Performance Analysis of IEEE 802.15.4a
BPSK/BPPM UWB Transmission

by
Zahra Ahmadian
B.Sc., Ajman University of Science and Technology, 2005

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE
in
THE FACULTY OF GRADUATE STUDIES
(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA
September 2007
© Zahra Ahmadian, 2007
The ultra-wideband (UWB) technology has recently attracted considerable attention in industry and academia. This is due to the great potentials of the license exempt operation with UWB signals. These include high data rate, low power consumption, robustness to multipath propagation, good penetration properties and the ability for accurate localization and ranging.

The IEEE working group 802.15 set up two task groups (TGs) for the standardization of UWB physical layers for short-range communication: the IEEE TG 802.15.3a for high data-rate transmission which was officially formed in December 2007 and the IEEE TG 802.154a for low data-rate that became an official task group in March 2004. While the TG 802.15.3a did not succeed and was finally disbanded in January 2006, the TG 802.15.4a approved a draft standard in March 2007. This standard prescribes a rather unique coding and modulation scheme, namely the concatenation of an outer Reed-Solomon and an inner convolutional encoder with a mixed binary phase-shift keying and pulse position modulation (BPSK/BPPM) and time-varying spreading and position hopping. The decoding for and performance analysis of this coding and modulation scheme are the subjects of this thesis.

First, we study the inner convolutional coded BPSK/BPPM in isolation. We suggest an optimal symbol-wise decoding metric, which replaces the sub-optimal bit-wise metric previously suggested in standardization documents, and we define semi-analytical ex-
pressions for the bit-error rate (BER) performance with both optimal and sub-optimal decoding metrics. It is shown through analytical and simulated results that using the optimal symbol-wise metric results in significant performance gains, of e.g. 2 dB at BER of $10^{-3}$, while decoding complexity is identical to that with bit-wise decoding metric. Based on our semi-analytical results, we also quantify the performance loss due to RAKE combining with a limited number of fingers as opposed to ideal combining.

Next, we investigate the entire concatenated coded BPSK/BPPM scheme, including the outer RS code and the inner convolutional code, and we suggest an improved decoding scheme by introducing reliability information generated by the inner decoder. More specifically, two different soft output Viterbi algorithms (SOVA) are considered and compared for generation of reliability information. For the conventional setup of inner Viterbi and outer RS decoder, we define semi-analytical expressions for the frame-error rate (FER) of the overall system. These expressions are highly valuable for quick performance assessment, since simulating the system's performance is extremely time consuming.

In addition to decoding assuming perfect channel state information at the receiver (coherent receiver) considered so far, we also study the performance of decoding without channel state information (non-coherent receiver) to detect the BPPM data bit. This decoding mode is explicitly envisioned by the standard. Again, analytical expressions for BER and FER are obtained, and the performance of non-coherent and coherent receivers are compared.
# Contents

Abstract ........................................................... ii

Contents ............................................................ iv

List of Figures ....................................................... vii

List of Abbreviations and Symbols ................................. x

Acknowledgments ..................................................... xii

1 Introduction ......................................................... 1
  1.1 Background and Motivation ................................... 2
  1.2 Contributions .................................................. 4
  1.3 Thesis Organization ............................................. 5

2 UWB System Model ................................................. 7
  2.1 Transmitter ..................................................... 7
    2.1.1 Channel Coding ........................................... 8
    2.1.2 Modulator .................................................. 9
    2.1.3 BPSK/BPPM Symbol Structure ............................ 10
  2.2 Channel Model ................................................ 13
  2.3 Receiver ........................................................ 14

3 Convolutional Coded BPSK/BPPM ................................ 16
3.1 Equivalent Channel Model ........................................ 17
3.2 Decoding Metrics ..................................................... 17
  3.2.1 Symbol-wise Metric ........................................... 18
  3.2.2 Bit-wise Metric ............................................... 19
3.3 Viterbi Decoding Algorithm ...................................... 20
3.4 BER Analysis for Convolutional Code Decoding .................. 21
  3.4.1 BER Analysis for Symbol-wise Decoding Metric ............ 22
  3.4.2 BER Analysis for Bit-wise Decoding Metric ............... 25
3.5 BER Numerical and Simulated Results ............................ 29
  3.5.1 BER Performance for Symbol-wise and Bit-wise Decoding Metrics 30
  3.5.2 RAKE Receiver ............................................... 31
  3.5.3 Effect of Time-varying Scrambling Sequence .......... 32

4 Concatenated Coded BPSK/BPPM .................................. 37
  4.1 Soft Output Viterbi Algorithm (SOVA) .......................... 38
    4.1.1 Symbol-based SOVA ...................................... 39
    4.1.2 Bit-based SOVA ........................................... 40
  4.2 Decoding of the Concatenated Code ............................. 41
  4.3 FER Analysis for Decoding of the Concatenated Code ......... 43
  4.4 Performance of Concatenated BPSK/BPPM ....................... 45
    4.4.1 FER Using Symbol-wise Decoding Metric ................. 46
    4.4.2 FER Using Bit-wise Decoding Metric .................... 47
    4.4.3 Improved Decoding Scheme with Symbol-based SOVA ...... 50
    4.4.4 Improved Decoding Scheme with Bit-based SOVA .......... 53

5 Non-Coherent BPPM Detection .................................... 57
  5.1 Non-Coherent Detection .......................................... 57
  5.2 BER Analysis of Non-Coherent BPPM ........................... 59
  5.3 FER Analysis of Non-Coherent BPPM with RS-Decoding ....... 60
1.1 FCC limits for indoor UWB emission. source: www.fcc.gov ................. 3
2.1 Block diagram of the transmitter. ............................................... 8
2.2 Block diagram of the systematic convolutional rate-1/2 code. ............. 9
2.3 The symbol structure of a UWB signal. Durations are for mandatory trans-
mission mode of IEEE 802.15.4a with pulse repetition frequency of 15.44 MHz. 11
2.4 The linear feedback shift register structure of scrambler. ................... 12
2.5 Block diagram of the considered receiver. .................................... 14
3.1 Trellis of the convolutional code. ............................................. 18
3.2 Trellis of the convolutional codeword. ..................................... 21
3.3 Two error events with respect to the all-zero codeword. The error events
involve the same set of BPSK/BPPM symbols, but in a different order. ... 24
3.4 Illustration of dominant error events. ....................................... 25
3.5 BER for decoding with symbol-wise and bit-wise metrics. Average over 100
CM2 channels and ARAKE combining. Lines: Analytical results. Markers:
Simulation results. ............................................................... 32
3.6 BER for SRAKE combining with \( L_s = 12 \) and \( L_s = 24 \) fingers. BER for
ARAKE as reference. Decoding with symbol-wise metrics. Average over 100
CM2 channels. Lines: Analytical results. Markers: Simulation results. ... 33
3.7 SNR(ARAKE)−SNR(SRAKE) (in dB) required for a BER of \( 10^{-4} \) for SRAKE
combining with different \( L_s \). Decoding with symbol-wise metrics. Average
over 100 CM2 channels. Analytical results. ............................... 34
3.8 BER\((p)\) [Eq.(3.16)] for ten different trellis positions \(p\). One CM2 channel realization and ARAKE combining. Decoding with symbol-wise metrics.

3.9 Simulated BER for ARAKE combining with and without time-varying scrambler for symbol-wise metric.


4.3 FER and FER_{cc} for ARAKE combining for single channel realization. Lines: Analytical results. Markers: Simulation results.

4.4 Effect of number of paths on calculated FER in Bit-wise decoding metric. Lines: Analytical results. Markers: Simulation results.


4.6 Probability of correct decoding for reliability information in the range 0 to 1 for symbol-based SOVA.

4.7 Probability of correct decoding for reliability information in the range 0.9 to 1 for symbol-based SOVA.

4.8 FER vs \(E_b/N_0\) for the iterative erasures-and-errors decoding with a maximum of \(e\) erasures. Symbol-based SOVA. One channel realization.

4.9 FER vs \(E_b/N_0\) for the non-iterative erasures-and-errors decoding with \(e\) erasures. Symbol-based SOVA. One channel realization.

4.10 FER vs \(E_b/N_0\) for the iterative erasures-and-errors decoding with a maximum of \(e\) erasures. bit-based SOVA. One channel realization.

4.11 FER vs \(E_b/N_0\) for the non-iterative erasures-and-errors decoding with \(e\) erasures. Symbol-based SOVA. One channel realization.
4.12 FER vs $E_b/N_0$ for the iterative erasures-and-errors decoding with a maximum of $e$ erasures. Dotted: symbol-based SOVA, dashed: bit-based SOVA. One channel realization. ........................................ 55

4.13 FER vs $E_b/N_0$ for the non-iterative erasures-and-errors decoding with $e$ erasures. Dotted: symbol-based SOVA, dashed: bit-based SOVA. One channel realization. ........................................ 56

5.1 BER for non-coherent and coherent detection with RAKE receivers with $L_s = 33$ fingers. .................................................. 62

5.2 FER for non-coherent and coherent detection with RAKE receivers with $L_s = 33$ fingers. .................................................. 63

5.3 SNR required at BER = $10^{-2}$ for SRAKE receiver with SLC combining with different $L_s$. .................................................. 63
List of Abbreviations

Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARAKE</td>
<td>All RAKE Combining</td>
</tr>
<tr>
<td>AWGN</td>
<td>Additive White Gaussian Noise</td>
</tr>
<tr>
<td>BER</td>
<td>Bit-Error Rate</td>
</tr>
<tr>
<td>BMA</td>
<td>Berlekamp-Massey Algorithm</td>
</tr>
<tr>
<td>BPPM</td>
<td>Binary Pulse Position Modulation</td>
</tr>
<tr>
<td>BPSK</td>
<td>Binary Phase-Shift Keying</td>
</tr>
<tr>
<td>CM</td>
<td>Channel Model</td>
</tr>
<tr>
<td>CRC</td>
<td>Cyclic Redundancy Check</td>
</tr>
<tr>
<td>CSS</td>
<td>Chirp Spread Spectrum</td>
</tr>
<tr>
<td>FCC</td>
<td>Federal Communications Commission</td>
</tr>
<tr>
<td>FEC</td>
<td>Forward Error Correction</td>
</tr>
<tr>
<td>FER</td>
<td>Frame-Error Rate</td>
</tr>
<tr>
<td>GF</td>
<td>Galois Field</td>
</tr>
<tr>
<td>GPR</td>
<td>Ground Penetrating Radar</td>
</tr>
<tr>
<td>IR-UWB</td>
<td>Impulse Radio Ultra-wideband</td>
</tr>
<tr>
<td>LFSR</td>
<td>Linear Feed-back Shift Register</td>
</tr>
<tr>
<td>LLR</td>
<td>Log-Likelihood Ratio</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium Access Control</td>
</tr>
<tr>
<td>ML</td>
<td>Maximum Likelihood</td>
</tr>
<tr>
<td>MRC</td>
<td>Maximal-Ratio Combining</td>
</tr>
<tr>
<td>PEP</td>
<td>Pairwise Error Probability</td>
</tr>
<tr>
<td>pdf</td>
<td>Probability Density Function</td>
</tr>
<tr>
<td>RFID</td>
<td>Radio-Frequency Identification</td>
</tr>
<tr>
<td>RRC</td>
<td>Root Raised Cosine</td>
</tr>
<tr>
<td>RS</td>
<td>Reed-Solomon</td>
</tr>
<tr>
<td>SER</td>
<td>Symbol-Error Rate</td>
</tr>
<tr>
<td>SLC</td>
<td>Square-Law Combining</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal-to-Noise Ratio</td>
</tr>
<tr>
<td>SOVA</td>
<td>Soft Output Viterbi Algorithm</td>
</tr>
<tr>
<td>SRAKE</td>
<td>Selective RAKE Combining</td>
</tr>
<tr>
<td>TG</td>
<td>Task Group</td>
</tr>
<tr>
<td>UWB</td>
<td>Ultra-wideband</td>
</tr>
<tr>
<td>VA</td>
<td>Viterbi Algorithm</td>
</tr>
<tr>
<td>WPAN</td>
<td>Wireless Personal Area Network</td>
</tr>
</tbody>
</table>
Acknowledgments

Undoubtedly, the completion of this thesis came from the support of many individuals. I would like to convey my gratitude first and foremost to my supervisor Professor Lutz Lampe for his invaluable guidance and continuous support. Professor Lampe provided much of the initial motivation for pursuing this investigation and also provided priceless feedback that has improved this work in nearly every aspect.

I would also like to thank my parents who have always encouraged me in my quest for higher education and especially my husband Hani for his persistent support.

Finally, I would like to extend my thanks to the colleagues at the Department of Electrical and Computer Engineering, UBC, for creating a stimulating and a friendly environment at work.

Zahra Ahmadian

The University of British Columbia
Vancouver, Canada
September 2007
Chapter 1

Introduction

The ultra-wideband (UWB) technology, which is based on transmission of short duration pulses with very high bandwidth, has recently shown its excellent potential as a suitable alternative for many wireless communication scenarios. Besides the application for data communication in wireless personal area networks (WPAN), the small, inexpensive and low power structure of UWB transceivers makes them an excellent candidate for wireless sensor networks [1] and radio frequency identification (RFID). The very wide frequency spectrum of UWB signals gives them high penetration capabilities which is useful in the UWB-based ground-penetrating radars (GPR). The GPR under investigation, can be used by disaster recovery teams for under rubble and through-wall detection [2]. Furthermore, the fine time-delay in UWB signals allows accurate ranging and positioning for inventory control and asset management. In addition to the above mentioned applications, UWB has medical applications in both medical imaging in the form of detecting respiratory and cardiac functions as well as body area networks.
1.1 Background and Motivation

The earliest records of the use of UWB signals go back to 1901 when Guglielmo Marconi transmitted Morse code sequences across the Atlantic. Approximately fifty years later, modern UWB radio was born by the introduction of impulse radio radars by Ross and Benneth [3] and Harmuth [4]. From the 1960s to the 1990s, this technology was restricted to military applications. It was not until February 2002 that the US Federal Communications Commission (FCC) approved the use of UWB for data communications as well as radar and imaging applications under strict power emission limits [5].

UWB signals are defined as the signal with bandwidth of greater than 500 MHz or with a fractional bandwidth larger than 0.2 at all times of transmission. The FCC rulings have allocated the frequency spectrum from 3.1 GHz to 10.6 GHz to UWB communication. However, the maximum power available to a transmitter over the 7.5 GHz bandwidth is only 0.5 mW. Figure 1.1 shows the FCC emission mask for indoor UWB systems.

UWB systems for data communication are typically divided into two classes: UWB systems for high data-rate transmission (above 100 Mbps) and UWB systems for low data-rate transmission (typically below 1 Mbps). The standardization of high data-rate UWB transmission was considered by the IEEE 802.15.3a task group, which however disbanded due to unsolvable disputes between two industry groups.

The standardization of the low data-rate option has successfully been completed under the auspices of the IEEE 802.15.4a task group. This task group was formed to amend the previous IEEE 802.15.4 standard for low-rate WPANs in March 2004 and terminated its work in June 2007 upon successful approval of the IEEE 802.15.4a standard. The standard supports three independent bands of operation below 1 GHz, which is a single channel, (249.6 – 749.6 MHz), the low-band consisting of four channels.
between 3.1 GHz and 4.8 GHz, and the high-band which consists of eleven channels between 6.0 GHz and 10.6 GHz. The modulation format is combined binary phase shift keying (BPSK) and binary pulse position modulation (BPPM) which is denoted as BPSK/BPPM in the following. The forward error correction (FEC) is a concatenated coding scheme with an outer systematic Reed-Solomon (RS) and inner systematic convolutional code. The use of systematic codes allows detection of the transmitted bits without decoding and this utilizes non-coherent detection of the transmitted BPPM bit through energy detection which is explicitly envisioned in the standard. Similar to other proposals time-hopping and spreading are performed for spectral smoothing and multi-user interference rejection. The spreading sequence is a time-varying pseudo-random binary sequence generated by a linear feedback shift register (LFSR) of length 15. In addition to UWB, the standard supports optional chirp spread spectrum (CSS) at 2.450 MHz to support long-range links and links to mobile devices moving at higher speeds [6]. The low power, low complexity and low cost of UWB transceivers com-
plying with this standard make them very attractive for most of the above mentioned low-rate applications such as wireless sensor networks.

The particular coding and modulation scheme prescribed by the IEEE 802.15.4a standard is rather unique. It is therefore interesting to investigate optimum and sub-optimum decoding for IEEE 802.15.4a systems. Furthermore, the performance evaluation for such (and other) coded UWB systems is extremely time consuming, since simulations need to be repeated for many, say 100, different transmission channels generated according to the IEEE 802.15.4a model. Hence, analytical expressions for the error-rate performance of these UWB systems are highly desirable. Therefore, in this work we investigate the decoding for the IEEE 802.15.4a transmission system, and derive semi-analytical expressions for the bit-error rate (BER) and frame-error rate (FER) of these systems and present performance comparisons for different receiver structures.

1.2 Contributions

The main contributions of the present research work are as follows:

- We propose an optimal symbol-wise decoding metric for Viterbi decoding of the convolutional code, which outperforms the bit-wise decoding metric suggested in [7].

- We derive semi-analytical expressions for the BER performance of the convolutional coded BPSK/BPPM transmission, for both symbol-wise and bit-wise decoding metrics, taking into account the effect of the time-varying spreading.

- We derive semi-analytical expressions for the FER of the concatenated coding system according to the IEEE 802.15.4a standard.
Based on the evaluation of the derived expressions and complemented by simulation results, we compare the performances of Viterbi decoding with bit- and symbol-wise metrics and quantify the effect of RAKE combining with a finite number of fingers.

We devise the use of soft-output Viterbi decoding to take advantage of the erasures-and-errors decoding capability of the outer RS decoder. An iterative and a non-iterative decoding scheme are suggested and the performance gains over convolutional decoding are studied through simulation results.

Finally, we investigate the non-coherent reception of the UWB BPPM signal as envisioned in the IEEE 802.15.4a standard and derive analytical results for the BER before and the FER after RS decoder.

Overall, this thesis provides a comprehensive investigation and analysis of the decoding for the IEEE 802.15.4a standard with novel schemes for improved decoding and new expressions for quick performance analysis.

1.3 Thesis Organization

This thesis is organized as follows: In Chapter 2, the UWB system model is presented. The IEEE 802.15.4a transmission system blocks together with the UWB channel model are explained.

Chapter 3 presents the two decoding metrics for Viterbi decoding of the convolutional coded BPSK/BPPM. Expressions for the BER for both decoding metrics are derived using a modified truncated union bound approach.

Chapter 4 considers the entire concatenated coded BPSK/BPPM transmission system according to IEEE 802.15.4a standard. Two soft output Viterbi algorithms (SOVAs)
are introduced and decoding of the concatenated code using reliability information from the inner code is investigated. The performance of the concatenated coded system is then analyzed and frame error rate (FER) approximations are provided.

Chapter 5 introduces the non-coherent detection of the BPPM UWB signal. Analytical results on coherent and non-coherent detection are given and performance comparison for the two schemes is presented.

Finally, this thesis is summarized in Chapter 6 and possible future work is suggested.
Chapter 2

UWB System Model

In this chapter, the overall IEEE 802.15.4a transmission system, including transmitter, forward error correction, symbol structure, channel model and receiver structure are discussed. Section 2.1 provides the details of the transmitter, forward error correction scheme and transmission pulse shape, as specified in the IEEE 802.154a standard. Section 2.2 describes the communication channel model as developed for UWB transmission by the task group TG4a. The receiver structure is described in Section 2.3.

2.1 Transmitter

The IEEE 802.15.4a transmitter is shown in Fig. 2.1. It consists of four main blocks: forward error correction encoder, modulator, time hopping/spreading sequence generator and pulse shaping filter. Each of these blocks will be explained in the following sub-sections.
2.1.1 Channel Coding

The IEEE 802.15.4a standard specifies a concatenated forward error correction coding. The concatenated coding consists of an outer systematic Reed-Solomon rate-55/63 (RS) and an inner systematic rate-1/2 convolutional encoder. There is no interleaving between the two constituent encoders.

**Systematic RS encoder**: RS-codes are a member of polynomial codes and are based on polynomial arithmetic. The specified systematic rate-55/63 RS code is formed over the Galois field GF($2^6$) with generator polynomial $g(x) = \prod_{k=0}^{7} (x + \alpha^k)$ and primitive binary polynomial $p(x) = x^6 + x + 1$. The primitive binary polynomial has a root $\alpha$ such that $p(\alpha) = 0$. The term "systematic" means that the original information sequence is part of the generated codeword. Thus, the 8 parity RS symbols are appended at the beginning of the information sequence to form the RS codeword. The encoding process is performed by processing the information polynomial $(m(x))$ with the generator polynomial $(g(x))$ as follows

$$c(x) = x^{n-k} m(x) + [x^{n-k} m(x) \mod g(x)]$$

where, $c(x)$ is the generated RS-codeword, $x^{n-k} m(x)$ represents cyclic shift by $k$ positions and $x^{n-k} m(x) \mod g(x)$ represents the remainder of the polynomial division $m(x)/g(x)$. The RS encoder accepts 330 data bits. If fewer than 330 data bits are available, the remaining bits will be zero-padded. Then the bits will be converted into
55 RS-symbols defined over GF(2^6), i.e. $m = 6$ bits form one RS symbol. Next, the 8 parity RS symbols will be appended at the beginning of the codeword. The RS codeword is then converted into 378 bits, which are forwarded to the convolutional encoder.

**Systematic convolutional encoder:** The inner code is a rate-1/2 convolutional code with constraint length $K = 3$. The encoder is a four-state machine as shown in Fig. 2.2, with a memory of $l = 2$ and a pair of systematic and non-systematic code bits at the output. The constraint length $K = l + 1$ is the number of input bits that affect one output bit. The generator vectors of the two bits are defined as: $g_1 = [0 1 0]$ and $g_2 = [1 0 1]$. In the encoding process, the systematic position bit is obtained by multiplying the input bit and the contents of the two shift registers with $g_1$, i.e., it is the delayed input bit. The non-systematic sign bit is similarly obtained by multiplying the input bit and the contents of the two shift registers with $g_2$. More details on Reed-Solomon and convolutional codes can be found in [8, 9, 10, 11].

### 2.1.2 Modulator

The modulation is a combination of binary pulse-position modulation (BPPM) and binary phase shift keying (BPSK), which is referred to as BPSK/BPPM in the following. The BPPM maps the value of the bit to pre-defined positions within the transmit
signal. The systematic code bit of the convolutional encoder is the input to the BPPM modulator and represents the position bit. The BPSK modulation chooses the polarity of the transmitted signal based on the input bit. The non-systematic code bit is the input to the BPSK modulator and represents the sign bit.

The combination of BPPM/BPSK is advantageous in both complexity and performance of the system. The BPPM modulated signal is detectable by both coherent and non-coherent receivers, where a non-coherent receiver does not require channel estimation at the receiver. The systematic code bit is modulated into the position of the pulse and since, the non-coherent receiver can recover the original data without the need for complex channel estimation. In case of a coherent receiver, which assumes reliable channel estimation, detection of both sign and position bits is accomplished by means of Viterbi decoding algorithm that improves the performance by correcting and recovering the transmitted bits prior to forwarding them to the RS-decoder. It is worth noting that the use of BPPM signaling reduces the periodicity of the transmitted signal and therefore is effective in spectrum smoothing.

2.1.3 BPSK/BPPM Symbol Structure

The BPSK/BPPM symbols are transmitted as bursts of N pulses or chips with pulse shape $g_c(t)$. Each pulse is multiplied with a coefficient $c_i \in \{\pm1\}$ of a time-varying binary spreading sequence. In addition, each burst is time-hopped among pre-defined hopping positions. Figure 2.3 shows schematically, the structure and timing of such a symbol for the mandatory transmission mode with pulse repetition frequency of 15.44 MHz.

The spreading sequences are the $N$ consecutive outputs of a scrambler. The scrambler is a pseudo-random binary sequence generator that is defined by a linear feedback shift register of length 15 with period $2^{15} - 1$ as shown in Fig. 2.4. The generator polyno-
Figure 2.3: The symbol structure of a UWB signal. Durations are for mandatory transmission mode of IEEE 802.15.4a with pulse repetition frequency of 15.44 MHz.

The polynomial of this linear feedback shift register is: $g(x) = 1 + x^{14} + x^{15}$. Using the spreading sequence results in spectral smoothing as well as interference suppression in multiple user scenarios. Multi-user interference rejection is improved by time hopping the position of the burst. The time hopping position is obtained from the linear feedback shift register as: $d_k = s_j 2^0 + s_{j-1} 2^1 + s_{j-2} 2^2$, with $s_j$ being the current state of the shift register (see Fig. 2.4).

Overall, the transmit signal can be written as

$$s(t) = \sum_{k=-\infty}^{\infty} a_k p_k(t - kT_s - b_k \Delta) ,$$

(2.2)

where $a_k \in \{\pm 1\}$ and $b_k \in \{0, 1\}$ are the $k^{th}$ BPSK and BPPM data symbols respectively, $T_s$ is the symbol interval, $\Delta$ is the PPM delay and $p_k(t)$ is a burst of $N$ chips. Denoting the chip interval by $T_c$, the burst signal is given by

$$p_k(t) = \sum_{l=0}^{N-1} c_{kN+l} g_c(t - lT_c - d_k NT_c) .$$

(2.3)

Throughout this work, we consider the mandatory transmission mode with a data rate of 851 kbps, where $T_c \approx 2$ ns and $T_s = 512 T_c \approx 1$ µs. Since $\Delta = T_s/2$ is true for all modes, we have $\Delta = 512$ ns for the mandatory mode. The two mandatory pulse repetition frequencies are 15.44 MHz and 3.90 MHz, for which $N = 16$, $d_k \in$
The chip pulse $g_c(t)$ is constrained by its normalized cross-correlation with a root raised cosine (RRC) pulse $g_r(t)$ with a roll-off factor $\beta = 0.6$. The normalized cross-correlation is given by

$$\Phi(\tau) = \frac{1}{\sqrt{E_{g_r} E_{g_c}}} \Re \left\{ \int_{-\infty}^{\infty} g_r(t + \tau) g_c(t) \, dt \right\}, \quad (2.4)$$

where, $E_{g_r}$ and $E_{g_c}$ are the energies of the $g_r(t)$ and $g_c(t)$ pulses respectively. For the mandatory transmission mode, the standard specifies the main-lobe width $T_w = 2$ ns for the transmission pulse $g_c(t)$. A transmitter compliant with standard should have a transmitted pulse that has $|\Phi(\tau)| \geq 0.8$ in the main lobe and $|\Phi(\tau)| \geq 0.3$ for each of the side lobes. Assuming the set of points $\tau_i$ for $i = 1, 2, \cdots$ as the points at which $\frac{d}{d\tau} |\Phi(\tau)| = 0$, the maximum of function $(\tau_{\max})$ occurs at one of the $\tau_i$ points. Therefore, the restriction requires $|\Phi(\tau)| > 0.8$ for the duration $T_w$ and $|\Phi(\tau_i)| \leq 0.3$ for all $\tau_i$ [6].

For simplicity, in the following it is assumed that $g_c(t)$ is an RRC pulse with $\beta = 0.6$. 
2.2 Channel Model

The transmission environment affects a UWB signal in a different way, compared to a narrow-band signal. Modeling the transmission channel is based on calculating and measuring the statistical parameters of the channel. The IEEE 802.15.4a channel model adopted in this work was developed by the TG4a task group for both indoor and outdoor environments. The final report and a MATLAB program to generate the channel impulse responses was released in November 2004 [12].

The aforementioned channel model is based on the Saleh-Valenzuela model with some modifications to account for the properties of the measured UWB channels. The impulse response of the multi-path channel consists of $L_c$ clusters of $L_r$ rays and can be expressed in the complex baseband as follows:

$$h(t) = \sum_{l=0}^{L_c} \sum_{k=0}^{L_r} a_{k,l} e^{-j2\pi f_c(T_i + \tau_{k,l})} \delta(t - T_i - \tau_{k,l}),$$

(2.5)

where $a_{k,l}$ is the tap weight of the $k^{th}$ component in the $l^{th}$ cluster, $T_i$ is the delay of the $l^{th}$ cluster, $\tau_{k,l}$ is the delay of the $k^{th}$ multi-path component relative to the $l^{th}$ cluster arrival time $T_i$ and $f_c$ is the baseband transformation frequency. The statistics of these parameters are specified in eight channel models (CMs) according to different environments which are categorized as: residential, indoor office, outdoor and industrial. In generating the channel realizations using MATLAB program, the parameters of each environment are given to the program and based on the statistical characteristics, the multi-path components of the channel are generated. Some of the parameters to be specified are cluster arrival times, cluster decay rates and the multi-path component arrival times within a cluster which is modeled as a Poisson random variable or mixture of two Poisson processes. More details on statistical characteristics of the channel can be found in [13] and [12].
2.3 Receiver

The block diagram of the receiver is shown in Fig. 2.2, where the received signal is

\[ r(t) = s(t) * h(t) + n(t) \quad (2.6) \]

where \( n(t) \) is the white Gaussian noise process and \( * \) denotes the convolution.

The RAKE receiver front-end is commonly used for detection in IR-UWB systems [14, 15, 16]. The operation of a RAKE receiver is based on collecting the signal energy from \( L_s \) different multi-path components of the signal and combining them to make a decision on the transmitted data. We assume that noise reduction is accomplished by the use of chip-matched filtering with \( g_c^*(-t) \) and that the RAKE fingers are spaced by integer multiples of the chip interval \( T_c \) [15]. In case of coherent detection, the RAKE receiver requires knowledge of both amplitude and phase of the multi-path channel gain coefficients.

The received samples from the fingers corresponding to the \( L_s \) strongest resolvable paths are optimally combined using maximal-ratio combining (MRC). If \( L_s \) equals the total number of the resolvable paths, this combiner is referred to as all RAKE (ARAKE), otherwise it is referred to as selective RAKE (SRAKE) [15]. Note that due to the large guard time of about 256 ns, it can be safely assumed that no interference occurs between the two PPM positions and between different symbols, i.e. orthogonal BPPM and intersymbol-interference-free BPSK/BPPM is assumed.

In case of non-coherent detection, the SRAKE can be used with square-law combining (SLC) [16] to capture the signal energy in each of the two PPM positions and no
information about amplitude and phase of the channel coefficients is required.

The output of the RAKE front-end is passed to the Viterbi decoder. Viterbi and RS decoding are discussed in detail in Chapters 3 and 5 assuming coherent RAKE detection. Non-coherent detection and decoding is studied in Chapter 5.
Chapter 3

Convolutional Coded BPSK/BPPM

In this chapter, we consider and analyze the convolutional-coded BPSK/BPPM transmission part of the IEEE 802.15.4a standard. For this purpose, Section 3.1 defines an equivalent channel model which is used in Section 3.2 to describe the decoding metrics. We will introduce two methods to define the decoding branch metrics based on maximum likelihood (ML) and an approximate ML decoding. The Viterbi decoding algorithm using these branch metrics is briefly reviewed in Section 3.3. Next, a detailed bit-error rate (BER) analysis is performed for both ML and approximate ML decoding in Section 3.4. The obtained expressions are evaluated and discussed together with BER simulations in Section 3.5. While the expressions and results derived in this chapter are interesting in its own right, they will be useful when considering the overall concatenated coding scheme in Chapter 4. Throughout this chapter and Chapter 4, we assume perfect channel information being available at the receiver.
3.1 Equivalent Channel Model

It is convenient to derive a simplified, equivalent channel model for BPSK/BPPM transmission as described in Chapter 2. To this end, we note two outputs corresponding to the two BPPM positions are generated per bit interval. Then, it readily follows that the channel between BPSK/BPPM modulation and the SRAKE outputs for one bit interval can be represented by the real-valued two-dimensional vector model as follows:

\[ r_k = a_k s_k^b + n_k, \tag{3.1} \]

where \( r_k \) is the two-dimensional SRAKE output vector, \( n_k \) is the two-dimensional additive white Gaussian noise (AWGN) vector and the vectors \( s_k^0 = [\sqrt{P_k} 0]^T \) and \( s_k^1 = [0 \sqrt{P_k}]^T \) represent the BPPM data bit. The noise has a variance \( \sigma_n^2 = N_0/2 \) per dimension, where \( N_0 \) denotes the two-sided power spectral density of the complex-baseband noise process. The normalized effective channel energy can be defined as

\[ P_k = \frac{\sum_{i=1}^{L_k} |p_k(t) * h(t) * g_c(-t)|_{t_i}^2}{g_c(t) * g_c(-t)|_{t=0}^2}, \tag{3.2} \]

where \( t_i \) are the sampling instances and \((*)\) denotes convolution operation. Note that, due to the time-varying spreading sequence, \( P_k \) depends on the symbol-time index \( k \).

3.2 Decoding Metrics

In this section, we devise two methods of decoding using the Viterbi algorithm [8]. The first method is based on a maximum likelihood (ML) decoding approach employing symbol-wise branch metrics. The second method is an approximate ML decoding that uses bit-wise branch metrics in the form of log-likelihood ratios. The reason to consider the latter is that (i) it was suggested during the standardization in [7] and (ii) bit-wise
metrics are often applied when convolutional coded linear (e.g. QAM) modulations are considered.

### 3.2.1 Symbol-wise Metric

Considering the equivalent channel model in Eq.(3.1), the maximum log-likelihood metric $\lambda_k(a, b)$ associated with a hypothetical BPSK symbol $a$ and a hypothetical BPPM symbol $b$ at time $k$ readily follows as

$$
\lambda_k(a, b) = a \cdot r_k^T s_k^b, \quad a \in \{\pm1\}, \quad b \in \{0, 1\},
$$

where we have exploited the fact that $\|a s_k^b\|^2 = P_k$ and $\|r_k\|^2$ are independent of the transmitted $a$ and $b$.

Considering the particular mapping of the encoder outputs to BPSK and BPPM signal elements and the equivalent data vectors $a_k s_k^b$ in Eq.(3.1), Fig. 3.1 shows the corresponding trellis diagram with the branches labeled with these vectors.
3.2.2 Bit-wise Metric

The second decoding metric definition which is referred to as bit-wise branch metric was described in the standardization document [7]. It is in the form of log-likelihood ratios (LLRs). The LLR for the BPSK bit is given by

\[
\mu'_k = \log \left( \frac{\Pr\{a_k = -1\}}{\Pr\{a_k = +1\}} \right) = \log \left( \frac{p(r_k | s_k^0) + p(r_k | s_k^1)}{p(r_k - s_k^0) + p(r_k - s_k^1)} \right)
\]

\[
= \log \left( \frac{e^{-\|r_k - s_k^0\|^2/(2\sigma^2_k)} + e^{-\|r_k - s_k^1\|^2/(2\sigma^2_k)}}{e^{-\|r_k + s_k^0\|^2/(2\sigma^2_k)} + e^{-\|r_k + s_k^1\|^2/(2\sigma^2_k)}} \right).
\]

(3.4)

where \(p(r_k | s_k^b)\) denotes the probability density function (pdf) of \(r_k\) given \(s_k^b\). The LLR for the BPPM bit is given by

\[
\nu'_k = \log \left( \frac{\Pr\{b_k = 0\}}{\Pr\{b_k = 1\}} \right) = \log \left( \frac{p(r_k | s_k^0) + p(r_k - s_k^0)}{p(r_k | s_k^1) + p(r_k - s_k^1)} \right)
\]

\[
= \log \left( \frac{e^{-\|r_k - s_k^0\|^2/(2\sigma^2_k)} + e^{-\|r_k + s_k^0\|^2/(2\sigma^2_k)}}{e^{-\|r_k + s_k^1\|^2/(2\sigma^2_k)} + e^{-\|r_k - s_k^1\|^2/(2\sigma^2_k)}} \right).
\]

(3.5)

To simplify the bulky expressions for \(\mu'_k\) and \(\nu'_k\), one can apply the nearest-neighbor (or “max-log”) approximation \(\log(e^a + e^b) = \max(a, b) + \log(1 + e^{-|a-b|}) \approx \max(a, b)\) and the simplified bit-wise metrics become

\[
\mu_k = \max (\|r_k^T - s_k^0\|^2, \|r_k^T - s_k^1\|^2) - \max (\|r_k^T + s_k^0\|^2, \|r_k^T + s_k^1\|^2)
\]

\[
= r_k^T s_k^0 + r_k^T s_k^1,
\]

(3.6)

and

\[
\nu_k = \max (\|r_k^T - s_k^0\|^2, \|r_k^T + s_k^0\|^2) - \max (\|r_k^T - s_k^1\|^2, \|r_k^T + s_k^1\|^2)
\]

\[
= |r_k^T s_k^0| - |r_k^T s_k^1|.
\]

(3.7)
The bit-wise decoding metric can readily be defined as sum of two LLRs as follows

\[ \lambda_k(a, b) = \frac{1 + a}{2} \mu_k + (1 - b) \nu_k, \]

(3.8)

where the LLRs are added only for zero sign and position bits.

The defined approximate ML decoding metrics provide simple and direct soft decisions on the sign and position bits separately. However, as it will be shown in the simulation and analytical results in Section 3.5, the bit-wise metric shows considerable performance loss compared to the optimum symbol-wise decoding metric.

### 3.3 Viterbi Decoding Algorithm

The Viterbi decoding algorithm provides an efficient method for ML decoding of a sequence. Following the standard specifications [6], we assume that the encoder starts at all-zero state and returns to all-zero state at the end of the transmitted codeword, i.e. the codeword is padded by two zeros. Fig. 3.2 shows the complete trellis of the inner convolutional codeword of length \( c \). The solid lines represent a transmitted information bit of 0 and dashed lines, a 1. For an arbitrary codeword of length \( c \), the decoding starts by calculating the branch metrics along the trellis starting at time \( k = 1 \), where the two possible paths diverge from the all-zero state. The decoder uses the previously defined metrics in Section 3.2 to calculate the two possible branch metrics arriving at nodes 0 and 2. These two paths diverge into four different paths at time \( k = 2 \). From this point forward, there are two paths arriving at each state node and eight possible path metrics are calculated. The decoder has to decide upon the four survivor paths at the four state nodes and discard the other four paths. The decision is made in favor of the path with the maximum accumulated path metric. The path metric at time index \( k \) and state node \( i \) is defined in terms of the branch metrics \( \lambda^i \) as

\[ \Lambda^i_k = \sum_{t=1}^{k} \lambda^i_t(a_t, b_t). \]

(3.9)
At each step, the path metric for two arbitrary paths A and B arriving at a state node is updated according to the following equation:

\[ \Lambda_k^i = \max(\Lambda_{k-1}^A + \lambda_k^A, \Lambda_{k-1}^B + \lambda_k^A). \]  

(3.10)

At time \( k = c - 1 \), the four survivor paths merge into two paths and only one path with the maximum accumulated path metric is selected as the survivor path arriving at the state node 0 at time \( k = c \). The Viterbi algorithm was implemented for both symbol-wise and bit-wise decoding metrics.

### 3.4 BER Analysis for Convolutional Code Decoding

It is desirable to obtain analytical BER results for coded IR-UWB transmission to avoid the inefficiency of simulations in terms of computational complexity, resources
and time. This is particularly pronounced for UWB transmission, as simulations have to be repeated for many different channel realizations to obtain statistically significant results. In this section, we derive analytical results for both symbol-wise and bit-wise decoding metrics. The analytical results are obtained in the form of modified truncated union bounds in both cases.

### 3.4.1 BER Analysis for Symbol-wise Decoding Metric

The classical method to analyze the error rate of convolutionally coded transmission with Viterbi decoding is to apply the union bound over codeword pairwise error probabilities (PEP) together with the extended path enumerator of the encoder [8, Ch. 4]. Due to the linearity of the code and since each trellis branch corresponds to one BPSK/BPPM symbol, the error probability is the same for all possible codewords. Hence it suffices to assume that the all-zero word \( c_0 \), i.e., a sequence of \( (+s^0) \) symbols, was transmitted. Considering the symbol-wise metric \( \lambda_k(a,b) \) in Eq.(3.3) and the equivalent AWGN channel model in Eq.(3.1), the PEP with respect to any alternative codeword can be calculated by combining the three possible error probabilities which result from incorrect decoding of \((+s^0_k \rightarrow -s^0_k, +s^1_k, -s^1_k)\). In the following we calculate each one of these probabilities and then combine them to obtain PEP for codewords.

For the error \((s^0_k \rightarrow -s^0_k)\) we find

\[
P_c(s^0_k \rightarrow -s^0_k) = \Pr\{r_k \cdot s^0_k < r_k \cdot (-s^0_k)\}
= \Pr\{2\|s^0_k\|^2 + 2s^0_k \cdot n_k < 0\}
= Q\left(\sqrt{\frac{\|s^0_k\|^2}{\sigma_n^2}}\right),
\]

where \(Q(\cdot)\) is the Gaussian Q-function. Similarly the probability of error for the trans-
mitted $s^0$ decoded as $s^1$ is given by

$$P_e(s^0_k \rightarrow s^1_k) = \Pr \{r_k \cdot s^0_k < r_k \cdot s^1_k\} = \Pr \{|s^0_k| - (s^0_k - s^1_k) \cdot n_k < 0\}. \quad (3.12)$$

Since $s^0_k$ and $s^1_k$ are orthogonal, $|s^0_k - s^1_k|^2 = |s^0_k + s^1_k|^2 = 2|s^0_k|^2$. Therefore $P_e(s^0_k \rightarrow s^1_k) = P_e(s^0_k \rightarrow -s^1_k)$ and the probability of error in the position bit follows as

$$P_e(s^0_k \rightarrow \pm s^1_k) = Q \left( \sqrt{\frac{|s^0_k|^2}{2\sigma^2_n}} \right). \quad (3.13)$$

Next, using equations (3.11) and (3.13), the PEP with respect to an alternative codeword $c_i$, $i > 0$, representing a single error event starting at $k = p$ and terminating at $k = q + 1$ and corresponding to $q - p + 1$ BPSK/BPPM symbols ($a_p s^b_p \ldots a_q s^b_q$), can be formulated as

$$P_e(c_0 \rightarrow c_i) = Q \left( \sqrt{\sum_{k=p}^{q} |s^0_k - a_k s^b_k|^2 / (4\sigma^2_n)} \right). \quad (3.14)$$

Furthermore, since (i) $a_k s^b_k = -s^0_k$ only for $k = p$ and $k = q$ (see Fig. 3.1), (ii) $|s^0_k|^2 = |s^1_k|^2 = P_k$, and (iii) $|s^0_k - s^1_k|^2 = |s^0_k + s^1_k|^2 = 2P_k$, this expression simplifies to

$$P_e(c_0 \rightarrow c_i) = Q \left( \sqrt{\frac{q}{\sigma^2_n} + \sum_{k=p+1}^{q-1} P_k b_k} \right). \quad (3.15)$$

If $P_k$ was a constant, i.e., $P_k = P$, the PEP in Eq.(3.15) would only be a function of the Hamming weight $d_H$, of the position bits of $c_i$ and given by $Q(\sqrt{(2 + d_H/2)P/\sigma^2_n})$. However, because of the time variance of $P_k$ due to the time-varying spreading sequences, the PEP depends on the positions of the bits in which $c_0$ and $c_i$ differ. Fig. 3.3 illustrates this for the two error events $[-s^0_p, -s^1_{p+1}, s^1_{p+2}, -s^1_{p+3}, s^0_{p+4}, s^1_{p+5}, -s^0_{p+6}]$ and $[-s^0_p, s^1_{p+1}, s^0_{p+2}, -s^1_{p+3}, s^1_{p+4}, -s^1_{p+5}, -s^0_{p+6}]$, which only differ in the order of BPSK/
Figure 3.3: Two error events with respect to the all-zero codeword. The error events involve the same set of BPSK/BPPM symbols, but in a different order.

Therefore, a different approach is chosen. First, in the calculation of PEP only dominant error events that include \( d_{s_1} \leq L \) symbols (±s\(_1^1\)), are considered. Figure 3.4 shows these error events for \( d_{s_1} = 1 \) (left) and \( d_{s_1} = 2 \) (right). Second, the truncated union bound is calculated for the BER using these error events and assuming a given starting position \( p \),

\[
\text{BER}(p) = \sum_{d_{s_1}=1}^{L} \sum_{d_{s_1}\leq L} d_{s_1} P_0(c_0 \rightarrow c_i),
\]

where, the second sum is over all codewords \( c_i \), such that the error event starts at a time \( p \) and traverses along \( d_{s_1} \) symbols (±s\(_1^1\)).

The time varying scrambling sequences result in a different bit error probability at each starting position \( p \). Therefore, the effect of different scrambling sequences is
Figure 3.4: Illustration of dominant error events.

accounted for by an average BER which is obtained as

$$\text{BER} = \frac{1}{2^{15} - 1} \sum_{p=1}^{2^{15} - 1} \text{BER}(p), \quad (3.17)$$

where $2^{15} - 1$ is the period of the linear feedback shift register of length 15 which is used for generation of spreading sequences. It will be seen in Section 3.5, that the calculated average BER obtained from Eq.(3.17) coincides well with the simulated BER.

### 3.4.2 BER Analysis for Bit-wise Decoding Metric

Similar to the symbol-wise decoding metric the probability that the transmitted $s_k^0$ is incorrectly decoded can be found by forming the difference metric in terms of LLRs (see the LLR expressions in equations (3.6) and (3.7)). The results can be combined to obtain the PEP of the transmitted all-zero codeword with respect to any alternative codeword.
The individual error probabilities are given by

\[ P_e(s^0 \rightarrow -s_0) = \Pr\{\mu_k + \nu_k < \nu_k\} = \Pr\{\mu_k < 0\} \],

\[ P_e(s^0 \rightarrow s_1) = \Pr\{\mu_k + \nu_k < \mu_k\} = \Pr\{\nu_k < 0\} \],

\[ P_e(s^0 \rightarrow -s_1) = \Pr\{\mu_k + \nu_k < 0\} \].

(3.18)

Combining the above probabilities, the probability of error for a codeword \( c_i \) deviating from \( c_0 \) at \( k = p \) and merging at \( k = q + 1 \) with \( q = p + 1 \) BPSK/BPPM symbols \( (a_p s_p^b \ldots a_q s_q^b) \) along the detour is given by

\[ P_e(c_0 \rightarrow c_i) = \Pr\left\{ \sum_{k=p}^q \left[ \frac{(a_k - 1)}{2} \mu_k - b_k \nu_k \right] > 0 \right\} \].

(3.19)

Unlike the case of decoding with symbol-wise metrics (see Eq.(3.14)), there is no closed-form solution to the PEP expression in (3.19). We therefore provide a numerically efficient approach to evaluate (3.19) for the dominant error events.

In the following, the probability of error for the shortest error event, which is shown in Fig. 3.4 (left side), is calculated to illustrate the proposed procedure. The accumulated metric difference for this error event is

\[ \Delta_p = -\mu_p - \nu_{p+1} - \mu_{p+2} \]

\[ = -(r_p^T s_p^0 + r_{p+1}^T s_{p+1}^0) - (r_{p+1}^T s_{p+1}^0 p_{p+1}) - (r_{p+2}^T s_{p+2}^0 + r_{p+2}^T s_{p+2}^1) \]

\[ = -\|s_p^0\|^2 - (s_p^0)^T n_p - (s_{p+1}^0)^T n_{p+1} - (s_{p+1}^0)^T n_{p+1} \]

\[ + (s_{p+1}^1)^T n_{p+1} - \|s_{p+2}^0\|^2 - (s_{p+2}^0)^T n_{p+2} - (s_{p+2}^1)^T n_{p+2} \]

\[ \triangleq N_{p,1} + |N_{p,2}| - |N_{p,3}| \],

(3.20)

where \( N_{p,1}, N_{p,2}, \) and \( N_{p,3} \) are real-valued Gaussian random variables

\[ N_{p,1} = -\|s_p^0\|^2 - (s_p^0)^T n_p - (s_{p+1}^0)^T n_{p+1} - (s_{p+2}^0)^T n_{p+2} - (s_{p+2}^1)^T n_{p+2} \],

(3.21)

\[ N_{p,2} = |(s_{p+1}^1)^T n_{p+1}| \],

(3.22)

\[ N_{p,3} = | \|s_{p+1}^0\|^2 + (s_{p+1}^0)^T n_{p+1} | \].

(3.23)
with mean values,

\[
m_{N_{p,1}} = -(P_p + P_{p+2}),
\]
\[
m_{N_{p,2}} = 0,
\]
\[
m_{N_{p,3}} = P_{p+1},
\]

(3.24)

and variances,

\[
\sigma^2_{N_{p,1}} = \sigma^2_n(2P_p + 2P_{p+2}),
\]
\[
\sigma^2_{N_{p,2}} = \sigma^2_nP_{p+1},
\]
\[
\sigma^2_{N_{p,3}} = \sigma^2_nP_{p+1}.
\]

(3.25)

The PEP for this error event is \( \Pr\{\Delta_p > 0\} \), which can be expressed as

\[
\Pr\{\Delta_p > 0\} = \int_0^{\infty} p_{\Delta_p}(x)dx = \frac{1}{2\pi j} \int_{c-j\infty}^{c+j\infty} \frac{\Phi_{\Delta_p}(s)}{s} ds,
\]

(3.26)

where \( p_{\Delta_p}(x) \) is the probability density function (pdf) of \( \Delta_p \), \( \Phi_{\Delta_p}(s) \) is the Laplace transform of \( p_{\Delta_p}(x) \), and \( c < 0 \) lies in the region of convergence of \( \Phi_{\Delta_p}(s) \). Since \( N_{p,1} \), \( N_{p,2} \), and \( N_{p,3} \) are independent random variables, \( \Phi_{\Delta_p}(s) \) is given by the product of the Laplace transforms of the pdfs of \( N_{p,1} \), \( |N_{p,2}| \), and \( -|N_{p,3}| \)

\[
\Phi_{\Delta_p}(s) = \Phi_{N_{p,1}}(s)\Phi_{|N_{p,2}|}(s)\Phi_{(-|N_{p,3}|)}(s).
\]

(3.27)

Using the expressions for the pdfs of Gaussian, Rayleigh, and Ricean distributed random variables [17], the Laplace transforms on the right-hand side of (3.27) are quickly found as

\[
\Phi_{N_{p,1}}(s) = e^{sm_{N_{p,1}}+s^2\sigma^2_{N_{p,1}}/2},
\]

(3.28)

\[
\Phi_{|N_{p,2}|}(s) = 2e^{s^2\sigma^2_{N_{p,2}}/2} Q(s\sigma_{N_{p,2}}),
\]

(3.29)

\[
\Phi_{(-|N_{p,3}|)}(s) = e^{-sm_{N_{p,3}}+s^2\sigma^2_{N_{p,3}}/2} \left[ Q\left(\frac{s\sigma_{N_{p,3}} - m_{N_{p,3}}}{\sigma_{N_{p,3}}}ight) + e^{2rs} Q\left(\frac{s\sigma_{N_{p,3}} + m_{N_{p,3}}}{\sigma_{N_{p,3}}}ight)\right],
\]

(3.31)
where the Q-function is defined for complex-valued arguments and is related to the error function through $\text{erf}(x) = 1 - 2Q(\sqrt{2x})$ [18].

While a closed-form solution of the integral (3.26) cannot be obtained for $\Phi_{\Delta_p}(s)$ in equations (3.27) with (3.28)-(3.31), efficient numerical integration based on Gauss-Chebyshev quadratures is possible [19, 18] and $P_r\{\Delta_p > 0\}$ is approximated as

$$Pr\{\Delta_p > 0\} \approx \frac{1}{K} \sum_{k=1}^{K/2} \left( \Re \left[ \Phi_{\Delta_p}(c + j\pi) \right] + \pi_k \Im \left[ \Phi_{\Delta_p}(c + j\pi) \right] \right),$$

(3.32)

where $\pi_k = \tan[(2k - 1)\pi/(2K)]$, and $K$ is the number of nodes at which the function is evaluated. A value of $K$ equal to 200 was found to yield satisfactory results. (The approximation can be made arbitrary precise by increasing $K$.)

The PEP for other error events involving symbols $-s_k^0$ and $s_k^1$ for various $k$ can be obtained similarly, since the corresponding Laplace transform $\Phi_{\Delta_p}(s)$ is a product of the transforms in equations (3.28)-(3.31). If the error event also traverses along a branch corresponding to $-s_{p+i}$, for example for one of the error events on the right-hand side of Fig. 3.4 for $i = 1$ and $i = 2$, the procedure is slightly different. In the following we calculate the probability of error for the error event of length four with $d_{s_i} = 2$ (see Fig.3.4). The accumulated metric difference for this error event can be written as

$$\Delta_p = -\mu_p - \mu_{p+1} - \nu_{p+1} - \mu_{p+2} - \nu_{p+2} - \mu_{p+3}$$

$$= -(r_p^T s_p^0 + r_{p+1}^T s_{p+1}^0) - (r_{p+1}^T s_{p+1}^0 + r_{p+2}^T s_{p+2}^1)$$

$$-(|r_{p+1}^T s_{p+1}^0|) - (r_{p+2}^T s_{p+2}^0 + r_{p+2}^T s_{p+2}^1)$$

$$-(|r_{p+2}^T s_{p+2}^0|) - (r_{p+3}^T s_{p+3}^0 + r_{p+3}^T s_{p+3}^1).$$

(3.33)

In the above, two independent random variables

$$N_{p,4} = -|P_{p+i} + n_{p+i}^T s_{p+i}^0| - (P_{p+i} + n_{p+i}^T s_{p+i}^0)$$

(3.34)

and

$$N_{p,5} = |n_{p+i}^T s_{p+i}^1| - n_{p+i}^T s_{p+i}^1$$

(3.35)
occur along with (some of) the other distance terms \(N_{p,1}, |N_{p,2}|, -|N_{p,3}|\) in the metric difference \(\Delta_p\). Denoting \(u(\cdot)\) as the unit-step function, \(N_{p,4}\) and \(N_{p,5}\) can be written as

\[
N_{p,4} = -2(P_{p+i} + P_{p+i}^T s_{p+i}^0)u(P_{p+i} + P_{p+i}^T s_{p+i}^0) ,
\]

\[
N_{p,5} = -2n_{p+i}^T s_{p+i}^1 u(-n_{p+i}^T s_{p+i}^1) .
\]

The Laplace transforms of the pdfs of \(N_{p,4}\) and \(N_{p,5}\) are obtained as

\[
\Phi_{N_{p,4}}(s) = e^{2P_{p+i}s(1+\sigma_n^2)}Q\left(1 + 2\sigma_n^2\sqrt{\frac{P_{p+i}}{\sigma_n^2}}\right) + Q\left(\sqrt{\frac{P_{p+i}}{\sigma_n^2}}\right)
\]

and

\[
\Phi_{N_{p,5}}(s) = e^{2P_{p+i}s\sigma_n^2}Q\left(2\sigma_n \sqrt{P_{p+i}\sigma_n^2}\right) + \frac{1}{2}
\]

respectively, thus the PEPs for the error events involving \(-s_{p+i}^1\) can also be numerically evaluated using (3.32).

The PEPs for the \(L\) dominant error events are used in the truncated union bound (3.16) and the individual BER approximations are averaged with respect to the spreading sequences according to (3.17).

### 3.5 BER Numerical and Simulated Results

In this section we present the numerical and simulated results for both decoding metrics to show the good convergence of approximations which are a valuable tool for quick performance evaluation. In addition we compare the performance of the two decoding metrics. For both analysis and simulation, we have used the UWB impulse responses defined in Section 2.2 according to the IEEE 802.15.4a channel model generated by the MATLAB code provided in [12]. As an interesting example, Channel Model 2 (CM2) is assumed, which represents residential non-line-of-sight environments. The average
signal-to-noise ratio (SNR) for each channel is given by

\[ \frac{E_b}{N_0} = \frac{1}{2^{15} - 1} \sum_{k=1}^{2^{15}-1} \frac{P_k}{2\sigma_k^2}. \]  

(3.40)

After chip-matched filtering SRAKE combining is applied by selecting the \(L_s\) strongest paths, assuming perfect channel estimation. If all paths are selected, i.e., \(L_s\) equals the number of all resolvable paths, SRAKE combining becomes ARAKE combining.

### 3.5.1 BER Performance for Symbol-wise and Bit-wise Decoding Metrics

In this section, we compare the BER performances for convolutional BPSK/BPPM decoding with symbol-wise and bit-wise metrics, using ARAKE and SRAKE combining. For this purpose we evaluate the BER expressions derived in Section 3.4 and present the results together with the simulated BERs.

For numerical BER evaluations we include \(L = 15\) dominant error events when considering symbol-wise metrics and \(L = 4\) dominant error events for decoding with bit-wise metrics. The smaller number in the latter case is motivated by the fact that the contributions from PEPs for longer error events to the union bound diminish very quickly for decoding with bit-wise metrics.

Figure 3.5 compares the average BER performances of the coded BPSK/BPPM, being decoded with symbol-wise and bit-wise metrics over 100 channel realizations. Both analytical and simulation results are shown, and ARAKE combining is performed. It can be seen that the analytical (lines) and simulated (markers) BER curves match very well and thus corroborate the validity of the BER approximations developed in Section 3.4. It is worth pointing out that while the BER simulation takes approximately 4-5 hours per channel realization for reaching a BER \(\approx 10^{-5}\), the analytical results, for example, for the symbol-wise decoding metric are obtained in less than a minute per
channel realization for the same number of $E_b/N_0$ points. Thereby, the amount of time consumed for simulation results increases dramatically with lower target BERs, whereas the calculation time of the analytical results is independent of the $E_b/N_0$ and thus of the target BER.

We further observe that the use of bit-wise metrics incurs a considerable loss in power efficiency of, for example, 2.2 dB at a BER of $10^{-4}$, which increases for lower target error rates. It should be noted that decoding with symbol-wise metrics requires about the same computational complexity as decoding with bit-wise metrics that is two correlations and addition of eight branch metrics per BPSK/BPPM symbol are required (see equations (3.3), (3.6) and (3.7)). Therefore, we conclude that symbol-wise metrics should be applied rather than bit-wise metrics suggested in [7].

### 3.5.2 RAKE Receiver

Next, we investigate the effect of SRAKE combining with a fixed number $L_s$ of fingers. Figure 3.6 shows the analytical (lines) and simulated (markers) BER results for SRAKE combining with $L_s = 12$ and $L_s = 24$ fingers. As a reference, the results for ARAKE combining are included. Once again, the analytical approximations and the simulation results match very well. We observe that SRAKE with $L_s = 24$ achieves a performance very close to that of ARAKE, while degradations of about 1-2 dB occur for $L_s = 12$. This performance penalty due to SRAKE combining is made more explicit in Fig. 3.7, where the difference in SNR between ARAKE and SRAKE combining for a target BER of $10^{-4}$ is plotted as a function of $L_s$. While with only $L_s = 5$ fingers the ARAKE performance is approached within about 2 dB, rather large $L_s (> 20)$ are needed to make the performance loss incurred by SRAKE combining disappear. We note, however, that CM2 models a residential non-line-of-sight environment with relatively many multi-path components and that other environments [13] may be less demanding.
3.5.3 Effect of Time-varying Scrambling Sequence

The effect of time-varying spreading sequences on the BER is illustrated in Fig. 3.8 for one CM2 channel realization and ARAKE combining. The ten BER curves correspond to BER\((p)\) (see Eq.\((3.16)\)) for ten different, rather arbitrarily chosen trellis positions \(p = 2, 7, 12, \ldots, 47\). Apparently, the variation of the effective channel energy \(P_k\) due to different spreading sequences are quite significant, since the "local" error rates \(BER(p)\) range over several orders of magnitude. Hence, this effect must not be neglected for
performance evaluation, whether simulative or analytical.

In considering the effect of the scrambling sequence, it is interesting to look at the system performance without scrambler. For this purpose, we fix the scrambler coefficients to $c_i = -1$. Figure 3.9 shows the simulated BER with disabled scramble in comparison to the normal scrambled operation for symbol-wise decoding metric; a performance degradation of approximately 0.6 dB is evident. It should be noted that we have not considered the multi-user interference in this work. We expect that in a multi-user scenario the performance gain due to scrambling is considerably larger.
Figure 3.7: SNR(ARAKE)−SNR(SRAKE) (in dB) required for a BER of $10^{-4}$ for SRAKE combining with different $L_n$. Decoding with symbol-wise metrics. Average over 100 CM2 channels. Analytical results.
Figure 3.8: BER(p) [Eq.(3.16)] for ten different trellis positions $p$. One CM2 channel realization and ARAKE combining. Decoding with symbol-wise metrics.
Figure 3.9: Simulated BER for ARAKE combining with and without time-varying scrambler for symbol-wise metric.
Chapter 4

Concatenated Coded BPSK/BPPM

In this chapter, we consider the decoding and analysis of the concatenated BPSK/BPPM scheme specified for the IEEE 802.15.4a standard as described in Chapter 2. Since the early introduction of concatenated coding schemes by Forney [20], combinations of two error correcting codes have been vastly used in communication systems from space application to data transfer and multimedia to improve error correction capability of the system [10]. In order to improve the performance of concatenated systems of inner convolutional and outer RS codes, the passing of reliability information from the inner decoder to the outer one and iterations between them have been considered in [21, 22]. The outer RS decoder can use the reliability information to perform erasures-and-errors decoding, which improves the error-correction capability if indeed errors of the inner decoder are transformed into erasures. In this context, it is desirable to align the reliability output from the inner decoder with the boundaries of the RS symbols [21]. As for the iterative decoding, it is typically assumed that the convolutional and RS codes are connected through an interleaver and that the corresponding de-interleaving at the receiver side distributes error bursts from the inner decoder over multiple RS codewords [21, 22]. This is unfortunately not the case in the standardized UWB sys-
tem. Considering the above, we propose an improved decoding procedure that uses the soft-outputs generated by a modified Viterbi Algorithm (VA). Two modified Viterbi decoders with soft output reliability information are described in Section 4.1. The entire decoding scheme using inner code soft output reliability information is presented in Section 4.2. Based on the BER analysis of the inner convolutional code in Chapter 3, we derive analytical results for the frame error rate (FER) performance of the entire concatenated coding scheme in Section 4.3. The analytical and simulation results are presented and discussed in Section 4.4.

4.1 Soft Output Viterbi Algorithm (SOVA)

The VA as the optimum decoder for convolutional codes accepts soft decisions from the channel and generates hard output data. For decoding a concatenated code with inner convolutional code, a so-called soft output Viterbi algorithm (SOVA) is necessary. The SOVA generates soft output, i.e., reliability information, for the decoded bits, along with hard decoding decision. There are several versions of the SOVA which have been introduced in the literature, e.g. [23, 24, 25]. Since the quality of soft outputs is directly related to decoding performance of RS erasures-and-errors decoding following the SOVA, we compare two methods of generating soft output, which mainly differ in the way the reliability output from the inner decoder is aligned with the boundaries of the RS symbols. The first one is adopted from the work by Huber and Rüppel [23] which directly matches the reliability information to the sequence of RS-symbols. The second method is based on work by Hagenauer and Hoher [24] and generates bit-based soft outputs, which are then mapped to RS-symbols reliabilities in a separate, sub-optimum step. The two SOVAs are briefly described in the following two sections. It should be noted that both SOVA implementations in the following sections and the results presented in Section 4.4, use the symbol-wise decoding metric and the terms
symbol-based and bit-based only refer to the two different SOVAs.

4.1.1 Symbol-based SOVA

The symbol-based SOVA devised in [23] calculates the probability of error based on the difference in the metric of the two paths merging at every state node and updates these probabilities for the RS symbol to which the bit information belongs. The hard decision Viterbi decoding procedure stays intact but the symbol-wise metric for which we implement the SOVA, needs to be modified. Considering the equivalent AWGN channel model from Eq.(3.1), the log-likelihood probability is obtained as

$$\log (p(r_k|a,b)) = -\frac{||r_k - a s_k^b||^2}{\sigma_n^2} + \frac{1}{2}\log \left( \frac{1}{\pi \sigma_n^2} \right)$$  

The reliability difference measure only depends on $(2/\sigma_n^2) a^T r_k s_k^b$. Therefore, the previously defined symbol-wise metric notation is multiplied by the factor $2/\sigma_n^2$.

In the following we consider the modified decoding steps for two survivor paths, A and B, merging to produce a new path C at time k. Without loss of generality, we assume that A is the survivor path when C is produced. Denoting $\Lambda_k^X$ as the partial path metric ending in trellis state X and $\Lambda_k^{XY}$ as the branch metric for the branch from X to Y at epoch k, the decoder will select the path with maximum accumulated metric, A, as survivor and the reliability difference is calculated as

$$\Delta_k^C = |\Lambda_k^A + \Lambda_k^{AC} - (\Lambda_k^B + \Lambda_k^{BC})|.$$  

The probability that the decision for path A over path B was correct is

$$p_k^C = \frac{1}{1 + e^{-\Delta_k^C}},$$  

where $p_k^C$ varies between 0.5, if $\Delta_k^C \approx 0$, and 1, for large values of $\Delta_k^C$. Furthermore, we introduce the vectors

$$d_k^X = [d_k^X[0], d_k^X[1], \ldots, d_k^X[\kappa]]$$
$$P_k^X = [P_k^X[0], P_k^X[1], \ldots, P_k^X[\kappa]]$$
which contain the \( \kappa = [k/m] \) RS symbols associated with the partial path to state \( X \)
and their reliability after decoding until time \( k \), respectively. If \( k \) is not a multiple of
\( m \), then \( d_k^X[\kappa] \) represents a partial RS symbol. When two paths \( A \) and \( B \) merge to
produce the new path \( C \), where we assume that \( A \) is the survivor, the reliability vector
\( P_{k+1}^C \) is updated as [23]

\[
P_{k+1}^C[l] = \begin{cases} 
P_k^A[l] \cdot P_k^C, & \text{if } d_k^A[l] \neq d_k^B[l] \\
P_k^A[l], & \text{if } d_k^A[l] = d_k^B[l] 
\end{cases}, \quad 0 \leq l \leq \kappa .
\]

(4.4)

Note that \( P_k^A[\kappa] \) is initialized by 1 at the beginning of the symbol.

### 4.1.2 Bit-based SOVA

The bit-based SOVA was introduced in [24]. Considering the two paths defined in
Section 4.1.1 and the reliability difference, \( \Delta_k^C \), defined in Eq.(4.2), the probability of
selecting the wrong survivor path is

\[
\bar{p}_k^C = 1 - p_k^C = \frac{1}{1 + e^{\Delta_k^C}},
\]

(4.5)

where \( p_k^C \) approaches 0.5 if \( \Delta_k^C \approx 0 \) and 0 if \( \Delta_k^C \gg 1 \). The \( \bar{p}_k^C \) is introduced to
keep the equations consistent with [24]. Under the assumption that path \( A \) has been
selected, the Viterbi decoder has made errors with probability \( \bar{p}_k^C \) in all the \( e_b \)
positions where the information bits of path \( B \) differ from path \( A \). The two vectors
\( d_k^X = [d_k^X[0], d_k^X[1], \ldots, d_k^X[k]] \) and \( P_k^X = [P_k^X[0], P_k^X[1], \ldots, P_k^X[k]] \) are defined differently now as the decoded binary information bits and the probability of previous erroneous decisions with the path to state \( X \) up to decoding instant \( k \). The probabilities of erroneous decisions are updated according to [24]

\[
P_{k+1}^C[l] = \begin{cases} 
P_k^A[l] \cdot (1 - \bar{p}_k^C) + (1 - P_k^A[l]) \cdot \bar{p}_k^C, & \text{if } d_k^A[l] \neq d_k^B[l] \\
P_k^A[l], & \text{if } d_k^A[l] = d_k^B[l] 
\end{cases}, \quad 0 \leq l \leq k .
\]

(4.6)
where $P_k^c[l]$ varies in the range $0 \leq P_k^c[l] \leq 0.5$ and $P_k^c[l] \approx 0$ represents a highly reliably decoding decision and vice versa. The vector $P_k^c$ is initialized as zero for all the four survivor paths propagating through the trellis at the beginning of the decoding process. The generated soft-output values belong to information bits and they are mapped to RS symbols by selecting the maximum bit reliability within a symbol as the symbol reliability [10].

The two SOVAs differ in the way the reliability information are updated as well as the mapping of reliability information to bits.

### 4.2 Decoding of the Concatenated Code

We now turn to the decoding of the concatenated code, using the SOVA as inner decoder. Note that the conventional VA is a special case of the SOVA, if the extra reliability information is ignored by the RS decoder. The hard decoded bits from the SOVA are converted into symbols to be fed to the RS decoder. Each RS symbol represents $m = 6$ binary symbols. The maximum error correcting capability of the RS code is $t = [(d_{\text{min}} - 1)/2] = 4$ RS-symbols for an error only decoding scheme, where $d_{\text{min}} = 9$ is the minimum distance of the RS code. In addition to error decoding, the RS decoder can perform erasures-and-errors decoding when the location of some errors are known. Denoting the number of declared erasures as $e$, the error correcting capability of the code is changed to $[(d_{\text{min}} - 1 - e)/2]$. Therefore, every two declared erasure positions reduce the number of correctable unknown errors by one. On the other hand, if erasure declared error-positions correspond to actual error positions, then the error correcting capability is improved at most twice.

Among several different algorithms for decoding RS codes, the Berlekamp-Massey Algorithm (BMA) [26] is the most widely used. The BMA is an iterative decoding
algorithm that computes the error-syndrome polynomial and then iteratively searches for the minimum linear feed-back shift register (LFSR) polynomial that can reproduce the error-syndrome polynomial. The roots of the LFSR polynomial are equal to the inverses of error locations. For details about the BMA we refer to [11].

In this work, we have both implemented the BMA for RS decoding, and have as well considered the FER simulation by counting the number of symbol errors after SOVA decoding: If this number exceeded \( [(d_{\text{min}} - 1 - e)/2] \), a frame error was declared.

Based on the soft output of the SOVA decoder, the e RS-symbols with the lowest probabilities in symbol-based SOVA and highest probabilities in bit-based SOVA, are declared as erasures and the RS decoder applies erasures-and-errors decoding. Since the choice of e is critical, we suggest to perform several decoding attempts with the RS decoder for different even numbers of erasures e, \( 0 \leq e \leq d_{\text{min}} - 1 = 8 \), until the correct information sequence is found. For the (63,55)-RS code the probabilities of decoding error can be defined as the probabilities that an incorrect codeword is decoded if more than \( [\frac{d_{\text{min}} - 1 - e}{2}] \) RS-symbol errors occur after Viterbi decoding and considering the erasures. These probabilities are 0.03, 0.14, 0.46, 0.97 and 1.00 for e equal to 0, 2, 4, 6 and 8, respectively [27]. Thus we cannot rely on error-only detection by the RS decoder. Instead we assume that the cyclic redundancy check (CRC) code applied to the data stream at the medium access control layer (MAC) can be utilized for this task. We further note that, unlike the systems considered in [21, 22], feedback from the RS decoder output to the Viterbi decoder cannot improve the performance due to the absence of interleaving between the convolutional and RS encoders.
4.3 FER Analysis for Decoding of the Concatenated Code

In this section, we present analytical expressions for the FER of the concatenated coded scheme. For analytical tractability, we assume only error decoding, i.e., no reliability information is exploited by the RS decoder.

The error-rate analysis for concatenated convolutional and RS coded systems is typically based on the symbol-error rate (SER), that is the probability of at least one error in a sequence of \( m \) consecutive bits, at the output of the Viterbi decoder assuming (ideal) symbol de-interleaving before RS decoding [28], or the use of a Markov chain to model the errors produced by Viterbi decoding and finite de-interleaving [29]. Since the considered concatenated coded system does not employ an interleaver, the bit and consequently symbol errors are no longer independent. Therefore error-gap distribution measures can not be used for calculation of the SER. On the other hand, given the relatively simple trellis-structure of the convolutional code, a different approach can be used to analyze Viterbi and RS decoding jointly.

To this end, as in Section 3.4, the analysis is restricted to the dominant error events of the Viterbi decoding. Furthermore, a simplifying assumption is made such that the error events do not extend over more than two adjacent RS symbols. This procedure can be extended further to include more error events affecting more than \( 2m \) bits. In spite of the simplifying assumption, the calculated FER gives a close match with the performance of the simulated system. Hence, it is reasonable to keep the assumption and reduce the number of calculations. With this assumption, it suffices to distinguish the two cases that an error event falls completely within an RS code symbol and that it extends over two adjacent RS symbols, which are referred to as single-symbol errors and double-adjacent-symbol errors, respectively.
Denoting the RS symbol time-index by \( k \) and considering, as in Section 3.4, an error event \( c_i \) starting at \( k = \kappa m + p \), \( 0 \leq p < m \), and terminating at \( k = \kappa m + q + 1 \), which corresponds to \( q-p+1 \) BPSK/BPPM symbols, the probability of single-symbol errors can be upper bounded by

\[
\text{SER}_1(\kappa) = \sum_{p=0}^{m-1} \sum_{d_a=1}^L \sum_{i = \kappa m + p, q < \kappa m + q + 1} P_2(c_0 \rightarrow c_i),
\]

where the inner sum is such that the error event starts at time \( \kappa m + p \), traverses along \( d_a \) symbols \( (\pm s^1_k) \), and is not longer than \( m \) symbol intervals. The pairwise error event probability is the same as the results obtained in Section 3. Likewise, the probability for double-adjacent-symbol errors can be upper bounded by

\[
\text{SER}_2(\kappa) = \sum_{p=0}^{m-1} \sum_{d_a=1}^L \sum_{i = \kappa m + p, m \leq q < 2m} P_2(c_0 \rightarrow c_i).
\]

For the purpose of FER calculation, all combinations of \( e_1 \) single- and \( e_2 \) double-adjacent-symbol errors with \( e_1 + 2e_2 > t \) would need to be considered for all \( \left( \frac{n}{e_1+e_2} \right) \) possible positions within the RS codeword, which is clearly infeasible. Here \( n \) is the length of the RS codeword and is equal to 63. Also the effect of time-varying spreading sequence has to be taken into account. Therefore, averages of \( \text{SER}_1(\kappa) \) and \( \text{SER}_2(\kappa) \) are calculated first with respect to \( \kappa \) and then the FER is determined based on the averaged single- and double-adjacent-symbol error probabilities. In doing so, the time-varying spreading is taken into account for calculating the symbol-error probabilities, but its effect on the probability of multiple symbol errors is neglected. Since the period of the spreading sequence with respect to RS symbols is \( 2^{15} - 1 \), the average symbol-error probability becomes:

\[
\overline{\text{SER}}_\nu = \frac{1}{2^{15} - 1} \sum_{\kappa=0}^{2^{15}-1} \text{SER}_\nu(\kappa), \quad \nu = 1, 2.
\]

The RS decoder can correct up to \( t \) symbol errors, which could result from single- and
double-adjacent symbol errors. The probability of having \( j \) double-adjacent-symbol errors and up to \( t - 2j \) single-symbol errors can be formulated as

\[
P_{e,j} = \binom{n-1}{j} (\text{SER}_2)^j (1 - \text{SER}_2)^{n-1-j} \sum_{i=0}^{t-2j} \binom{n}{i} (\text{SER}_1)^i (1 - \text{SER}_1)^{n-i}. \tag{4.10}
\]

Since \( t = 4 \) for the (63,55)-RS code, the FER is finally obtained as

\[
\text{FER} = 1 - \sum_{j=0}^{2} P_{e,j} \tag{4.11}
\]

We note that the FER calculation is possible for both symbol-wise and bit-wise decoding metrics.

As a special case of Eqs.(4.10) and (4.11), we can calculate the FER at the output of Viterbi decoder. Since, the probability of correct decoding is the probability of having no single- and no double-adjacent symbol error, the FER at the output of the Viterbi decoder is given by

\[
\text{FER}_{cc} = 1 - P_{e,0}, t=0 = 1 - [(1 - \text{SER}_2)^{n-1} (1 - \text{SER}_1)^{n}]. \tag{4.12}
\]

### 4.4 Performance of Concatenated BPSK/BPPM

In this section, we evaluate the FER expression derived above and compare the results with FER simulations. Furthermore, we present the results for error-and-erasures decoding, making use of the reliability information from the different SOVA implementations.
4.4.1 FER Using Symbol-wise Decoding Metric

The analytical results were calculated using the probabilities of the $L$ dominant errors previously obtained in Section 3.4.1 in Eqs.(4.7) and (4.8). The effect of the number of error events on the accuracy of the analytical results can be observed in Fig. 4.1 for the symbol-wise decoding metric using $L = 4, 7, 15$ dominant error events. It is shown that use of $L = 15$ error events results in close match with simulation. Figure 4.2 shows the analytical and simulation results for three different channel realizations, ARAKE combining is assumed. We observe the very good match between analytical and simulation results (note the scaling on the x-axis). An even earlier (at lower $E_b/N_0$) convergence of analytical and simulated curves is obtained for higher performance channels. The time required to obtain simulation results for the eight $E_b/N_0$ points with $\text{FER} > 10^{-4}$ using symbol-wise decoding metric was approximately 24 hours per channel realization while the calculated analytical results were still obtained in less than a minute. The simulation time extends to more than 24 hours for a single $E_b/N_0$ at $\text{FER} \leq 10^{-4}$. This motivates use of analytical approximations over simulation.

Figure 4.3 compares the FER at the output of the Viterbi decoder in the first decoding stage and the FER resulting from the second stage RS-decoder. The calculated FER at the output of Viterbi decoder in Eq.(4.12) is validated by observing the close match between the simulated and analytical results. The error correction capability of the RS-decoder provides a performance gain of approximately 2 dB at $\text{FER} = 10^{-1}$, which increases for lower FERs. The analytical average FER over 100 channels are shown for both convolutional and RS decoder. We note that obtaining simulation results for 100 channel realizations would require more than 100 days if the simulations were run serially for the 100 channels, this is totally infeasible in terms of resources, and therefore we can not show the simulation results for this case. Instead, we use the analytical results to calculate the average FER over 100 channels in less than an hour. The average FER over 100 channels is very close to the single channel realization result
that is because the FER curves for the 100 channel realizations are very similar.

4.4.2 FER Using Bit-wise Decoding Metric

Similar to the results presented for symbol-wise decoding metric, the analytical results are calculated using the probabilities of the $L$ dominant errors previously obtained in Section 3.4.2 in Eqs. (4.7) and (4.8). Figure 4.4 shows the effect of number of error events for $L = 4, 7, 15$ using bit-wise decoding metric. It is shown that using the probabilities of error for 15 error events produces results that match very well with the simulation. Including a higher number of error events in the calculation provides more accurate result at the cost of higher calculation complexity. This is significantly important in bit-wise decoding metric, since analytical calculations involve enumeration of complex
Figure 4.2: FER for ARAKE combining for 3 channels. Lines: Analytical results. Markers: Simulation results.

Figure 4.3: FER and FER$_{cc}$ for ARAKE combining for single channel realization. Lines: Analytical results. Markers: Simulation results.
Figure 4.4: Effect of number of paths on calculated FER in Bit-wise decoding metric. Lines: Analytical results. Markers: Simulation results.

valued Q-function and numerical integration.

Figure 4.5, compares the FER performances of the concatenated coded BPSK/BPPM, being decoded with symbol-wise and bit-wise decoding metrics for one channel realization. Both analytical (lines) and simulation (markers) are shown. Again, similar to the BER results presented in Section 3.5.1, it is observed that the use of bit-wise metrics incurs a considerable loss in power efficiency of, e.g. 1.7 dB at BER of $10^{-2}$. Since the Viterbi decoding procedures for the two decoding metrics involve same number of calculations, considering the BER results in Section 3.5.1 and FER results in Fig. 4.5, we conclude that symbol-wise metrics should be used instead of bit-wise metric.
4.4.3 Improved Decoding Scheme with Symbol-based SOVA

Next, we present FER results with erasures-and-errors RS decoding, which utilizes reliability information provided by the SOVA. In this section, we consider the symbol-based SOVA, introduced in Section 4.1.1. Since an analysis of erasures-and-errors decoding seems infeasible, we resort to showing simulation results throughout this section. As mentioned before, the generation of reliability information requires the application of symbol-wise decoding metrics.

First, it is insightful to consider whether the SOVA soft output represents a faithful reliability information. For this purpose, Fig. 4.6 shows the reliability information $P_k^X$ defined in Section 4.1.1 versus the probability of correct decoding for the RS-symbol $d_k^X$ associated with $P_k^X$. In other words, the x-axis represents the value of the reliability information of RS-symbol $d_k^X$ estimated by SOVA and the y-axis represents the actually
measured reliability of the decoded symbol. Figure 4.7 shows $P_k^X$ versus probability of correct decoding for the RS-symbol $d_k^X$ in the range of $0.9 \leq P_k^X \leq 1$ with a higher resolution on the x-axis. The approximately linear relation (with slope 1) shows that the reliability information $P_k^X$ estimated by the SOVA is almost identical to the actually measured reliability for RS-symbol $d_k^X$.

Next, we consider the FER performance of erasures-and-errors RS decoding. Iterative erasures-and-errors decoding was suggested in Section 4.2. In this method, additional erasures are declared if there exist more errors after decoding with previously declared erasures. Figure 4.8 shows the simulated result of such scheme for a single channel realization. Based on the reliability of the generated soft output values, increasing the number of declared erasures $e$ enhances the performance. This is the case even for $e = 8$ at which point, the decoder is an erasure-only RS decoder. The FER at the output of the Viterbi decoder is shown for better comparison.
Figure 4.7: Probability of correct decoding for reliability information in the range 0.9 to 1 for symbol-based SOVA.

Figure 4.8: FER vs $E_b/N_0$ for the iterative erasures-and-errors decoding with a maximum of $e$ erasures. Symbol-based SOVA. One channel realization.
When it is not possible to use the CRC code for error detection for iterative erasures-and-errors decoding, we suggest picking a fixed number of erasures that results in the best decoding performance in most of the cases. Figure 4.9 shows the simulated FER for the non-iterative decoding scheme, in which a fixed number of erasures are declared for all frames. It can be observed that declaring two to four erasures can result in moderate gain. However, if more than four erasures are declared, the performance degrades again. This is due to the fact that, if the erased bits do not correspond to the actual errors, the overall error-correction capability of the RS-decoder is reduced.

4.4.4 Improved Decoding Scheme with Bit-based SOVA

In this section, we consider erasures-and-errors decoding with the bit-based SOVA introduced in Section 4.1.2. Figures 4.10 and 4.11 show the FER for iterative and
Figure 4.10: FER vs $E_b/N_0$ for the iterative erasures-and-errors decoding with a maximum of $e$ erasures. bit-based SOVA. One channel realization.

non-iterative erasures-and-errors decoding. Similar to the symbol-based SOVA, the performance gain increases with increasing number of declared erasures for iterative decoding. In the non-iterative decoding scheme, again similar to the symbol-based case, two and four declared erasures yield a performance improvement. Hence, we suggest declaring $e = 4$ erasures in cases where reliable error detection and therefore, iterative erasures-and-errors decoding is not possible.

Finally, figures 4.12 and 4.13 compare the performances with the two SOVAs when declaring 2 and 8 erasures in iterative and non-iterative decoding, respectively. Perhaps surprisingly, we observe that the performance merit of using symbol-based soft outputs instead of bit-based soft output is quite insignificant.
Figure 4.11: FER vs $E_b/N_0$ for the non-iterative erasures-and-errors decoding with $e$ erasures. Symbol-based SOVA. One channel realization.

Figure 4.12: FER vs $E_b/N_0$ for the iterative erasures-and-errors decoding with a maximum of $e$ erasures. Dotted: symbol-based SOVA, dashed: bit-based SOVA. One channel realization.
Figure 4.13: FER vs $E_b/N_0$ for the non-iterative erasures-and-errors decoding with $e$ erasures. Dotted: symbol-based SOVA, dashed: bit-based SOVA. One channel realization.
Chapter 5

Non-Coherent BPPM Detection

The IEEE 802.15.4a standard strongly emphasizes on simple transmitter and possible simple receiver structure with low complexity and low cost. In cases where complexity and cost do not allow implementing a reliable channel estimator, non-coherent detection of the BPPM symbol can be performed. More specifically, since the convolutional encoder is systematic, the data can be recovered using a simple energy detector with subsequent RS decoding. Section 5.1 describes such a non-coherent detection of the BPPM IR-UWB signal using square-law RAKE combining. The BER and FER analysis for this detector are given in Sections 5.2 and 5.3, respectively. For completeness, the error-rate expansion for coherent detection is stated in Section 5.4. Finally, analytical and simulated performance results are presented and discussed in Section 5.5.

5.1 Non-Coherent Detection

Non-coherent receivers are generally low cost and simple compared to the coherent receivers. However, non-coherent receivers exhibit a significant drop in performance compared to coherent receivers. A popular choice for non-coherent reception of BPPM
signals is energy detection through auto-correlation or RAKE receivers [16, 30]. In the auto-correlation method, the received signal is correlated with itself or a previously received one. This is often done based on the analog received signal. In case of non-coherent RAKE-based reception, the entire received signal is not used for energy detection, but only samples at certain delays are considered. Since channel estimation required for optimal maximum-ratio combining is not available, suboptimal square-law combining (SLC), which does not require knowledge of the channel information, is often applied. In SLC the multi-path components of the received vector are squared and summed to form the decision variables for each of the two possible pulse positions. This form of energy detection is highly dependent on the tap selection. For simplicity, we assume that the taps corresponding to the $L_s$ strongest resolvable paths are selected at the receiver, i.e., the receiver has some knowledge about the energy-delay profile of the channel.

Assuming again the chip-matched filtering front-end introduced in Section 2.3, and defining $L_t = T_s/T_c$, the discrete time equivalent channel model

$$r_k = s_k^0 + n_k, \quad (5.1)$$

results, where $r_k = [r_{k,1}, r_{k,2}, \ldots, r_{k,L_s}]$ is the vector of received samples at the output of RAKE receiver, $n_k = [n_{k,1}, n_{k,2}, \ldots, n_{k,L_s}]$ is the vector of noise samples with variance $\sigma_n^2$, and $s_k^0 = [s_{k,1}, s_{k,2}, \ldots, s_{k,L_s/2}, 0, \ldots, 0]$ and $s_k^1 = [0, \ldots, 0, s_{k,1}, s_{k,2}, \ldots, s_{k,L_s/2}]$ represent the BPPM data bits. The data bit samples are given by

$$s_{k,i} = \frac{p_k(t) * h(t) * g_c(t)|_{t=t_i} r_c}{\sqrt{g_c(t) * g_c(-t)}}. \quad (5.2)$$

The SLC selects the $L_s$ samples $r_{k,i}$ corresponding to the largest $|s_{k,i}|$. Specifying the corresponding sample incidences by $l(i), i = 1, 2, \ldots, L_s$, the SLC decision variable follows as

$$d_k = \sum_{i=1}^{L_s} (r_{k,i})^2 - \sum_{i=1}^{L_s} (r_{k, L_s/2 + l(i)})^2. \quad (5.3)$$
If \( d_k \geq 0 \), the BPPM data bit is decided as 0, and if \( d_k < 0 \), the data bit is decided to be 1.

### 5.2 BER Analysis of Non-Coherent BPPM

In this section, we derive an analytic expression for the BER at the output of the SLC. To this end, without loss of generality, we assume that \( s_k^0 \) was transmitted. The probability of error can be written as

\[
P_b = P_t \left\{ d_k < 0 \right\} = P_t \left\{ \sum_{i=1}^{L_s} (-1)^{k(i)} n_{k(i)}^2 - (n_{k(i)})^2 < 0 \right\},
\]

(5.4)

The above equation can be easily mapped to a quadratic form of Gaussian random variables

\[
D = \sum_{n=1}^{L_s} A|X_i|^2 + B|Y_i|^2 + CX_iY_i^* + C^*X_i^*Y_i,
\]

(5.5)

where \( A, B, \) and \( C \) are constants and \( X_i \) and \( Y_i \), are mutually independent. In mapping the two equations, \( A = 1, B = -1, C = 0 \), \( X_i = r_{k(i)} \) and \( Y_i = r_{k(i) + 1} \). The solution of Eq.(5.4) for such quadratic form is given in [31, Appendix 9A] as

\[
P_b = \frac{1}{2} + \sum_{n=0}^{L_s - 1} \left( \frac{2L_s - 1}{L_s - 1} \right) \left[ Q_t(a, b) - Q_t(b, a) \right],
\]

(5.6)

where \( a = 0, b = \sqrt{\frac{\sum_{i=1}^{L_s} (s_{k(i)})^2}{2\sigma^2_k}} \) and \( \sum_{i=1}^{L_s} (s_{k(i)})^2 = ||s_k^0||^2 = P_k \) in this case. The Marcum Q-function \( Q_t(a, b) \) for the special case of \( a = 0 \) can be written in series form as

\[
Q_t = (0, b) = \sum_{n=0}^{l-1} \exp \left( -\frac{b^2}{2} \right) \frac{(b^2/2)^n}{n!}.
\]

(5.7)

By substituting Eq.(5.7) in Eq.(5.6), the solution of the quadratic form in Eq.(5.4) can be written as

\[
P_b(k) = \frac{1}{2} + \sum_{l=1}^{L_s} \sum_{i=1}^{L_s} \left( \frac{2L_s - 1}{L_s - 1} \right) \left[ \sum_{n=0}^{l-1} \exp \left( -\frac{P_k}{4\sigma^2_n} \right) \frac{(-P_k/4\sigma^2_n)^n}{n!} - 1 \right].
\]

(5.8)
Since the normalized effective energy $P_k$ is time-varying, the scrambling sequence has direct impact on Eq.(5.8), which is emphasized by writing $P_k$ as a function of $k$. The average BER is then given as

$$\overline{\text{BER}} = \frac{1}{2^{15} - 1} \sum_{k=0}^{2^{15} - 1} P_k(k).$$

(5.9)

### 5.3 FER Analysis of Non-Coherent BPPM with RS-Decoding

In this section, we extend the analytical results obtained in Section 5.2 to calculate the FER at the output of the RS-decoder. For the BPPM transmission, since the errors are independent, the SER can be written as a function of average bit error probability $\overline{\text{BER}}$, more specifically, the SER is the complement of probability of having no error in any of the $m = 6$ bits within an RS-symbol and is given by

$$\overline{\text{SER}} = 1 - (1 - \overline{\text{BER}})^6.$$

(5.10)

Since the RS-decoder can correct up to $t = 4$ symbol errors within a frame, the probability of frame error is calculated as a sum over probabilities of having more than $t = 4$ errors in a frame

$$\overline{\text{FER}} = \sum_{i=5}^{63} \binom{63}{i} \overline{\text{SER}}^i (1 - \overline{\text{SER}})^{63-i}.$$

(5.11)

### 5.4 BER and FER for Coherent BPPM Detection

For completeness, we state the BER and FER expressions for coherent BPPM detection, i.e., RAKE with MRC is applied. For MRC the decision variable is given by

$$d_k = \sum_{i=1}^{L_x} r_{k,l,i} s_{k,l,i} - \sum_{i=1}^{L_x} r_{k,43+l,i} s_{k,43+l,i}.$$

(5.12)
Therefore, assuming $s_k^0$ is transmitted, the probability of error can be written as

$$P_b(k) = Q\left(\sqrt{\frac{\|s_k^0\|^2}{2\sigma_n^2}}\right) = Q\left(\sqrt{\frac{P_k}{2\sigma_n^2}}\right).$$

(5.13)

The SER for RS-symbols and the FER with RS-decoding are again given by equations (5.10) and (5.11) substituting $P_b(k)$ from Eq. (5.13).

### 5.5 Results and Discussion

In this section we present analytical and simulation results for non-coherent BPPM detection and compare them with those for coherent detection. For the results, $L(s) = 33$ is assumed.

Figure 5.1 shows the analytical and simulated BER results. We observe an excellent match between the analytical and simulated curves. This also justifies the suggested averaging over possible values of the effective normalized energy, $P_k$.

As mentioned before, the absence of channel information can cause a significant performance degradation for the BPPM transmission considered here, a loss of approximately 7 dB can be observed at BER $= 10^{-2}$.

Figure 5.3 shows the FER at the output of the RS-decoder. Due to the error correcting capability of the RS-code, the frame error rate decreases with a steep slope after 14 dB in the non-coherent and after 7 dB in the coherent case. At FER $= 10^{-2}$, coherent receiver has a considerable performance gain of approximately 6 dB.

The effect of number of fingers on performance of the RAKE receiver with square-law combining is explicitly shown in Fig. 5.2. The $E_b/N_0$ at which the target BER of $10^{-2}$ is obtained is plotted as a function of $L_s$ for one channel realization. As it can be seen, initially, increasing the number of fingers, the performance is improved and the target BER is achieved at lower $E_b/N_0$. However, at $L_s \approx 50$ the performance starts
Figure 5.1: BER for non-coherent and coherent detection with RAKE receivers with $L_s = 33$ fingers.

to degrade slightly. This is due to the fact that the effect of additional white Gaussian noise becomes dominant starting at $L_s \approx 50$. The performance loss highly increases for $L_s > 70$.

In conclusion, there are significant losses in power efficiency due to non-coherent decoding if channel information is not available. Hence, obtaining channel state information is highly desirable for the UWB system under consideration. In addition, to fully benefit from the concatenated coding scheme applied to the data at the transmitter, channel state information is needed.
Figure 5.2: FER for non-coherent and coherent detection with RAKE receivers with $L_s = 33$ fingers.

Figure 5.3: SNR required at BER $= 10^{-2}$ for SRAKE receiver with SLC combining with different $L_s$. 

63
Chapter 6

Conclusions and Future Work

6.1 Conclusion

In this thesis we have investigated decoding for UWB transmission systems complying with the IEEE 802.15.4a standard. The main contributions and results can be summarized as follows:

- The probability of error for the convolutional coded BPSK/BPPM system for decoding with symbol-wise and bit-wise metrics, taking into account the time-varying spreading prescribed by the standard, has been analyzed. The presented numerical and simulation results have shown that (i) the BER approximations are tight over a wide range of BERs, (ii) decoding with symbol-wise metrics is clearly advantageous in terms of performance-complexity trade off over bit-wise metrics, and (iii) SRAKE combining with about 5-10 fingers is sufficient to approach the optimal performance within 1-2 dB for highly dispersive UWB transmission channels.

- The performance of the concatenated coded BPSK/BPPM in the absence of
interleaving and the presence of time-varying spreading has been considered and analytical results for FER have been obtained. The analytically obtained FERs closely match the simulated results, which renders the derived expressions highly valuable for quick performance evaluation.

- Improved decoding schemes using erasures-and-errors decoding of the outer RS code have been suggested. The Viterbi decoding algorithm has been modified to produce soft output reliability values that are passed to the RS decoder for erasure declaration. Two different modified SOVA have been implemented, where one is based on assigning reliability information to RS-symbols and the other to bits. By means of simulation results, it has been found that (i) the two SOVAs show approximately the same FER (ii) using SOVA 0.4 dB gain can be obtained over the conventional decoding.

- Non-coherent detection of the UWB BPPM signal without channel estimation using a RAKE receiver with square-law combining has been investigated and the corresponding bit and frame error probabilities have been analyzed. A Comparison of the coherent and non-coherent receivers has shown that a performance degradation of approximately 6 dB is incurred by the absence of channel information in non-coherent receivers.

6.2 Recommendations on Future Works

We have investigated the performance of and have provided analytical tools for performance evaluation of the basic UWB system complying with the recently approved standard IEEE 802.15.4a. There are various extensions that could be considered. Perhaps most interestingly are

- To study the effect of channel estimation errors on the system performance in case
of coherent decoding using RAKE receiver and MRC combining with practical channel estimation.

- The inclusion of multi-user interference and its effect on the system performance.
Bibliography


