagonal clustering with centroid cluster-heads. The single-hop and multi-hop routing schemes will be studied when our proposed new clustering method is adopted. Besides, the effect of path-loss exponent on the lifetime of a sensor network will also be investigated.

5.2 Clustering and Cluster-Head Selection for Sensor Networks

5.2.1 Network Model

Throughout this chapter, a WSN is considered to consist of a BS located at some specified position near or within the coverage area and sensor nodes are arbitrarily scattered over the coverage area. The network can be mathematically modeled by a graph. A *graph* consists of a set of vertices denoted by $\mathcal{V} = \{v_0, v_1, \ldots, v_N\}$ and a set of edges denoted by $\mathcal{E} = \{e_1, e_2, \ldots, e_M\}$, where N is the total number of sensor nodes, M is the total number of edges, and v_0 indicates the base station. In our network model, the vertex set $\mathcal{V}$ represents the BS and all other sensor nodes. The set of weighted edges $\mathcal{E}$ represents the wireless communication links between the BS and each sensor node as well as the links between any pair of sensor nodes. Within this underlying network topology, the set of sensor coordinates is denoted by $\mathcal{S} = \{\mathbf{p}|\mathbf{p} = (x_i, y_i)$ represents the sensor v_i's coordinate, $i = 1, 2, \ldots, N\}$.

The general idea of clustering and cluster-head selection is depicted in Figure 5.1. The

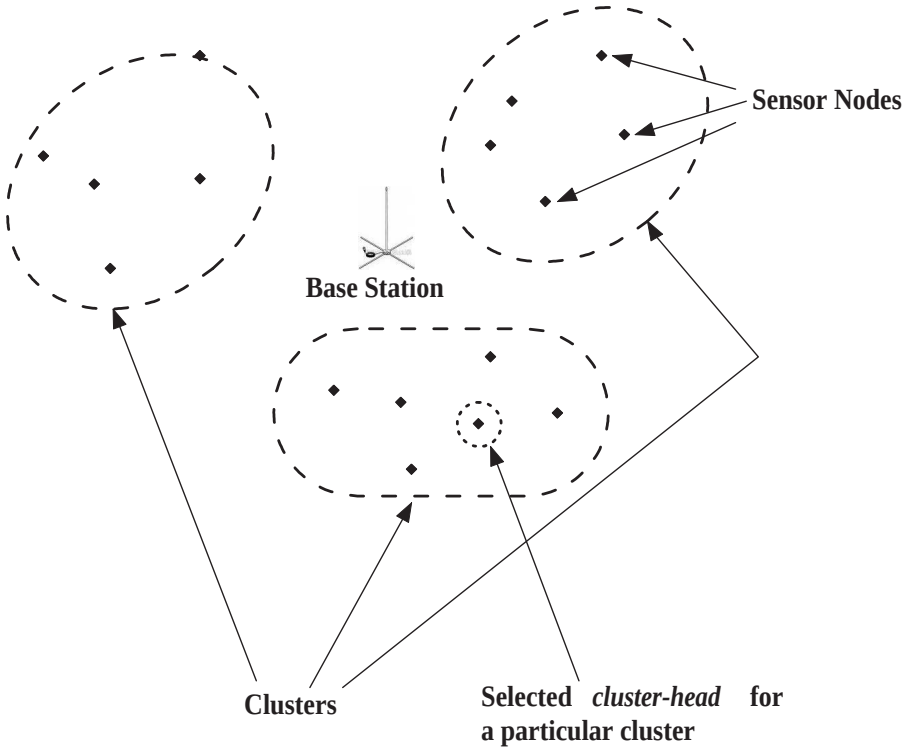

Figure 5.1: The illustration of the clustering and cluster-head selection problem for a sensor network.

problem of clustering and cluster-head selection can be mathematically expressed as follows. The clustering problem is to design a *partition* $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_L$ of $\mathcal{S}$ such that $\bigcup_{l=1}^{L} \mathcal{S}_l = \mathcal{S}$ and $\mathcal{S}_l \subset \mathcal{S}, \forall l$. The cluster-head selection problem is to find a sensor's coordinate, say $(x_{i_l^*}, y_{i_l^*})$, within the l^{th} cluster to transmit all the messages for other sensors therein. Here we introduce hexagonal tessellation based clustering, where each set $\mathcal{S}_l$ is composed of points (sensor locations) within a given hexagonal area (cluster). This idea is manifested by Figure 5.2. Similarly, our proposed cluster-head selection scheme is also based on further partitioning of the hexagonal cluster into smaller hexagons (cells). It is illustrated by Figure 5.3.

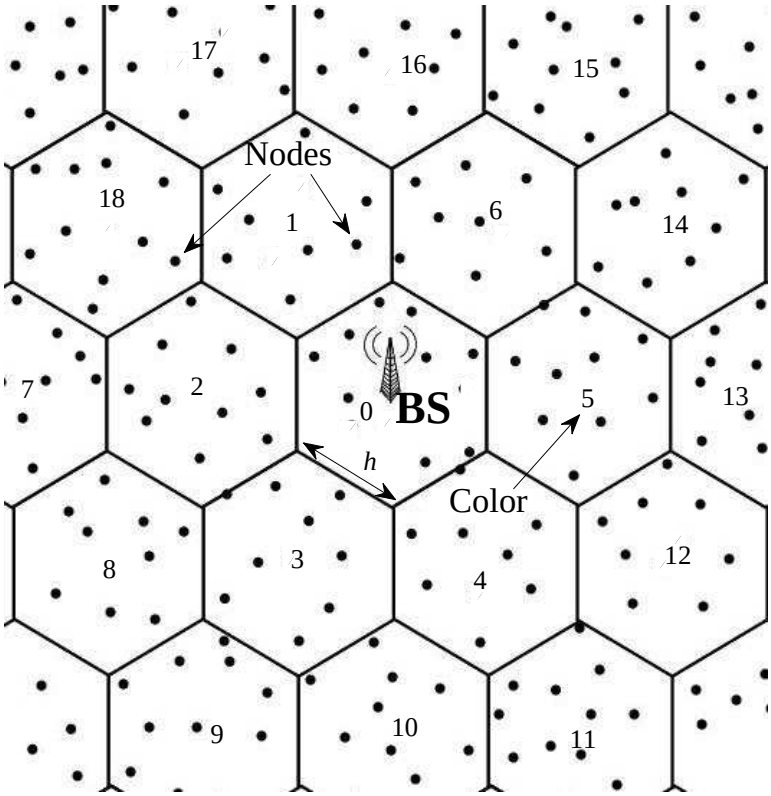

Figure 5.2: The illustration of hexagonal clustering. The black dots are the sensor nodes. The integers in the hexagons are the color indices.

5.2.2 Hexagonal Tessellation

In this subsection, we will present a hexagonal tessellation technique to be used for partitioning the coverage area of a sensor network. Consider $\mathbf{u_1}$ and $\mathbf{u_2}$ to be any two linearly-independent vectors in a two-dimensional Euclidean space (see Figure 5.4). The set of all points given by

$$\Gamma = \{\mathbf{p} \mid \mathbf{p} = a_1\mathbf{u}_1 + a_2\mathbf{u}_2;\ a_1, a_2 \in \mathbb{Z}\} \tag{5.1}$$

constitutes a lattice [85], where $\mathbf{u_1}$ and $\mathbf{u_2}$ are referred to as the *basis vectors* of this lattice. Any point on the lattice has a unique coordinate representation (a_1, a_2). If every point on

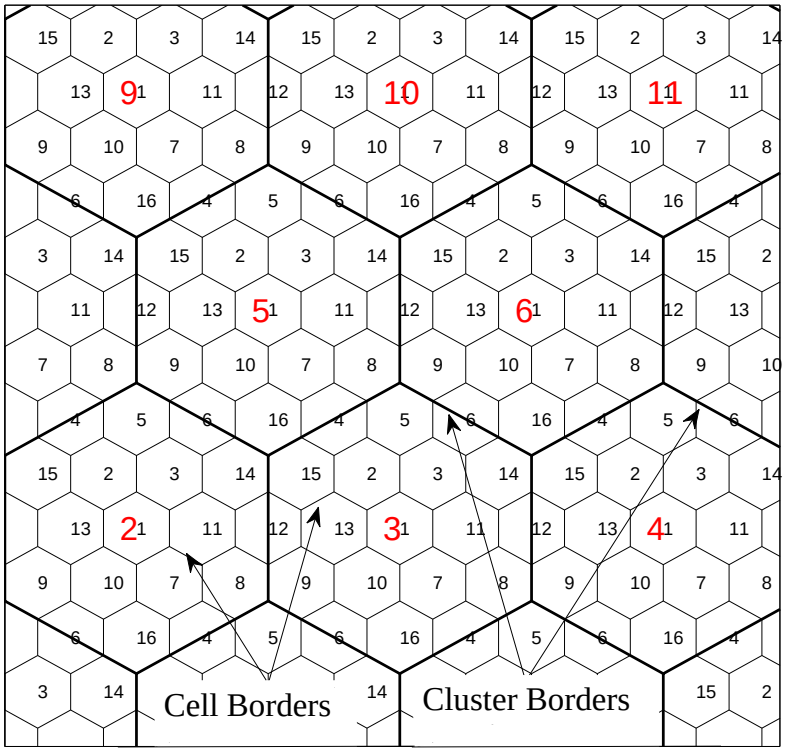

Figure 5.3: Hexagonal Clustering and Cluster-Head Selection: Large hexagons represent the clusters. Different clusters have different "*colors*" (indexed by red integers). For each large hexagon, we further divide it into small hexagons. The cluster-head for a cluster (large hexagon) will be selected from the small hexagon centered at this cluster.

a lattice Λ is also a point on another lattice Γ, then we say that Λ is a *sublattice* of Γ [85].

For a given lattice, the basis vectors may not be unique. If $\mathbf{u_1}$ and $\mathbf{u}_2$ are basis vectors, then $\mathbf{u}'_\mathbf{i} = \sum_j \nu_{i,j}\mathbf{u}_j$, $j = 1, 2$, $i = 1, 2$, are also basis vectors when $\nu_{i,j} \in \mathbb{Z}$ and

$$\left| \det \left\{ \begin{bmatrix} \nu_{11} & \nu_{12} \\ \nu_{21} & \nu_{22} \end{bmatrix} \right\} \right| = 1$$

.

For a given lattice, a *fundamental parallelogram* with respect to this lattice is defined as

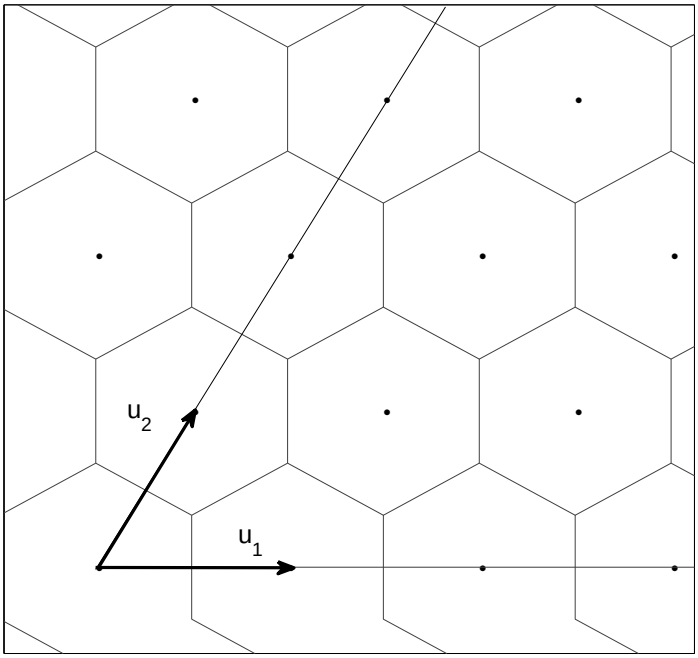

Figure 5.4: Lattice points which are also centers of hexagons. These hexagons partition the two-dimensional space (network coverage area). The basis vectors of this lattice, $\mathbf{u}_1$ and $\mathbf{u}_2$, are also delineated.

a set of all points $\mathcal{P}$, which is given by [85]:

$$\mathcal{P} = \{\mathbf{p} | \mathbf{p} = \beta_1 \mathbf{u}_1 + \beta_2 \mathbf{u}_2;\ 0 \leq \beta_1, \beta_2 < 1\}. \tag{5.2}$$

Here, we focus on the equal-length lattice basis vectors such that $\|\mathbf{u_1}\| = \|\mathbf{u}_2\| = r$. Let's consider the *normalized basis vectors* with $r = 1$. We choose a particular basis vector set such that the two vectors impose an angle of $\frac{\pi}{3}$ radians between them. A lattice based on these vectors will form a hexagonal tessellation. Thus the basis vectors can be given by

$$\mathbf{u}_1 = \begin{pmatrix} 1 \\ \\ 0 \end{pmatrix}, \quad \mathbf{u}_2 = \begin{pmatrix} \frac{1}{2} \\ \\ \frac{\sqrt{3}}{2} \end{pmatrix}. \tag{5.3}$$

According to [86], the lattice points generated by the basis vectors in (5.3) will coincide with the centers of the hexagonal regions and they are at a unit distance from each other.

These hexagonal regions, centered at every point of the lattice, partition the x-y plane. We will exploit this partitioning or *tessellation* to divide an underlying coverage area into the previously mentioned clusters and cells (see Figure 5.4). Furthermore, we assume that any sensor node can have a complete knowledge of its own coordinate and hence it can easily identify exactly in which cluster and cell it resides.

According to the hexagonal tessellation illustrated in Figure 5.4, we can partition the entire network region (or x-y plane) into small hexagonal clusters and cells by using different pairs of basis vectors which are of different lengths. This scheme is further illustrated by Figure 5.3. According to Figure 5.3, each wireless sensor node needs to carry out two things. First, it needs to figure out in which cluster (a large hexagon) it resides based on the *cluster basis vectors* $\mathbf{v}_1$ and $\mathbf{v}_2$ (the basis vectors for large hexagons). These basis vectors are given by Eq. (5.4) later on. Next, each sensor node should determine in which cell it is located. Within each cluster, a different set of basis vectors $\mathbf{u}_1$ and $\mathbf{u}_2$ can be chosen to further decompose the cluster (large hexagon) into cells (small hexagons). At every sensor node, we implement the method introduced in [86] for the purpose of deciding the cluster and the cell where that node belongs to.

This kind of partitioning should be undertaken to address two crucial issues: (i) each sensor should take turn to serve as a cluster-head for other nodes within the same cluster in order to balance the energy consumption and elongate the network lifetime; (ii) the cluster-head should always reside within the cell at the center of each cluster to avoid distant and energy-consuming intra-cluster transmissions and save the overall transmission energy within each cluster. To address the two aforementioned issues, we propose a dynamic clustering approach which can guarantee the cluster-heads to be always within the cells at the centers

of the corresponding clusters. Such a dynamic clustering scheme can be established by a *coloring algorithm*, which colors the various cells in each cluster. The cells with an identical relative position to the corresponding cluster centers have the same color or they are indexed by the same number as illustrated in Figure 5.3. The cluster-heads, which generally have the same cell color will transmit the messages to the base station in the same time slots.

The transition between any two cells of identical color on the Euclidian plane can be given by $j\mathbf{u}_1 + k\mathbf{u}_2$, for integers j and k [86]. According to [86], the set of these identically colored cell centers will form a lattice with the basis vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ given by

$$\begin{aligned} \mathbf{v}_1 &= j\mathbf{u}_1 + k\mathbf{u}_2, \\ \mathbf{v}_2 &= -k\mathbf{u}_1 + (j+k)\mathbf{u}_2. \end{aligned} \tag{5.4}$$

The lattice generated by these new basis vectors will then be

$$\Lambda = \{\mathbf{p} \mid \mathbf{p} = b_1\mathbf{v}_1 + b_2\mathbf{v}_2;\ b_1, b_2 \in \mathbb{Z}\}. \tag{5.5}$$

One interesting property of these basis vectors is that $\mathbf{v}_2$ is a $\frac{\pi}{3}$-rads rotation of $\mathbf{v}_1$. This property does exist for the aforementioned basis vectors $\mathbf{u}_1$ and $\mathbf{u}_2$ as well.

Lattice Λ is a sublattice of Γ given by (5.1), and all points in Λ have the same color. It means that if a sensor node knows its relative position within a certain parallelogram, which is a translation of the fundamental parallelogram given by (5.2), then it can determine its appropriate cell color. The total number of cells in a cluster is the total number of lattice points of Γ that are in the fundamental parallelogram of lattice Λ. This number is equal to the order of the *quotient group* (according to [86]) Γ/Λ such that

$$\Gamma/\Lambda \stackrel{\text{def}}{=} \{\gamma + \Lambda \mid \gamma \in \Gamma\}, \tag{5.6}$$

where $\gamma + \Lambda \stackrel{\text{def}}{=} \{\gamma + \lambda | \lambda \in \Lambda\}$.

The order Q of this quotient group, referred to as the index of Λ in Γ, is given by (see [85])

$$\begin{aligned} Q &= \frac{\begin{vmatrix} j & k \\ k & j+k \end{vmatrix}}{\begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}} \\ &= j^2 + jk + k^2. \end{aligned} \tag{5.7}$$

Eq. (5.7) determines the total number of colors we need to use for a given tessellation (or clustering) in terms of j and k.

5.2.3 Topology-Dependent Distributed Clustering

Our proposed topology-dependent clustering and cluster-head selection scheme is presented here. Since each sensor node can determine its own cluster and cell (color) independently as previously discussed, our scheme is completely *distributed* (*sensor-centric*). This new topology-dependent distributed method involves three major steps, namely (*i*) hexagonal cluster-tessellation, (*ii*) hexagonal cell-tessellation, and (*iii*) cluster-head selection. These steps are introduced in the subsequent subsections.

Hexagonal Cluster-Tessellation

The first procedure of our proposed method is to determine the cluster size and the total number of colors to be used. Since each cluster will have one cluster-head, we determined the cluster size based on the required number of cluster-heads. The optimal number depends on the data-reporting schemes and the communication range between nodes. Empirical results

in [75] suggested that the optimal number of the cluster-heads should be 5% of the nodes. Based on the number of cluster-heads needed, the size of a cluster can be determined to divide the entire network regime into similar hexagons (clusters). According to Eq. (5.4), we can obtain j and k subject to the appropriate size of each hexagon. Thus, the maximum number of clusters, Q given by Eq. (5.7), can be found. We assign a natural number between 1 and Q to each cluster (using the coloring method in [86]) and all sensors within such a cluster should store this cluster index.

Hexagonal Cell-Tessellation

For the cell-tessellation, we first need to determine the number of cells in a cluster, say $Q' = j'^2 + j'k' + k'^2$, according to Eq. (5.7) again. It depends on the sensor population distribution. Consequently, the radius of the cells will be $\frac{\text{(the radius of a cluster)}}{\sqrt{Q'}}$, and each cluster will be covered by Q' cells as depicted in Figure 5.3 ($Q' = 16$ in this figure). Each sensor node uses the information of the cell radius, the values of j' and k', and the coordinate of its own position to determine its cell color based on the method in [86]. Once a cell color is determined by a node, it will not change throughout the node's life; therefore it needs to be carried out only once when the entire sensor network is deployed.

Since j' and k' (or Q') are arbitrary integers to be selected, it is always possible to make the expected number of nodes in a cell as small as required. Hence, the nodes in a given cell can negotiate with each other to become the cluster-head alternately. This can be done with negligible energy loss, not only because the negotiation is undertaken within very few sensor nodes but also because the communication links are very short and energy-efficient for this kind of negotiations. Moreover, negotiations need to be done only once in a network life.

Cluster-Head Selection

We define the lifetime of a network is the duration between the network deployment and the first death of a node. In such a period, our proposed cluster-head selection algorithm is given as below.

Algorithm 3 Cluster-Head selection algorithm

while no node's death **do**
  Q' is determined according to Section 5.2.3.
  for $i = 1$ to Q' **do**
    Step 1: Each cell with color i will contribute a cluster-head sensor.
    Step 2: Inorder to undertake clustering, set the the geometric origin at a lattice point in Γ which has color i {This will also make the cell which is contributing a cluster-head be at the center of its cluster.}
    Step 3: Select a cluster-head within each cell having color i by negotiation.
    Step 4: Transmit the aggregated data to the base station through one of the data-reporting schemes.
  end for
end while

Make sure that the cluster-head is always at the vicinity of the cluster center in Step 2. The cell coloring is fixed throughout the network life, but the clustering performed in Step 2 is continuously changed in order to keep the cluster centers at the respective vicinities of the cluster-heads.

5.3 Lifetime Anlaysis

5.3.1 Data-Reporting Methods

For the purpose of lifetime analysis, the WSN is considered to be composed of a BS located at the center of the circular coverage area and sensor nodes arbitrarily scattered over the coverage area. The distance from the geometric origin (center of the coverage area) to

each node v_i is denoted by $\mathbf{d}_i$. The reference point or the origin is set as the base station location, and therefore $\mathbf{d}_0 = 0$.

Reporting Without Clustering

When reporting without using any clustering method, each individual node transmits the sensed information to the base station directly. For a wireless communication channel, both measurement-based empirical studies and theoretical derivations suggest that the received signal power decreases exponentially with the distance [24]. Hence, the signal power $P_r(d)$ received at a distance d is inversely proportional to d^α, where α is the *path-loss exponent* for the given wireless environment. It yields

$$P_r(d) \propto \left(\frac{1}{d}\right)^\alpha. \tag{5.8}$$

According to Eq. (5.8), we can determine the average transmitted power P_i from a sensor node v_i to the BS as

$$P_i = P \times d^\alpha, \tag{5.9}$$

where P accounts for the channel gain, antenna gains, and sensitivity of the receiver. Consequently, the direct reporting scheme is associated with a weighted edge set $\{e_{j_1}, e_{j_2}, \ldots, e_{j_N}\}$, where e_{j_i} is the link between node v_i and the base station and its weight is specified as the transmitted power P_i.

Reporting via Random Cluster-Heads in the Hexagonal Clustering

1. Direct Reporting:

In this data-reporting case, we will divide the coverage area into hexagons of equal radius,

h, as shown in Figure 5.2. The number in each hexagon represents the particular color index of that cluster [86]. Each node in a hexagonal cluster will determine its own color (identifies which cluster it actually belongs to) based on its location, and each color class (all nodes having the same color index) will form a cluster. It is assumed that all nodes within a given cluster will negotiate and become cluster-heads alternately. While reporting, each node transmits its data to the cluster-head, and the cluster-head will send the aggregated data to the BS. During data aggregation, a cluster-head may process the received data so that it can transmit the compressed data.

2. Multi-hop Routing:

For multi-hop routing, the *geographic greedy routing* [87] is proposed here, where the network is considered to be composed of cluster-heads, and each node relays the routed message to its neighbor closest to the base station.

Reporting via Cluster-Heads Located at the Centers of Hexagonal Clusters

Direct Reporting: In this scheme, the coverage area is partitioned into the regular hexagonal cells. Then, once the data is required to be reported, the coverage area is divided into hexagonal clusters, where each cluster contains many cells within it (see Figure 5.3). The sensor nodes within a cell which is located at the center of a hexagonal cluster negotiate to be cluster-heads alternately. Each selected cluster-head is responsible for transmitting the data collectively for all other nodes within the same cluster. Multi-hop Routing: Similar to Section 5.3.1, the geographic greedy routing method is also used here. The only difference is that in this case the cluster-heads are located in an approximate equidistance from each other.

5.3.2 Network Lifetime Analysis

In our work, the lifetime of a sensor network is considered to be the time elapsed between the deployment of the network and the first death of any node. Besides, the sensing and the reporting will be assumed to be schedule-based, that is, each node reports the sensed data routinely. In the following subsections, we will discuss the network lifetime subject to different data-reporting schemes.

Reporting without Clustering

In this case, each node transmits the sensed information to the base station directly. From Eq. (5.9), we can calculate the amount of transmitted power by each node. If the length of the transmission slot is given by T, then the energy consumed during a single report by node v_i is

$$\begin{aligned} E_i &= TP\mathbf{d}_i^{\alpha} \\ &= E_0\mathbf{d}_i^{\alpha}, \end{aligned} \tag{5.10}$$

where $E_0 \stackrel{\text{def}}{=} TP$.

The lifetime of a sensor network depends on the first death of the first node. This corresponds to the node that has consumed the maximum energy, $E_{\max}$ given by

$$\begin{aligned} E_{\max} &= \max_i E_i \\ &= E_0 \left(\max_i \mathbf{d}_i\right)^{\alpha}. \end{aligned} \tag{5.11}$$

Thus, we first have to determine the distribution of $\mathbf{d}_i$. Since a circular coverage area of radius R is considered and the BS is at the geometric center, the probability density function

$f_D(\mathbf{d}_i)$ with respect to the node-distance from the BS, $\mathbf{d}_i$, is given by

$$f_D(\mathbf{d}_i) = \frac{2\mathbf{d}_i}{R^2}. \tag{5.12}$$

Denote z by $z \stackrel{\text{def}}{=} \max_{1 \le i \le N} (\mathbf{d}_i)$. The distribution of z is thus given by

$$f_Z(z) = \left(\frac{2N}{R^{2N}}\right) z^{2N-1}. \tag{5.13}$$

Then, the statistical average $E_{\text{ave}} \stackrel{\text{def}}{=} \mathbf{E}[E_{\max}]$ in Eq. (5.11) is

$$\begin{aligned} E_{\text{ave}} &= E_0 \times \mathbf{E}\left[z^{\alpha}\right] \\ &= E_0 \left(\frac{2N}{2N+\alpha}\right) R^{\alpha}. \end{aligned} \tag{5.14}$$

Therefore, for a sensor network consisting of nodes with an initial battery energy E, the average lifetime of the network L_{ave} will approximately be

$$\begin{aligned} L_{\text{ave}} &= \frac{E}{E_{\text{ave}}} \\ &= \frac{E}{E_0}\left(\frac{2N+\alpha}{2N}\right) R^{-\alpha} \\ &= L_0 \left(\frac{2N+\alpha}{2N}\right) R^{-\alpha}, \end{aligned} \tag{5.15}$$

where $L_0 \stackrel{\text{def}}{=} \frac{E}{E_0}$. For a large number of nodes (when N is large), the average lifetime of the network will approximately be $L_{\text{ave}} \approx L_0 R^{-\alpha}$ according to Eq. (5.15).

Reporting via Random Cluster-Heads in the Hexagonal Clustering

Assume that we have τ tiers around the cluster containing the BS. Then the total number of clusters, say C, and the average number of nodes per cluster, say n, will be given by

$$C = 1 + \sum_{i=1}^{\tau} 6i, \tag{5.16}$$

$$n = \frac{N}{C}. \tag{5.17}$$

1. Direct Reporting:

In this scenario, each node becomes a cluster-head once per n times of data-reporting, while in the other $n-1$ times, it will just forward the sensed information to the cluster-head served by another node. Hence, the average energy required per data-reporting by node v_i can be approximated as

$$\begin{aligned} E_i &\approx \frac{1}{n}\left[(n-1)E_0\,\mathbf{d}_{i,\mathrm{CH}}^{\alpha} + \gamma n E_0\,\mathbf{d}_i^{\alpha}\right] \\ &= \left(\frac{n-1}{n}\right) E_0\,\mathbf{d}_{i,\mathrm{CH}}^{\alpha} + \gamma E_0\,\mathbf{d}_i^{\alpha}. \end{aligned} \tag{5.18}$$

In Eq. (5.18), the first term $\left(\frac{n-1}{n}\right) E_0\,\mathbf{d}_{i,\mathrm{CH}}^{\alpha}$ approximates the energy consumption during the intra-cluster communication, where $\mathbf{d}_{i,\mathrm{CH}}$ is the distance between node v_i and the cluster-head. The second term $\gamma E_0\,\mathbf{d}_i^{\alpha}$ is the energy consumption while the aggregated information is reported to the BS, where γ is the compression-ratio. If $\gamma = 1$, the cluster-head forwards the received signal from other sensor nodes without any processing. When $\gamma = 1$, this scheme does not appear to be any better than the reporting method without any clustering. Nevertheless in practice when the node density is high, since sensor nodes in a single cluster will be in a close vicinity, their sensed data will be correlated. Therefore, aggregation can result in a compression-ratio less than one.

To estimate the life time of a network consider the worst case scenario within a cluster, that is, the cluster-head is right at the border of a cluster. Let's denote the distance from the cluster-head to any node within this cluster by $\mathbf{d}_{\mathrm{CH,Border}}$. Then the energy consumed

by this node will be

$$\begin{aligned} E_{\max} &= \max_i (E_i) \\ &= \left(\frac{n-1}{n}\right) E_0 \, \mathbf{d}^{\alpha}_{\mathrm{CH,Border}} + \gamma E_0 \, z^{\alpha}. \end{aligned} \tag{5.19}$$

Thus, the average maximum energy consumed by these nodes that determine the lifetime of a network can be expressed as

$$E_{\mathrm{ave}} = \left(\frac{n-1}{n}\right) E_0 \, \mathbf{E}\left[\mathbf{d}^{\alpha}_{\mathrm{CH,Border}}\right] + \gamma E_0 \, \mathbf{E}\left[z^{\alpha}\right]. \tag{5.20}$$

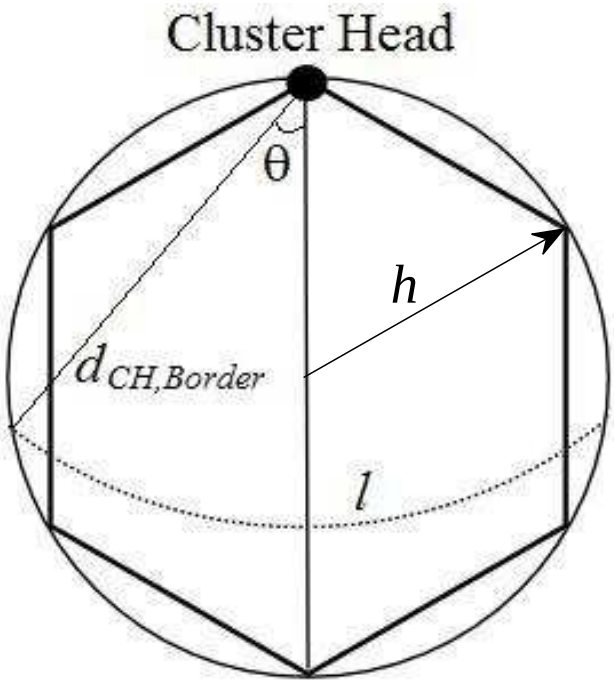

Figure 5.5: The illustration of a distance between a cluster-head and a node within the same cluster.

To determine $\mathbf{E}\left[\mathbf{d}^{\alpha}_{\mathrm{CH,Border}}\right]$, we approximate a hexagonal cluster area as a circular area shown in Figure 5.5. All nodes falling on the arc l will share the same distance $\mathbf{d}_{\mathrm{CH,Border}} = 2h\cos(\theta)$ and the probability density function value of this distance (or with respect to θ in

Figure 5.5) is proportional to the length of the arc l. Thus, we can express the probability density function of θ as

$$\begin{aligned} f_{\Theta}(\theta) &= \frac{2\theta \times 2h\cos(\theta)}{\int_{\theta=0}^{\frac{\pi}{2}} 2\theta \times 2h\cos(\theta)d\theta} \\ &= \left(\frac{1}{\frac{\pi}{2}-1}\right)\theta\cos(\theta). \end{aligned} \tag{5.21}$$

Based on Eq. (5.21), we can determine $\mathbf{E}\left[\mathbf{d}^{\alpha}_{\text{CH,Border}}\right]$ as

$$\begin{aligned} \mathbf{E}\left[\mathbf{d}^{\alpha}_{\text{CH,Border}}\right] &= \mathbf{E}\left[(2h)^{\alpha}\cos^{\alpha}(\theta)\right] \\ &= \frac{(2h)^{\alpha}}{\left(\frac{\pi}{2}-1\right)}\int_{\theta=0}^{\frac{\pi}{2}}\theta\cos^{\alpha+1}(\theta)d\theta. \end{aligned} \tag{5.22}$$

On the other hand, according to Eq. (5.13), the term $\mathbf{E}\left[z^{\alpha}\right]$ in Eq. (5.20) can be determined as

$$\begin{aligned} \mathbf{E}\left[z^{\alpha}\right] &= \frac{2N}{R^{2N}}\int_{0}^{R} z^{2N+\alpha-1}d(z) \\ &= \left(\frac{2N}{2N+\alpha}\right)R^{\alpha}. \end{aligned} \tag{5.23}$$

Therefore, the average maximum energy consumed is given by

$$E_{\text{ave}} = \left(\frac{n-1}{n}\right)E_0\frac{(2h)^{\alpha}}{\left(\frac{\pi}{2}-1\right)}\int_{\theta=0}^{\frac{\pi}{2}}\theta\cos^{\alpha+1}(\theta)d\theta + \gamma E_0\left(\frac{2N}{2N+\alpha}\right)R^{\alpha}. \tag{5.24}$$

Thus, the average lifetime of the network L_{ave} will approximately be

$$L_{\text{ave}} = \frac{L_0}{\left(\frac{n-1}{n}\right)\frac{(2h)^{\alpha}}{\left(\frac{\pi}{2}-1\right)}\int_{\theta=0}^{\frac{\pi}{2}}\theta\cos^{\alpha+1}(\theta)d\theta + \gamma\left(\frac{2N}{2N+\alpha}\right)R^{\alpha}}. \tag{5.25}$$

2. Multi-hop Routing:

When multi-hop routing is considered, at tier t, we have $6t$ hexagonal clusters. Any cluster-head at tier t receives data from tier $t+1$ and forwards it to tier $t-1$. A cluster-head at tier 1 may directly forward the routed data to the base station. The average number of

messages forwarded per node at tier t, namely $\mathcal{M}(t)$, can be calculated in terms of the total number of tiers, τ:

$$\begin{aligned}\mathcal{M}(t) &= 1 + \frac{\text{number of clusters in tiers beyond } t}{\text{number of clusters in tier } t} \\ &= 1 + \frac{\sum_{i=t+1}^{\tau} 6\,i}{6\,t} \\ &= \frac{1}{2}\left(\frac{\tau}{t} + 1\right)(\tau - t + 1)\,.\end{aligned}$$

It is clear that the nodes at the first tier forward the maximum number of messages, which is $\frac{1}{2}\tau(\tau+1)$. Therefore, the nodes at the first tier will limit the lifetime of the sensor network. Thus, the lower limit of the network lifetime can be set as the lifetime of a node at the first tier, that is the farthest from the base station. Let the distance from the BS to node v_i be $\mathbf{d}_{\mathrm{BS,i}}$. Then the node which determines the lifetime of the network in this multi-hop routing is the one with the distance $\mathbf{d}_{\mathrm{MH}} = \max_{v_i} \mathbf{d}_{\mathrm{BS,i}}$ from the BS, where v_i belongs to the 1^{st} tier. Then the energy consumed by such a node is given by

$$\begin{aligned}E_{\max} &= \max_i (E_i) \\ &= \left(\frac{n-1}{n}\right) E_0\,\mathbf{d}^{\alpha}_{\mathrm{CH,Border}} + \frac{1}{2}\tau(\tau+1)\,\gamma E_0\,\mathbf{d}^{\alpha}_{\mathrm{MH}}. \qquad (5.26)\end{aligned}$$

Accordingly, the average maximum energy consumed can be expressed as

$$E_{\mathrm{ave}} = \left(\frac{n-1}{n}\right) E_0\,\mathbf{E}\left[\mathbf{d}^{\alpha}_{\mathrm{CH,Border}}\right] + \frac{1}{2}\tau(\tau+1)\,\gamma\,E_0\,\mathbf{E}\left[\mathbf{d}^{\alpha}_{\mathrm{MH}}\right]. \qquad (5.27)$$

We have determined $\mathbf{E}\left[\mathbf{d}^{\alpha}_{\mathrm{CH,Border}}\right]$ in Eq. (5.22). Here we will derive the expression for $\mathbf{E}\left[\mathbf{d}^{\alpha}_{\mathrm{MH}}\right]$. According to Eq. (5.13), the term $\mathbf{E}\left[\mathbf{d}^{\alpha}_{\mathrm{MH}}\right]$ in Eq. (5.27) can be determined by considering the total number of nodes in the first tier to be $6n$ and the maximum distance

from the base station to be $2h$. It yields

$$\begin{aligned}\mathbf{E}\left[\mathbf{d}_{\mathrm{MH}}^{\alpha}\right] &= \frac{2\times 6n}{(2h)^{2\times 6n}}\int_{0}^{2h}\mathbf{d}_{\mathrm{MH}}^{2\times 6n+\alpha-1}d(\mathbf{d}_{\mathrm{MH}}) \\ &= \left(\frac{12n}{12n+\alpha}\right)(2h)^{\alpha}. \end{aligned} \tag{5.28}$$

And L_{ave} will approximately be

$$L_{\mathrm{ave}} = \frac{L_0}{\left(\frac{n-1}{n}\right)\frac{(2h)^{\alpha}}{\left(\frac{\pi}{2}-1\right)}\int_{\theta=0}^{\frac{\pi}{2}}\theta\cos^{\alpha+1}(\theta)d\theta+\gamma\left(\frac{12n}{12n+\alpha}\right)(2h)^{\alpha}}. \tag{5.29}$$

Reporting via Cluster-Heads Located at the Centers of Hexagonal Clusters

The difference of this data-reporting method from the method stated in Section 5.3.2 is that here the cluster-head is always at the vicinity of the center of the cluster. Thus Eq. (5.20) can be modified as

$$E_{\mathrm{ave}} = \left(\frac{n-1}{n}\right)E_0\,\mathbf{E}\left[\mathbf{d}_{\mathrm{CH,Center}}^{\alpha}\right]+\gamma E_0\,\mathbf{E}\left[z^{\alpha}\right], \tag{5.30}$$

where $\mathbf{E}\left[z^{\alpha}\right]$ is given by Eq. (5.23) and $\mathbf{d}_{\mathrm{CH,Center}}$ is the distance from a node to its cluster-head when the cluster-head is at the center of the cluster. The probability density function of this distance is given by

$$f_{\mathbf{d}_{\mathrm{CH,Center}}}(\mathbf{d}_{\mathrm{CH,Center}}) = \frac{2\mathbf{d}_{\mathrm{CH,Center}}}{h^2}. \tag{5.31}$$

Hence,

$$\begin{aligned}\mathbf{E}\left[\mathbf{d}_{\mathrm{CH,Center}}^{\alpha}\right] &= \frac{2}{h^2}\int_{0}^{h}\mathbf{d}_{\mathrm{CH,Center}}^{\alpha+1}d(\mathbf{d}_{\mathrm{CH,Center}}) \\ &= \left(\frac{2}{\alpha+2}\right)h^{\alpha}. \end{aligned} \tag{5.32}$$

Then, L_{ave} can be expressed as

$$L_{\text{ave}} = \frac{L_0}{\left(\frac{n-1}{n}\right)\left(\frac{2}{\alpha+2}\right) h^\alpha + \gamma \left(\frac{2N}{2N+\alpha}\right) R^\alpha}. \tag{5.33}$$

Similarly for the multi-hop routing, the average consumed energy by the nodes that limit lifetime of the network is given by

$$E_{\text{ave}} = \left(\frac{n-1}{n}\right) E_0 \, \mathbf{E}\left[\mathbf{d}^\alpha_{\text{CH,Center}}\right] + \frac{1}{2}\tau(\tau+1)\,\gamma\, E_0 \, \mathbf{E}\left[\mathbf{d}^\alpha_{\text{MH}}\right], \tag{5.34}$$

where $\mathbf{E}\left[\mathbf{d}^\alpha_{\text{CH,Center}}\right]$ is given by Eq. (5.32) and $\mathbf{E}\left[\mathbf{d}^\alpha_{\text{MH}}\right]$ is given by Eq. (5.28). And, L_{ave} will be

$$L_{\text{ave}} = \frac{L_0}{\left(\frac{n-1}{n}\right)\left(\frac{2}{\alpha+2}\right) h^\alpha + \frac{1}{2}\tau(\tau+1)\,\gamma\, \left(\frac{12n}{12n+\alpha}\right)(2h)^\alpha}. \tag{5.35}$$

5.4 Results and Summary

5.4.1 Theoretical Comparison

In this subsection, we will depict the average lifetimes of a particular network subject to hexagonal clustering. Consider a WSN consisting of 800 nodes on a circular area with a radius of 400 meters. The base station is considered to be located at the center of the coverage area. Assume that much power is consumed by the data transmissions. The transmission power per bit period over a distance of 1 meter is considered to be 10^{-10} Watts, according to [75]. During each data gathering, every node transmits 20 bits and initially each sensor node is assumed to be equipped with a battery of total energy in 20 Joules. Accordingly, we have

$$L_0 = \frac{E}{T\,P} = \frac{20}{20 \times 10^{-10}}.$$

While using hexagonal clustering, we will have two tiers (in a total of 19 hexagonal clusters). To implement hexagonal-tessellation based cluster-head selection, we set $k' = 0$ and $j' = 4$ and hence there will be 16 cells in every cluster. Cluster-heads compress the data using a compression-ratio of 0.1.

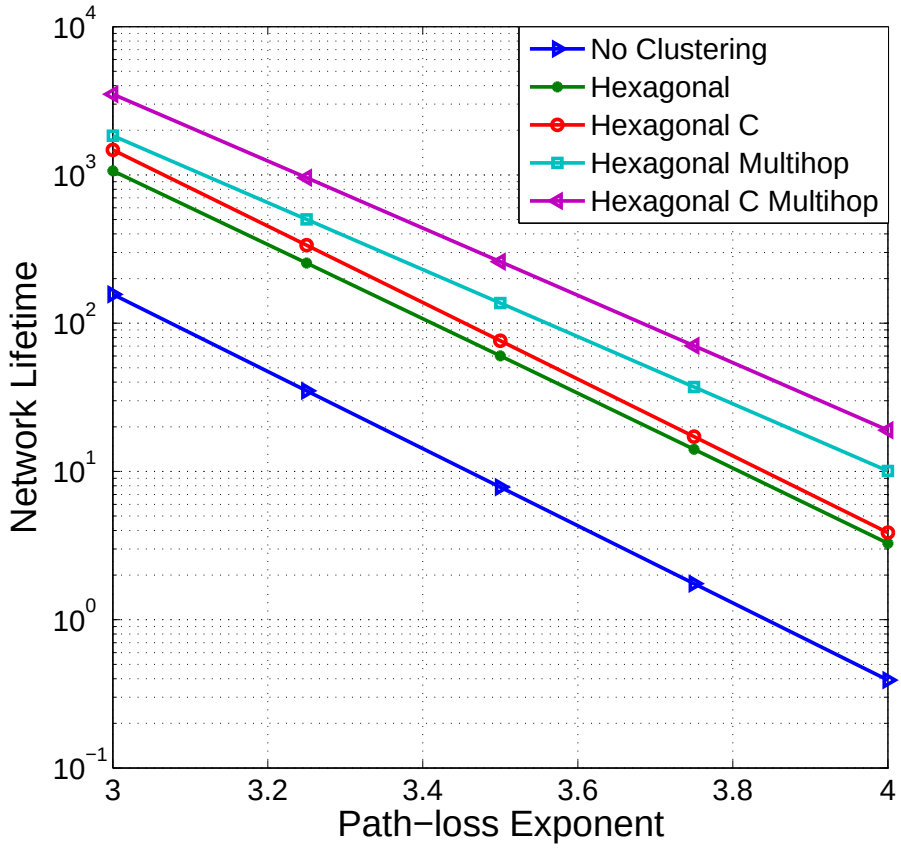

Figure 5.6: The analytical network lifetime versus the path-loss exponent α for different data-reporting schemes. Note that the unit of lifetime is the number of data reports.

The resultant plot of network lifetime versus path-loss exponent is shown in Figure 5.6. From now on, the curves for the single-hop transmission by each node directly without clustering are denoted by "No Clustering". The curves for the single-hop and multi-hop routing schemes with hexagonal clustering are denoted by "Hexagonal" and "Hexagonal Multihop",

respectively. The curves for hexagonal clustering and hexagonal-tessellation based cluster-head selection are denoted by "Hexagonal C" (single-hop routing) and "Hexagonal C Multihop" (multi-hop routing).

From Figure 5.6, one can observe that clustering leads to a better performance in terms of long network lifetime. The main reason is that the clustering technique enables the data compression at the cluster-heads. One can also observe the fact that the multi-hop routing can lead to a longer network lifetime than the single-hop routing; the lifetime difference between these two methods increases as the path-loss exponent increases. Moreover, the network lifetime can be further elongated when the hexagonal-tessellation based cluster-head selection is employed because the link distances for intra-cluster communications can be greatly reduced thereby.

5.4.2 Numerical Evaluation

For all simulations presented in this chapter, an arbitrary network with 800 nodes and a base station right at the center of the circular coverage area which has a radius of 400 meters are considered. Each node's position is randomly generated such that the nodes' population distribution is uniform over the coverage area. The transmission power level can be adjusted so as to use the minimum energy required to reach the intended next-hop receiver and thus the energy consumed per transmission depends on the choice of the next hop. If the distance between the transmitter and the next-hop receiver is d, then the required transmission energy per bit period will be $10^{-10} \times d^{\alpha}$ Joules. For data-reporting, each node generates a 20-bit message, and nodes are all equipped with the 20-Joule batteries. Therefore, we have $L_0 = 10^{10}$. The subsequent subsections will illustrate the actual lifetimes

measured from the simulations for different network parameters.

LEACH Algorithm versus Our Proposed Approach

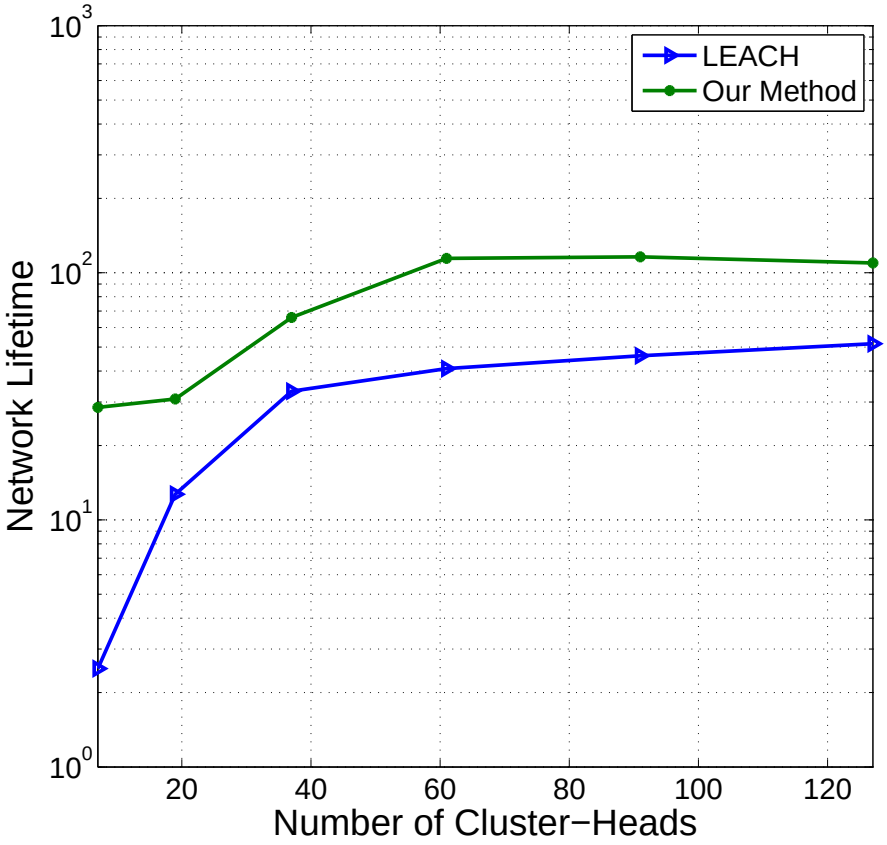

Figure 5.7: The network lifetime comparison between the LEACH algorithm and our proposed approach for various numbers of cluster-heads (clusters). The path-loss exponent is fixed at $\alpha = 3.5$. Note that the unit of lifetime is the number of data reports before any node in the network dies.

For this experiment, the path-loss exponent of the wireless channel is considered as 3.5 and the compression ratio at the cluster-heads is 0.1. In Figure 5.7, the lifetimes of the network versus the number of cluster-heads are plotted for the LEACH algorithm [75] and our approach which employs hexagonal tessellation for both clustering and cluster-head selection and the data-reporting scheme is *multi-hop greedy geographic routing*. According

to Figure 5.7, our approach performs better than the LEACH algorithm. In addition to leading to a longer lifetime, our approach also has the advantage in computational efficiency. For the LEACH algorithm, each node has to search for the nearest cluster-head before it forwards its message. Nevertheless, in our approach, when nodes communicate with their cluster-heads, the search procedure is not necessary; the cluster-head determination of any node depends on the same color index and hence it requires a constant time no matter how many sensor nodes reside in the same cluster.

Comparison between Random Cluster-Head Selection and Hexagonal -Tessellation Based Cluster-Head Selection: Network Lifetime versus Number of Cluster-Heads

For this experiment, the path-loss exponent of the wireless channel is considered as 4 and the compression ratio at the cluster-heads is 0.1. In Figure 5.8, we depict the lifetime curves for the hexagonal clustering method with random cluster-head selection resulting from both theoretical analysis (according to Eq. (5.25)) and simulations as well as those lifetime curves for the hexagonal clustering method which depends on the hexagonal-tessellation based cluster-head selection resulting from both theoretical analysis (according to Eq. (5.33)) and simulations. As one can observe from the plots in Figure 5.8, generally, the hexagonal clustering method which employs the hexagonal-tessellation based cluster-head selection gives better results than the hexagonal clustering method with random cluster-head selection. This phenomenon is reflected by both analytical and simulation results. Note that the more the clusters, the longer the lifetimes for both methods. It results from the fact that the more the clusters, the smaller the communication link distances and then the smaller the power consumption by sensor nodes.

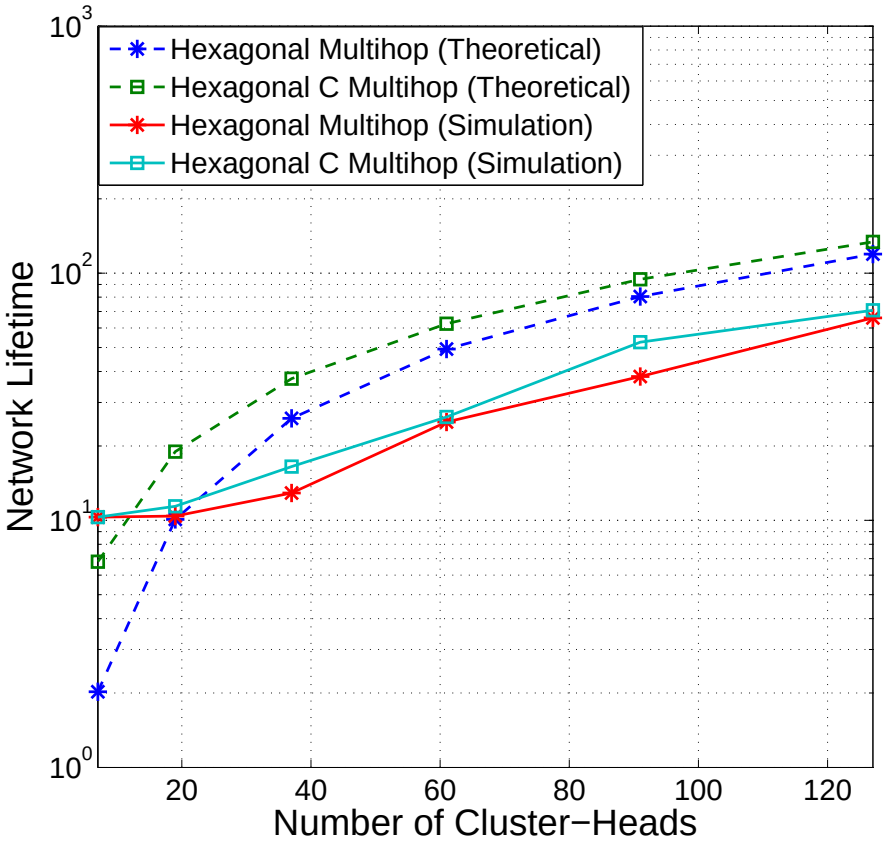

Figure 5.8: The network lifetime comparison between the hexagonal clustering method with random cluster-head selection and the hexagonal clustering method with the hexagonal-tessellation based cluster-head selection for various numbers of cluster-heads (clusters). The path-loss exponent is fixed at $\alpha = 4$. Note that the unit of lifetime is the number of data reports before any node in the network dies.

Comparison between Random Cluster-Head Selection and Hexagonal-Tessellation Based Cluster-Head Selection: Network Lifetime versus Path-loss Exponent

In this experiment, the number of tiers is fixed at 6 so that we have 127 cluster-heads (127 clusters), and the compression ratio at the cluster-heads is 0.1. Figure 5.9 delineates the network lifetime versus the path-loss exponent. The hexagonal-tessellation based cluster-head selection scheme is advantageous over the random cluster-head selection method when the multi-hop routing is considered. In all curves shown by Figure 5.9, the network lifetimes

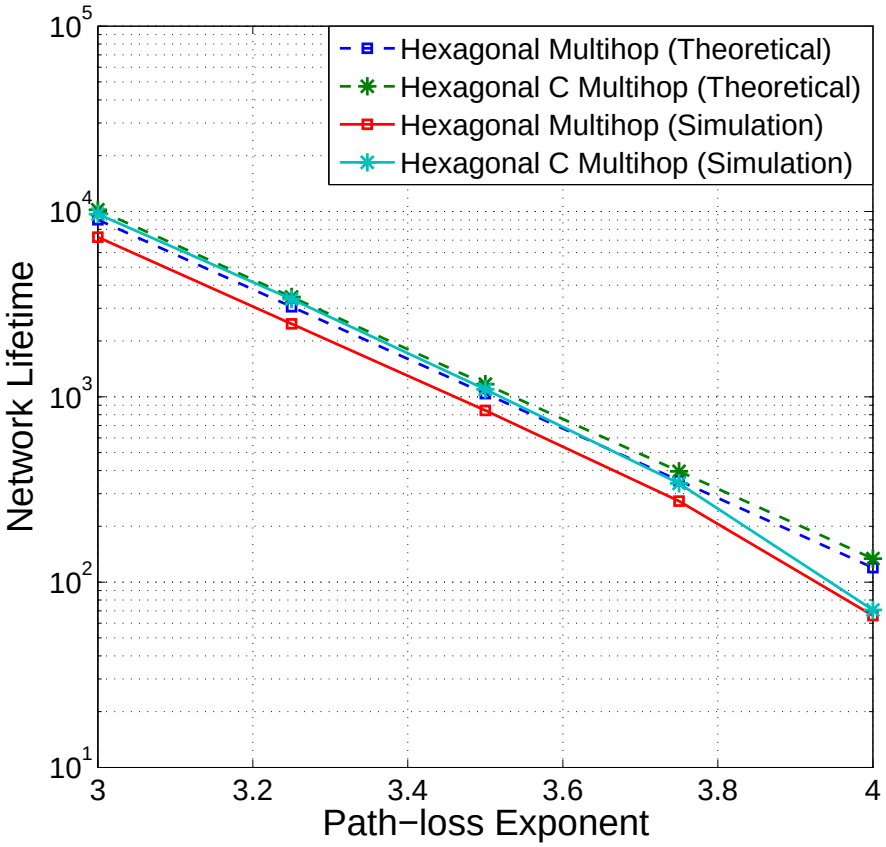

Figure 5.9: The network lifetime comparison between the hexagonal clustering method with random cluster-head selection and the hexagonal clustering method with the hexagonal-tessellation based cluster-head selection. The number of tiers is fixed at 6 while the path-loss exponent α varies. Note that the unit of lifetime is the number of data reports before any node in the network dies.

decrease with the increase in the path-loss exponent for both methods. The reason is obvious. The larger the path-loss exponent, the larger the signal attenuation and then the larger the required transmission power.

To summarize, in this chapter, we present a novel hexagonal clustering method in a distributed manner, where each node determines which cluster it belongs to. Our proposed method does not involve any search algorithm, which is generally very time-consuming. In addition, our proposed method is insensitive to the network dimension and hence it is

well scalable. Moreover, we theoretically analyze the lifetime of a wireless sensor network under different data-reporting schemes. Our newly derived lifetime measures can be utilized as an important tool for the management and coordination of any sensor network. The new lifetime measures can also be used for selecting an appropriate data-reporting scheme for a particular sensor network subject to a given compression-ratio. Our new theoretical results are validated by the simulation results as well. The theoretical analysis matches the simulation results accordingly.

6. CONCLUSION

In this dissertation work, we investigate the important reliability and efficiency issues encountered in wireless communication systems. We proposed and examined different coding/decoding techniques and energy-saving schemes to enhance the reliability and efficiency for different practical applications using wireless systems.

The primary topics addressed in this dissertation are *variable-rate convolutional coding*, *local-content detection through a modified turbo decoder*, *blind turbo encoder estimation*, and *hexagonal-clustering for wireless sensor networks*.

The first contribution of this work is to provide an innovative approach for the performance improvement of mobile communication systems operating over time-varying wireless channels. This system performance enhancement has been achieved through the implementation of a new variable-rate convolutional encoder. This new adaptive channel coding scheme lowers the data rate in an adverse channel condition so as to maintain a low error probability; on the other hand, it increases the throughput in a good channel condition. The resultant low error probability actually reduces the processing delay incurred by retransmissions, commonly encountered in current wireless communication systems.

Another major contribution of this dissertation is to facilitate a novel design of the modified turbo decoder to decode the local content and the development of a computationally-efficient scheme to remove the global content from the received signal in DVB-SH systems.

For most practical purposes, the error performance of our proposed new scheme is as good as that of the standard approach while our scheme can achieve much better computational-efficiency and smaller latency.

Moreover, we have investigated a very new topic, namely the blind estimation of encoder parameters for turbo codes. We propose to separate the feedback components from the forward path in a recursive convolutional encoder so as to blindly estimate the parameters. Our new scheme does not need the knowledge of the encoder or the transmission of training symbols. The average blind encoder estimation performance exhibits 90% accuracy for the channel condition at 4 dB signal-to-noise ratio.

In addition to the coding/decoding studies, we have also investigated the energy efficiency of wireless sensor networks and propose a clustering and cluster-head selection algorithm which significantly improves the lifetime of the entire network. Our new clustering and cluster-head selection method is based on hexagonal tessellation. The individual nodes determine if they can serve as cluster-heads simply subject to their cell colors. Overall, our proposed technique is quite computationally-efficient. This new clustering and cluster-head selection strategy would be very convenient in practice.

Generally speaking, this dissertation addresses crucial issues regarding the reliability and efficiency of wireless communication systems. Based on our theoretical and algorithmic studies, we provide scientific and engineering solutions to the practical problems incurred in the modern wireless communication technologies. The results of this dissertation work would be a valuable addition to future coding, modulation, estimation, and detection in wireless communication systems.

REFERENCES

[1] *Framing Structure, Channel Coding and Modulation for Satellite Services to Handheld Devices (SH) Below 3 GHz,* ETSI EN 302 583 V1.1.1 (2008-03), Digital Video Broadcasting (DVB).

[2] C. E. Shannon, "A mathematical theory of communication," *Bell Sys. Tech. J.*, vol. 27, pp. 379–423, 623–656, July 1948.

[3] J. G. Proakis, *Digital Communications*. McGraw-Hill, 4 ed., 1989.

[4] T. M. Cover and J. A. Thomas, *Information Theory*. Wiley-Interscience, 1991.

[5] S. Lin and D. J. Costello, *Error Control Coding*. Prentice Hall, 2 ed., 2005.

[6] R. W. Hamming, "Error detecting and error correcting codes," *Bell System Technical Journal*, vol. 26, no. 2, pp. 147–160, 1950.

[7] M. J. E. Golay, "Notes on digital coding," in *Proceedings of the IRE*, vol. 37, p. 657, June 1949.

[8] D. E. Muller, "Application of boolean algebra to switching circuit design and to error detection," *IRE Transactions on Electronic Computers*, vol. 3, pp. 6–12, September 1954.

[9] I. S. Reed, "A class of multiple-error-correcting codes and the decoding scheme," *IRE Transactions on Information Theory*, vol. 4, pp. 38–49, September 1954.

[10] E. Prange, "Cyclic error-correcting codes in two symbols," *Technical report AFCRC-TN-57-103 Air Force Cambridge Research Center Cambridge Mass*, september 1957.

[11] A. Hocquenghem, "Codes correcteurs d'erreurs," *Chiffres*, vol. 2, pp. 147–156, 1959.

[12] R. C. Bose and D. K. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Information and Control*, vol. 3, pp. 68–79, March 1960.

[13] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, pp. 300–304, June 1960.

[14] E. Berlekamp, "Nonbinary BCH decoding," *IEEE Transactions on Information Theory*, vol. 14, no. 2, p. 242, 1968.

[15] E. R. Berlekamp, *Algebraic Coding Theory*. McGraw-Hill, 1968.

[16] P. Elias, "Coding for noisy channels," in *IRE Convention Record*, pp. 37–46, March 1955.

[17] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, pp. 260–269, April 1967.

[18] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Transactions on Information Theory*, vol. 28, no. 1, pp. 55–67, 1982.

[19] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding," in *International Conference on Communications*, pp. 1064–1070, 1993.

[20] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1261–1271, 1996.

[21] D. J. Costello, J. Hagenauer, H. Imai, and S. B. Wicker, "Applications of error-control coding," *IEEE Transactions on Information Theory*, vol. 44, pp. 2531–2560, October 1998.

[22] D. J. Costello and G. D. Forney, "Channel Coding: The Road to Channel Capacity," *Proceedings of the IEEE*, vol. 95, pp. 1150–1177, June 2007.

[23] B. Sklar, "Rayleigh fading channels in mobile digital communication systems part i: Characterization," *IEEE Communications Magazine*, vol. 35, pp. 90–100, July 1997.

[24] T. S. Rappaport, *Wireless Communications Principles and Practice*. New Jersey, USA: Prentice Hall, 2nd ed., 2002.

[25] *DVB-SH Implementation Guidelines Issue 2,* ETSI TS 102 584 V1.1.2 (2010) Digital Video Broadcasting (DVB).

[26] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, pp. 102–114, November 2002.

[27] A. Goldsmith and S.-G. Chua, "Adaptive coded modulation for fading channels," *IEEE Transactions on Communications*, vol. 46, pp. 595–602, May 1998.

[28] A. Goldsmith, "Adaptive modulation and coding for fading channels," in *Proceedings of the 1999 IEEE Information Theory and Communications Workshop*, pp. 24–26, 1999.

[29] S. Vishwanath and A. Goldsmith, "Adaptive turbo-coded modulation for flat-fading channels," *IEEE Transactions on Communications*, vol. 51, pp. 964–972, June 2003.

[30] M.-S. Alouini, X. Tang, and A. Goldsmith, "An adaptive modulation scheme for simultaneous voice and data transmission over fading channels," in *48th IEEE Vehicular Technology Conference, 1998. VTC 98.*, vol. 2, pp. 939–943, May 1998.

[31] H.-C. Wu and S. Y. Chang, "Constellation subset selection: Theories and algorithms," *IEEE Transactions on Wireless Communications*, vol. 9, pp. 2248–2257, July 2010.

[32] S. C.-H. Huang, H.-C. Wu, and S. Y. Chang, "Novel efficient algorithms for symmetric constellation subset selection," in *Proceedings of International Conference on Communications (ICC)*, pp. 1–5, May 2010.

[33] S. Falahati, A. Svensson, M. Sternad, and M. Hong, "Adaptive trellis-coded modulation over predicted flat fading channels," in *IEEE 58th Vehicular Technology Conference, 2003. VTC 2003-Fall.*, vol. 3, pp. 1532 – 1536, October 2003.

[34] G. D. Forney, "Coset codes-part I: Introduction and geometrical classification," *IEEE Transactions on Information Theory*, vol. 34, pp. 1123–1151, September 1988.

[35] S. M. Alamouti and S. Kallel, "Adaptive trellis-coded multiple-phase-shift keying for rayleigh fading channels," *IEEE Transactions on Communications*, vol. 42, pp. 2305–2314, June 1994.

[36] S. Massoumi and S. Kallel, "Adaptive trellis coded modulation for mobile communications," in *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, vol. 2, pp. 538–541, May 1991.

[37] M. Ferrari and S. Bellini, "Rate variable, multi-binary turbo codes with controlled error-floor," *IEEE Transactions on Communications*, vol. 57, pp. 1209–1241, May 2009.

[38] J. Hagenauer, N. Seshadri, , and C.-E. W. Sundberg, "The performance of rate-compatible punctured convolutional codes for digital mobile radio," *IEEE Transactions on Communications*, vol. 38, pp. 966–980, July 1990.

[39] A. Lientz and J. Villasenor, "Very low variable-rate convolutional codes for unequal error protection in ds-cdma systems," *IEEE Transactions on Communications*, vol. 45, pp. 753–755, July 1997.

[40] F. Gagnon, D. Haccoun, and C. Leung, "An analysis of convolutional coding for land mobile channels," in *IEEE 39th Vehicular Technology Conference*, vol. 2, pp. 666–670, May 1989.

[41] J.-J. Chang, D.-J. Hwang, and M.-C. Lin, "Some extended results on the search for good convolutional codes," *IEEE Transactions on Information Theory*, vol. 43, pp. 1682–1697, September 1997.

[42] J. Hagenauer and L. Erich, "Forward error correction- coding for fading compensation in mobile satellite channels," *IEEE Journal On Selected Areas In Communications*, vol. SAC-5, pp. 215–225, February 1987.

[43] J. Hagenauer, "Viterbi decoding of convolutional codes for fading and burst channel," in *Proc. of the 1980 Zurich Seminar Digital Communication*, pp. G2.1–G2.7, 1980.

[44] S. A. Wilkus, J. D. Bailey, D. G. Brown, R. Dave, R. L. Dorn, J. Hanriot, M. Hoffman, A. Kulkarni, C. S. Lee, L. R. Meader, J. M. Polakovic, and J. Sullivan, "Field measurements of a hybrid dvb-sh single frequency network with an inclined satellite orbit," *IEEE Transactions on Broadcasting*, vol. 56, pp. 523–531, December 2010.

[45] H. Jiang and P. Wilford, "A hierarchical modulation for upgrading digital broadcast systems," *IEEE Transactions on Broadcasting*, vol. 51, pp. 223–229, June 2005.

[46] H. Jiang, P. A. Wilford, and S. A. Wilkus, "Providing local content in a hybrid single frequency network using hierarchical modulation," *IEEE Transactions on Broadcasting*, vol. 56, pp. 223–229, December 2010.

[47] C.-S. Lee, T. Keller, and L. Hanzo, "Ofdm-based turbo-coded hierarchical and non-hierarchical terrestrial mobile digital video broadcasting," *IEEE Transactions on Broadcasting*, vol. 46, pp. 1–22, March 2000.

[48] G. Gur, S. Bayhan, F. Alagoz, and A. Jamalipour, "On the use of wimax as the terrestrial segment for dvb-sh networks," in *IEEE International Workshop on Satellite and Space Communications. IWSSC 2008.*, pp. 326–330, October 2008.

[49] S. M. S. Sadough and P. Duhamel, "On the interaction between channel coding and hierarchical modulation," in *IEEE International Conference on Communications, 2009. ICC '09*, June 2009.

[50] H. Meric, J. Lacan, C. Amiot-Bazile, F. Arnal, and M. Boucheret, "Generic approach for hierarchical modulation performance analysis: Application to dvb-sh," in *Wireless Telecommunications Symposium (WTS), 2011*, April 2011.

[51] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information Theory*, vol. 20, pp. 284–287, March 1974.

[52] W. E. Ryan, "A turbo code tutorial." 1997.

[53] X. Wang, H.-C. Wu, S. Y. Chang, Y. Wu, and J.-Y. Chouinard, "Efficient non-pilot-aided channel length estimation for digital broadcasting receivers," *IEEE Transactions on Broadcasting*, vol. 55, pp. 633–641, September 2009.

[54] H.-C. Wu, Y. Wu, J. Principe, and X. Wang, "Robust switching blind equalizer for wireless cognitive receivers," *IEEE Transactions on Wireless Communications*, vol. 7, pp. 1461–1465, May 2008.

[55] X. Huang, H.-C. Wu, and Y. Wu, "Novel pilot-free adaptive modulation for wireless OFDM systems," *IEEE Transactions on Vehicular Technology*, vol. 57, pp. 3863–3867, November 2008.

[56] H.-C. Wu, X. Huang, and D. Xu, "Pilot-free dynamic phase and amplitude estimations for wireless ICI self-cancellation coded OFDM systems," *IEEE Transactions on Broadcasting*, vol. 51, pp. 94–105, March 2005.

[57] H.-C. Wu, X. Huang, and D. Xu, "Novel semi-blind ICI equalization algorithm for wireless OFDM systems," *IEEE Transactions on Broadcasting*, vol. 52, pp. 211–218, June 2006.

[58] R. Moosavi and E. G. Larsson, "A fast scheme for blind identification of channel codes," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM 2011)*, December 2011.

[59] F. Wang, Z. Huang, and Y. Zhou, "A method for blind recognition of convolution code based on euclidean algorithm," in *Proceedings of International Conference on Wireless Communications, Networking and Mobile Computing (WiCom 2007)*, pp. 1414–1417, September 2007.

[60] M. Marazin, R. Gautier, and G. Burel, "Blind recovery of k/n rate convolutional encoders in a noisy environment," *EURASIP Journal on Wireless Communications and Networking*, 2011.

[61] J. Dingel and J. Hagenauer, "Parameter estimation of a convolutional encoder from noisy observations," in *Proceedings of IEEE International Symposium on Information Theory (ISIT 2007)*, pp. 1776–1780, June 2007.

[62] T. K. Moon, "Maximum-likelihood binary shift-register synthesis from noisy observations," *IEEE Transactions on Information Theory*, vol. 48, pp. 2096–2104, July 2002.

[63] T. K. Moon, "Maximum likelihood binary shift-register synthesis from noisy observations," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01)*, vol. 6, pp. 3977–3980, May 2001.

[64] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257–286, February 1989.

[65] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, pp. 6–28, December 2004.

[66] Y. T. Hou, Y. Shi, J. Pan, and S. F. Midkiff, "Maximizing the lifetime of wireless sensor networks through optimal single-session flow routing," *IEEE Transactions on Mobile Computing*, vol. 5, pp. 1255–1266, September 2006.

[67] H. Wang, D. Peng, W. Wang, H. Sharif, and H.-H. Chen, "Cross-layer routing optimization in multirate wireless sensor networks for distributed source coding based applications," *IEEE Transactions on Wireless Communications*, vol. 7, pp. 3999–4009, October 2008.

[68] A. H. Shuaib and A. H. Aghvami, "A routing scheme for the ieee-802.15.4-enabled wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 52, pp. 5135–5151, November 2009.

[69] J.-H. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, pp. 609–619, August 2004.

[70] H. M. Ammari and S. K. Das, "Promoting heterogeneity, mobility, and energy-aware voronoi diagram in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, pp. 995–1008, May 2008.

[71] H. Yang and B. Sikdar, "Optimal cluster head selection in the leach architecture," in *IEEE International Conference on Performance, Computing, and Communications IPCCC 2007*, pp. 93–100, April 2007.

[72] V. Rajendran, K. Obraczka, and J. Garcia-Luna-Aceves, "Energy-efficient, collision-free medium access control for wireless sensor networks," *Wireless Networks*, vol. 12, pp. 63–78, Feburary 2006.

[73] H. M. Ammari, "Csi: An energy-aware cover-sense-inform framework for k-covered wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, pp. 651–658, April 2012.

[74] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Journal of Computer Communications*, vol. 30, pp. 2826–2841, October 2007.

[75] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, January 2000.

[76] M. Youssef, A. Youssef, and M. Younis, "Overlapping multihop clustering for wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, pp. 1844–1856, December 2009.

[77] M. J. Handy, M. Haase, and D. Timmermann, "Low energy adaptive clustering hierarchy with deterministic cluster-head selection," in *Proceedings of the 4th International Workshop on Mobile and Wireless Communications Network (MWCN '02)*, pp. 368–372, September 2002.

[78] C.-S. Nam, H.-J. Jeong, and D.-R. Shin, "The adaptive cluster head selection in wireless sensor networks," in *IEEE International Workshop on Semantic Computing and Applications IWSCA '08*, pp. 147–149, July 2008.

[79] T. C. Wong, S. Y. Chang, and H.-C. Wu, "Optimal energy-efficient pair-wise cooperative transmission scheme for WiMax mesh networks," *IEEE Journal on Selected Areas in Communications*, vol. 27, pp. 191–201, February 2009.

[80] T. Kaur and J. Baek, "A strategic deployment and cluster-header selection for wireless sensor networks," *IEEE Transactions on Consumer Electronics*, vol. 55, pp. 1890–1897, November 2009.

[81] A. E. Rhazi and S. Pierre, "A tabu search algorithm for cluster building in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 8, pp. 433–444, April 2009.

[82] P. Guo, T. Jiang, K. Zhang, and H.-H. Chen, "Clustering algorithm in initialization of multi-hop wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 8, pp. 5713–5717, December 2009.

[83] O. Younis and S. Fahmy, "Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, pp. 366–379, October-December 2004.

[84] M. Demirbas, A. Arora, V. Mittal, and V. Kulathumani, "A fault-local self-stabilizing clustering service for wireless ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, pp. 912 – 922, September 2006.

[85] J. W. S. Cassels, *An introduction to the geometry of numbers*. Berlin: Springer-Verlag, 1997.

[86] A. Gamst, "Homogeneous distribution of frequencies in a regular hexagonal cell system," *IEEE Transactions on Vehicular Technology*, vol. 31, pp. 132–144, August 1982.

[87] F. Kuhn, R. Wattenhofer, and A. Zollinger, "An algorithmic approach to geographic routing in ad hoc and sensor networks," *IEEE/ACM Transactions on Networking*, vol. 16, pp. 51–62, February 2008.

VITA

Yonas Debessu received a Bachelor of Science degree in Electrical Engineering from Jimma University, Ethiopia, in 2002. He got a Master of Technology in Electrical Engineering from Indian Institute of Technology, Madras, in 2006. He is currently pursuing the degree of Doctor of Philosophy in the Department Electrical and Computer Engineering Department, Louisiana State University, Baton Rouge. His research interests are in the areas of wireless communications and error control coding.