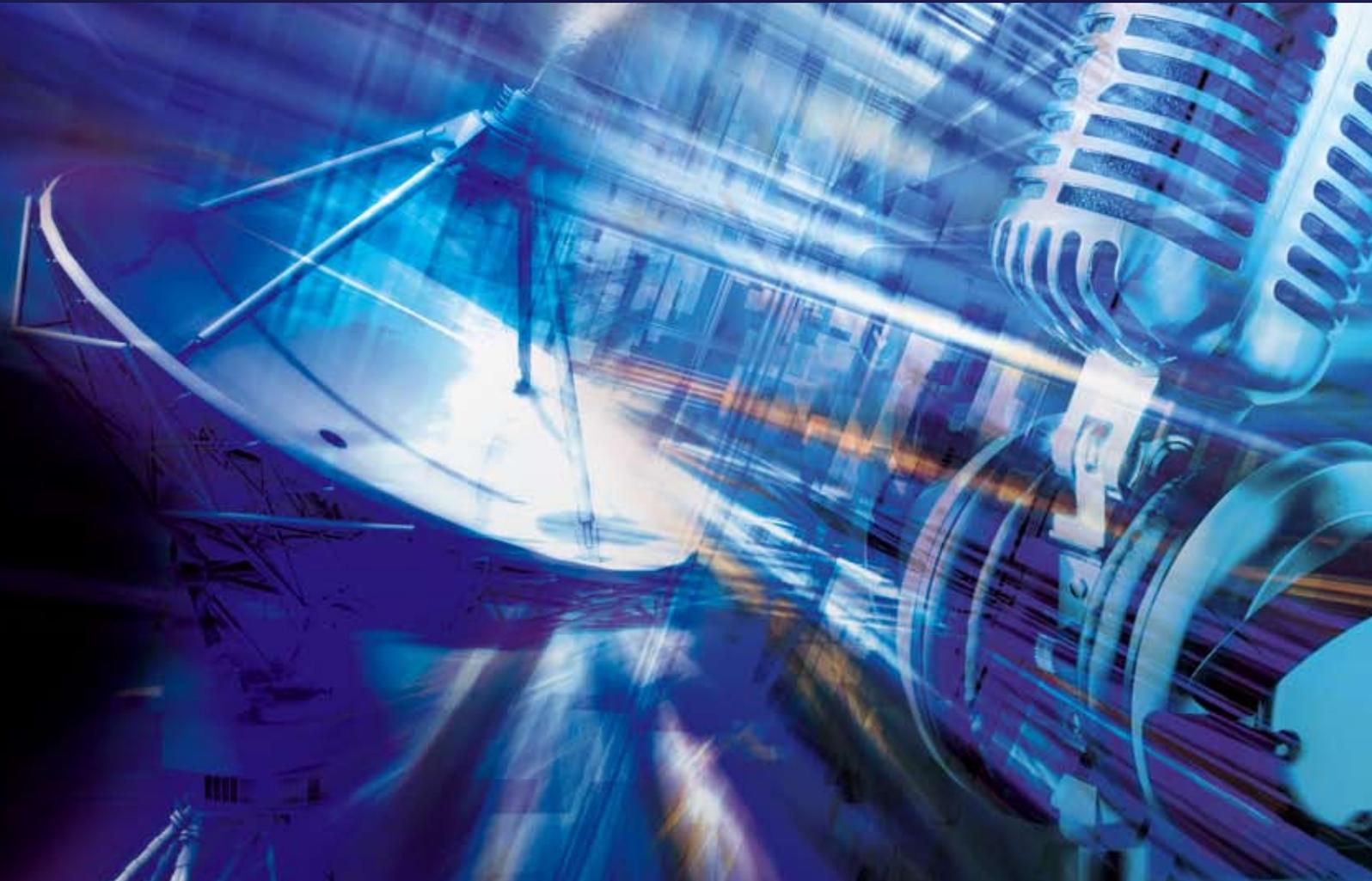


Iterative Decoding and Cross-Layering Techniques for Multimedia Broadcasting and Communications

Guest Editors: Fred Daneshgaran,
Massimiliano Laddomada, and Marina Mondin





Iterative Decoding and Cross-Layering Techniques for Multimedia Broadcasting and Communications

International Journal of Digital Multimedia Broadcasting

Iterative Decoding and Cross-Layering Techniques for Multimedia Broadcasting and Communications

Guest Editors: Fred Daneshgaran,
Massimiliano Laddomada, and Marina Mondin



Copyright © 2008 Hindawi Publishing Corporation. All rights reserved.

This is a special issue published in volume 2008 of "International Journal of Digital Multimedia Broadcasting." All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Editor-in-Chief

Fa-Long Luo, Element CXI, USA

Associate Editors

Sos S. Aghaian, USA
Jörn Altmann, Korea
Ivan V. Bajic, Canada
A. Bouzerdoun, Australia
Hsiao-Hwa Chen, Taiwan
Gerard Faria, France
Borko Furht, USA
Rajamani Ganesh, India
Jukka Henriksson, Finland
Shuji Hirakawa, Japan
Yu-Hen Hu, USA

Jiwu Huang, China
Jenq-Neng Hwang, USA
Daniel Iancu, USA
Thomas Kaiser, Germany
Dimitra Kaklamani, Greece
M. Kampmann, Germany
Alexander Korotkov, Russia
Harald Kosch, Germany
Massimiliano Laddomada, USA
Ivan Lee, Canada
Jaime Lloret-Mauri, Spain

Thomas Magedanz, Germany
Guergana S. Mollova, Austria
Algirdas Pakstas, UK
Kiran Ranga Rao, USA
M. Rocchetti, Italy
Peijun Shan, USA
Ravi S. Sharma, Singapore
Tomohiko Taniguchi, Japan
Wanggen Wan, China
Fujio Yamada, Brazil

Contents

Iterative Decoding and Cross-Layering Techniques for Multimedia Broadcasting and Communications, Fred Daneshgaran, Massimiliano Laddomada, and Marina Mondin
Volume 2008, Article ID 262148, 2 pages

Lossy Joint Source-Channel Coding Using Raptor Codes, O. Y. Bursalioglu, M. Fresia, G. Caire, and H. V. Poor
Volume 2008, Article ID 124685, 18 pages

A Simple Scheme for Belief Propagation Decoding of BCH and RS Codes in Multimedia Transmissions, Marco Baldi and Franco Chiaraluce
Volume 2008, Article ID 957846, 12 pages

Design of a VLSI Decoder for Partially Structured LDPC Codes, Fabrizio Vacca, Libero Dinoi, and Guido Masera
Volume 2008, Article ID 245305, 12 pages

Turbo Decoder for Low-Power Ultrawideband Communication Systems, Esam A. Obiedat and Lei Cao
Volume 2008, Article ID 897069, 7 pages

System Level Evaluation of Innovative Coded MIMO-OFDM Systems for Broadcasting Digital TV, Y. Nasser, J.-F. Hélar, and M. Crussière
Volume 2008, Article ID 359206, 12 pages

Stationary Interference Cancellation in Upstream Coordinated DSL Using a Turbo-MMSE Receiver, Issam Wahibi, Meryem Ouzzif, Jérôme Le Masson, and Samir Saoudi
Volume 2008, Article ID 428037, 8 pages

Iterative Mean Removal Superimposed Training for SISO and MIMO Channel Estimation, O. Longoria-Gandara, R. Parra-Michel, M. Bazdresch, and A. G. Orozco-Lugo
Volume 2008, Article ID 535269, 9 pages

Block Iterative/Adaptive Frequency-Domain Channel Estimation for Cyclic-Prefixed Single-Carrier Broadband Wireless Systems, Jong-Seob Baek and Jong-Soo Seo
Volume 2008, Article ID 509171, 7 pages

Real-Time Communications in Large-Scale Wireless Networks, Liang Song and Dimitrios Hatzinakos
Volume 2008, Article ID 586067, 16 pages

Editorial

Iterative Decoding and Cross-Layering Techniques for Multimedia Broadcasting and Communications

Fred Daneshgaran,¹ Massimiliano Laddomada,² and Marina Mondin³

¹ *Department of Electrical and Computer Engineering, School of Engineering, Computer Science and Technology, California State University, Los Angeles, CA 90032, USA*

² *Department of Electrical Engineering, Texas A&M University, Texarkana, TX 75505, USA*

³ *Third School of Engineering - Information Technologies, Politecnico di Torino, 10129 Torino, Italy*

Correspondence should be addressed to Massimiliano Laddomada, mladdomada@tamut.edu

Received 19 October 2008; Accepted 19 October 2008

Copyright © 2008 Fred Daneshgaran et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The explosive growth of multimedia applications over the Internet and the ever-increasing users' demands over commercial terrestrial digital multimedia broadcasting all over the world call for efficient physical and cross-layer techniques able to mitigate the potential problems limiting broadband services over wireless networks. In this scenario, mobile multimedia is expected to be one of the key services of future wireless mobile networks. Meanwhile, recent advances in digital communications have paved the way to a variety of standards aimed at providing multimedia services over terrestrial broadband networks. For example, DVB-H, T-DVB, T-DMB, wireless LANs, and wireless MANs are some of the most recent standards enabling such technology.

Iterative decoding techniques for both source, channel, and joint source-channel coding and decoding and cross-layering techniques have proven to be very effective for providing a viable means of achieving capacity-approaching performance at very reduced computational burden.

This special issue is aimed at highlighting state-of-the-art techniques on the most recent research advances enabling digital multimedia services over broadband wireless networks, focused on physical and cross-layering solutions.

The first paper "Lossy joint source-channel coding using raptor codes" by O. Y. Bursalioglu, M. Fresia, G. Caire, and H. V. Poor presents a joint source-channel coding scheme that combines the advantages and simplicity of entropy-coded quantization with the robustness of linear codes into a single linear encoding stage. Raptor codes are exploited in order to guarantee a continuum of coding rates with a single basic encoding algorithm.

The second paper "A simple scheme for belief propagation decoding of BCH and RS codes in multimedia transmissions" by Marco Baldi and Franco Chiaraluce proposes a new technique for decoding classic binary as well as nonbinary codes through the use of the belief propagation algorithm, which has proven to be effective in the decoding of low-density parity-check codes. The focus of the paper is on Reed-Solomon codes included in the CDMA2000 standard.

The third paper "Design of a VLSI decoder for partially structured LDPC codes" by Fabrizio Vacca, Libero Dinoi, and Guido Masera focuses on the development of a new class of partially structured low-density parity-check codes, and discusses the design of the VLSI decoder, as well as a novel decoding technique called split decoding.

The fourth paper "Turbo decoder for low power ultra-wideband communication systems" by Esam Obiedat, and Lei Cao proposes a new method to reduce the computational complexity of turbo decoding in ultra-wideband systems employing the orthogonal frequency division multiplexing (OFDM) modulation. A bit-level stopping technique based on the constrained decoding method is presented with the aim of detecting code convergence on each OFDM subcarrier.

The fifth paper "System level evaluation of innovative coded MIMO-OFDM systems for broadcasting digital TV," by Youssef Nasser, Jean-Francois Helard, and Matthieu Crussière investigates the application of space-time codes in single frequency networks for digital TV broadcasting. A 3D space-time-space block code is proposed in the context of

the future terrestrial digital TV systems using coded MIMO-OFDM techniques.

The sixth paper “Stationary interference cancellation in upstream coordinated DSL using a turbo-MMSE receiver” by Issam Wahibi, Meryem Ouzzif, Jérôme Le Masson, and Samir Saoudi investigates the use of stationary interference cancellation in upstream digital subscriber lines (DSLs) employing turbo-MMSE techniques. The use of noise whitening techniques coupled with a mean-squared error iterative receiver is proposed in order to approach the matched filter bound of a DSL-coordinated system.

The seventh paper “Iterative Mean Removal Superimposed Training for SISO and MIMO Channel Estimation” by Omar Humberto Longoria-Gandara, Ramón Parra-Michel, Luis Miguel Bazdresch, and Aldo Orozco describes a novel iterative radio channel estimation algorithm based on a superimposed training estimation technique. The proposed algorithm is well tailored to SISO as well as MIMO channels.

The eighth paper “Block iterative/adaptive frequency-domain channel estimation for cyclic-prefixed single-carrier broadband wireless systems” by Jong-Seob Baek and Jong-Soo Seo presents a new block iterative/adaptive frequency-domain channel estimation algorithm. The idea consists in iteratively estimating the frequency response of the channel by employing a weighted element-wise block adaptive frequency-domain channel estimation scheme that relies on the soft information obtained by a soft-input soft-output decoder.

The last paper of this special issue “Real time communications in large scale wireless networks” by Liang Song and Dimitrios Hatzinakos investigates the use of large-scale cognitive networking methods in multihop wireless networks. Based on the proposed method, data packets travel along opportunistic paths using opportunistically available spectrum in every hop. The proposed algorithm is key to guaranteeing quality of service in wireless mesh networks for broadband networks supporting multimedia services, and visual sensor networks for surveillance.

ACKNOWLEDGMENTS

The authors are grateful to the reviewers for their invaluable work and to the authors of the papers collected in this special issue.

Fred Daneshgaran
Massimiliano Laddomada
Marina Mondin

Research Article

Lossy Joint Source-Channel Coding Using Raptor Codes

O. Y. Bursalioglu,¹ M. Fresia,² G. Caire,¹ and H. V. Poor²

¹Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089, USA

²Department of Electrical Engineering, Princeton University, Princeton, NJ 08544, USA

Correspondence should be addressed to G. Caire, caire@usc.edu

Received 1 May 2008; Accepted 17 June 2008

Recommended by Massimiliano Laddomada

The straightforward application of Shannon's separation principle may entail a significant suboptimality in practical systems with limited coding delay and complexity. This is particularly evident when the lossy source code is based on entropy-coded quantization. In fact, it is well known that entropy coding is not robust to residual channel errors. In this paper, a joint source-channel coding scheme is advocated that combines the advantages and simplicity of entropy-coded quantization with the robustness of linear codes. The idea is to combine entropy coding and channel coding into a single linear encoding stage. If the channel is symmetric, the scheme can asymptotically achieve the optimal rate-distortion limit. However, its advantages are more clearly evident under finite coding delay and complexity. The sequence of quantization indices is decomposed into bitplanes, and each bitplane is independently mapped onto a sequence of channel coded symbols. The coding rate of each bitplane is chosen according to the bitplane conditional entropy rate. The use of systematic raptor encoders is proposed, in order to obtain a continuum of coding rates with a single basic encoding algorithm. Simulations show that the proposed scheme can outperform the separated baseline scheme for finite coding length and comparable complexity and, as expected, it is much more robust to channel errors in the case of channel capacity mismatch.

Copyright © 2008 O. Y. Bursalioglu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

A stationary ergodic source can be transmitted over an information-stable channel with end-to-end average distortion D with *bandwidth expansion factor* b not lower than $R(D)/C$ channel symbols per source sample, where $R(D)$ is the source rate distortion function and C is the channel capacity. Shannon's source-channel *separation principle* [1] ensures that this optimal performance can be approached by independently designing the source coding and the channel coding schemes. The bandwidth expansion factor b is defined as the number of channel symbols per source symbol. If a block of K source symbols is transmitted through the channel in N channel uses, then $b = N/K$. This provides a definite architectural advantage in practical systems, where typically (lossy) source coding is implemented at the application layer, while channel coding is designed and optimized for the physical layer.

On the other hand, this separated source-channel coding (SSCC) approach may incur substantial suboptimality, due to the nonideal behavior of finite length, finite complexity, source and channel codes. In fact, source codes designed

without taking into account the presence of channel decoding errors are typically very fragile and this might impose unnecessarily restrictive constraints on the performance of channel coding. In such cases, joint source-channel coding (JSCC) may lead to performance improvement (i.e., a better (b, D) operating point) for the same level of complexity.

Most practical lossy source coding schemes for natural sources (e.g., images, audio, video) are based on the idea of *transform coding* [2]. Source blocks are projected onto a suitable basis by a linear transformation, such that the source is well described by only a small number of significant transform coefficients. Then, the coefficients are scalar-quantized, and finally the resulting sequence of quantization indices are entropy coded. The theoretical foundation of this approach relies on the universality of entropy-coded quantization, and dates back to the work of Ziv [3]. In general, the linear transform is adapted to the given class of sources (e.g., wavelet transforms for images [4]). The statistics of the quantization indices is not known a priori. However, the memory structure of the underlying discrete source is fixed and it is typically described as a finite-memory tree-source (e.g., the context structure of JPEG2000 [5, 6]). Then,

data compression is obtained by using an adaptive entropy coding scheme that estimates the transition probabilities of the source statistical model. For example, arithmetic coding [7] with Krichevsky-Trofimov (KT) sequential probability estimation is a common choice [8].

For the sake of simplicity, this paper treats only independent and identically distributed (i.i.d.) sources with known statistics, that is, it neither deals with the transform coding aspect, nor with the universal implementation of entropy coding. However, our results can be generalized along the lines of what done in [5, 6]. Even in the nonuniversal case, classical lossless compression is *catastrophic*: a small Hamming distortion (number of bits in error) in the entropy-coded sequence is mapped into a large distortion in the reconstructed source sequence. This imposes a very strict target error probability on the channel coding stage, thus involving both complex channel coding and operating points that may be quite far from the theoretical limits. This is even more evident in applications where the coding delay is limited, thus preventing the use of very large block lengths.

It was shown in [9] that fixed-to-fixed length data compression of a discrete source with linear codes is asymptotically optimal, in the sense that compression up to the source entropy rate can be achieved. This is strongly related to transmission using the same linear code on a discrete additive noise channel where the noise has the same statistics as the discrete source. This analogy can be exploited in order to design a JSCC scheme. We wish to maintain the simplicity of the transform coding approach while improving the robustness of the scheme. The rationale behind the proposed design is the following: since linear codes achieve the entropy rate of discrete sources and the capacity of symmetric channels, we can combine the entropy coding stage and the coding stage into a single linear encoding stage. The advantage of this approach is that the design of noncatastrophic linear encoders is very well understood. Therefore, the proposed scheme can approach the optimal separation limit for large block length, while achieving better robustness to channel errors at finite decoding delay and complexity.

In [5], this JSCC approach was applied to the transmission of JPEG2000-like encoded images (in the sense that the wavelet transform, the quantization scheme and the tree-source memory structure were borrowed from JPEG2000), by using a family of progressively punctured turbo codes to map directly the redundant quantization bits into channel symbols. As stated above, here we focus on simpler i.i.d. sources with perfectly known statistics (i.e., the nonuniversal case) and investigate in greater detail the performance analysis and the comparison with the baseline SSCC approach. In this work, we use raptor codes [10] in order to map the redundant quantization bits into channel-coded symbols.

Our scheme works as follows. A source block of length K is quantized symbol by symbol. The sequence of quantization indices, represented as binary vectors, are partitioned into bitplanes. The bitplanes are separately encoded into channel symbols by a bank of binary raptor encoders. Each bitplane is encoded at a rate that depends on its conditional

entropy rate given the bitplanes previously encoded. At the decoder, the bitplanes are decoded in sequence using a multistage decoder, where in each stage we use a belief propagation (BP) iterative decoder that takes into account both the already decoded bits from previous planes, and the a priori statistics of the current bitplane as well as the received channel output.

Raptor codes are a particularly useful class of rateless codes. The advantage of using a rateless code is clear: with a single basic encoding machine we can generate a continuum of coding rates. Therefore, the scheme can adapt naturally to the entropy rate of the source and to the capacity of the channel. Although we do not pursue the universal setting in this work, we notice here that the proposed architecture allows a very fine rate matching between the (unknown a priori) source entropy and the channel capacity without resorting to a library of progressively punctured codes as is done in [5].

We express the performance of a source-channel coding scheme in terms of its peak signal-to-noise ratio (PSNR), expressed in dB, defined as

$$\text{PSNR} \triangleq -10 \log_{10}(D). \quad (1)$$

In particular, we will focus on a standard Gaussian i.i.d. source $S \sim \mathcal{N}(0, 1)$ and on the mean-squared distortion $D = \mathbb{E}[|S - \hat{S}|^2]$. In this case, the distortion-rate function is $D = 2^{-2R}$. At the Shannon separation limit, that is, letting $R = bC$, we have

$$\text{PSNR}_{\text{Shannon}} = 20 \log_{10}(2) bC = (6 \text{ dB}) \times bC. \quad (2)$$

Our aim is to design a family of practical schemes that operate close to the curve $\text{PSNR}_{\text{Shannon}}$ versus b . Notice that we do not pursue here the design of embedded schemes, that is, of single coding schemes that achieve multiple (PSNR, b) points. Nevertheless, the bitplane layered structure of the proposed encoder and the proposed multistage decoder lend themselves quite naturally to the design of embedded JSCC schemes. We leave this aspect for future work and comment on it further in the concluding section.

The rest of this paper is organized as follows. In Section 2, we review the limits of scalar entropy-coded quantization and define the target “operational Shannon limit” of our scheme. In Section 3 we present a comprehensive analysis of the baseline SSCC scheme which represents our term of comparison. Section 4 presents the details of the proposed scheme, its analysis and an algorithm for progressive incremental redundancy in order to optimize the coding rates at each bitplane. Section 5 presents some additional numerical comparisons between the baseline SSCC and the JSCC schemes, and in Section 5 we present some concluding remarks. Raptor codes, BP decoding, EXIT chart analysis and some ancillary results are presented in the appendices.

2. ENTROPY-CODED SCALAR QUANTIZATION

A source sequence of length K , $\mathbf{s} \in \mathbb{R}^K$, is quantized by applying componentwise the scalar quantizer $\mathcal{Q}_B : \mathbb{R} \rightarrow \mathbb{F}_2^{B+1}$,

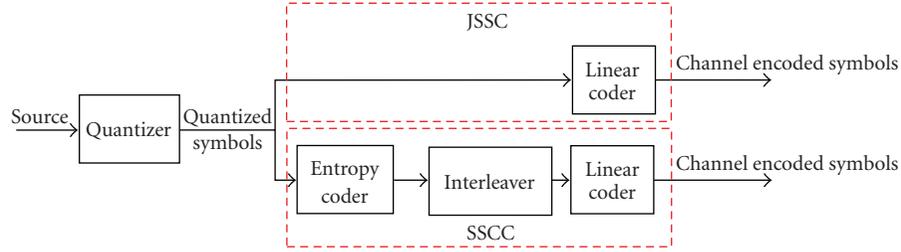


FIGURE 1: Conceptual block diagram of the conventional SSCC and the proposed JSCC schemes. The two schemes coincide but for the fact that the concatenation of entropy coding and channel coding (SSCC) is replaced by a single linear encoding block (JSCC).

where B bits are used to represent the magnitude and one bit represents the sign. Let $\mathbf{u} = \mathcal{Q}_B(\mathbf{s})$ denote the sequence of quantization indices and let $u_{p,k} : p = 0, \dots, B$ denote the bits forming the k th index. The sequence \mathbf{u} can be thought as a $(B + 1) \times K$ binary array, where each row is called a “bitplane.” Without loss of generality, we associate the 0th row with the sign bit and the rows from 1 to B with the magnitude bits, with the convention that the first bitplane is the least significant and the B th bitplane is the most significant.

As anticipated in the Introduction, we fix the quantizer and compare the performance of an SSCC approach based on the concatenation of a conventional entropy coding stage with a conventional channel code, with the performance of a JSCC that merges the two operations into a single linear encoding map. Therefore, in the absence of channel residual errors, both schemes achieve the same minimum distortion due to the quantizer, denoted by $D_{\mathcal{Q}}(B)$. Letting $H_B(U)$ denote the entropy rate of \mathbf{u} , measured in bit per quantization index, we have that the point

$$\left(b = \frac{H_B(U)}{C}, \text{PSNR} = -10 \log_{10}(D_{\mathcal{Q}}(B)) \right) \quad (3)$$

is the best achievable point for any scheme based on the fixed quantizer \mathcal{Q}_B . Following [2], we refer to this point as the “operational Shannon limit” for schemes with fixed quantizers.

We consider uniform scalar quantizers where the interval size is chosen in order to minimize the mean-squared distortion of the Gaussian unit variance i.i.d. source, for a fixed number 2^{B+1} of intervals. In [3], Ziv showed that a coding scheme formed by a scalar uniform quantizer followed by entropy coding yields a rate penalty of no more than 0.754 bits per sample with respect to the $R(D)$ limit. Thus, constraining the quantizer to be a uniform scalar quantizer should cost no more than a $0.754/C$ channel symbols per source symbols.

In Figure 2 we compare the PSNR versus b curves for the Shannon limit, the Ziv bound and the operational Shannon limit for a family of optimized uniform scalar quantizers with $B = 1, 2, 3, 4, 5$, and 6, and for channel capacity $C = 0.5$. All results in this paper make use of this family of quantizers.

3. ANALYSIS OF THE BASELINE SSCC SCHEME

In this section, we study the performance of nonideal SSCC. First, we consider the performance degradation due to

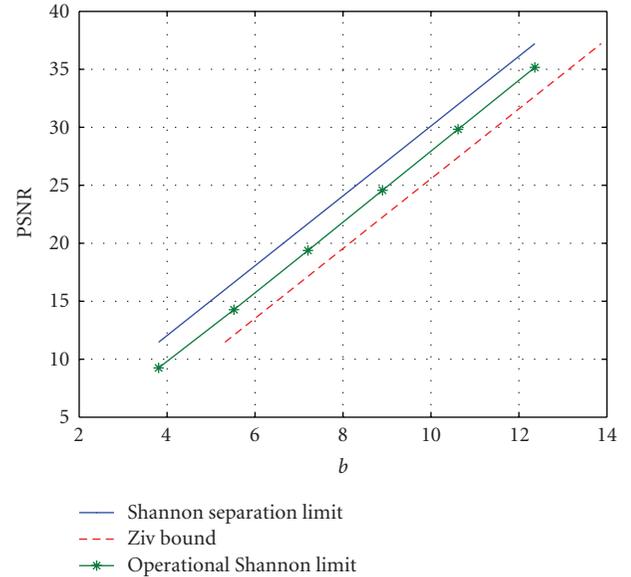


FIGURE 2: PSNR versus b for the Shannon Limit, the Ziv bound and the operational Shannon limit for the considered family of scalar quantizers for $B = 1, \dots, 6$ and for channel capacity $C = 0.5$.

nonideal source and channel codes that operate at source coding rate $R_s = R(D) + \delta_s$ and channel coding rate $R_c = C - \delta_c$, respectively, where δ_s and δ_c are positive rate gaps. This analysis assumes no errors at the output of the channel decoder.

Then, we introduce the channel decoding error probability, and obtain a distortion upper bound as a function of δ_c and δ_s , closely following the analysis of [11]. This analysis is based on the random coding exponent for channel codes, and essentially validates the error-free rate-gap analysis even for moderately large block length K .

Finally, we consider a very practical scheme, based on the concatenation of arithmetic entropy coding and a conventional binary raptor code. We provide a very accurate semi-analytic approximation for the achievable PSNR of this scheme and show that the achieved results follow closely the error free rate-gap analysis by matching the parameters δ_c and δ_s . We also notice that for finite block length K the practical scheme suffers from an additional performance degradation, especially visible at high resolution (large PSNR). We quantify this additional performance

degradation by looking at the finite-length versus infinite-length error performance of raptor codes.

3.1. Rate-gap analysis

Consider a separated scheme that makes use of channel coding at rate $R_c = C - \delta_c$ and source coding at rate $R_s = R(D) + \delta_s$, where $\delta_c, \delta_s > 0$ are rate gaps, and where the residual bit-error rate (BER) at the output of the channel decoder is (essentially) zero. Using $D = 2^{-2(R_s - \delta_s)}$ and $b = R_s/R_c$ we obtain

$$\text{PSNR} = (6 \text{ dB}) \times (b(C - \delta_c) - \delta_s). \quad (4)$$

We notice that the slope of the straight line characterizing PSNR versus b decreases with the channel coding gap δ_c , while the source coding rate gap involves only a horizontal shift. As a result, an SSCC whose channel coding stage achieves negligible BER works further and further away from the Shannon limit as PSNR increases (high resolution).

3.2. SSCC with codes achieving positive error exponent

In order to take into account channel decoding errors, we modify slightly the approach of [11] and obtain the achievable PSNR lower bound (we omit the details since the derivation follows trivially from [11]):

$$\text{PSNR} \geq (6 \text{ dB}) \times (b(C - \delta_c) - \delta_s) - 10 \log_{10}(1 + 2^{-KbE_r(C - \delta_c) + 1 + 2b(C - \delta_c) - 2\delta_s}), \quad (5)$$

where $E_r(R_c)$ denotes the random coding error exponent for a given coding ensemble over the considered transmission channel. Notice that $E_r(R_c) > 0$ for all $R_c < C$, and therefore, the error exponent is positive for all rate gaps $\delta_c > 0$. For values of $K, \delta_c, \delta_s, b, C$ such that $2^{-KbE_r(C - \delta_c) + 1 + 2b(C - \delta_c) - 2\delta_s} \ll 1$, (5) essentially coincides with (4).

Figure 3 compares (4) and (5) for different values of δ_c and for block length $K = 10000$ (which is the finite source block length that we will use throughout this paper), and $\delta_s = 0.3821$. This value of δ_s is chosen in order to match the rate gap attained by the quantizers (see Figure 5). In these results, we considered a binary symmetric channel (BSC) with capacity $C = 0.5$ (cross-over probability $\epsilon = 0.11$). The exponent $E_r(R_c)$ for the BSC can be found, for example, in [12]. For the parameters of Figure 3 we notice that (4) and (5) do not coincide for too small δ_c (e.g., $\delta_c = 0.01$ in the figure) while they coincide for large enough δ_c (in this case, $\delta_c \geq \delta_{c*} = 0.016$). For finite but large block lengths, as in this case, the threshold δ_{c*} is given by the minimum value of the channel coding rate gap above which the exponent $T(C, \delta_c, \delta_s, K, b) \triangleq -KbE_r(C - \delta_c) + 1 + 2b(C - \delta_c) - 2\delta_s$ becomes negative. In Figure 4, we plot $T(C, \delta_c, \delta_s, K, b)$ and $E_r(C - \delta_c)$ versus δ_c , when the parameters of Figure 3 are considered. It has been observed that for different values of b given in the range of Figure 3, δ_c is constant. Thus in Figure 4, we use $b = 7.2036$.

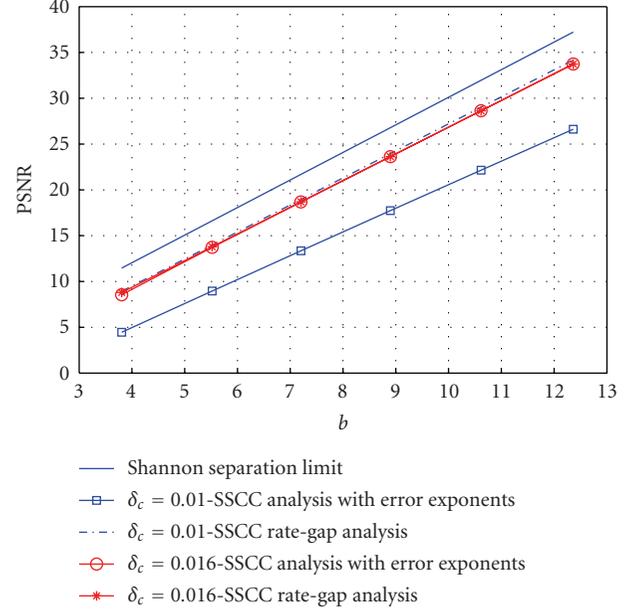


FIGURE 3: The SSCC rate-gap approximation (4) is compared with the PSNR upperbound (5) for $C = 0.5$ and $\delta_s = 0.3821$.

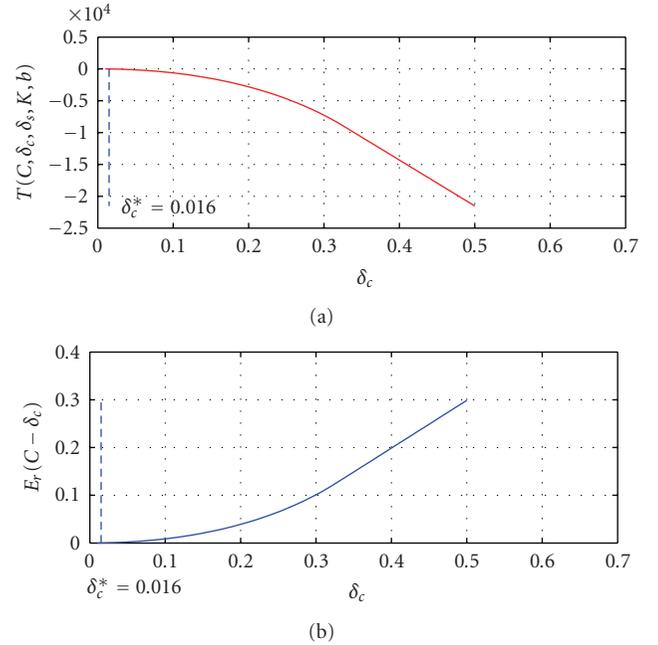


FIGURE 4: $T(C, \delta_c, \delta_s, K, b)$ and $E_r(R_c)$ versus δ_c for $C = 0.5$, $K = 10000$, $\delta_s = 0.03821$, and $b = 7.2036$.

3.3. SSCC with arithmetic coding and raptor codes

We provide an accurate approximated analysis of the performance of a practical SSCC scheme that can be regarded as our baseline scheme, since its encoding and decoding complexity is very similar to that of the JSCC scheme examined in the next section. With reference to the block diagram of Figure 1, we consider the concatenation of an

optimized uniform scalar quantizer with $B + 1$ quantization bits with an arithmetic encoder. The resulting entropy-coded bits are then channel encoded using a raptor code of suitable rate.

A sufficiently large interleaver is placed between the entropy coding and the channel coding stages, such that the bit decoding errors at the input of the arithmetic decoder can be considered to be i.i.d.. Since the arithmetic encoder has perfect knowledge of the probability distribution of the discrete source \mathbf{u} at the quantizer output, it can approach very closely the source entropy rate $H_B(U)$ even for moderate source block length K .

We approximate the performance of such a scheme by assuming that the arithmetic decoder produces random data after the first bit error at its input. Let $M = KH_B(U)$ denote the number of entropy-coded bits produced by the arithmetic encoder. These bits are channel encoded and decoded. Let m denote the position of the last correctly decoded bit before the first bit error at the arithmetic decoder input. Under the assumption of i.i.d. bit errors, m is a truncated geometric random variable with probability mass function

$$P(m = i) = (1 - P_b)^i P_b \quad (6)$$

for $i = 0, 1, \dots, M - 1$, and $P(m = M) = 1 - \sum_{i=0}^{M-1} (1 - P_b)^i P_b$, where P_b denotes the BER at the output of the channel decoder. We approximate the number of correctly decoded quantization indices by $m/H_B(U)$ (neglecting integer effects). After the first bit error, the arithmetic decoder produces random symbols distributed as the quantization indices (i.e., according to the given discrete-source probability distribution) but essentially statistically independent of the source sequence. Therefore, the average distortion in this case is given by

$$\sigma^2 = \mathbb{E}[(S - \tilde{S})^2] = \mathbb{E}[S^2] + \mathbb{E}[\tilde{S}^2] = 1 + \sigma_{\tilde{S}}^2, \quad (7)$$

where \tilde{S} denotes a random variable distributed as the quantizer reconstruction points, and $\sigma_{\tilde{S}}^2$ denotes its variance. On the other hand, before the first bit error, the system reconstructs the correct quantization points \hat{S} correctly; therefore the average distortion in this case coincides with the quantization distortion $D_{\mathcal{Q}}(B)$.

Eventually, the total average distortion of the system is approximated by

$$D \approx P(m = M)D_{\mathcal{Q}}(B) + \sum_{i=0}^{M-1} (1 - P_b)^i P_b \left[D_{\mathcal{Q}}(B) \frac{i/H_B(U)}{K} + \sigma^2 \frac{K - i/H_B(U)}{K} \right]. \quad (8)$$

The approximate analysis requires the evaluation of the residual BER at the channel decoder output. This can be obtained by simulation of the stand-alone raptor code with given finite length, or by using any suitable approximation or semi-analytic technique, such as Density Evolution or EXIT chart methods [13–17]. In particular, we make use of the EXIT chart approximation, reviewed in Appendix B.

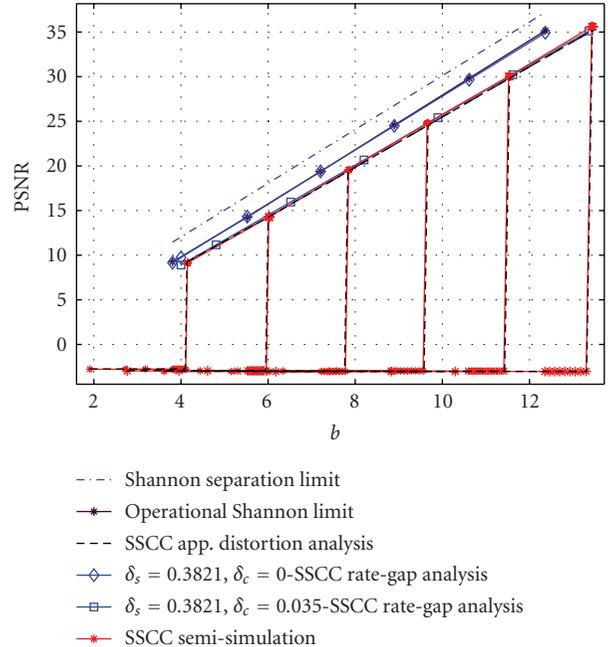


FIGURE 5: Performance of the concatenation of arithmetic coding and raptor code for infinite channel coding length, source length $K = 10000$, and a BSC with $C = 0.5$. $\delta_s = 0.3821$, and $\delta_c = 0$ are the values of rate-gaps for the operational Shannon limit (4). For the base line SSCC, these rate-gap values are empirically found to be $\delta_s = 0.3821$ and $\delta_c = 0.035$.

In Figure 5 we report the PSNR versus b (obtained by using (8)) for different values of B when $K = 10000$, a BSC with capacity $C = 0.5$ and where the raptor code output BER is approximated via the EXIT chart method. These results assume implicitly infinite channel coding block length. In order to validate the approximated distortion analysis of (8), we run simulations of the arithmetic decoder and quantization reconstruction, fed with the quantization bits corrupted by independent bit errors at a rate equal to the raptor code output BER. As seen in Figure 5, the results of the simulated arithmetic decoder match remarkably well the approximation (8), thus showing that the arithmetic decoder indeed produces approximately random data after the first bit error.

The results in Figure 5 show that for the case of very large channel coding block length the performance of the baseline SSCC scheme is remarkably close to the operational Shannon limit and therefore the scheme is hard to beat by any scheme using the same set of quantizers. However, the picture changes when we consider a finite channel coding block length. In particular, we consider independent encoding of each source block, so that the system latency is dictated by the source block length K and not by the channel coding block length. This corresponds to choosing the raptor code input bits block length equal to $M = KH_B(U)$. For the system parameters as before, the PSNR results in this case are shown in Figure 6. We notice that the finite channel coding block length yields an additional degradation that increases with PSNR.

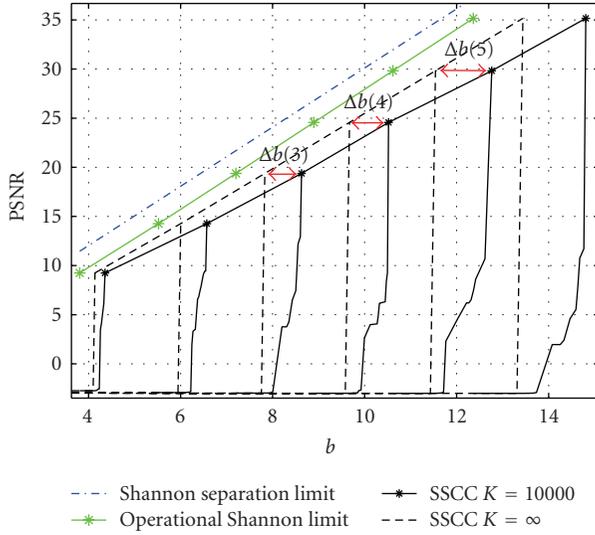


FIGURE 6: Performance of the concatenation of arithmetic coding and a raptor code for finite channel coding length, source length $K = 10000$, and a BSC with $C = 0.5$.

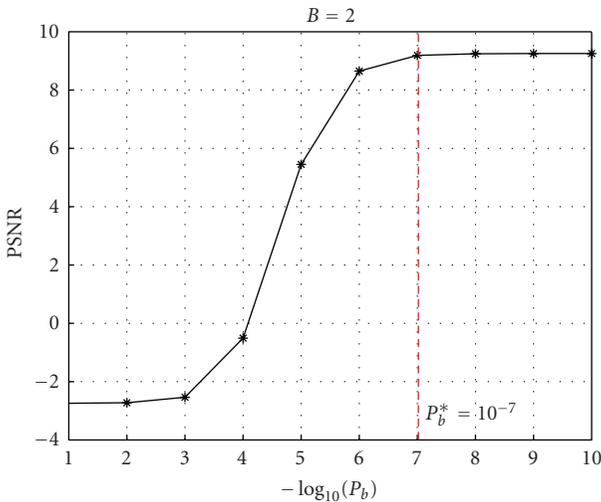


FIGURE 7: Output PSNR as a function of the channel decoding residual BER P_b .

We can explain and quantify the increasing bandwidth expansion gap $\Delta b(B)$ shown in Figure 6 as follows. Let R_{inf} and R_{fin} denote the channel coding rates needed by the raptor code to reach a small BER such that the effective distortion is virtually identical to the quantization distortion. For example, Figure 7 plots the PSNR corresponding to the distortion (8) as a function of P_b . We notice that for $P_b = 10^{-7}$ the quantization distortion (corresponding to the maximum achievable PSNR) is essentially reached. Then, Figure 8 plots the raptor code BER for the BSC with capacity $C = 0.5$, as a function of the reciprocal of the channel coding rate $1/R_c$. Notice that the raptor code is a rateless code, and therefore we can generate as many coded symbols as we like. In order to generate Figure 8 we keep the channel parameter $\epsilon = 0.11$ fixed (corresponding to $C = 0.5$) and run encoding

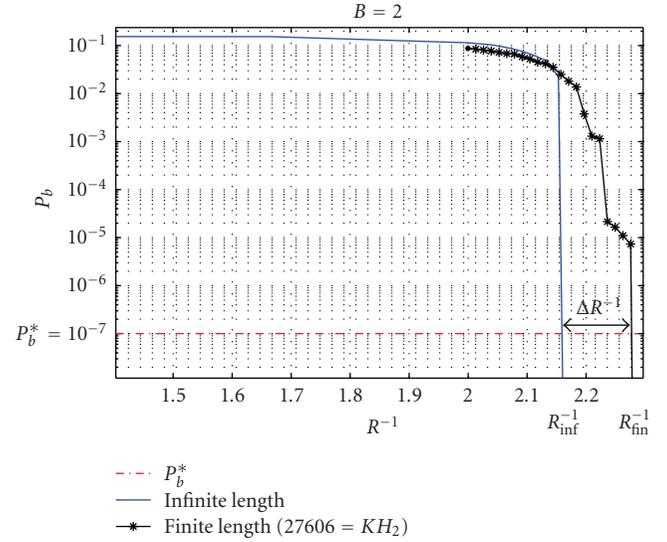


FIGURE 8: Raptor code output BER for the infinite block length case (EXIT approximation) and for the finite length case, obtained by simulation, as a function of the reciprocal of the coding rate for $C = 0.5$. The Finite length is taken to be $27606 = KH_2(U)$ which is the approximate number of bits at the output of arithmetic encoder, since the Gaussian block has entropy rate 2.7606 when quantized with \mathcal{Q}_2 .

and decoding for smaller and smaller coding rates. The infinite block length case is obtained by using the EXIT chart approximation.

Figure 8 shows that the target BER of 10^{-7} is reached at certain rates R_{inf} and R_{fin} for the cases of infinite and finite block length, respectively, and allows us to find the difference $1/R_{\text{inf}} - 1/R_{\text{fin}}$, shown in the figure.

Finally, we can quantify the bandwidth expansion gaps shown in Figure 6 by noticing that, since $b = H_B(U)/R_c$, we have

$$\Delta b(B) = H_B(U) \left(\frac{1}{R_{\text{inf}}} - \frac{1}{R_{\text{fin}}} \right). \quad (9)$$

It is clear that the gap $\Delta b(B)$ is increasing with the quantizer resolution B , and therefore with PSNR. This is a further confirmation of the fact that, in practice, it becomes more and more difficult to approach the Shannon limit as the resolution increases.

4. JOINT SOURCE-CHANNEL CODING SCHEME

In this section we describe the encoder and decoder of the proposed JSCC scheme. Then, we discuss an incremental redundancy rate allocation procedure that allows the optimization of the scheme. We hasten to say that this rate allocation procedure is run off-line, and serves to design the coding scheme for given source and channel statistics. More generally, an adaptive scheme that allocates coded bits to the bitplanes on the fly, depending on the empirical entropy rate of the source and on the capacity of the channel may be envisaged in a universal JSCC setting, where the source statistics are not known a priori and are learned instead from

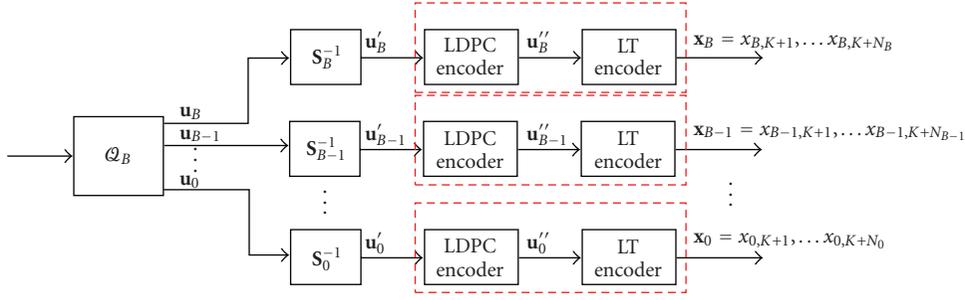


FIGURE 9: Diagram of the proposed JSCC encoder.

the source sequence itself. However, we do not pursue this approach here.

Figure 9 shows the encoder block diagram. Each bitplane (row of the binary array \mathbf{u} of quantization indices produced by the quantizer), is mapped into a sequence of coded symbols. Here we consider binary coding, and a BSC. Letting \mathbf{u}_p denote the p th row of \mathbf{u} , the corresponding block of coded symbols is given by $\mathbf{x}_p = \mathbf{u}_p \mathbf{G}_p$, where \mathbf{G}_p is a suitable encoding matrix of size $K \times N_p$. Then, the encoded blocks $\mathbf{x}_0, \dots, \mathbf{x}_B$ are transmitted in sequence over the BSC. The resulting bandwidth expansion factor is

$$b = \frac{\sum_{p=0}^B N_p}{K}. \quad (10)$$

Given the source symmetry, it is clear that the sign bit is equiprobable and has entropy $H(U_0) = 1$. Furthermore, it is independent of the magnitude bits. Hence, the target nominal rate for the encoder of the sign bit is $K/N_0 = C$. As for the p th magnitude bit, we allocate a nominal target rate equal to $K/N_p = C/H(U_p | U_{p+1}, \dots, U_B)$, where $H(U_p | U_{p+1}, \dots, U_B)$ denotes the conditional entropy rate of the p th bitplane, conditioned on the bitplanes $p+1, p+2, \dots, B$. It follows that the nominal bandwidth expansion is given by

$$\begin{aligned} b &= \frac{K/C + \sum_{p=1}^B KH(U_p | U_{p+1}, \dots, U_B)/C}{K} \\ &= \frac{1 + \sum_{p=1}^B H(U_p | U_{p+1}, \dots, U_B)}{C} \\ &= \frac{H_B(U)}{C}, \end{aligned} \quad (11)$$

which is optimal.

In order to be able to decode at these rates, we consider a multistage decoder as shown in Figure 10, that considers the bitplanes in sequence. The sign bit is independently decoded. The magnitude bits are decoded in sequence, starting from the B th plane. At each decoding stage p , the hard decisions of the already decoded planes are used by the BP decoder to compute the conditional a priori probabilities of the p th bitplane, as explained in Appendix A. Assuming that at each level p , the previous levels are correctly decoded, then the rates $K/N_p = C/H(U_p | U_{p+1}, \dots, U_B)$ are achievable.

In practice, due to the fact that the raptor codes do not achieve sufficiently low BER if their rate is too close to the

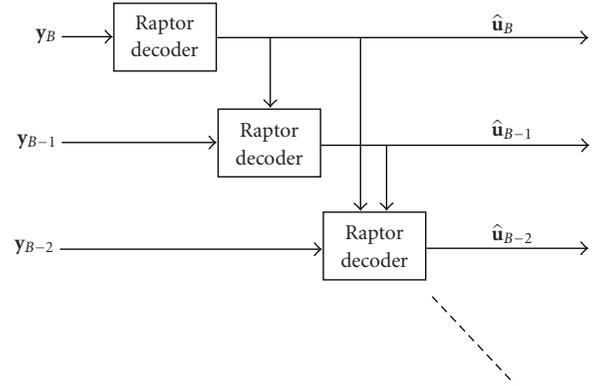


FIGURE 10: Multistage decoder for successive bitplane decoding and source reconstruction.

nominal rate limit, we must allocate the rates allowing for some gap. The rate allocation problem is made more complicated by the fact that in the multistage decoder the decoding of the different planes is not independent. In particular, if the p th plane fails with many bits in errors, then it is very likely that all the planes $p-1, p-2, \dots, 1$ will also fail, since their decoders are fed with incorrect a priori conditional probabilities. We will address the problem of rate allocation for the multistage decoder at the end of this section.

Next, let us examine in more detail how encoding of the p th plane is implemented with raptor codes [10]. Raptor codes can substantially be viewed as an extension of Luby Transform codes (LT codes) [18], since they are based on the concatenation of an outer linear code (in our case we consider low-density-parity check (LDPC) codes) with an inner LT code (see Appendix A for details). We use raptor codes in systematic form. In particular, let \mathbf{S}_p be a $K \times K$ full-rank binary matrix given by $\mathbf{S}_p = \mathbf{G}_{\text{ldpc}} \mathbf{A}_p$, where \mathbf{A}_p is a submatrix of the LT code generator matrix at encoding level p and \mathbf{G}_{ldpc} is the generator matrix of the LDPC code (see [10] for details). The encoder produces a vector of K intermediate symbols, denoted by $\mathbf{u}'_p = \mathbf{u}_p \mathbf{S}_p^{-1}$. Then, the intermediate symbols are expanded by high-rate LDPC encoding, into $\mathbf{u}''_p = \mathbf{u}'_p \mathbf{G}_{\text{ldpc}}$. Finally, the encoded symbols \mathbf{x} are obtained from \mathbf{u}'' , by applying nonsystematic rateless encoding, that is, the symbols x_1, x_2, \dots, x_{N_p} are produced in sequence, and each x_i is given as the sum of elements of \mathbf{u}'' selected at random according to the LT degree distribution Ω .

Notice that $\mathbf{u}_p = \mathbf{u}'_p \mathbf{S}_p$. Therefore, in the Tanner graph representing the code [19, 20] the nodes corresponding to source symbols \mathbf{b}_p have a degree distribution identical to that of a standard nonsystematic raptor code. Furthermore, although \mathbf{S}_p is sparse, its inverse is sufficiently dense such that the symbols \mathbf{u}'_p are close to being uniform and random i.i.d. Notice that this is essential to the scheme, since in order to drive the channel with the correct input distribution we need to send the nonsystematic symbols \mathbf{x}_p through the channel, and their distribution should be as close as possible to i.i.d. and equiprobable.

A key component in the systematic raptor code design consists of finding a suitable nonsingular $K \times K$ matrix \mathbf{S}_p , with given column weight distribution, and such that its inverse looks as much as possible like a random binary matrix. As for the LDPC code (often referred to as the “precode” in the raptor coding literature), we used a regular code with parameters (2, 100).

Let us focus now on decoding and source reconstruction. The multistage decoder of Figure 10 is based on BP at each stage p in order to approximately compute the symbol-by-symbol posterior marginal Log-Likelihood Ratios (LLRs) $\{\lambda_{p,k} : k = 1, \dots, K\}$ defined as

$$\lambda_{p,k} = \log \frac{P(u_{p,k} = 0 \mid \mathbf{y}_p, \mathbf{u}_{p+1}, \dots, \mathbf{u}_B)}{P(u_{p,k} = 1 \mid \mathbf{y}_p, \mathbf{u}_{p+1}, \dots, \mathbf{u}_B)}, \quad (12)$$

where \mathbf{y}_p denotes the channel output corresponding to the input \mathbf{x}_p , and the conditioning is with respect to the already decoded bitplanes. This is obtained by feeding the hard decisions from the planes $p + 1, \dots, B$ to the BP decoder at level p . An iterative version of the multistage decoder where soft messages in the form of a posteriori LLRs are exchanged instead of hard decisions was also considered, but it was observed that this does not provide any significant improvement and therefore was not pursued further, given its much greater complexity.

The information about the already decoded bitplanes is incorporated into the BP decoder for bitplane p in the following way. As explained in Appendix A, the BP algorithm is initialized with input messages at all the source and coded nodes in the Tanner graph of the code. The coded nodes (corresponding to the coded symbols \mathbf{x}_p), receive their input message from the corresponding channel observation. In the case of a BSC, this is given by

$$\mu_{p,i} = (-1)^{y_{p,i}} \log \frac{1 - \epsilon}{\epsilon}, \quad i = 1, \dots, N_p. \quad (13)$$

The source nodes (corresponding to the source bits \mathbf{u}_p), are associated with the input messages

$$\nu_{p,k} = \log \frac{P(u_{p,k} = 0 \mid \hat{u}_{p+1,k}, \dots, \hat{u}_{B,k})}{P(u_{p,k} = 1 \mid \hat{u}_{p+1,k}, \dots, \hat{u}_{B,k})}, \quad (14)$$

where $\hat{u}_{p+1,k}, \dots, \hat{u}_{B,k}$ are the hard decisions obtained from the previous stages.

The BP decoder at each stage runs for a given desired number of iterations, and eventually outputs both hard decisions to be passed to the next stage and soft outputs in the

form of the posterior LLRs given by (12). Once all bitplanes have been decoded, the source is reconstructed as follows. Consider without loss of generality the inverse quantization mapping

$$\mathcal{Q}_B^{-1}(u_k) = (-1)^{u_{0,k}} \frac{\Delta_B}{2} \left(\left[\sum_{p=1}^B u_{p,k} 2^p \right] + 1 \right) \quad (15)$$

that yields the mid-point of each quantization interval given the set of quantization bits.

Then, we can either consider hard reconstruction, which consists of using the hard decisions $\hat{u}_{p,k}$ in (15), or soft reconstruction, which makes use of the (approximate) posterior LLRs in order to compute the minimum-mean-square-error (MMSE) estimator of the source samples given the channel output, that is, the conditional mean estimator $\hat{s}_k = \mathbb{E}[s_k \mid \mathbf{y}]$. Treating the decoder estimated posterior LLRs as if they were the true posterior LLRs, we obtain

$$\hat{s}_k = \frac{\Delta_B}{2} \tanh \left(\frac{\lambda_{0,k}}{2} \right) \left(\left[\sum_{p=1}^B \frac{2^p}{1 + e^{\lambda_{p,k}}} \right] + 1 \right). \quad (16)$$

In Appendix A, we prove an interesting isomorphism between the BP decoder of the joint source-channel problem as described above and a related standard channel coding problem. Let us focus on a single binary independent source sequence \mathbf{u} of length K , with probabilities $p_k \triangleq P(u_k = 1)$ for $k = 1, \dots, K$. This is encoded into a binary codeword $\mathbf{x} = \mathbf{u}\mathbf{G}$, of length N , where \mathbf{G} is a $K \times N$ raptor encoder as previously described. Let us transmit \mathbf{x} through a BSC with cross-over probability ϵ , and let $\mathbf{y} = \mathbf{x} \oplus \mathbf{e}$ denote the corresponding output. The result holds for any binary input symmetric output channel, but here we focus on the BSC for simplicity of exposition. Then, the BP decoder for this problem is isomorphic to a decoder for the following related channel coding problem: consider transmission of the all-zero codeword from the systematic code with generator matrix $[\mathbf{I} \mid \mathbf{G}]$, of size $K \times (K + N)$ over a channel that for the first K components operates as

$$y_k = x_k \oplus u_k, \quad k = 1, \dots, K, \quad (17)$$

where u_k is the k th source symbol, and for the remaining N components operates as

$$y_{K+n} = x_{K+n} \oplus e_n, \quad n = 1, \dots, N. \quad (18)$$

In other words, there exists a one-to-one mapping of the messages of the BP decoder for the first problem (joint source channel) and the messages of the BP decoder for the second problem (channel only), for every edge of the decoder graph and every decoder iteration.

This means that the source-channel BP decoding can be analyzed, for example, by using the EXIT chart method, by considering the associated “virtual” channel, where the all-zero codeword from the associated systematic code is transmitted partly on a binary additive noise channel with noise realization identical to the source realization of the source-channel problem, and partly on the *same* BSC

(with the same noise realization) of the source-channel problem. We use this BP isomorphism result in order to derive a simple EXIT chart analysis of the BP decoder at each stage of the multistage decoder, under the assumption that the hard decisions from previous stages are correct.

4.1. Rate allocation algorithm

The rate allocation of each bitplane encoder is established offline by using the greedy algorithm described below. Again, we notice that we do not consider adaptive rate allocation: given the source and channel statistics, we run the greedy allocation algorithm in order to design the JSCC coding scheme.

Allocating the number of coded symbols according to the optimallimits, that is,

$$N_p = H(U_p | U_{p+1}, \dots, U_B) \frac{K}{C} \quad (19)$$

yields very bad performance even at very large block length, since it is known that raptor codes converge to very small BER at a fixed (small) gap from capacity on general binary-input symmetric output channels [14]. Therefore, we have to allow for some increment in the coded block lengths, normally referred to as “overhead” in the raptor coding literature. The problem is how to allocate a total overhead among the $B + 1$ stages. In order to do so, we propose the following greedy overhead allocation algorithm.

We initialize the lengths $N_p^{(0)}$ according to their nominal value given by (19). At each iteration of the allocation algorithm, we allocate a given number ΔN of additional coded symbols to one of the $B + 1$ codes. Let $D(N_0, \dots, N_B)$ denote the achieved average distortion of the JSCC scheme when coding lengths N_0, \dots, N_B are used and let $D^{(0)} = D(N_0^{(0)}, \dots, N_B^{(0)})$. Then, for iterations $i = 1, 2, \dots$, do the following.

- (i) For all $p = 0, \dots, B$, compute

$$D_p^{(i)} = D(N_0^{(i-1)}, \dots, N_p^{(i-1)} + \Delta N, \dots, N_B^{(i-1)}). \quad (20)$$

- (ii) Find $\hat{p} = \arg \min_{p=0, \dots, B} D_p^{(i)}$.

- (iii) Let $N_p^{(i)} \leftarrow N_p^{(i-1)}$ for all $p \neq \hat{p}$, and $N_{\hat{p}}^{(i)} \leftarrow N_{\hat{p}}^{(i-1)} + \Delta N$.

- (iv) If $|D_{\hat{p}}^{(i)} - D_Q(B)| \leq \delta$, exit. Otherwise, let $D^{(i)} \leftarrow D_{\hat{p}}^{(i)}$ and go back to 1.

The quantity $\delta > 0$ is the tolerance within which we wish to achieve the target quantization distortion.

In essence, the above algorithm allocates at each iteration a packet of ΔN -coded bits to the bitplane raptor encoder that yields the largest decrease in the overall average distortion. The distortion can be computed either by Monte Carlo simulation, or by using the EXIT chart approximation. The latter method is much faster, but cannot take into account the effect of finite block length and the error propagation between the stages of the multistage decoder.

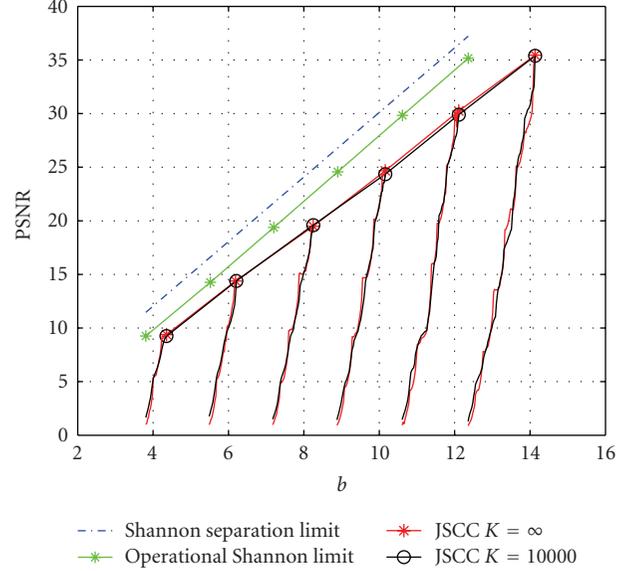


FIGURE 11: PSNR versus b comparison of JSCC scheme for finite block length simulation and infinite block length EXIT approximation for $C = 0.5$.

In Figure 11, we report the comparison between the finite length simulation and the infinite length EXIT approximation for the same setting of B ranging from 1 to 6, the BSC with capacity $C = 0.5$, and source block length $K = 10000$ used throughout the paper. As we can see, the two cases yield almost identical results. This allows us to use the infinite length EXIT approximation to estimate (with very good approximation) a suitable rate allocation among the $B + 1$ stages for the finite length case. Finally, in Figure 12, for the case of $B = 3$, we report the relative overhead $N_p^{(i)}/N_p^{(0)}$ versus $\text{PSNR}^{(i)} = -\log_{10} D^{(i)}$ produced by the greedy allocation algorithm. As one might expect, the greedy algorithm starts increasing the overhead of the sign bitplane and then continues from the most significant to least significant magnitude bitplanes. Eventually, each bitplane is allocated a coding length between 12% and 18% larger than the nominal length (19), in line with standard raptor coding reported results. Furthermore, notice that this scheme tends to give larger overheads to most significant bitplanes, that is, it implicitly implements unequal error protection across the layers, which is a very well-known design approach for multilevel coded modulation with multistage decoding [21].

5. NUMERICAL RESULTS

In this section, we provide both finite length and infinite length results. We considered source block length $K = 10000$ for the finite length results. In all the numerical results of this paper, we considered raptor codes with the “LT” degree distribution [14]

$$\begin{aligned} \Omega(x) = & 0.008x + 0.494x^2 + 0.166x^3 + 0.073x^4 + 0.083x^5 \\ & + 0.056x^8 + 0.037x^9 + 0.056x^{19} + 0.025x^{65} + 0.003x^{66}. \end{aligned} \quad (21)$$

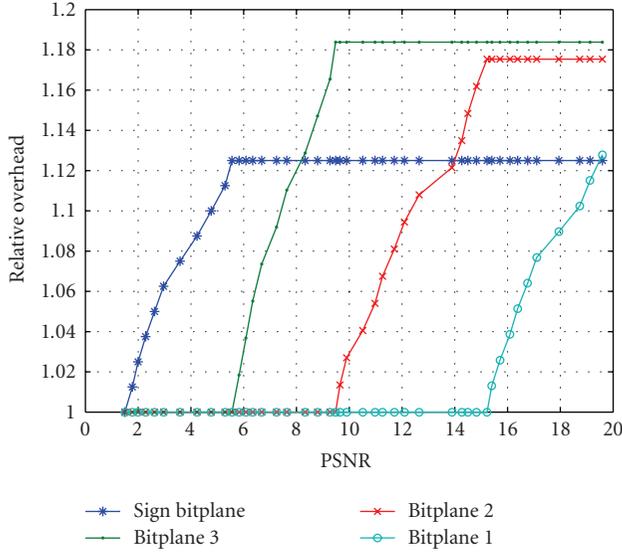


FIGURE 12: Relative overhead $N_p^{(i)}/N_p^{(0)}$ versus $\text{PSNR}^{(i)} = -10 \log_{10} D^{(i)}$ produced by the greedy allocation algorithm for the case $B = 3$. We notice that the bitplane coding overheads are incremented one at a time, in sequence.

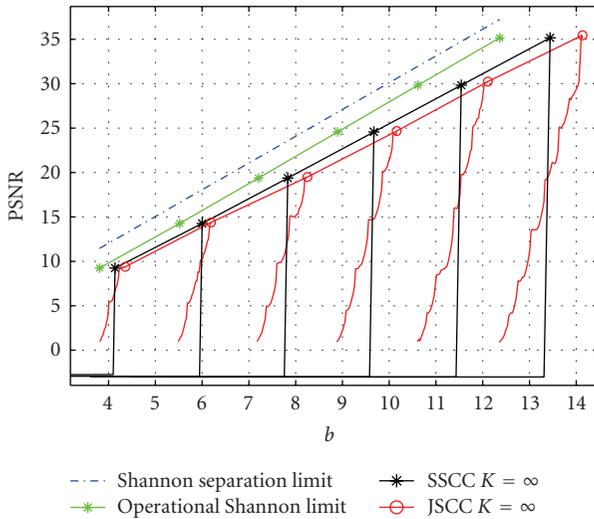


FIGURE 13: JSCC and SSCC infinite block length comparison for $B = 1, 2, 3, 4, 5, 6$ and $C = 0.5$.

As outer code we used a regular high-rate LDPC code with degrees $(2, 100)$ and rate 0.98. The source symbols are estimated after running 100 iterations of the decoding algorithm.

We would like to stress the fact that the LT and LDPC degree distribution polynomials have been chosen without considering any optimization method and that we have averaged over the ensemble of randomly generated raptor codes with the given parameters. In practice, one would carefully design an LDPC graph with good properties for the desired length K and degree distributions.

This section is subdivided in two parts. In the first part we described the results obtained by varying of the bandwidth

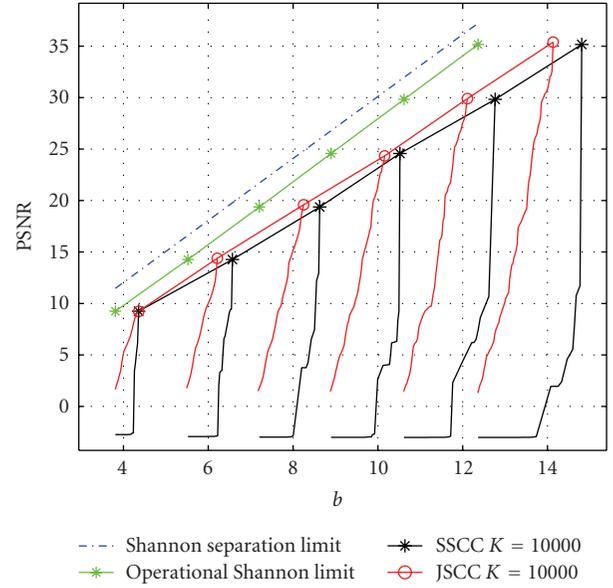


FIGURE 14: JSCC and SSCC finite block length comparison for $B = 1, 2, 3, 4, 5, 6$ and $C = 0.5$.

expansion factor, when the capacity of the BSC is fixed to $C = 0.5$, corresponding to crossover probability $\epsilon = 0.11$. The aim of this section is to compare the performance of families of SSCC and JSCC codes for different values of b , and to see how they approach the operational Shannon limit.

In the second part we examine the behavior of a *single* fixed code, designed for a nominal channel crossover probability and target PSNR, when we vary the channel crossover probability. This set of results illustrates the robustness of a given coding scheme to nonideal channel conditions.

In both subsections we provide results for infinite and finite codeword length cases. The infinite case results have been generated by using the EXIT chart approximation of Appendix B.

5.1. Approaching the operational Shannon limit

In Figure 13, we plot the performance comparison between the proposed JSCC scheme and the SSCC scheme, when infinite codeword length is considered. In this case, the SSCC scheme outperforms the proposed scheme in the sense that it reaches the quantization distortion at slightly lower values of b , for all $B = 1, \dots, 6$. The SSCC schemes show a very sharp transition (“all or nothing” behavior). In contrast, the JSCC schemes reach their quantization PSNR more gradually: as we increase the overhead, the performance improves.

The situation radically changes when we consider finite codeword length. In Figure 14 we plot the performance of JSCC and SSCC schemes for finite block length. In this case, the JSCC schemes outperform their SSCC counterpart. In particular, as we have already remarked, the JSCC performance is almost identical to that for infinite block length, while the SSCC suffers much more evidently from

the residual BER due to finite length practical codes. This also hints that the EXIT approximated analysis yields very faithful results for the JSCC scheme, while it provides optimistic results for the SSCC scheme. This can be explained by the fact that the BER performance of infinite length codes exhibits a very sharp “waterfall” threshold, beyond which the BER is zero, while for finite length the waterfall is smoother.

An important advantage of the JSCC is that the PSNR value gradually increases as b increases, while a sharp threshold effect can be seen in the case of SSCC. In [5] it was shown that, with natural sources such as images, PSNR values lower than peak value were still perceptually acceptable for the JSCC scheme, while the SSCC scheme degrades abruptly also from the perceptual viewpoint.

5.2. Robustness

In the previous set of results, we have fixed the channel capacity and the (quantized) source entropy rate and we have examined families of codes operating at different (b, PSNR) points. Now, we take a complementary view and fix the channel code while letting the channel capacity vary. This setting is relevant when a given code, designed for some nominal channel conditions, is used on a channel of variable quality, and therefore we are interested in the robustness of the performance with respect to the channel parameters. Also, this setting is more akin to the standard way of studying the performance of channel coding, where the BER is plotted as a function of the channel parameters (ϵ in the case of a BSC), for a given channel code.

In order to have a fair comparison between the two schemes, the bandwidth expansion factor (i.e., the code used) has been fixed in the following way: we keep the minimum value of b such that both schemes reach the quantization PSNR on the previous set of results (see Figure 14). In particular we keep $b = 4.3565$ and $b = 14.8079$ for $B = 1$ and $B = 6$, respectively. Since the JSCC scheme needs lower values of b to reach the quantization PSNR in both cases, we add some extra bits to reach the same values of b .

We have examined the two extreme cases of low resolution ($B = 1$) and high resolution ($B = 6$). In Figures 15 and 16 we notice that in both cases the JSCC scheme outperforms the SSCC scheme in terms of PSNR. Moreover, as expected, the PSNR of the SSCC scheme degrades sharply, while the PSNR of the JSCC scheme degrades gradually as the channel crossover probability increases. For example, considering $B = 6$, if ϵ increases from its nominal value 0.11 to a higher value 0.115 the JSCC scheme loses about 6 dB in PSNR, while the SSCC loses 24 dB. We interpret this sharp degradation as an effect of the catastrophic behavior of the entropy coding stage in SSCC, which is greatly mitigated by the linear coding stage in the proposed JSCC scheme.

6. CONCLUSIONS

Unlike most JSCC schemes presented in the literature, which are carefully targeted for specific source and channel pairs, the scheme proposed here can closely approach the rate-distortion separation limit for virtually any well-behaved

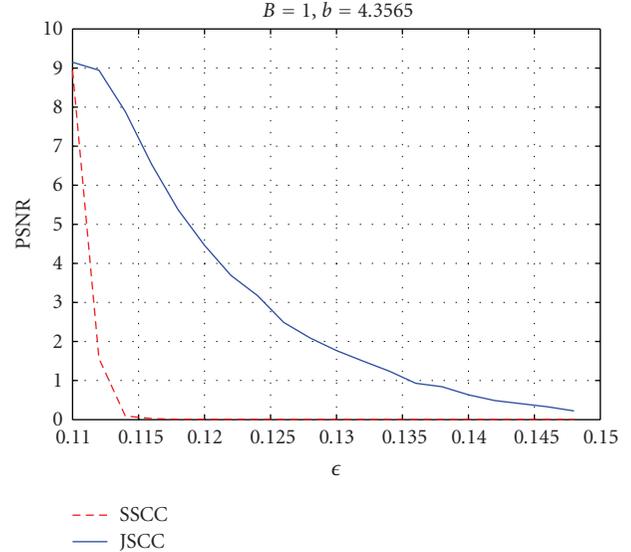


FIGURE 15: Comparison of performance degradation of JSCC and SSCC as the cross-over probability of the BSC increases for $K = 10000$ and $B = 1$.

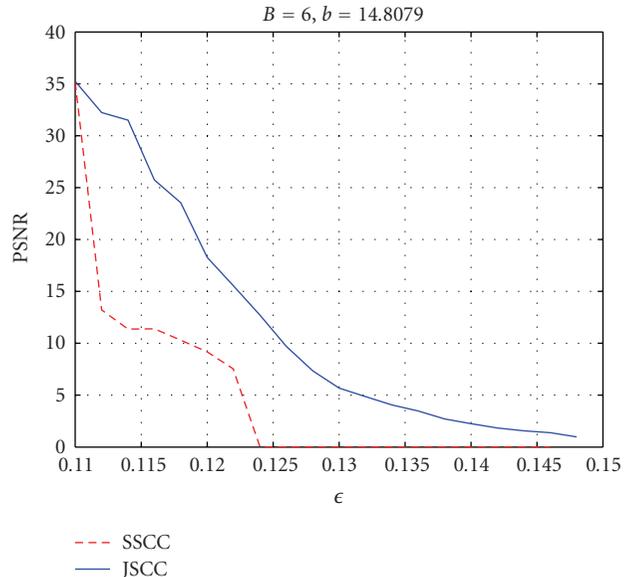


FIGURE 16: Comparison of performance degradation of JSCC and SSCC as the cross-over probability of the BSC increases for $K = 10000$ and $B = 6$.

source under quadratic distortion and any symmetric channel, owing to the universality of entropy-coded quantization and the optimality of linear codes for both data compression and channel coding. Furthermore, we have demonstrated that beyond operating close to optimal, the proposed scheme is better and more robust than a separated approach, especially in the practical case of finite block length.

We wish to conclude this paper with some considerations for future work. Following [5], the JSCC scheme can be applied to any class of sources for which efficient transform

coding has been designed. In particular, images, audio and video are natural and relevant candidates. The scheme takes advantage of the know-how and careful source statistical characterization developed in designing lossy coding standards, and preserves the structure of the transform coder. This makes it easy to introduce the JSCC scheme into practical applications, for example, by introducing a trans-coding stage at the physical layer, while preserving the network architecture and the source coding standards developed at the application layer.

Although we have not pursued this aspect here, the bit-plane layered encoding and multistage successive decoding architectures of the proposed scheme lend themselves quite naturally to a multiresolution, or “embedded,” implementation. For example, it is sufficient to use an embedded scalar quantizer in order to obtain such a scheme: bitplanes will be transmitted in sequence, and the resolution of the reconstructed source improves at each additional layer received.

A different route for future investigation involves the use of nonbinary linear codes. Also for the proposed JSCC scheme, the gap from the Shannon limit increases with the PSNR (resolution). This is due to the fact that each layer needs to be encoded with a fixed overhead, such that the overall overhead increases with the number of layers. As an alternative, we may wish to use a nonbinary raptor code operating over symbols of $B+1$ bits, and mapping directly the quantization indices over the channel symbols. The hope is that the overhead of such nonbinary codes does not depend (or at least depends in a sublinear way) on B . This may lead to better bandwidth expansion gaps at high resolution.

APPENDICES

A. RAPTOR CODES AND BP DECODING

Raptor codes [10] are a class of rateless codes designed for transmission over erasure channels with unknown capacity. They are an extension of Luby Transform codes (LT codes) [18], since they are based on the concatenation of an outer linear code (precode) with an inner LT code. To be compliant with the raptor codes terminology, let us define the *input symbols* as the symbols generated from the *source symbols* by the linear precode encoder, and *output symbols* as the symbols generated from the input symbols by the LT encoder.

Formally a raptor code is defined by the triplet $(K, \mathcal{C}, \Omega(x))$, where K is the source symbols length, \mathcal{C} is a linear encoder $\mathcal{C} : \mathbb{F}_2^K \rightarrow \mathbb{F}_2^n$ and $\Omega(x) = \sum_{j=1}^n \Omega_j x^j$ represents the generating function of the probability distribution $\Omega_1, \dots, \Omega_n$ on $\{1, \dots, n\}$ that generates the LT codewords.

The $(n, \Omega(x))$ LT code ensemble corresponds to the ensemble of $n \times N$ binary matrices, for all $N = 1, 2, \dots$, with columns randomly generated according to $\Omega(x)$, where each matrix yields an encoding mapping.

The operations to generate a generic column of an LT encoding matrix can be summarized in two steps:

- (1) sample the distribution $\{\Omega_1, \dots, \Omega_n\}$ to obtain a weight w between 1 and n ;

- (2) generate the column (v_1, \dots, v_n) uniformly at random from all $\binom{n}{w}$ binary vectors of weight w and length n ;

As shown in [14], it is possible to adapt raptor codes for transmission over memoryless symmetric channels. The decoding is performed by using the classical belief propagation algorithm (see [14] for details).

In this paper, we exploit a high rate LDPC code as the precoder, then the n input nodes can also be seen as the n bitnodes of the LDPC code.

A.1. BP decoder isomorphism

As anticipated in Section 4, there is an interesting isomorphism between the standard channel coding problem when an all zero codeword is transmitted (we refer to this as Scheme A) and the joint source-channel coding problem as defined at each stage of the multistage decoder (we refer to this as Scheme B).

Consider the following unified scheme. Let the vector $\begin{bmatrix} \mathbf{w} \\ \mathbf{z} \end{bmatrix}$ be the output block when a vector \mathbf{w} of length K is channel coded with a systematic raptor code and where \mathbf{z} has length N (i.e., the raptor code rate is equal to $(K/(K+N))$). Let us assume that the output block is transmitted over a hybrid channel such that the first K output symbols are distorted by noise vector \mathbf{u} where $p_k = P(u_k = 1)$ for $k = 1, \dots, K$ and the remaining N output symbols are distorted by the BSC channel noise vector \mathbf{e} where $P(e_k = 1) = \epsilon$, $k = 1, \dots, N$. Then, the hybrid channel is characterized by many BSCs with crossover probabilities p_1, \dots, p_K and ϵ . The channel observation block is then composed of

$$\mathbf{y} = \begin{bmatrix} \mathbf{w} \oplus \mathbf{u} \\ \mathbf{z} \oplus \mathbf{e} \end{bmatrix}. \quad (\text{A.1})$$

Notice that when $\mathbf{w} = \mathbf{0}$ then $\mathbf{z} = \mathbf{0}$, and $\mathbf{y} = \begin{bmatrix} \mathbf{u} \\ \mathbf{e} \end{bmatrix}$. In this case the unified scheme becomes Scheme A. On the other hand, when $\mathbf{w} = \mathbf{u}$, then $\mathbf{y} = \begin{bmatrix} \mathbf{0} \\ \mathbf{z} \oplus \mathbf{e} \end{bmatrix}$, and the unified scheme becomes Scheme B. Let us consider the l th iteration of the BP decoder. We use the following notation (see Figure 17):

- (i) $m_{v,o}^{(l)}$, $m_{o,v}^{(l)}$ are the messages passed from the v th input node to the o th output node and from the o th output node to the v th input node, respectively, of the LT-decoder;
- (ii) $m_{v,c}^{(l)}$, $m_{c,v}^{(l)}$ are the messages passed from the v th input node (the so called variable node in classical LDPC notations) to the c th check node and from the c th check node to the v th input node, respectively, of the LDPC decoder;
- (iii) $\delta_{\text{ldpc}}^{(l),v}$ is the message generated from the v th LDPC input node and passed to the corresponding input node of the LT-decoder;
- (iv) $\delta_{\text{lt}}^{(l),v}$ is the message generated from the v th LT input node and passed to the corresponding input node of the LDPC decoder; and

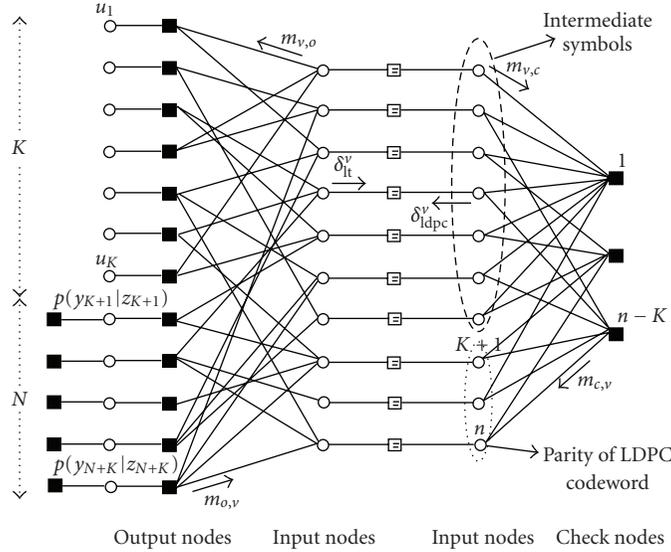


FIGURE 17: Raptor code factor graph for the application of belief propagation.

- (v) Z_o is the LLR of the o th output symbol received from noisy channel; notice for $o = 1, \dots, K$, $Z_o = (-1)^{u_o \oplus w_o} \log((1-p_o)/p_o)$ while $Z_o = (-1)^{e_o \oplus z_o} \log((1-\epsilon)/\epsilon)$ for $o = K+1, \dots, N+K$.

Using the notation above, we can define the updating rules for the LT and the LDPC decoders separately.

For the LT decoder, at the l th iteration, we have

$$\begin{aligned} \tanh\left(\frac{m_{o,v}^{(l)}}{2}\right) &= \tanh\left(\frac{(-1)^{u_o \oplus w_o} \log((1-p_o)/p_o)}{2}\right) \\ &\quad \cdot \prod_{v' \neq v} \tanh\left(\frac{m_{v',o}^{(l)}}{2}\right), \quad o = 1, \dots, K, \\ \tanh\left(\frac{m_{o,v}^{(l)}}{2}\right) &= \tanh\left(\frac{(-1)^{e_o \oplus z_o} \log((1-\epsilon)/\epsilon)}{2}\right) \\ &\quad \cdot \prod_{v' \neq v} \tanh\left(\frac{m_{v',o}^{(l)}}{2}\right), \quad o = K+1, \dots, N+K, \\ m_{v,o}^{(l+1)} &= \delta_{\text{ldpc}}^{(l),v} + \sum_{o' \neq o} m_{o',v}^{(l)}, \quad v = 1, \dots, n, \end{aligned} \quad (\text{A.2})$$

where the product is taken over all input nodes adjacent to o other than v and the summation is taken over all output nodes adjacent to v other than o . For $l = 0$, we set $m_{v,o}^{(0)} = 0$ for $v = 1, \dots, n$.

For the LDPC decoder, at the l th iteration, we have

$$m_{v,c}^{(l)} = \begin{cases} 0 & \text{if } l = 0, \\ \delta_{\text{lt}}^{(l),v} + \sum_{c' \neq c} m_{c',v}^{(l-1)} & \text{if } l \neq 0, \quad v = 1, \dots, n, \end{cases} \quad (\text{A.3})$$

$$\tanh\left(\frac{m_{c,v}^{(l)}}{2}\right) = \prod_{v' \neq v} \tanh\left(\frac{m_{v',c}^{(l)}}{2}\right), \quad c = 1, \dots, n-K. \quad (\text{A.4})$$

The messages $\delta_{\text{lt}}^{(l),v}$ and $\delta_{\text{ldpc}}^{(l),v}$ passed from the LT to the LDPC decoder and from the LDPC to the LT decoder, respectively, are defined by

$$\delta_{\text{lt}}^{(l),v} = \sum_o m_{o,v}^{(l)}, \quad v = 1, \dots, n, \quad (\text{A.5})$$

$$\delta_{\text{ldpc}}^{(l),v} = \sum_c m_{c,v}^{(l)}, \quad v = 1, \dots, n, \quad (\text{A.6})$$

where the summation is taken over all output nodes adjacent to v or overall checknodes adjacent to v .

The overall factor graph (FG) of the proposed decoding algorithm is displayed in Figure 17 for the case of JSCC $\mathbf{w} = \mathbf{u}$. We use Wiberg's notation (see [20]), that is, the FG is a bipartite graph with variable nodes (circles) and function nodes (boxes). A variable node is connected to a function node if the corresponding variable is an argument of the corresponding factor [20]. In our case, the variable nodes correspond to the input symbols of the LT code and to the input symbols of the LDPC code. The function nodes correspond to the output symbols of the LT code and to the check nodes of the LDPC code. To explicitly represent the messages passed between the two decoders at each stage, we split the graph into two parts connected to each other by "equality constraints." Finally, to distinguish between channel outputs received from the equivalent channel and channel outputs received from the noiseless channel, we explicitly represent the source symbols $\mathbf{u} = (u_1, \dots, u_K)$, and the output $\mathbf{y} = (y_{K+1}, \dots, y_{K+N})$ of the noisy channel with input \mathbf{z} . Let us also denote the input block by \mathbf{i} .

As we can see from the updating rules described above and from the factor graph, the decoder can be modeled as two independent factor graphs that exchange information between themselves after each iteration.

Theorem 1. *The magnitude of the BP messages exchanged between input and output symbols for the same Tanner graph is the same for both Schemes A and B. In particular, at BP round*

l , the relationship between the messages passed in Schemes A and B is

$$\begin{aligned} {}_B m_{v,o}^{(l)} &= (-1)^{i_v} {}_A m_{v,o}^{(l)}, \\ {}_B m_{o,v}^{(l+1)} &= (-1)^{i_v} {}_A m_{o,v}^{(l+1)}, \end{aligned} \quad (\text{A.7})$$

(where ${}_A m$ is used to denote messages for Scheme A and ${}_B m$ is used for Scheme B).

Belief propagation equations (A.2)–(A.4) can be also written in an explicit form by using a map γ from the real numbers $(-\infty, \infty)$ to $F_2 \times [0, \infty)$ defined by $\gamma(x) \triangleq (\text{sgn}(x), -\ln \tanh(|x|/2))$. Clearly γ is bijective and there exists an inverse γ^{-1} . Moreover, $\gamma(xy) = \gamma(x) + \gamma(y)$ where addition is component-wise in F_2 and in $[0, \infty)$. Another important property is as follows:

$$\gamma^{-1}\left(\sum_i \gamma((-1)^{b_i} B_i)\right) = \prod_i (-1)^{b_i} \gamma^{-1}\left(\sum_i \gamma(B_i)\right). \quad (\text{A.8})$$

Rewriting (A.2), (A.4) in terms of the γ mapping and using (A.8), we have

$$\begin{aligned} m_{o,v}^{(l)} &= \gamma^{-1}\left(\sum_{v' \neq v} \gamma(m_{v',o}^{(l)}) + \gamma((-1)^{u_o \oplus w_o} \mathcal{P}_o)\right), \\ m_{o,v}^{(l)} &= (-1)^{e_o \oplus z_o} \gamma^{-1}\left(\sum_{v' \neq v} \gamma(m_{v',o}^{(l)}) + \gamma(\xi)\right), \quad (\text{A.9}) \\ m_{c,v}^{(l)} &= \gamma^{-1}\left(\sum_{v' \neq v} \gamma(m_{v',c}^{(l-1)})\right), \end{aligned}$$

where $\mathcal{P}_o \triangleq \log((1 - p_o)/p_o)$ and $\xi \triangleq \log((1 - \epsilon)/\epsilon)$.

Similarly, we have

$$m_{o,v}^{(l)} = (-1)^{n_o \oplus z_o} \gamma^{-1}\left(\sum_{v' \neq v} \gamma(m_{v',o}^{(l)}) + \gamma(\xi)\right), \quad (\text{A.10})$$

$$m_{c,v}^{(l)} = \gamma^{-1}\left(\sum_{v' \neq v} \gamma(m_{v',c}^{(l-1)})\right). \quad (\text{A.11})$$

Proof. To prove the theorem, the BP equations for each scheme will be given explicitly and then starting with the 0th round, the relationship between the messages corresponding to different schemes will be verified. The proof follows by induction, after showing that, given the rule holds for round (l) , it also hold for round $(l+1)$.

BP for Scheme A: In this case we have

$$\begin{aligned} {}_A m_{o,v}^{(l)} &= (-1)^{u_o} \gamma^{-1}\left(\sum_{v' \neq v} \gamma({}_A m_{v',o}^{(l)}) + \gamma(\mathcal{P}_o)\right), \\ {}_A m_{o,v}^{(l)} &= (-1)^{e_o} \gamma^{-1}\left(\sum_{v' \neq v} \gamma({}_A m_{v',o}^{(l)}) + \gamma(\xi)\right), \quad (\text{A.12}) \\ {}_A m_{v,o}^{(l+1)} &= \sum_{o' \neq o} {}_A m_{o',v}^{(l)} + {}_A \delta_{\text{ldpc}}^{(l),v}. \end{aligned}$$

BP for Scheme B: In this case, we have

$$\begin{aligned} {}_B m_{o,v}^{(l)} &= \gamma^{-1}\left(\sum_{v' \neq v} \gamma({}_B m_{v',o}^{(l)}) + \gamma(\mathcal{P}_o)\right), \\ {}_B m_{o,v}^{(l)} &= (-1)^{e_o \oplus z_o} \gamma^{-1}\left(\sum_{v' \neq v} \gamma({}_B m_{v',o}^{(l)}) + \gamma(\xi)\right), \quad (\text{A.13}) \\ {}_B m_{v,o}^{(l+1)} &= \sum_{o' \neq o} {}_B m_{o',v}^{(l)} + {}_B \delta_{\text{ldpc}}^{(l),v}. \end{aligned}$$

Note that in the above equations, we have provided two different versions of equations for $m_{o,v}$ for both Scheme A and Scheme B for values of $1 \leq o \leq K$ and for $K+1 \leq o \leq K+N$. We call these ranges of o the first block and the second block, respectively.

By applying the BP rules at round zero, we have the following relationships between Scheme A and Scheme B:

$$\begin{aligned} {}_A m_{o,v}^{(0)} &= {}_B m_{o,v}^{(0)} = 0, \\ {}_A m_{v,o}^{(1)} &= {}_B m_{v,o}^{(1)} = 0. \end{aligned} \quad (\text{A.14})$$

Then for round zero (A.7) are satisfied.

Now let us assume that the theorem holds for the (l) th round. Then we have the following equations for Round l

$$\begin{aligned} {}_B m_{o,v}^{(l)} &= (-1)^{i_v} {}_A m_{o,v}^{(l)}, \\ {}_B m_{\tilde{o},\tilde{v}}^{(l)} &= (-1)^{i_{\tilde{v}}} {}_A m_{\tilde{o},\tilde{v}}^{(l)}, \quad (\text{A.15}) \\ {}_B m_{v,o}^{(l+1)} &= (-1)^{i_v} {}_A m_{v,o}^{(l+1)}. \end{aligned}$$

Consequently, the equations for Round $(l+1)$ can be written as follows. Letting o and \tilde{o} denote any output symbols, from the first and the second output blocks, respectively, and letting v and \tilde{v} denote any adjacent input nodes, we can write:

$$\begin{aligned} {}_A m_{o,v}^{(l+1)} &= (-1)^{u_o} \gamma^{-1}\left(\sum_{v' \neq v} \gamma({}_A m_{v',o}^{(l+1)}) + \gamma(\mathcal{P}_o)\right), \\ {}_A m_{\tilde{o},\tilde{v}}^{(l+1)} &= (-1)^{e_{\tilde{o}}} \gamma^{-1}\left(\sum_{\tilde{v}' \neq \tilde{v}} \gamma({}_A m_{\tilde{v}',\tilde{o}}^{(l+1)}) + \gamma(\xi)\right), \quad (\text{A.16}) \\ {}_B m_{o,v}^{(l+1)} &= \gamma^{-1}\left(\sum_{v' \neq v} \gamma({}_B m_{v',o}^{(l+1)}) + \gamma(\mathcal{P}_o)\right), \end{aligned}$$

$${}_B m_{\tilde{o},\tilde{v}}^{(l+1)} = (-1)^{e_{\tilde{o}} \oplus z_{\tilde{o}}} \gamma^{-1}\left(\sum_{\tilde{v}' \neq \tilde{v}} \gamma({}_B m_{\tilde{v}',\tilde{o}}^{(l+1)}) + \gamma(\xi)\right).$$

Using the assumption, we can write

$$\begin{aligned} {}_B m_{o,v}^{(l+1)} &= \gamma^{-1}\left(\sum_{v' \neq v} \gamma((-1)^{i_{v'}} {}_A m_{v',o}^{(l+1)}) + \gamma(\mathcal{P}_o)\right), \\ {}_B m_{\tilde{o},\tilde{v}}^{(l+1)} &= (-1)^{e_{\tilde{o}} \oplus z_{\tilde{o}}} \gamma^{-1}\left(\sum_{\tilde{v}' \neq \tilde{v}} \gamma((-1)^{i_{\tilde{v}'}} {}_A m_{\tilde{v}',\tilde{o}}^{(l+1)}) + \gamma(\xi)\right). \end{aligned} \quad (\text{A.17})$$

In order to find a relationship similar to what obtained before, we need to apply (A.8). By applying (A.8), summation coefficient terms such as $(-1)^{i_{v'}}$ or $(-1)^{i_{v'}}$ can be separated from the other summands. By (A.8), it is known that the number of terms in the summation is important. For any o , denote by \mathcal{I}_o the adjacent input node set for the output node o . Then

$$\prod_{v' \in \mathcal{I}_o, v' \neq v} (-1)^{i_{v'}} = \left(\prod_{v' \in \mathcal{I}_o} (-1)^{i_{v'}} \right) (-1)^{i_v} = (-1)^{u_o} (-1)^{i_v}, \quad (\text{A.18})$$

since the \oplus summation of all of the input nodes should give the value of the corresponding output node without additive noise. Similarly, for any \tilde{o} and \tilde{v} , define $\mathcal{I}_{\tilde{o}}$ as the set of adjacent input nodes for the output node \tilde{o} . Then

$$\begin{aligned} \prod_{\tilde{v}' \in \mathcal{I}_{\tilde{o}}, \tilde{v}' \neq \tilde{v}} (-1)^{i_{\tilde{v}'}} &= \left(\prod_{\tilde{v}' \in \mathcal{I}_{\tilde{o}}} (-1)^{i_{\tilde{v}'}} \right) (-1)^{i_{\tilde{v}}} = (-1)^{z_{\tilde{o}}} (-1)^{i_{\tilde{v}}} \\ &\Rightarrow \\ {}_B m_{o,v}^{(l+1)} &= (-1)^{u_o} (-1)^{i_v} \gamma^{-1} \left(\sum_{v' \neq v} \gamma(A m_{v',o}^{(l+1)}) + \gamma(\mathcal{P}_0) \right), \\ {}_B m_{\tilde{o},\tilde{v}}^{(l+1)} &= (-1)^{z_{\tilde{o}}} (-1)^{i_{\tilde{v}}} (-1)^{e_{\tilde{o}} \oplus z_{\tilde{v}}} \gamma^{-1} \left(\sum_{\tilde{v}' \neq \tilde{v}} \gamma(A m_{\tilde{v}',\tilde{o}}^{(l+1)}) + \gamma(\xi) \right) \\ &\Rightarrow \end{aligned} \quad (\text{A.19})$$

$${}_B m_{o,v}^{(l+1)} = (-1)^{i_v} {}_A m_{o,v}^{(l+1)}, \quad (\text{A.20})$$

$${}_B m_{\tilde{o},\tilde{v}}^{(l+1)} = (-1)^{i_{\tilde{v}}} {}_A m_{\tilde{o},\tilde{v}}^{(l+1)}. \quad (\text{A.21})$$

It is worth noting that by applying the l th round hypotheses to (A.5), we obtain ${}_B \delta_{\text{ldpc}}^{(l),v} = (-1)^{i_v} {}_A \delta_{\text{ldpc}}^{(l),v}$; that is, when we only consider the LDPC iterations, Scheme A and Scheme B differ only in the signs of the channel observations. It can easily be shown that, with such an input relationship between two schemes, the messages will be also closely related as follows:

$$\begin{aligned} {}_B m_{v,c}^{(l)} &= (-1)^{i_v} {}_A m_{v,c}^{(l)}, \\ {}_B m_{c,v}^{(l)} &= (-1)^{i_v} {}_A m_{c,v}^{(l)}, \end{aligned} \quad (\text{A.22})$$

for any l .

Then by (A.6) we obtain

$${}_B \delta_{\text{ldpc}}^{(l),v} = (-1)^{i_v} {}_A \delta_{\text{ldpc}}^{(l),v}, \quad (\text{A.23})$$

so that we can write

$$\begin{aligned} {}_A m_{v,o}^{(l+2)} &= \sum_{o' \neq o} {}_A m_{o',v}^{(l+1)} + {}_A \delta_{\text{ldpc}}^{(l+1),v}, \\ {}_B m_{v,o}^{(l+2)} &= \sum_{o' \neq o} {}_B m_{o',v}^{(l+1)} + {}_B \delta_{\text{ldpc}}^{(l+1),v}. \end{aligned} \quad (\text{A.24})$$

Applying (A.23) for round $(l+1)$,

$$\begin{aligned} {}_B m_{v,o}^{(l+2)} &= \sum_{o' \neq o} (-1)^{v_b} {}_A m_{o',v}^{(l+1)} + (-1)^{i_v} {}_A \delta_{\text{ldpc}}^{(l+1),v}, \\ {}_B m_{v,o}^{(l+2)} &= (-1)^{i_v} \left(\sum_{o' \neq o} {}_A m_{o',v}^{(l+1)} + {}_A \delta_{\text{ldpc}}^{(l+1),v} \right), \\ {}_B m_{v,o}^{(l+2)} &= (-1)^{i_v} {}_A m_{v,o}^{(l+2)}. \end{aligned} \quad (\text{A.25})$$

Equations (A.20), (A.21), and (A.25) are identical to the ones assumed in (A.15) of the (l) th round. This completes the proof by induction. \square

Due to Theorem 1, the BER of the pure channel coding scheme (assuming the all zero codeword) is equal to the BER of the source bits in the JSCC scheme. Based on this result, we can obtain an EXIT chart by considering the associated channel coding problem.

B. EXIT CHART APPROXIMATION

The standard analysis tool for graph-based codes under BP iterative decoding, in the limit of infinite block length, is density evolution (DE) [22, 23]. DE is typically computationally heavy, and numerically not very well conditioned. A much simpler approximation of DE consists of the so-called EXIT chart, which corresponds to DE by imposing the restriction that message densities are of some particular form. In particular, the EXIT with Gaussian approximation (GA) assumes that at every iteration the BP message distribution is Gaussian having a particular symmetry condition, which imposes that the variance is equal to 2 times the mean [13]. At this point, densities are uniquely identified by a single parameter, and the approximate DE tracks the evolution of this single parameter across the decoding rounds.

In particular, the EXIT chart tracks the mutual information between the message on a random edge of the graph and the associated binary variable node connected to the edge. By the isomorphism proved before, we know that the JSCC scheme and the “two-channel” scheme have the same performance. For the sake of completeness, in this section we apply the EXIT chart analysis to the “two-channel” case. The resulting EXIT chart applies directly to the JSCC EXIT chart for a binary source. Finally, we briefly discuss how to apply the EXIT chart method to the multistage decoder used by our JSCC scheme. The resulting EXIT chart analysis provides very accurate approximations of the actual JSCC scheme performance, also in the finite (moderately large) block length case (see Figure 11).

For the graph induced by the raptor (LT) distribution, we define the input nodes (also called information bitnodes), the output nodes (also called coded bitnodes) and the checknodes. For LDPC codes, we define just the bitnodes and the checknodes, since any set of bitnodes that form an information set, can be taken as information bitnodes (see Figure 17).

There are different ways of scheduling for raptor decoder.

Practical schedule

Activate in parallel all input LT checknodes, then all LDPC bitnodes (corresponding to LT input nodes), then all LDPC checknodes, then back to the LDPC bitnodes. This forms a complete cycle of scheduling, which is repeated an arbitrarily large number of times. This is the scheduling that was used in our finite length simulation.

Conceptually simple schedule

Activate the LT checknodes. Then, reset the LDPC decoder and treat the messages generated by the LT checknodes as inputs for the LDPC decoder. Perform infinite iterations of the LDPC decoder. After reaching a fixed point of the LDPC decoder, take the LLRs produced for the bitnodes by the LDPC decoder at the fixed-point equilibrium and incorporate these messages as “virtual channel observations” for the input nodes of the LT code. Then, activate all LT input nodes. This provides a complete cycle of scheduling, which is repeated an arbitrarily large number of times. Our EXIT chart equations are obtained assuming this scheduling.

EXIT charts can be seen as a multidimensional dynamic system. We are interested in studying the fixed points and the trajectories of this system. As such, an EXIT chart has state variables. Proceeding to find an EXIT recursion for the conceptually simple schedule, we will denote by \mathbf{x} and \mathbf{y} the state variables of the LT EXIT chart, and by \mathbf{X} and \mathbf{Y} the corresponding state variables for the LDPC EXIT chart.

We use the following notations.

- (i) x_i denotes the mutual information between a message sent along an edge (v, o) with “left-degree” i and the symbol corresponding to the bitnode v , and \mathbf{x} the average of “ x_i ” over all edges (v, o) . Following standard parlance of LDPC codes, we refer to the degree of the bitnode connected to an edge as the left degree of that edge, and to the degree of the checknode connected to an edge as the right degree of that edge.
- (ii) y_j denotes the mutual information between a message sent along an edge (o, v) with “right-degree” j and the symbol corresponding to the bitnode v and \mathbf{y} denotes the average of y_j over all edges (o, v) .
- (iii) X_i denotes the mutual information between a message sent along an edge (v, c) with “left-degree” i and the symbol corresponding to the bitnode v , and \mathbf{X} denotes the average of X_i over all edge (v, c) .
- (iv) Y_j denotes the mutual information between a message sent along an edge (c, v) with “right-degree” j and the symbol corresponding to the bitnode v , and \mathbf{Y} denotes the average of Y_j over all edge (c, v) .
- (v) For an LDPC code, we let $\lambda(x) = \sum_i \lambda_i x^{i-1}$ and $\rho(x) = \sum_j \rho_j x^{j-1}$ denote the generating functions of the edge-centric left- and right-degree distributions, and we let

$$\Lambda(x) = \sum_i \Lambda_i x^i = \frac{\int_0^x \lambda(u) du}{\int_0^1 \lambda(u) du} \quad (\text{B.26})$$

denote the bit-centric left-degree distribution.

- (vi) For an LT code, we let $\iota(x) = \sum_i \iota_i x^{i-1}$ denote the edge-centric degree distribution of the input nodes, $\omega(x) = \sum_j \omega_j x^{j-1}$ denote the edge-centric degree distribution of the “output nodes” or, equivalently, the edge-centric degree distribution of the checknodes. The node-centric degree distribution of the checknodes, is given by

$$\Omega(x) = \sum_i \Omega_i x^i = \frac{\int_0^x \omega(u) du}{\int_0^1 \omega(u) du}. \quad (\text{B.27})$$

- (vii) For the concatenation of the LT code with the LDPC code we also have the node-centric degree distribution of the LT input nodes. This is given by

$$\mathfrak{I}(x) = \sum_i \mathfrak{I}_i x^i = \frac{\int_0^x \iota(u) du}{\int_0^1 \iota(u) du}. \quad (\text{B.28})$$

We consider the class of EXIT functions that make use of Gaussian approximation of the BP messages. Imposing the symmetry condition and Gaussianity, the conditional distribution of each message \mathcal{L} in direction $v \rightarrow c$ is Gaussian $\sim \mathcal{N}(\mu, 2\mu)$, for some value $\mu \in \mathbb{R}_+$. Hence, letting V denote the corresponding bitnode variable, we have

$$I(V; \mathcal{L}) = 1 - \mathbb{E}[\log_2(1 + e^{-\mathcal{L}})] \triangleq J(\mu), \quad (\text{B.29})$$

where $\mathcal{L} \sim \mathcal{N}(\mu, 2\mu)$.

In BP, the message on (v, o) is the sum of all messages incoming to v on all other edges. The sum of Gaussian random variables is also Gaussian, and its mean is the sum of the means of the incoming messages. It follows that

$$\mathbf{x}_i = J((i-1)J^{-1}(\mathbf{y}) + J^{-1}(C)), \quad (\text{B.30})$$

where C is the mutual information (capacity) between the bitnode variable and the corresponding LLR at the (binary-input symmetric output) channel output. In the raptor case, the bitnodes correspond to variables that are observed through a virtual channel by the LDPC decoder. Averaging with respect to the edge-degree distribution, we have

$$\mathbf{x} = \sum_i \iota_i J((i-1)J^{-1}(\mathbf{y}) + J^{-1}(C)). \quad (\text{B.31})$$

As far as checknodes are concerned, we use the well-known quasiduality approximation and replace checknodes with bitnodes by changing mutual information into entropy (i.e., replacing \mathbf{x} by $1 - \mathbf{x}$). Then

$$\mathbf{y}_j = 1 - J((j-1)J^{-1}(1 - \mathbf{x}) + J^{-1}(1 - C)). \quad (\text{B.32})$$

Let us consider now the “two-channel” scenario induced by the JSCC isomorphism. Let K denote the number of source bits, and N denote the number of parity bits. In the corresponding LT code, we have $M = K + N$ output nodes. The first K output nodes are “observed” through a channel with capacity $1 - H$ (i.e., the channel corresponds to the source statistics), while the second N output nodes

are observed through the actual transmission channel, with capacity C .

This channel feature is taken into account by an outer expectation in the EXIT functions. Therefore, the LT EXIT chart can be written in terms of the state equations as follows:

$$\begin{aligned} \mathbf{x} &= \sum_k \sum_i \Lambda_{ki} J((i-1)J^{-1}(\mathbf{y}) + J^{-1}(\downarrow c_k)) \\ &= \sum_k \sum_i \Lambda_{ki} J((i-1)J^{-1}(\mathbf{y}) + kJ^{-1}(\mathbf{Y})), \end{aligned} \quad (\text{B.33})$$

where $K/M = \beta$ and $N/M = 1 - \beta$, and

$$\begin{aligned} \mathbf{y} &= 1 - \sum_j \omega_j [\beta J((j-1)J^{-1}(1-\mathbf{x}) + J^{-1}(H)) \\ &\quad + (1-\beta)J((j-1)J^{-1}(1-\mathbf{x}) + J^{-1}(1-C))], \end{aligned} \quad (\text{B.34})$$

where $\downarrow c_k$ is the mutual information input by the LDPC graph into the LT code graph via the node v of degrees (i, k) as explained in the following.

Equation (B.34) follows from the fact that a random edge (o, v) is connected with probability β to a source bit (i.e., to the channel with capacity $1 - H$), while with probability $1 - \beta$ to a parity bit (i.e., to the channel with capacity C).

Consider an LDPC bitnode v that coincides with an input node of the LT code. The degree of this node with respect to the LDPC graph is k , while the degree of v with respect to the LT graph is i . For a randomly generated graph, and a random choice of v , k and i are independent random variables, with joint distribution given by

$$\prod_{i,k} \mathfrak{I}_i \Lambda_k. \quad (\text{B.35})$$

The mutual information input by the LT graph into the LDPC graph via the node v of degrees (i, k) is given by

$$\uparrow c_i = J(iJ^{-1}(\mathbf{y})). \quad (\text{B.36})$$

Therefore, the LDPC EXIT chart can be written in terms of the following state equations:

$$\begin{aligned} \mathbf{X} &= \sum_k \sum_i \lambda_k \mathfrak{I}_i J((k-1)J^{-1}(\mathbf{Y}) + J^{-1}(\uparrow c_i)) \\ &= \sum_k \sum_i \lambda_k \mathfrak{I}_i J((k-1)J^{-1}(\mathbf{Y}) + iJ^{-1}(\mathbf{y})), \\ \mathbf{Y} &= 1 - \sum_\ell \rho_\ell J((\ell-1)J^{-1}(1-\mathbf{X})). \end{aligned} \quad (\text{B.37})$$

The mutual information input by the LDPC graph into the LT graph via the node v of degrees (i, k) is given by

$$\downarrow c_k = J(kJ^{-1}(\mathbf{Y})). \quad (\text{B.38})$$

Equations (B.37), (B.33), and (B.34) form the state equations of the global EXIT chart of the concatenated LT-LDPC graph, where the state variables are \mathbf{x} , \mathbf{y} , \mathbf{X} , \mathbf{Y} , while

the parameters are H , C and β , and the degree sequences ω , ι , ρ , and λ .

Finally, in order to get the reconstruction distortion we need to obtain the conditional probability density function (pdf) of the LLRs output by BP for the source bits. Under the Gaussian approximation, the LLR is Gaussian. Let μ_j denote the mean of the LLR of a source bitnode corrected to a checknode of degree j , given by

$$\mu_j = J^{-1}(1 - J(jJ^{-1}(1 - \mathbf{x}))) + J^{-1}(1 - H). \quad (\text{B.39})$$

Then, we approximate the average BER of the source bits as

$$P_b = \sum_j \Omega_j Q\left(\sqrt{\frac{\mu_j}{2}}\right). \quad (\text{B.40})$$

B.1. Multilayer EXIT chart analysis

For each bitplane, at every location, the entropy of the bit depends on the realization of the bits at the previous (more significant) bitplanes. We are then in the presence of a ‘‘time-varying’’ memoryless channel in the corresponding channel coding problem. To develop the equations for the multilayer case, we use the same idea described in the previous section, namely, an outer expectation. At the $(B - p)$ th most significant level, the previous corresponding bit locations might have 2^p different combinations, with possibly different probabilities which are denoted by $\gamma_{B-p}(m)$ for $0 \leq m \leq 2^p - 1$.

Let $H(x_{B-p}|m)$ denote the conditional entropy of a bit at the $(B - p)$ th most significant plane given that the value of the corresponding more significant bits’ combination is m .

At the $(B - p)$ th most significant level, the channel has capacity C with probability $1 - \beta$, while it has capacity $1 - H(x_{B-p}|m)$ for $0 \leq m \leq 2^p - 1$ with probability $\beta \gamma_{B-p}(m)$.

Following this approach we can modify (B.34) for the decoding of the $(B - p)$ th magnitude plane. It is worth noting that when the $(B - p)$ th biplane is considered, we assume that the sign plane and p (from B to $B - p - 1$) magnitude planes have been processed. Since it is known that the magnitude plane model does not depend on the sign plane, we take into account 2^p different realizations.

That is, we have

$$\begin{aligned} \mathbf{y} &= 1 - \sum_j \omega_j \left\{ (1 - \beta)J((j-1)J^{-1}(1-\mathbf{x}) + J^{-1}C) \right. \\ &\quad \left. + \beta \left(\sum_{m=0}^{2^p-1} \gamma_{B-p}(m) J((j-1)J^{-1}(1-\mathbf{x}) + J^{-1}(H(x_{B-p}|m))) \right) \right\}, \\ &\quad p = 1, \dots, B - 1. \end{aligned} \quad (\text{B.41})$$

We would like to underline that the sign bitplane and the most important biplane do not depend on any other bitplane, and so $\gamma_B(0) = \gamma_0(0) = 1$.

Similar to (B.41), we can update (B.40) as follows:

$$P_b = \sum_j \Omega_j Q\left(\sqrt{\frac{\mu_j}{2}}\right), \quad (\text{B.42})$$

where

$$\mu_j = J^{-1}(1 - J(jJ^{-1}(1 - x))) + \sum_{m=0}^{2^p-1} \gamma_{B-p}(m) J^{-1}(1 - H(x_{B-p}|m)). \quad (\text{B.43})$$

Note that a genie-aided scheme was assumed for the EXIT analysis, where there is no error-propagation between layers. In fact, it is not possible to take into account error propagation using the EXIT chart, since the underlying assumption is that the message exchanged at each iteration of the BP is a true LLR, that is, an LLR computed on the basis of the correct conditional probabilities. Decision errors, instead, would feed the decoder at a lower stage with “false” a priori probabilities.

As was done in the finite-length case, we will use soft reconstruction for the infinite-length case where the conditional mean estimator of the reconstruction points will be calculated using the fact that the source bit LLRs have the symmetric Gaussian distribution. From the EXIT chart, we can obtain the mean μ of the Gaussian approximation of the conditional pdf of any LLR in the graph. Hence, this can be used to compute the MMSE of the soft-reconstruction estimator (16).

ACKNOWLEDGMENTS

This research was supported in part by the National Science Foundation under Grants ANI-03-38807, CNS-06-25637 and NeTS-NOSS-07-22073 and in part by the USC Annenberg Graduate Fellowship Program.

REFERENCES

- [1] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley-Interscience, New York, NY, USA, 1991.
- [2] A. Ortega and K. Ramchandran, “Rate-distortion methods for image and video compression,” *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 23–50, 1998.
- [3] J. Ziv, “On universal quantization,” *IEEE Transactions on Information Theory*, vol. 31, no. 3, pp. 344–347, 1985.
- [4] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards, and Practices*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2001.
- [5] M. Fresia and G. Caire, “A practical approach to lossy joint source channel coding,” in *Proceedings of the Information Theory and Applications Workshop (ITA '07)*, San Diego, Calif, USA, February 2007.
- [6] M. Fresia and G. Caire, “A practical approach to lossy joint source-channel coding,” submitted to *IEEE Transactions on Information Theory*.
- [7] I. H. Witten, R. M. Neal, and J. G. Cleary, “Arithmetic coding for data compression,” *Communications of the ACM*, vol. 30, no. 6, pp. 520–540, 1987.
- [8] R. E. Krichevsky and V. K. Trofimov, “The performance of universal encoding,” *IEEE Transactions on Information Theory*, vol. 27, no. 2, pp. 199–207, 1981.
- [9] G. Caire, S. Shamai, and S. Verdú, “Noiseless data compression with low-density parity-check codes,” in *Advances in Network Information Theory*, P. Gupta, G. Kramer, and A. J. van Wijngaarden, Eds., vol. 66 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pp. 263–284, American Mathematical Society, Providence, RI, USA, 2004.
- [10] A. Shokrollahi, “Raptor codes,” *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.
- [11] B. Hochwald and K. Zeger, “Tradeoff between source and channel coding,” *IEEE Transactions on Information Theory*, vol. 43, no. 5, pp. 1412–1424, 1997.
- [12] R. G. Gallager, *Information Theory and Reliable Communication*, John Wiley & Sons, New York, NY, USA, 1968.
- [13] S. T. Brink, “Convergence behavior of iteratively decoded parallel concatenated codes,” *IEEE Transactions on Communications*, vol. 49, no. 10, pp. 1727–1737, 2001.
- [14] O. Etesami and A. Shokrollahi, “Raptor codes on binary memoryless symmetric channels,” *IEEE Transactions on Information Theory*, vol. 52, no. 5, pp. 2033–2051, 2006.
- [15] T. J. Richardson and R. L. Urbanke, “The capacity of low-density parity-check codes under message-passing decoding,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 599–618, 2001.
- [16] A. Roumy, S. Guemghar, G. Caire, and S. Verdú, “Design methods for irregular repeat-accumulate codes,” *IEEE Transactions on Information Theory*, vol. 50, no. 8, pp. 1711–1727, 2004.
- [17] R. Palanki and J. S. Yedidia, “Rateless codes on noisy channels,” in *Proceedings of the Conference on Information Sciences and Systems (CISS '04)*, Princeton, NJ, USA, March 2004.
- [18] M. Luby, “LT codes,” in *Proceedings of the 43rd Annual Symposium on Foundations of Computer Science (FOCS '02)*, pp. 271–280, Vancouver, Canada, November 2002.
- [19] R. M. Tanner, “A recursive approach to low complexity codes,” *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, 1981.
- [20] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.
- [21] U. Wachsmann, R. F. H. Fischer, and J. B. Huber, “Multilevel codes: theoretical concepts and practical design rules,” *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1361–1391, 1999.
- [22] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, “Analysis of low density codes and improved designs using irregular graphs,” in *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC '98)*, pp. 249–258, Dallas, Tex, USA, May 1998.
- [23] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619–637, 2001.

Research Article

A Simple Scheme for Belief Propagation Decoding of BCH and RS Codes in Multimedia Transmissions

Marco Baldi and Franco Chiaraluce

Dipartimento di Elettronica, Intelligenza artificiale e Telecomunicazioni (DEIT), Facoltà di Ingegneria, Università Politecnica delle Marche, via Brecce Bianche, 60131 Ancona, Italy

Correspondence should be addressed to Marco Baldi, m.baldi@univpm.it

Received 1 April 2008; Accepted 30 June 2008

Recommended by Massimiliano Laddomada

Classic linear block codes, like Bose-Chaudhuri-Hocquenghem (BCH) and Reed-Solomon (RS) codes, are widely used in multimedia transmissions, but their soft-decision decoding still represents an open issue. Among the several approaches proposed for this purpose, an important role is played by the iterative belief propagation principle, whose application to low-density parity-check (LDPC) codes permits to approach the channel capacity. In this paper, we elaborate a new technique for decoding classic binary and nonbinary codes through the belief propagation algorithm. We focus on RS codes included in the recent CDMA2000 standard, and compare the proposed technique with the adaptive belief propagation approach, that is able to ensure very good performance but with higher complexity. Moreover, we consider the case of long BCH codes included in the DVB-S2 standard, for which we show that the usage of “pure” LDPC codes would provide better performance.

Copyright © 2008 M. Baldi and F. Chiaraluce. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

In spite of their age, classic families of linear block codes, like Bose-Chaudhuri-Hocquenghem (BCH) and Reed-Solomon (RS) codes, continue to be adopted in many telecommunication standards. For example, the most recent European standard for satellite digital video broadcasting (DVB-S2) includes an error correction scheme based on the concatenation of an outer BCH code followed by an inner low-density parity-check (LDPC) code [1]. Classic coding schemes are adopted also for broadcast services implemented over different networks, like packet-switched mobile networks: the American CDMA2000 standard includes RS codes for the deployment of high-rate broadcast data services [2].

Encoding and decoding of BCH and RS codes can be accomplished through very simple circuits that implement operations over finite fields. However, classic decoding techniques rely on hard-decision decoders that allow the correction of up to $\lfloor (d-1)/2 \rfloor$ errors, where d is the code minimum distance and $\lfloor x \rfloor$ the greatest integer smaller than or equal to x . On the contrary, the use of channel measurements in soft-decision decoders can improve significantly the error correction capability, thus approaching, for high

signal-to-noise ratios, the theoretical limit of correcting $d-1$ errors [3].

A good review of soft-decision decoding algorithms applied to linear block codes, and RS codes in particular, can be found in [4], where a new approach is also proposed, based on the iterative belief propagation (BP) algorithm. Thanks to the adoption of BP, LDPC codes can approach the maximum likelihood (ML) performance, while maintaining low decoding complexity [5].

The BP algorithm works on Tanner graphs that are bipartite graphs with variable nodes and check nodes corresponding to code bits and parity equations, respectively. An edge connecting the variable node v_i with the check node z_j exists if and only if the parity-check matrix associated with the Tanner graph has a 1 at position (j, i) .

In order to achieve a good performance, BP decoding needs a parity-check matrix with the following characteristics: (i) sparsity (that is, in fact, inherent in LDPC codes), (ii) absence of short cycles in the associated Tanner graph, and (iii) regular or optimized irregular row and column weight distributions. Such properties are rarely ensured by parity-check matrices of binary cyclic codes. For example, it can be shown that (n, k, d) -BCH codes, where $n = 2^m - 1$ is the

codeword length and k the number of information bits, with rate greater than or equal to $1/2$ and $3 \leq m \leq 8$, cannot have Tanner graphs free of length-4 cycles [6].

For these reasons, many alternative solutions have been proposed in the literature for effectively applying BP decoders to generic linear block codes, binary cyclic codes, or specific classes of cyclic codes [7–15]. All these techniques aim at finding, through different approaches, a graph representation for the code that is well suited for BP decoding.

In [7, 8], for example, the generalized parity-check matrix (GPCM) is adopted to reduce the number of short cycles. Such approach has been further investigated in [9], where an algorithm is presented that achieves a representation free of length-4 cycles. All techniques based on GPCMs, however, require the introduction of auxiliary bits that do not correspond to transmitted bits and, therefore, do not yield information on the channel status; this fact, in turn, may cause performance degradation. In [10], it is demonstrated that Vardy's technique can be used to find sparse parity-check matrices for Reed-Solomon codes.

Maybe the best technique for soft-decoding of linear block codes characterized by dense parity-check matrices is the "adaptive belief propagation" algorithm [4, 11]. The rationale of this method lies in varying the parity-check matrix at each iteration, according to the bit reliabilities, such that the unreliable bits correspond to a sparse submatrix, suitable for the BP algorithm. Actually, significant performance improvements with respect to hard-decision decoding and standard BP decoding can be achieved through this method. As a counterpart, its complexity is rather high, and often unsuitable for implementation in real-time (or almost-real-time) applications, as those required in many multimedia transmissions. As described in [4], this method requires to implement a Gaussian elimination, at each iteration of the decoding algorithm, that generally yields a great amount of operations. Complexity can be somehow reduced by combining this approach with the Koetter-Vardy algebraic soft-decision decoding algorithm [12], but it remains, in any case, rather high.

In [13], instead, a different approach is tempted: the author proposes to use the so-called extended parity-check matrix (EPCM) in order to obtain a regular Tanner graph associated with the code. The notion of EPCM will be reminded in Section 2.2; the method is very simple and allows to obtain matrices more suitable, in principle, for applying BP decoding. Unfortunately, however, for most codes, the performance achievable through this method is very poor. Examples will be given in Section 4.

Keeping in mind, on one hand, the simplicity of the EPCM-based techniques and, on the other hand, the astonishing results of adaptive BP, in this paper, we extend an alternative approach we have recently presented [14, 15], based on "spread" parity-check matrices. We improve such approach through the adoption of an adaptive version of the algorithm, where adaptation, however, is much simpler than in [4].

At first, we apply the new method to the case of short BCH codes where, we show, it is able to achieve very good

performance if compared with EPCM-based techniques. Short codes are often used in multimedia communications with very rigorous requests on delay and complexity [16]. On the other hand, as mentioned, some important telecommunication standards adopt nonbinary cyclic codes or very long codes for matching the length of LDPC codes in concatenated schemes. For this reason, we also study the applicability of the proposed procedure to RS codes, like those included in the CDMA2000 standard, and to long BCH codes, like the outer codes in the DVB-S2 standard.

The paper is organized as follows. In Section 2 we analyze the parity-check matrix of the considered codes and present some options for its modification. In Section 3 we describe the standard decoding algorithm and the new version working on the spread code. In Section 4 the proposed technique is assessed through numerical simulations. Finally, Section 5 concludes the paper.

2. PARITY-CHECK MATRICES OF LINEAR BLOCK CODES

In order to optimize the parity-check matrix for application of belief propagation decoding algorithms, we consider first binary cyclic codes that represent particular cases of linear block codes. We obtain an alternative representation of their parity-check matrix by considering its cyclic nature. The proposed technique can be applied to BCH codes and can be extended to other families of codes, as will be shown in the following sections.

Given a binary cyclic code $C(n, k)$ with length n , dimension k , and redundancy $r = n - k$, each codeword \mathbf{c} can be associated to a polynomial $c(x)$ over $\text{GF}_2[x] \text{ mod } (x^n + 1)$. Moreover, all the shifted versions of $c(x)$, that is, $x^i c(x)$, are valid codewords, due to the cyclic property of the code. Within the set of code polynomials in C , there is a unique monic polynomial $g(x)$, with minimal degree $r < n$, called the generator polynomial of C . Every codeword polynomial $c(x) \in C$ can be expressed uniquely as $c(x) = m(x)g(x) \text{ mod } (x^n + 1)$, where $m(x) \in \text{GF}_2[x]$ is a polynomial of degree $< k$. The generator polynomial $g(x)$ of C is a factor of $(x^n + 1)$, and there exists a parity polynomial with degree k , $h(x)$, such that $g(x)h(x) = x^n + 1$. Moreover, since $g(x)$ divides $c(x)$, the following relationship is satisfied:

$$c(x)h(x) \equiv 0 \text{ mod } (x^n + 1) \quad \forall c(x) \in C. \quad (1)$$

2.1. Standard parity-check matrix

The standard form of the parity-check matrix (PCM) of a binary cyclic code is as follows [17]:

$$\mathbf{H} = \begin{bmatrix} h_k & \cdots & h_1 & h_0 & 0 & \cdots & 0 \\ 0 & h_k & \cdots & h_1 & h_0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & h_k & \cdots & h_1 & h_0 \end{bmatrix}, \quad (2)$$

where h_i , $i = 0, \dots, k$, are the binary coefficients of $h(x)$.

The form (2) of the parity-check matrix is not suitable for BP decoding: it contains many length-4 cycles and it has irregular and nonoptimized column weights.

2.2. Extended parity-check matrix

The parity-check matrix (2) is a (nonsingular) submatrix of the extended parity-check matrix (EPCM) of a cyclic code that has the following form [13]:

$$\mathbf{H}^E = \begin{bmatrix} h_k & \cdots & h_1 & h_0 & 0 & \cdots & 0 \\ 0 & h_k & \cdots & h_1 & h_0 & 0 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \\ 0 & \cdots & 0 & h_k & \cdots & h_1 & h_0 \\ h_0 & 0 & \cdots & 0 & h_k & \cdots & h_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & h_1 & h_0 & 0 & \cdots & 0 & h_k \end{bmatrix}. \quad (3)$$

\mathbf{H}^E is a binary circulant matrix, where each row is obtained through a cyclic shift of the previous row. The form (3) of the parity-check matrix corresponds to a regular Tanner graph, so, at least in principle, it is more suitable for BP decoding.

However, such form of the parity-check matrix contains a number of short cycles even higher than matrix (2). If the number of nonnull coefficients of $h(x)$ increases (e.g., when long or high-rate codes are considered, like in the DVB-S2 standard [1]), \mathbf{H}^E has an extremely high number of short cycles that deteriorate performance.

We also observe that \mathbf{H}^E has the same density of \mathbf{H} , but its Tanner graph contains a larger number of edges; therefore, the decoding complexity is increased by a factor of n/r .

2.3. Reduced parity-check matrix

In order to find a sparser representation for the code parity-check matrix, it is possible to adopt a very simple iterative algorithm that aims at deriving, from the EPCM, a “reduced parity-check matrix” (RPCM), \mathbf{H}^R , whose density is lower than that of \mathbf{H}^E . This can be done by combining linearly (that is, summing up) couples of rows in \mathbf{H}^E . The algorithm relies on the observation that, for a circulant matrix, the number of overlapping 1’s between its first row and each other row can be easily computed in terms of the periodic autocorrelation function of the first row.

As an example, Figure 1 shows the periodic autocorrelation function of the first row of \mathbf{H}^E (denoted as \mathbf{h}_1 in the following) for the (127, 71)-BCH code. We observe that, for a null shift, the periodic autocorrelation function takes the (maximum) value of 48 that coincides with the Hamming weight of \mathbf{h}_1 , denoted as w_1 in the following. We also notice that, for a shift value equal to 4, the periodic autocorrelation function assumes its maximum out-of-phase (that is for a nonnull shift) value, which is equal to 32. It follows that, by summing up the fifth row of \mathbf{H}^E to its first row, we obtain a new vector, \mathbf{h}_2 , with Hamming weight $w_2 = 2(48 - 32) = 32$.

The new vector \mathbf{h}_2 provides a valid parity-check equation for the original code, since it is obtained as a linear combination of parity-check vectors. Due to the cyclic nature of the code, any cyclically shifted version of \mathbf{h}_2 is a parity-check vector as well. Therefore, \mathbf{h}_2 can be used to obtain a new parity-check matrix in circulant form, with reduced density with respect to \mathbf{H}^E . In general, given the vector \mathbf{h}_i ,

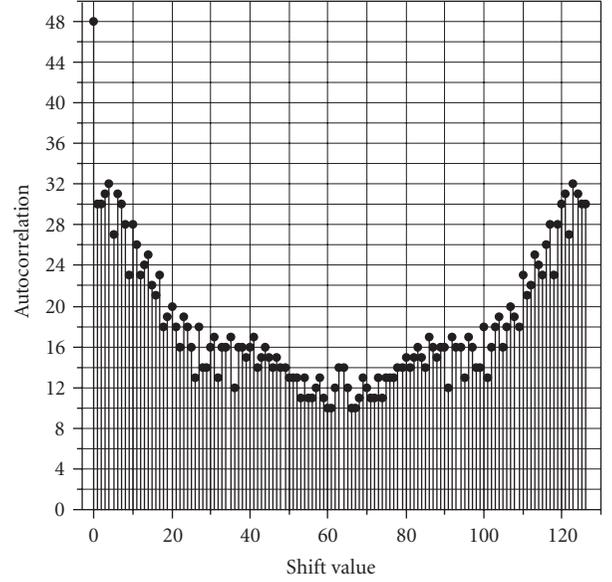


FIGURE 1: Periodic autocorrelation function of the first row of \mathbf{H}^E for the (127, 71)-BCH code.

it is possible to reduce its density through this procedure if its periodic autocorrelation function has a maximum value (out of the null shift) greater than half of its Hamming weight, $w_i/2$. So, we can apply an iterative density reduction algorithm as follows.

- (1) Set $i = 1$; initialize \mathbf{h}_1 as the first row of \mathbf{H}^E and w_1 as its Hamming weight.
- (2) Calculate the periodic autocorrelation function of \mathbf{h}_i and its maximum value a for a shift $\nu > 0$. If $a > w_i/2$, go to step (3), otherwise, stop and output \mathbf{h}_i .
- (3) Calculate $\mathbf{h}_{i+1} = \mathbf{h}_i + \mathbf{h}_i^\nu$ (where \mathbf{h}_i^ν represents the cyclically shifted version of \mathbf{h}_i by ν positions), and its Hamming weight $w_{i+1} = 2(w_i - a)$. Increment i and go back to step (2).

When the algorithm stops, it outputs a binary vector \mathbf{h}_i with density less than or equal to that of \mathbf{h}_1 . \mathbf{h}_i is used to obtain the reduced parity-check matrix in the form of a circulant matrix having \mathbf{h}_i as its first row.

We say that the algorithm is successful when the RPCM has a reduced density with respect to the EPCM, that is, the algorithm has executed step (3) at least once.

2.4. Spread parity-check matrix

After having derived the reduced parity-check matrix \mathbf{H}^R , the effectiveness of BP decoding can be further improved by “spreading” the code at the decoder by means of a simple s -times repetition of each codeword (affected by channel noise) of the original code. Obviously, the “spread code” must have a valid parity-check matrix. For this purpose, we

identify a set of s binary circulant matrices, \mathbf{H}_i^S , $i = 1, \dots, s$, that sum into \mathbf{H}^R . In formula

$$\mathbf{H}^R = \sum_{i=1}^s \mathbf{H}_i^S. \quad (4)$$

If \mathbf{c} is an n -bit codeword of the original code, it must be

$$\mathbf{H}^R \cdot \mathbf{c}^T = \left(\sum_{i=1}^s \mathbf{H}_i^S \right) \cdot \mathbf{c}^T = \mathbf{0}, \quad (5)$$

where superscript T denotes vector transposition, and $\mathbf{0}$ represents the $n \times 1$ null vector. Let us consider the following “spread parity-check matrix” (SPCM):

$$\mathbf{H}^S = [\mathbf{H}_1^S | \mathbf{H}_2^S | \dots | \mathbf{H}_s^S], \quad (6)$$

and the following spread codeword, obtained by repeating s times the generic codeword \mathbf{c} :

$$\mathbf{c}^S = [\mathbf{c} | \mathbf{c} | \dots | \mathbf{c}]. \quad (7)$$

It follows from these definitions that

$$\begin{aligned} \mathbf{H}^S \cdot (\mathbf{c}^S)^T &= [\mathbf{H}_1^S | \mathbf{H}_2^S | \dots | \mathbf{H}_s^S] \cdot [\mathbf{c} | \mathbf{c} | \dots | \mathbf{c}]^T \\ &= [\mathbf{H}_1^S \cdot \mathbf{c}^T + \mathbf{H}_2^S \cdot \mathbf{c}^T + \dots + \mathbf{H}_s^S \cdot \mathbf{c}^T] \\ &= \mathbf{H}^R \cdot \mathbf{c}^T = \mathbf{0}. \end{aligned} \quad (8)$$

Therefore, \mathbf{H}^S is a valid parity-check matrix for the spread code, and it is used by the modified decoding algorithm to work on a more efficient graph.

In order to minimize the density of 1 symbols in \mathbf{H}^S , we choose particular sets $\{\mathbf{H}_1^S, \mathbf{H}_2^S, \dots, \mathbf{H}_s^S\}$ where, according to (4), the blocks \mathbf{H}_i^S have Hamming weights that sum into the Hamming weight of \mathbf{H}^R . This way, the density of 1 symbols in \mathbf{H}^S is reduced by a factor s with respect to that of \mathbf{H}^R . We observe that, in this case, the number of edges in the Tanner graph relative to \mathbf{H}^S is the same in the Tanner graph relative to \mathbf{H}^R ; therefore, the decoding complexity is practically unchanged.

The spreading criterion we adopt corresponds to spreading the i th column of \mathbf{H}^R into s columns of \mathbf{H}^S (those at positions $i, i+n, i+2n, \dots, i+(s-1)n$) whose supports are contained in the support of the original column.

In other terms, we spread the 1 symbols in the i th column of \mathbf{H}^R among its corresponding s columns in \mathbf{H}^S . If we denote as d_i the Hamming weight of the i th column of \mathbf{H}^R , the Hamming weights of the corresponding set of columns in the spread matrix, at positions $i, i+n, i+2n, \dots, i+(s-1)n$, must take values such that $\sum_{j=0}^{s-1} d_{i+jn}^S = d_i$, where d_l^S denotes the Hamming weight of the l th column of \mathbf{H}^S . As for the values d_l^S , they are chosen in a nearly uniform way, that is, $d_{i+jn}^S \simeq d_i/s$, $j = 0, \dots, s-1$. More precisely, we fix $d_{i+jn}^S = d_i/s$ when s divides d_i ; otherwise, the d_{i+jn}^S values may be slightly different in order to ensure that they sum up to d_i .

It is important to observe that the original code and its transmission rate are not altered by the spreading: the spread

code is used only inside the decoder, with the aim of decoding better the original code.

It should be also noted that the proposed procedure for spreading the parity-check matrix represents a special case of column splitting, presented in [18]; the target of column splitting, however, is to design new finite-geometry LDPC codes, while our aim is to use the spread code to improve decoding of the original code.

2.5. Adaptive spread parity-check matrix

Inspired by the adaptive belief propagation approach [4], we have also implemented an adaptive version of our spread parity-check matrix that evolves during decoding iterations on the basis of the values of the bit reliabilities.

Adaptation of the SPCM consists in dynamically changing the “spreading profile” that is the set of values d_{i+jn}^S , $j = 0, \dots, (s-1)$ in such a way to produce unitary weight columns in the spread Tanner graph that correspond to the least reliable bits.

This only implies rearranging of some edges in the Tanner graph (i.e., changing the variable nodes these edges are connected to); thus, it does not require sums of rows and does not alter the total number of 1 symbols in the parity-check matrix that remains sparse. For these reasons, the adaptation technique we propose has very low complexity, contrary to that used in adaptive belief propagation that is based on Gaussian elimination.

For adapting the SPCM at each iteration, we propose the following criterion: the 1 symbols in each column of the RPCM corresponding to the r least reliable bits are spread in a 1-weight column in each block of the SPCM, except the last block, in which a column with weight greater than one can appear (due to the fact that it must contain all the remaining 1 symbols that are present in the RPCM column). In formulae:

$$d_{i+jn}^S = \min \left(1, d_i - \sum_{m=0}^{j-1} d_{i+mn}^S \right), \quad j = 0, \dots, (s-2), \quad (9)$$

$$d_{i+jn}^S = d_i - \sum_{m=0}^{s-2} d_{i+mn}^S, \quad j = s-1.$$

For the $k = n - r$ remaining bits, instead, we adopt again a uniform spreading profile, that is, $d_{i+jn}^S \simeq d_i/s$, $j = 0, \dots, (s-1)$. The spreading profile is updated at the end of each decoding iteration and the new SPCM, for the subsequent step, is obtained from the RPCM.

In the following, we will denote as ASPCM the adaptive version of the SPCM.

2.6. Application to Reed-Solomon codes

Reed-Solomon codes are nonbinary BCH codes, included in many telecommunication standards and in a huge variety of applications. Each RS code is defined over the finite field GF_{2^q} , with q a positive integer, and has length $N = 2^q - 1$, dimension K , and redundancy $R = N - K$. Its correction capability is $t = \lfloor (R+1)/2 \rfloor$ [19]. Shortened RS codes are

often used to adapt the code length to the values required in practical applications.

Given a primitive polynomial, $p(x)$, with degree q , and one of its roots, α , the latter is a primitive element of GF_{2^q} and, hence, any other element can be expressed as a power of α : $\{0, \alpha^0 = 1, \alpha^1 = \alpha, \alpha^2, \dots, \alpha^{2^q-2}\}$. The parity-check matrix of an RS code is an $R \times N$ matrix defined over GF_{2^q} :

$$\tilde{\mathbf{H}} = \begin{bmatrix} \tilde{h}_{0,0} & \tilde{h}_{0,1} & \dots & \tilde{h}_{0,N-1} \\ \tilde{h}_{1,0} & \tilde{h}_{1,1} & \dots & \tilde{h}_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{h}_{R-1,0} & \tilde{h}_{R-1,1} & \dots & \tilde{h}_{R-1,N-1} \end{bmatrix}, \quad (10)$$

where each $\tilde{h}_{i,j}$ represents the power α must be raised to for obtaining its corresponding element.

Although defined over GF_{2^q} , RS codes can be seen as binary codes by using their binary expansions that can be obtained on the basis of the primitive polynomial adopted. In order to derive a valid parity-check matrix for the binary expansion of an RS code, we can use the companion matrix, \mathbf{C} , of the primitive polynomial. For a q -degree polynomial, the companion matrix is a $q \times q$ matrix whose eigenvalues coincide with the roots of the polynomial. So, in the case of a monic binary polynomial $p(x) = p_0 + p_1x + \dots + p_{q-1}x^{q-1} + x^q$, the companion matrix assumes the form

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & \dots & p_0 \\ 1 & 0 & 0 & \dots & p_1 \\ 0 & 1 & 0 & \dots & p_2 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & p_{q-1} \end{bmatrix}. \quad (11)$$

When $p(x)$ is a primitive polynomial, we have $p_0 = 1$ and \mathbf{C} is a full-rank matrix.

A valid parity-check matrix for the binary expansion of an RS code can be obtained as follows:

$$\mathbf{H} = \begin{bmatrix} \tilde{\mathbf{C}}_{\tilde{h}_{0,0}} & \tilde{\mathbf{C}}_{\tilde{h}_{0,1}} & \dots & \tilde{\mathbf{C}}_{\tilde{h}_{0,N-1}} \\ \tilde{\mathbf{C}}_{\tilde{h}_{1,0}} & \tilde{\mathbf{C}}_{\tilde{h}_{1,1}} & \dots & \tilde{\mathbf{C}}_{\tilde{h}_{1,N-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{C}}_{\tilde{h}_{R-1,0}} & \tilde{\mathbf{C}}_{\tilde{h}_{R-1,1}} & \dots & \tilde{\mathbf{C}}_{\tilde{h}_{R-1,N-1}} \end{bmatrix}. \quad (12)$$

Matrix \mathbf{H} expressed by (12) is an $r \times n$ binary matrix (with $r = qR$ and $n = qN$) that can be used for decoding the binary expansion of the RS code. We will denote it as the “binary expansion parity-check matrix” (BXPCM) in the following.

In order to apply the proposed soft-decision decoding technique also to RS codes, we adopt the BXPCM in place of the EPCM used for binary cyclic codes. However, due to the lack of cyclic structure in the BXPCM, the density reduction algorithm must be slightly changed. The BXPCM, in fact, is not a circulant matrix; so, the number of overlaps between couples of rows cannot be obtained by means of the periodic autocorrelation function, but must be calculated by directly

resorting to the dot product among couples of rows. A single row is replaced every time a sparser version of the same row is found, since it is not possible to rebuild the whole parity-check matrix through cyclically shifted versions of a row.

Finally, the SPCM is derived from the RPCM by “spreading” its 1 symbols in s blocks, each with size $r \times n$, in such a way to minimize the number of short cycles in the associated Tanner graph.

3. THE DECODING ALGORITHM

We consider the sum-product algorithm with log-likelihood ratios (LLRs-SPA) [20] that is very common for decoding LDPC codes. This algorithm is well known, and its main steps are reminded next only for the sake of convenience.

Decoding is based on the exchange of messages between variable and check nodes: information on the reliability of the i th received bit c_i is sent as a message $\Gamma_{i \rightarrow j}(c_i)$ from the variable node v_i to the check node z_j , then elaborated, and sent back as a message $\Lambda_{j \rightarrow i}(c_i)$ from the check node z_j to the variable node v_i .

The algorithm starts by initializing both sets of messages, that is, $\forall i, j$ for which an edge exists between nodes v_i and z_j , we set

$$\Gamma_{i \rightarrow j}(c_i) = L(c_i) = \ln \left[\frac{P(c_i = 0 | y_i)}{P(c_i = 1 | y_i)} \right], \quad i = 1, \dots, n, \quad (13)$$

$$\Lambda_{j \rightarrow i}(c_i) = 0,$$

where $L(c_i)$ is the initial reliability value based on the channel measurement information, and $P(c_i = x | y_i)$, $x \in \{0, 1\}$, is the probability that the codeword bit c_i at position i is equal to x , given a received signal y_i at the channel output.

After initialization, the LLR-SPA algorithm starts iterating. During each iteration, messages sent from the check nodes to the variable nodes are calculated by means of the following formula:

$$\Lambda_{j \rightarrow i}(c_i) = 2 \tanh^{-1} \left\{ \prod_{l \in A(j) \setminus i} \tanh \left[\frac{1}{2} \Gamma_{l \rightarrow j}(c_l) \right] \right\}, \quad (14)$$

where $A(j) \setminus i$ represents the set of variable nodes connected to the check node z_j , with the exclusion of node v_i .

Messages sent from the variable nodes to the check nodes are then calculated as follows:

$$\Gamma_{i \rightarrow j}(c_i) = L(c_i) + \sum_{l \in B(i) \setminus j} \Lambda_{l \rightarrow i}(c_l), \quad (15)$$

where $B(i) \setminus j$ represents the set of check nodes connected to the variable node v_i , with the exclusion of node z_j . In addition, the following quantity is evaluated:

$$\Gamma_i(c_i) = L(c_i) + \sum_{l \in B(i)} \Lambda_{l \rightarrow i}(c_l), \quad (16)$$

where $B(i)$ is the whole set of check nodes connected to v_i . Equation (16) is used to obtain an estimate (\hat{c}) of the received codeword (c) as follows:

$$\hat{c}_i = \begin{cases} 0 & \text{if } \Gamma_i(c_i) \geq 0, \\ 1 & \text{if } \Gamma_i(c_i) < 0. \end{cases} \quad (17)$$

The estimated codeword \hat{c} is then multiplied by the parity-check matrix associated with the Tanner graph. If the parity-check is successful, the decoding process stops and gives the estimated codeword as its result. Otherwise, the algorithm reiterates using updated messages. In this case, a further verification is made on the number of decoding iterations: when a maximum number of iterations is reached, the decoder stops the estimation efforts and outputs the estimated codeword as its result. In this case, however, decoding is unsuccessful and the error is detected.

3.1. Adaptation to the spread code

In order to take advantage of spread parity-check matrices, we adopt a modified version of the standard BP decoding algorithm.

The demodulator and demapper block produces, for each received bit, the $L(c_i)$ used to initialize the decoding algorithm (see (13)). Then, the vector containing the $L(c_i)$ values is repeated s times to form the new vector of $L(c_i^S)$ values, valid for the spread code. This is used to initialize the LLR-SPA algorithm that works on the spread parity-check matrix; the algorithm starts iterating and, at each iteration, produces updated versions of the extrinsic $[\Gamma_{i,j}(c_i^S)]$ and a posteriori $[\Gamma_i(c_i^S)]$ messages. While the former are used as inputs for the subsequent iteration (if needed), the latter represent the decoder output, and serve to obtain an estimated codeword that is subject to the parity-check test. In addition, this version of the algorithm produces a posteriori messages also for the original codeword as follows:

$$\Gamma_i(c_i) = \sum_{l=0}^{s-1} \Gamma_{i+ln}(c_{i+ln}^S), \quad i = 1, \dots, n. \quad (18)$$

Two estimated codewords, \hat{c}^S and \hat{c} , are derived on the basis of the sign of $\Gamma_i(c_i^S)$ and $\Gamma_i(c_i)$, respectively, and their corresponding parity-check tests are executed (based on \mathbf{H}^S and \mathbf{H}^R). The test on $\Gamma_i(c_i^S)$ is passed if and only if the test is passed by all submatrices, while the test on $\Gamma_i(c_i)$ is passed if the test is passed by the sum of the a posteriori messages for all the replicas of each bit. When both tests are successful, the decoder stops iterating and outputs \hat{c} as the estimated codeword; otherwise, decoding continues until a maximum number of iterations is reached. This double parity-check test permits to reduce significantly the number of undetected errors (decoder failures), as we have verified through numerical simulations.

4. NUMERICAL SIMULATIONS

In order to assess the benefits of the proposed approach, we have simulated transmission over the additive white

Gaussian noise (AWGN) channel, in conjunction with binary phase shift keying (BPSK) modulation for different BCH and RS codes. In all simulations, we have used a maximum number of iterations equal to 100.

4.1. Short BCH codes

We consider two examples of short BCH codes with different length and dimension, namely, $(n, k) = (63, 57)$ and $(n, k) = (127, 71)$.

For the first code, the density reduction algorithm is unsuccessful. So we apply the spreading technique directly to the extended parity-check matrix. For the $(127, 71)$ -BCH code, instead, the density reduction algorithm is successful and, starting from \mathbf{h}_1 with Hamming weight 48, a vector \mathbf{h}_2 is obtained with Hamming weight 32, thus reducing by 1/3 the parity-check matrix density. Hence, spreading has been applied to the reduced parity-check matrix. The main features of the considered BCH codes are summarized in Table 1. The number of length-4 cycles has been calculated exhaustively by considering the overlapping ones between each couple of rows (or columns).

We notice that, for the $(63, 57)$ -BCH code, the spread parity-check matrix has a number of length-4 cycles higher than that of the classic parity-check matrix. This is because such code is characterized by a very small r , and this reflects in a matrix (2) with the smallest number of length-4 cycles. Figures 2 and 3 show the bit error rate (BER) and frame error rate (FER) as a function of the signal-to-noise ratio E_b/N_0 . The curves have been obtained, through numerical simulations, for the considered codes when decoding with the classic parity-check matrix (PCM), the reduced parity-check matrix (RPCM, in Figure 3 only, for the reasons explained above), the extended parity-check matrix (EPCM), and the spread parity-check matrix (SPCM). The figures report also curves for the union bound (UB) [21] that can be used as a reference for the error rate under ideal (maximum likelihood) decoding.

We observe from Figure 2 that, for the $(63, 57)$ -BCH code, the new technique outperforms those based on the classic PCM and EPCM, with a gain of more than 1 dB over the PCM and more than 1.5 dB over the EPCM. Furthermore, the curves obtained through the SPCM approach are practically overlaid to the union bound, and the SPCM decoder achieves almost ideal performance.

In the case of the $(127, 71)$ -BCH code, we have reported also the performance achieved by the ASPCM that offers the best result, at least in the region of explored BER and FER values, with a gain of more than 2 dB over the PCM-based algorithm and more than 3 dB over the EPCM approach.

However, for the $(127, 71)$ -BCH code, the curves are rather distant from the union bound, showing that further coding gain could be achieved, in principle. Actually, techniques based on the adaptive belief propagation can show better performance for the same code parameters. Figure 3 reports also the BER and FER curves obtained by using the software made publicly available in [22], showing that the adaptive belief propagation can achieve about 2 dB of further gain, though still not reaching the

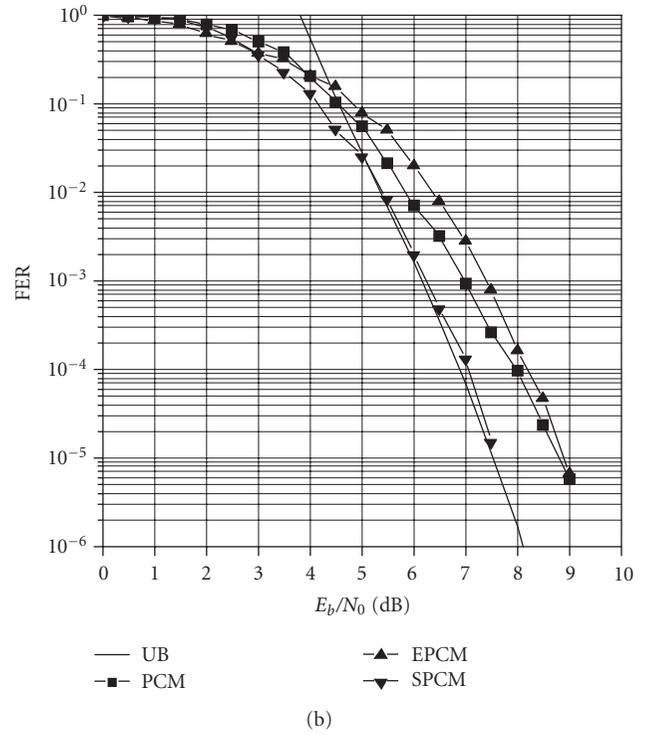
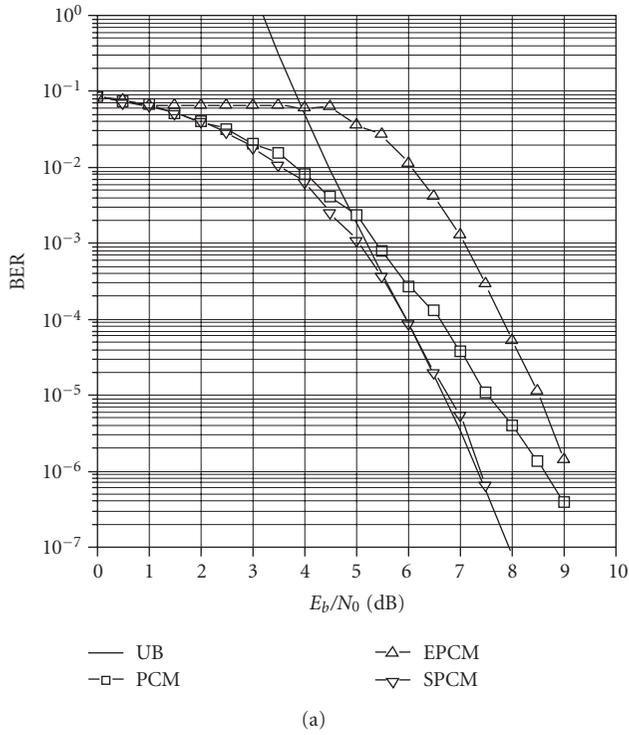


FIGURE 2: Simulated BER (a) and FER (b) for the (63, 57)-BCH code.

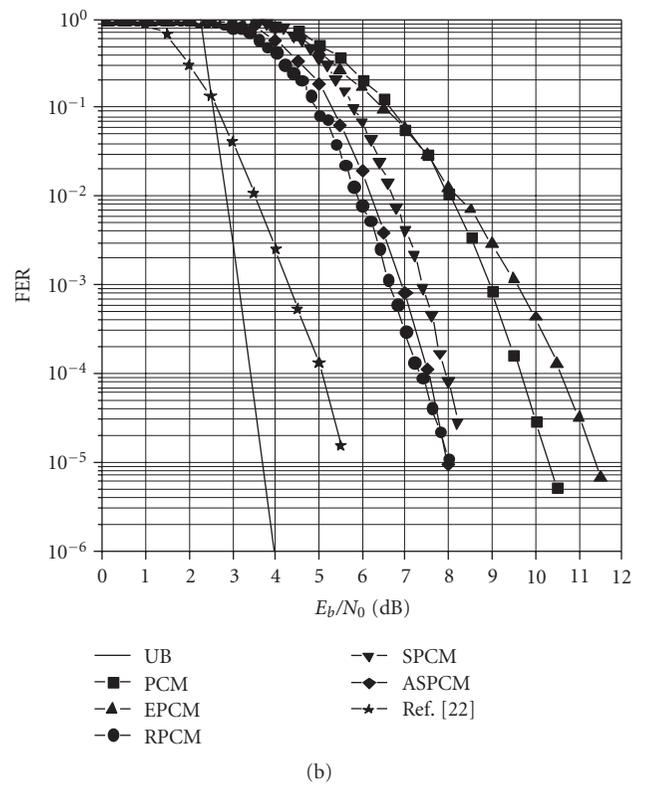
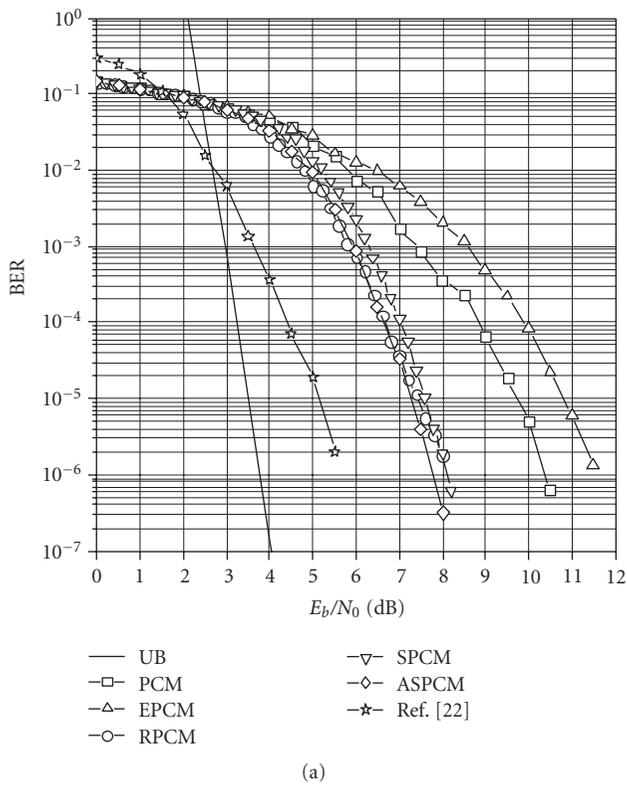


FIGURE 3: Simulated BER (a) and FER (b) for the (127, 71)-BCH code.

TABLE 1: Characteristics of the (63, 57) and (127, 71) BCH codes.

BCH code	Rate	PCM		EPCM		RPCM		SPCM	
		# 4-cycles	1 Symbols						
(63, 57)	0.9	1800	192	234360	2016	234360	2016	7749	2016
(127, 71)	0.56	378314	2688	1356614	6096	240284	4064	4699	4064

union bound. Moreover, as a drawback, such approach exhibits a much higher complexity than the one proposed here.

4.2. CDMA2000 Reed-Solomon codes

As an example of application of the proposed technique to RS codes, we have considered the codes included by the “third-generation partnership project 2” (3GPP2) in the

CDMA2000 specification for broadcast services in high-rate packet data systems [2].

The CDMA2000 standard adopts systematic RS codes defined over GF_{256} with the following choice of the parameters (N, K) : (16, 12), (16, 13), (16, 14), (32, 24), (32, 26), and (32, 28).

We have focused on the (16, 12) and (32, 28) RS codes that are characterized by the following parity-check matrices over GF_{256} [2]:

$$\tilde{\mathbf{H}}_{(16,12)} = \begin{bmatrix} 40 & 8 & 158 & 209 & 76 & 160 & 125 & 70 & 137 & 254 & 160 & 201 & 0 & -\infty & -\infty & -\infty \\ 138 & 196 & 4 & 123 & 226 & 142 & 19 & 87 & 169 & 192 & 57 & 246 & -\infty & 0 & -\infty & -\infty \\ 141 & 97 & 250 & 27 & 198 & 95 & 59 & 39 & 244 & 27 & 53 & 201 & -\infty & -\infty & 0 & -\infty \\ 8 & 158 & 209 & 76 & 160 & 125 & 70 & 137 & 254 & 160 & 201 & 0 & -\infty & -\infty & -\infty & 0 \end{bmatrix} \quad (19)$$

$$\tilde{\mathbf{H}}_{(32,28)} = [\tilde{\mathbf{H}}_{(32,28)}^a \tilde{\mathbf{H}}_{(32,28)}^b]$$

with

$$\tilde{\mathbf{H}}_{(32,28)}^a = \begin{bmatrix} 207 & 229 & 141 & 32 & 224 & 248 & 120 & 228 & 113 & 88 & 74 & 178 & 78 & 5 & 182 & 212 \\ 34 & 210 & 50 & 160 & 37 & 5 & 229 & 73 & 215 & 208 & 37 & 76 & 225 & 218 & 188 & 157 \\ 22 & 95 & 89 & 127 & 223 & 131 & 44 & 240 & 118 & 113 & 215 & 97 & 181 & 168 & 204 & 221 \\ 229 & 141 & 32 & 224 & 248 & 120 & 228 & 113 & 88 & 74 & 178 & 78 & 5 & 182 & 212 & 40 \end{bmatrix}, \quad (20)$$

$$\tilde{\mathbf{H}}_{(32,28)}^b = \begin{bmatrix} 40 & 8 & 158 & 209 & 76 & 160 & 125 & 70 & 137 & 254 & 160 & 201 & 0 & -\infty & -\infty & -\infty \\ 138 & 196 & 4 & 123 & 226 & 142 & 19 & 87 & 169 & 192 & 57 & 246 & -\infty & 0 & -\infty & -\infty \\ 141 & 97 & 250 & 27 & 198 & 95 & 59 & 39 & 244 & 27 & 53 & 201 & -\infty & -\infty & 0 & -\infty \\ 8 & 158 & 209 & 76 & 160 & 125 & 70 & 137 & 254 & 160 & 201 & 0 & -\infty & -\infty & -\infty & 0 \end{bmatrix},$$

where “ $-\infty$ ” represents the null element ($\alpha^{-\infty} = 0$).

From (19), the BXPCMs for the (16, 12) and (32, 28) RS codes can be easily obtained, as explained in Section 2.6, in the form of a 32×128 and a 32×256 binary matrix, respectively. The density reduction algorithm has been applied to the BXPCMs, thus obtaining two RPCMs with a reduced number of symbols 1. Finally, the RPCMs have been used as the starting point for the adaptive spreading algorithm that has been applied with $s = 2$. The features of the parity-check matrices for the considered RS codes are summarized in Table 2.

We observe that the density reduction algorithm is able to produce, in the RPCMs, a density reduction of about 6% for the (16, 12)-RS code and 11% for the (32, 28)-RS code,

with respect to the corresponding BXPCMs. This reflects on a lower number of short cycles in the associated Tanner graphs and in a more favorable performance, as shown in Figures 4 and 5. The ASPCM has a further reduced density of 1 symbols and, jointly with the spread version of the decoding algorithm, it is able to ensure the best performance. In particular, the BER curve in Figure 4(a), referred to the (16, 12)-RS code, exhibits a coding gain of more than 1 dB due to the adoption of the proposed approach, based on spread matrices, in comparison with the more conventional BXPCM approach. Instead, the coding gain for the (32, 28)-RS code is less than 1 dB (see Figure 5(a)).

In comparison with the algorithm based on adaptive belief propagation, the approach based on the ASPCM

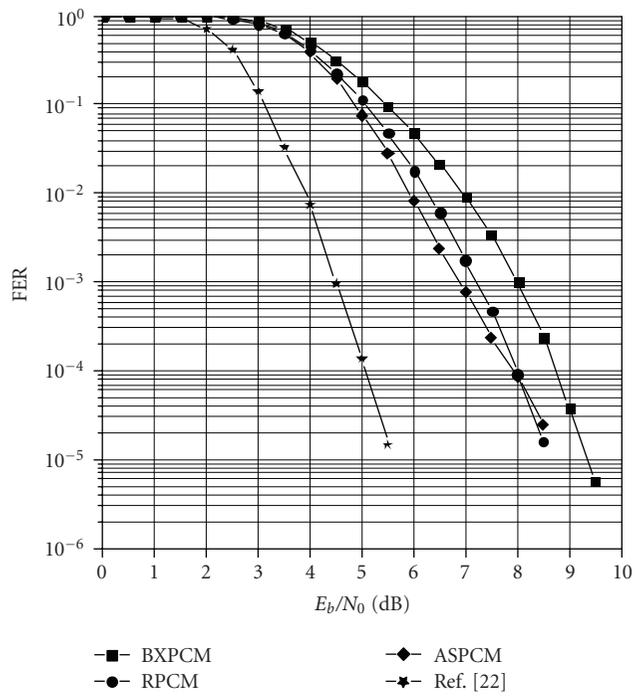
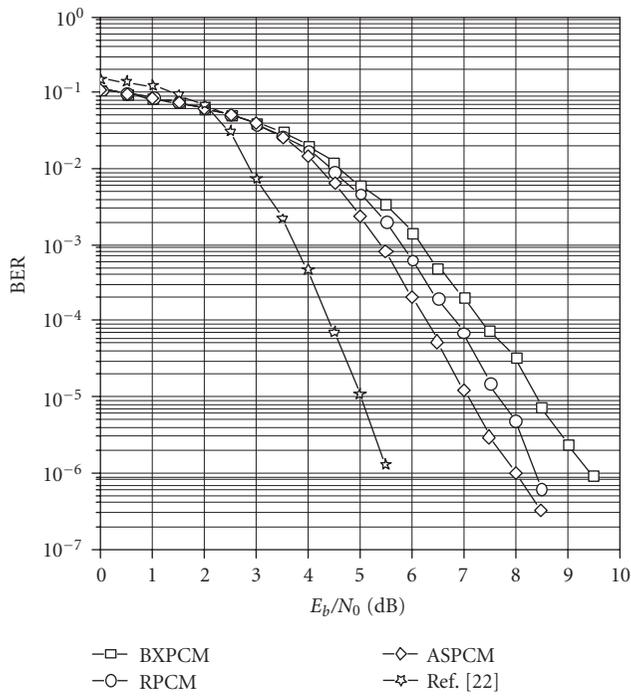


FIGURE 4: Simulated BER (a) and FER (b) for the (16, 12)-RS code over GF_{256} .

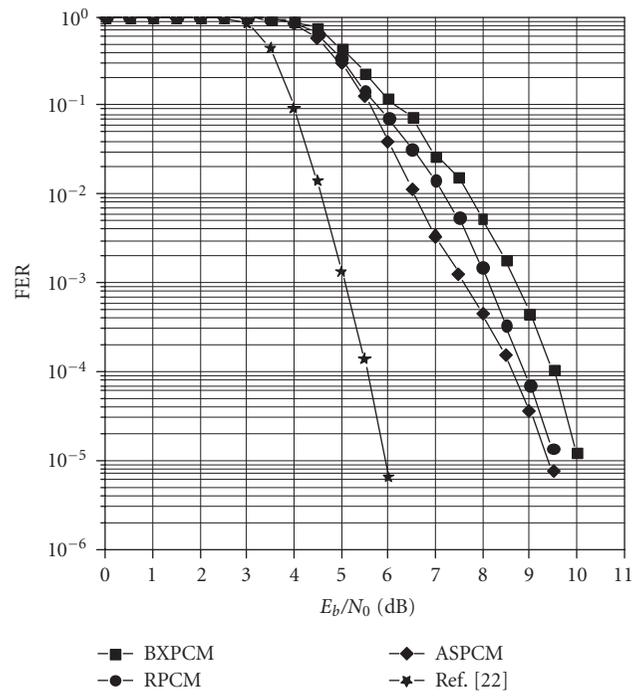
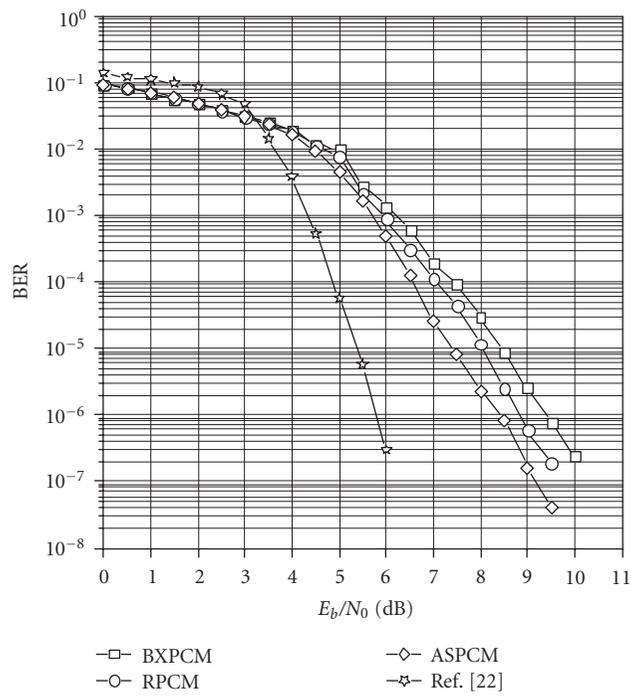


FIGURE 5: Simulated BER (a) and FER (b) for the (32, 28)-RS code over GF_{256} .

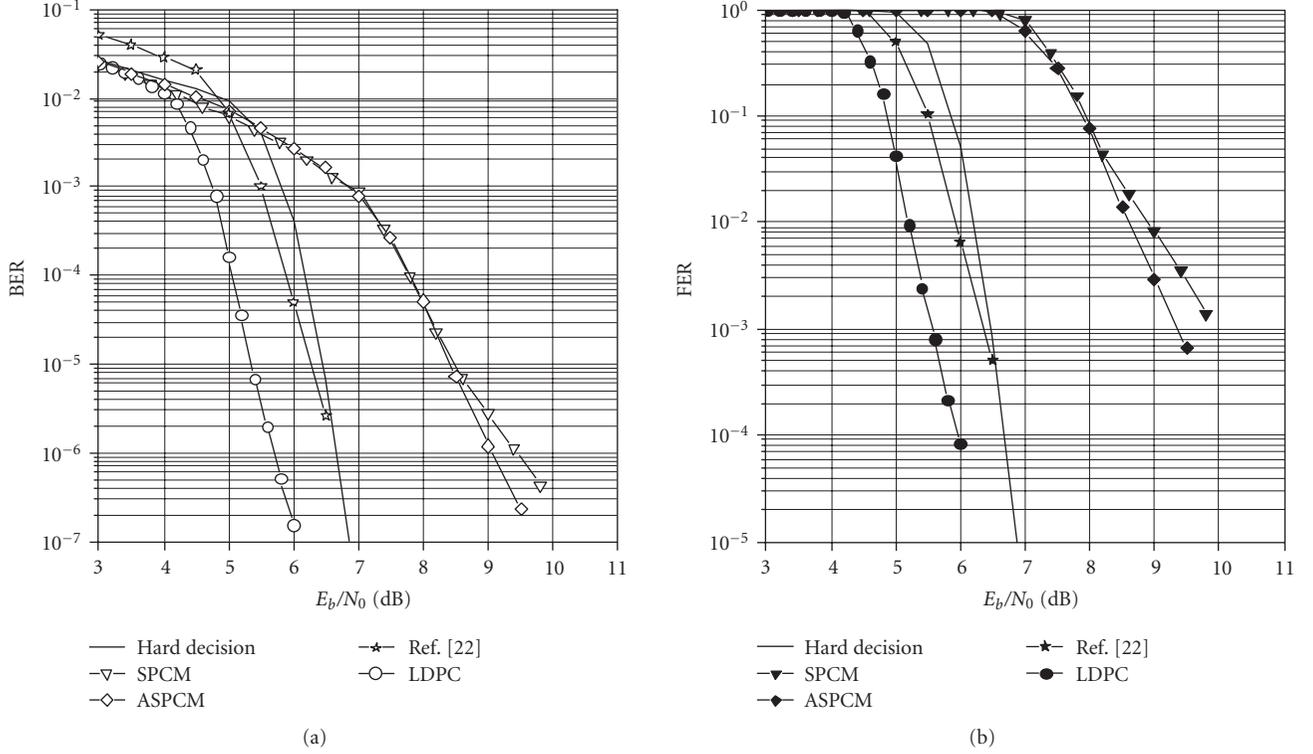


FIGURE 6: Simulated BER (a) and FER (b) for the DVB-S2 BCH code with $n_{\text{bch}} = 3240$. The performance of an LDPC code with the same length and rate is also shown for comparison.

TABLE 2: Characteristics of the parity-check matrices for the (16, 12) and (32, 28) RS Codes over GF_{256} .

RS code	Matrix	Rows	Columns	1 Symbols	# 4-cycles
(16, 12)	BXPCM	32	128	1621	126756
	RPCM	32	128	1519	82360
(32, 28)	BXPCM	32	256	3700	854034
	RPCM	32	256	3302	502161

exhibits, for the considered codes, a loss of more than 2 dB. This is not surprising, as the method proposed in [4] is significantly more involved than the approaches we have proposed.

4.3. DVB-S2 BCH codes

The second revision of the European standard for satellite digital video broadcasting (DVB-S2) adopts a forward error-correction (FEC) scheme based on the concatenation of BCH and LDPC codes [1]. The data stream is divided into k_{bch} -bit frames that are used as inputs for a systematic BCH encoder. This produces n_{bch} -bit frames by appending $n_{\text{bch}} = n_{\text{bch}} - k_{\text{bch}}$ redundancy bits to the input frame. According to the standard, n_{bch} can assume the following values: 128, 160, and 192 for normal frames, 168 for short frames. The output of the outer BCH encoder is given as input to an inner systematic LDPC encoder that produces n_{ldpc} -bit frames by

appending further $n_{\text{ldpc}} = n_{\text{ldpc}} - k_{\text{ldpc}} = n_{\text{ldpc}} - n_{\text{bch}}$ redundancy bits to each BCH-encoded frame.

The interest for applying an iterative soft-decision decoding to the BCH code, too, is in the possibility of uniforming the decoding procedure to that of the inner LDPC code, with expected hardware and software advantages. The result should be a significant reduction of the complexity that, even adopting hard decoding, is a critical issue for BCH codes of so large sizes. In addition, a performance improvement should also be expected, although we show that it is not simple to achieve it with the method proposed in the previous sections.

We consider, in our simulations, the short frame format that is characterized by $n_{\text{ldpc}} = 16200$, but the proposed techniques can also be applied to normal frames, with $n_{\text{ldpc}} = 64800$. The standard FEC parameters for short frames, together with encoding details, are reported in [1] and are omitted here for the sake of brevity.

BCH codes used for short DVB-S2 frames are able to correct $t = 12$ errors and have code length ranging between 3240 and 14400 bits. Actually, the standard adopts shortened BCH codes, all defined by the same generator polynomial that can be obtained as $g(x) = \prod_{i=1}^{12} g_i(x)$; the structure of $g_i(x)$ is also given in [1]. Each factor $g_i(x)$, $i = 1, \dots, 12$ has degree $r = 14$ and can be seen as the generator polynomial of a Hamming code with length $n = 2^{14} - 1 = 16383$ and redundancy $r = 14$. The corresponding parity-check polynomial can therefore be obtained as $h_i(x) = (1 + x^n)/g_i(x)$, $i = 1, \dots, 12$.

Each BCH code can be seen as a shortened version of a “mother” BCH code with length $n^* = 16n = 262128$, redundancy $r^* = 12r = 168$, and dimension $k^* = 261960$. In fact, it can be easily shown that $g(x)$ divides $1 + x^{n^*}$ and $h(x)$ can be derived as follows:

$$\begin{aligned} h(x) &= \frac{1 + x^{n^*}}{g(x)} = \frac{1 + x^{16n}}{g(x)} = \frac{(1 + x^n)^{16}}{g(x)} \\ &= \frac{(1 + x^n)^{16}}{\prod_{i=1}^{12} g_i(x)} = \prod_{i=1}^{12} h_i(x) \cdot (1 + x^n)^4. \end{aligned} \quad (21)$$

Once having obtained $h(x)$, the first row of \mathbf{H}^E , \mathbf{h}_1 , becomes available; it has Hamming weight $w_1 = 128640$. Starting from this dense vector, it is possible to execute 7 iterations of the reduction algorithm described in Section 2.3, thus obtaining a new vector, \mathbf{h}_8 , with Hamming weight $w_8 = 125568$. It must be said, however, that in the present case that considers a long code with very high rate, the density reduction algorithm is not able to produce immediately an excellent result: the reduced parity-check matrix has a density that is only 2.4% smaller than that of the extended parity-check matrix.

For each iteration of the algorithm, the shift ν has taken the following values: 213318, 215694, 106013, 171879, 40909, 85749, 761. When multiple choices for ν were possible (due to the fact that the autocorrelation function can assume its maximum out-of-phase value for more than one shift ν), a random selection criterion has been adopted, and the experiment has been repeated several times in order to find the best sequence among the selected ones.

Vector \mathbf{h}_8 has been used as the first row of the RPCM for the cyclic mother code. A valid parity-check matrix for each shortened code derived from the mother code can be obtained by selecting the first $r_{\text{bch}} = 168$ rows and the first n_{bch} columns of the RPCM so found. The shortened RPCM is then used as the starting point for the spreading algorithm that produces the SPCM.

We have considered the case of $n_{\text{bch}} = 3240$ and applied the spreading algorithm with $s = 20$. The results of numerical simulations based on the spreading technique are shown in Figure 6. Actually, performance is not particularly good: even by adopting the ASPCM, that outperforms the SPCM, the simulated curves are worse than those referred to a hard-decision decoder able to correct $t = 12$ errors. However, we guess that the unsatisfactory result is due to the difficulty in reducing the weight of the parity-check matrix when starting from a so dense parity-check matrix. Also in this case, the adoption of adaptive belief propagation permits to achieve better performance (with more than 2 dB of further gain) at the cost of increased complexity.

In Figure 6, we also show the performance of an LDPC code having the same parameters of the BCH code. It has been designed through the so-called LCO technique [23], that permits to avoid the presence of length-4 cycles but, except for this, the parity-check matrix has not been further optimized. So, we see that wide margins for improving

performance should exist, on condition to find more effective representations of the parity-check matrix than those considered so far. Work is in progress in such a direction.

5. CONCLUSION

We have studied the application of some new iterative soft-decision decoding algorithms based on belief propagation to BCH and RS codes. The essence of the method is in the possibility to overcome the drawbacks of the parity-check matrix of these codes, namely, the high density of 1 symbols and the presence of short length cycles in the Tanner graph that prevent effective application of the BP decoding algorithm. The naive idea of matrix extension, already proposed in the literature, has been refined through the introduction of additional “reduction” and “spreading” operations, the latter, eventually, in an adaptive implementation.

The procedure is very simple and quite suitable for application in multimedia transmissions. If applied to binary short codes, like those required in presence of stringent requirements on the decoding delay, the method achieves improved performance with respect to classic parity-check matrices. The proposed approach is still outperformed by adaptive belief propagation, particularly in the case of very long and high-rate codes. Its complexity, however, is always lower.

REFERENCES

- [1] ETSI EN 302 307 v1.1.2, “Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation system for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications,” June 2006.
- [2] 3GPP2 C.S0054-A v1.0, “CDMA2000 High Rate Broadcast-Multicast Packet Data Air Interface Specification,” February 2006.
- [3] D. Chase, “Class of algorithms for decoding block codes with channel measurement information,” *IEEE Transactions on Information Theory*, vol. 18, no. 1, pp. 170–182, 1972.
- [4] J. Jiang and K. R. Narayanan, “Iterative soft-input soft-output decoding of Reed-Solomon codes by adapting the parity-check matrix,” *IEEE Transactions on Information Theory*, vol. 52, no. 8, pp. 3746–3756, 2006.
- [5] T. J. Richardson and R. L. Urbanke, “The capacity of low-density parity-check codes under message-passing decoding,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 599–618, 2001.
- [6] T. R. Halford, A. J. Grant, and K. M. Chugg, “Which codes have 4-cycle-free Tanner graphs?” *IEEE Transactions on Information Theory*, vol. 52, no. 9, pp. 4219–4223, 2006.
- [7] J. S. Yedidia, J. Chen, and M. P. C. Fossorier, “Generating code representations suitable for belief propagation decoding,” Tech. Rep. TR-2002-40, Mitsubishi Electric Research Laboratories, Cambridge, Mass, USA, September 2002.
- [8] J. S. Yedidia, J. Chen, and M. P. C. Fossorier, “Representing codes for belief propagation decoding,” in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '03)*, p. 176, Yokohama, Japan, June-July 2003.

- [9] S. Sankaranarayanan and B. Vasic, "Iterative decoding of linear block codes: a parity-check orthogonalization approach," *IEEE Transactions on Information Theory*, vol. 51, no. 9, pp. 3347–3353, 2005.
- [10] B. Kamali and A. H. Aghvami, "Belief propagation decoding of Reed-Solomon codes; a bit-level soft decision decoding algorithm," *IEEE Transactions on Broadcasting*, vol. 51, no. 1, pp. 106–113, 2005.
- [11] A. Kothiyal and O. Y. Takeshita, "A comparison of adaptive belief propagation and the best graph algorithm for the decoding of linear block codes," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '05)*, pp. 724–728, Adelaide, Australia, September 2005.
- [12] M. El-Khamy and R. J. McEliece, "Iterative algebraic soft-decision list decoding of Reed-Solomon codes," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 3, pp. 481–490, 2006.
- [13] R. H. Morelos-Zaragoza, "Architectural issues of soft-decision iterative decoders for binary cyclic codes," Tech. Rep., Sony ATL, Atlanta, Ga, USA, August 2000.
- [14] M. Baldi, G. Cancellieri, and F. Chiaraluce, "Iterative soft-decision decoding of binary cyclic codes based on spread parity-check matrices," in *Proceedings of the 15th International Conference on Software, Telecommunications and Computer Networks (SoftCOM '07)*, Dubrovnik, Croatia, September 2007, Paper 7069.
- [15] M. Baldi, G. Cancellieri, and F. Chiaraluce, "Iterative soft-decision decoding of binary cyclic codes," submitted to *Journal of Communications Software and Systems*.
- [16] L. Zhang, V. O. K. Li, and Z. Cao, "Short BCH codes for wireless multimedia data," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '02)*, vol. 1, pp. 220–222, Orlando, FL, USA, March 2002.
- [17] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1994.
- [18] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low-density parity-check codes based on finite geometries: a rediscovery and new results," *IEEE Transactions on Information Theory*, vol. 47, no. 7, pp. 2711–2736, 2001.
- [19] S. B. Wicker and V. K. Bhargava, Eds., *Reed-Solomon Codes and Their Applications*, Wiley-IEEE Press, Piscataway, NJ, USA, 1999.
- [20] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 429–445, 1996.
- [21] R. H. Morelos-Zaragoza, *The Art of Error Correcting Coding*, John Wiley & Sons, New York, NY, USA, 2002.
- [22] J. Jiang, "Software simulator for the adaptive iterative RS decoding algorithm," <http://www.ece.tamu.edu/~jjiang>.
- [23] M. Baldi and F. Chiaraluce, "On the design of punctured low density parity check codes for variable rate systems," *Journal of Communications Software and Systems*, vol. 1, no. 2, pp. 88–100, 2005.

Research Article

Design of a VLSI Decoder for Partially Structured LDPC Codes

Fabrizio Vacca,¹ Libero Dinoi,² and Guido Masera¹

¹ *Dipartimento di Elettronica, Politecnico di Torino, 10129 Torino, Italy*

² *Elettronica S.p.A., 00131 Roma, Italy*

Correspondence should be addressed to Fabrizio Vacca, fabrizio.vacca@polito.it

Received 3 April 2008; Revised 1 July 2008; Accepted 27 August 2008

Recommended by Fred Daneshgaran

The starting point of this work is the development of a new class of partially structured LDPC codes, very well suited for hardware implementation. Specifically these codes are built so that the edges of their parity matrix can be partitioned into two disjoint sets, namely, the structured and the random ones. For the proposed class of codes a constructive design method is provided. To assess the value of this method the constructed codes performance are presented. From these results, a novel decoding method called split decoding is introduced. Finally, to prove the effectiveness of the proposed approach a whole VLSI decoder is designed and characterized.

Copyright © 2008 Fabrizio Vacca et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Low Density Parity Check [1, 2] (LDPC) codes are among the most powerful error correcting codes available; performance and decoding algorithms have been deeply explored in the last few years and LDPC codes have been proposed for application in several standards. However, their practical implementation is still a challenging subject of investigation. There are basically two aspects of LDPC that pose serious implementation problems: (1) the huge size of parity-check matrices that are of interest for high performance applications; (2) the high irregularity of these matrices, that is, the fact that they are very sparse with ones often distributed almost randomly.

From the implementation standpoint, the first aspect implies the allocation of a large number of processing elements and larger number of communication paths among them; the second one entails a very limited adjacency of processing elements, resulting in scarcely efficient and expensive communication structures [3]. In order to limit the implementation complexity of both the processing and the interconnect resources of the decoder, several partially parallel architectures have been proposed as feasible alternatives to the fully parallel approach [4–6].

In partially parallel architectures, processing elements (PE) are shared among multiple check and variable nodes: each PE is required to sequentially serve a number of rows

or columns of the parity-check matrix; since, at each instant of time, only a subset of the messages to be exchanged between variable and check nodes actually need to be moved from one PE to another one, this approach also reduces the number of physical interconnects in the decoder. Finally, instead of storing messages in independent registers, partially parallel architectures allow grouping them into more efficient memories. Of course, resource sharing implies a throughput scaling and the parallelism degree has to be selected according to the target throughput: complexity tradeoff.

While the partially parallel approach looks like a fully scalable solution able to flexibly adapt to different cost and throughput constraints, it raises the problem of collisions in the access to memories [7]. In the connection of P PEs with the same number of message memories, the possibility of simultaneously moving P messages between PEs and memories has a very low probability and in a large percentage of cases more than one required message must be read from or written to the same memory. Due to the limited adjacency of the parity-check matrix, the optimization of message partitioning among memories tends to have a poor effect on the number of occurred conflicts. PEs must then be frequently stalled to accommodate for multiple simultaneous accesses to memories and this severely affects the decoder throughput.

In order to cope with these problems, two different kinds of solution have been proposed.

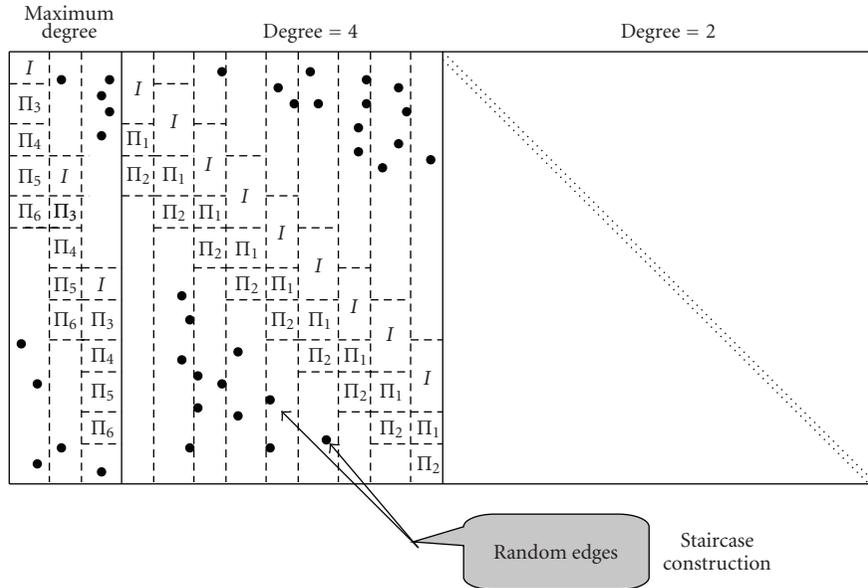


FIGURE 1: Parity-check matrix structure for code rate 1/2.

- (1) A few code-independent approaches have been formulated aiming at avoiding collisions for a generic LDPC code. This class of solutions leads to expensive implementation architectures, such as those proposed in [5, 8, 9].
- (2) Structured LDPC codes [10] have been proposed: in this approach, a structure is given to the parity-check matrix and all ones are properly distributed with the purpose of limiting the interconnect requirements while keeping good error correcting capabilities. Classes of structured codes have been studied and implemented successfully.

This paper presents a new approach based on the idea of designing parity-check matrices with edges partitioned into two classes, namely, structured ones, positioned in accordance with a repetitive fixture laid not far from the diagonal, and random ones, which can be placed freely in the whole matrix, with the purpose of achieving good performance in terms of error correcting capabilities. This approach has been previously partly presented in [11] where the authors proposed a partially structured 1/2 rate LDPC code and described the implementation of a decoder with two separate processing units: a dedicated part tailored to take advantage of the regularly placed ones, and a programmable application specific instruction set processor (ASIP) serving the whole random part of the parity-check matrix.

This work extends the previous one deeply investigating the benefits coming from the idea of designing a partitioned parity-check matrix, with a particular emphasis on the hardware architecture. This paper provides two major contributions: a decoding algorithm that takes advantage of the partitioned parity-check matrix to simplify the collision

problem and the VLSI design of a decoder implementing the described algorithm.

The rest of the paper is organized as follows. Section 2 introduces LDPC and eIRA codes, reviewing some code design issues. Then the proposed partially structured eIRA code class is presented and some constructive results are given. To prove the usefulness of this approach simulation performance is presented.

Then the decoding algorithm of these codes is presented in Section 3. Due to the particular structure of the parity-check matrix, some modifications to the traditional belief propagation algorithm are possible. The obtained algorithm, called split decoding presents remarkable savings with respect to the traditional one. These savings are deeply investigated in Section 4. The modular fixture of structured ones strongly simplifies the communication structure among processing elements, while the reduced number of “random” ones enables collision-free partitioning that eliminates the need for stall cycles in the decoding process. In the same Section, logical synthesis results are given and memory footprint of the designed VLSI core is analyzed. Lastly, in Section 5 conclusions will be drawn.

2. PARTIALLY STRUCTURED eIRA CODES

2.1. LDPC and eIRA codes background

An LDPC code is a linear block code defined by an $m \times n$ sparse parity-check matrix. It can be described also in terms of its Tanner graphs: each of the bits and of the parity-check equations, defined, respectively, as variable nodes (VNs) and check nodes (CNs), is represented by a vertex in the graph. The number of edges connected to a vertex of the graph is defined as the *degree* of that node; the VN (or CN) *degree distribution* of a code specifies how the edges are distributed

TABLE 1: VN degree distributions (node perspective).

Code rate	$\tilde{\lambda}_1$	$\tilde{\lambda}_3$	$\tilde{\lambda}_5$	$\tilde{\lambda}_6$	$\tilde{\lambda}_9$
1/3	0.6667	0.2222	—	—	0.1111
1/2	0.5000	0.3750	—	0.1250	—
2/3	0.3334	0.5556	—	0.1110	—

among the variable (or check) nodes of the code. In a *regular* (c, r) LDPC code all VNs have degree c and all CNs have degree r in the graph.

In [12], *irregular* LDPC codes have been shown to outperform regular LDPC codes. In this work, we will focus on eIRA codes [13–15]: a popular subset of LDPC codes. eIRA codes are characterized by the peculiar structure of the parity-check matrix corresponding to the nonsystematic bits, which is made of VNs with degree 2, arranged according to a chain-like structure, known in the literature as “staircase construction.” The main advantage of this structure is that the obtained code can be encoded in linear time. In [14], it is shown that such a constraint leads to negligible performance losses.

There are some specific design issues for eIRA codes. Typically, the project of irregular LDPC codes consists of the optimization of the VN and CN degree distributions and of the design of a parity-check matrix compliant with the obtained distributions. The degree distributions are optimized by means of several techniques such as density evolution [12] and its approximated version of [16]; however, when dealing with eIRA codes, these techniques cannot be directly applied. The problem of designing an efficient eIRA code is well presented in [17], where it is shown that the loss in performance introduced by the staircase construction is about 0.1 dB, in the waterfall region.

While the VN and CN degree distributions determine the convergence behavior of the code asymptotically (for an infinite block length), the actual structure of the parity-check matrix is crucial for the performance of finite length LDPC codes. Many design algorithms can be applied to obtain a good parity-check matrix [18–20]; specific design algorithms for eIRA codes are presented in [21, 22].

2.2. Proposed code design guidelines

As a general rule, highly structured matrices and a limited connectivity lead to low code performance; this effect is not present in [10] because regular LDPC with a VN degree of 3 does not suffer from high error floors. On the contrary, eIRA codes have a significant number of degree 2 VNs. This would likely result in a code with a high error floor due to the constraints on the limited connectivity. To avoid these effects we adopt a partial structure allowing some of the edges to be placed randomly.

Let us illustrate this concept with an example: in Figure 1 we depict the parity-check matrix of a rate 1/2 code. Three sections can be identified: the rightmost contains the “staircase” parity-check VNs, the leftmost one contains high-degree VNs and the central one degree 4 VNs. The latter two sections include edges disposed according to a predefined

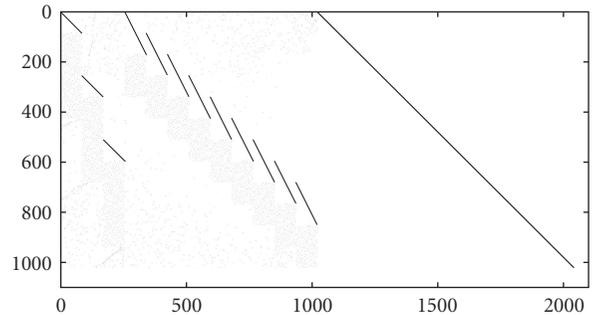


FIGURE 2: Parity-check matrix of a (2040, 1020) code.

TABLE 2: Edge categories distribution.

Code rate	Staircase	Pattern	Free
1/3	0.400	0.467	0.133
1/2	0.296	0.519	0.185
2/3	0.182	0.606	0.212

pattern, as well as some placed in a pseudorandom fashion. The optimal degree distribution obtained as explained in [17] contains a significant number of degree-3 VNs. We deliberately decided to forbid them to lower the error floor. Details on this expedient are available in [23].

As far as the structured part of the matrix is concerned, we use permuted versions of the identity matrix. In particular, the permutations used in the blocks labelled Π_i ($i = 1, 2, \dots$) in Figure 1 are triangular S-random interleavers built according to a tail-biting definition of the spread factor.

2.3. Three examples: rate 1/2, 2/3, and 1/3 codes

We will illustrate how the design guidelines can be applied to generate three different codes with three code rates, namely, 1/2, 2/3, and 1/3. For the rate 1/2 we consider a codeword length approximately equal to 2000 bits, in order to have a direct comparison with the code of [10], and we chose a degree distribution with a maximum VN degree of 7, which produces a total number of edges very similar to the one of [10]. The degree distribution of the code and its parity-check matrix are described, respectively, in Table 1 and in Figure 2.

The edges can be classified into three categories: they can be related to the staircase construction, to the deterministic pattern, or their position can be free. In Table 2 we list their distribution: the percentage of randomly placed edges is quite low, so that the burden on the slower ASIP part of the decoder is not excessive.

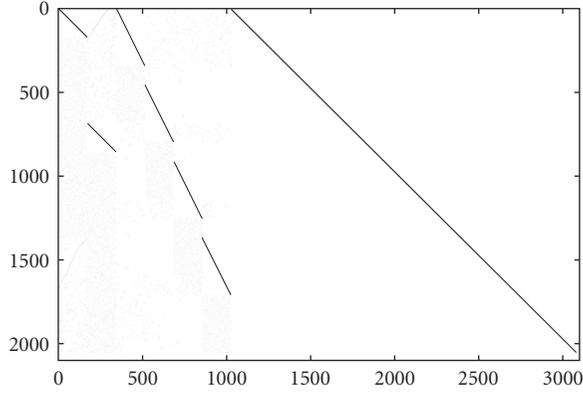


FIGURE 3: Parity-check matrix of a (3078, 1026) code.

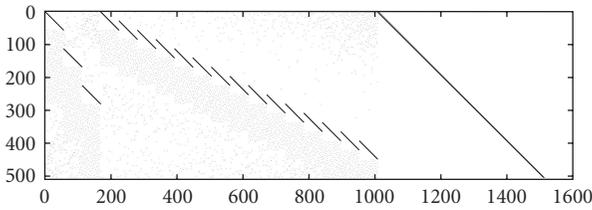


FIGURE 4: Parity-check matrix of a (1512, 1008) code.

According to the scheme of Figure 1, there are nine macro-columns in the central section. If $D = (n - k)/12$, for the i th ($i = 0, 1, \dots, D-1$) column of the j th ($j = 0, 1, \dots, 8$) macro-column, the three edges are placed in the following rows:

- (i) $jD + 2i + (1 - \text{mod}(D, 2))$;
- (ii) $(2 + j)D + \Pi_1(i)$;
- (ii) $(3 + j)D + \Pi_2(i)$.

Based on the equations above, we chose Π_1 and Π_2 as tail-biting S-random interleavers as explained in [11].

The same philosophy holds also for code rates different from 1/2. In particular, we present the cases 1/3 and 2/3 as other examples.

The degree distributions are reported in Table 1, while Table 2 describes the distribution of the edges in the different categories. The two generated LDPC codes have an information word length k of 1026 and 1008, respectively, while the codeword length n is 3078 and 1512. Their parity-check matrixes are depicted in Figures 3 and 4, respectively.

Also in these cases, it is crucial to use tail-biting S-random interleavers as building blocks of the parity-check matrix.

2.4. Simulation results

To verify the validity of the designed algorithm, we compared the rate 1/2 code described in Section 2.3 with two similar codes coming from IEEE 802.11n and IEEE 802.16e standards.

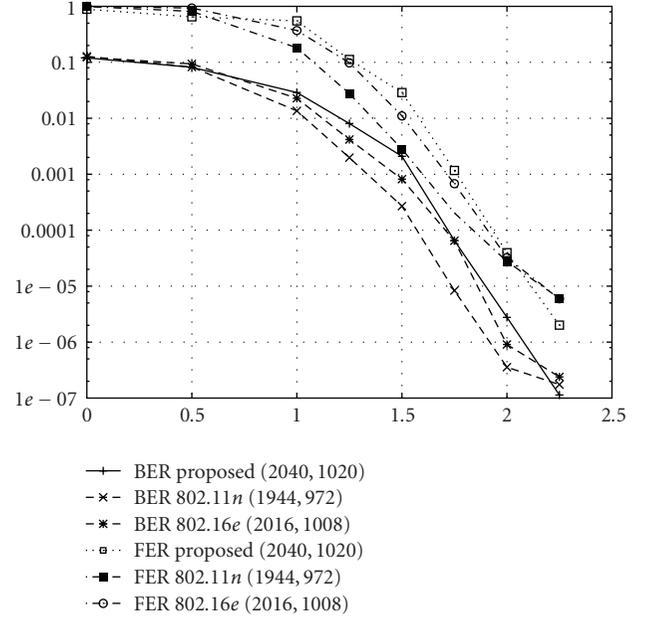


FIGURE 5: Simulation results for code rate 1/2.

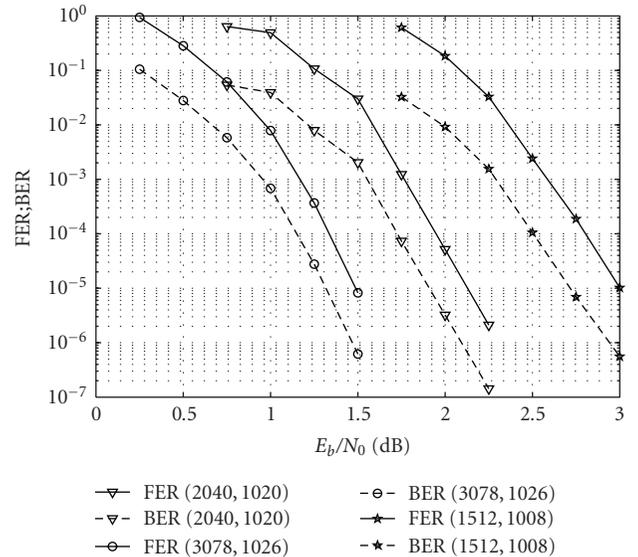


FIGURE 6: FER and BER simulation results for the 1/2, 1/3, and 2/3 code rates.

The simulation results of Figure 5 show that our code does not suffer from high error floor, despite its strong structure (is better than IEEE 802.11n and IEEE 802.16e at high SNR as far as FER is concerned).

Finally, in Figure 6 we provide the simulations results of the LDPC codes generated in the previous subsections. It is reasonably safe to conclude that the structured design approach suggested in this paper leads to good results for a wide range of code rates, both in terms of convergence threshold and of low error floors.

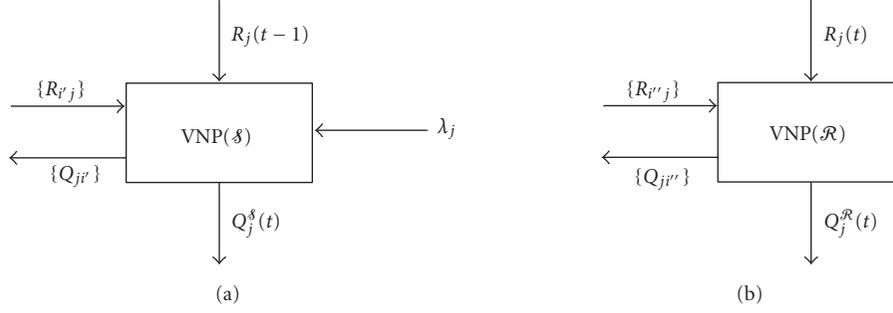


FIGURE 7: Variable node processor input and outputs.

3. DECODING ALGORITHM FOR PARTIALLY STRUCTURED LDPC CODES

The belief propagation (BP) algorithm is one of the most popular LDPC decoding methods, giving optimal performance in the case of H matrices with no cycles. In BP, the following update rule is applied for VN_j :

$$Q_{j,i} = \lambda_j + \sum_{k \in \mathcal{M}(j)/i} R_{k,j}, \quad (1)$$

where λ_j is the intrinsic information, which depends on the channel variance and on the j th received symbol of the codeword, $R_{i,j}$ is the message sent by CN_i to VN_j , $\mathcal{M}(j)$ is the set of nodes connected to VN_j , and $Q_{j,i}$ is the message sent by VN_j to CN_i . All but one input messages R_{ij} are summed in (1) to output a single variable to check message $Q_{j,i}$.

Check node i receives $Q_{j,i}$ messages as inputs and generates output messages according to the following equation:

$$R_{i,j} = \Omega_{k \in \mathcal{N}(i)/j} (R_{k,i}), \quad (2)$$

where $\mathcal{N}(i)$ is the set of nodes connected to $\text{CN } i$ and the Ω operator, for two operands, is defined as

$$\Omega(a, b) = \log \left(\frac{e^a + e^b}{1 + e^{a+b}} \right). \quad (3)$$

For more than two operands, the Ω operator can be applied recursively; for example, for three operands, we have

$$\Omega_{k \in [0 \dots 2]} a_k = \Omega(a_0, \Omega(a_1, a_2)). \quad (4)$$

The BP algorithm is usually executed in two phases, repeated at each decoding iteration: first all variable nodes sample their input messages and process them, then check nodes receive messages and generate their outputs.

We propose here a new decoding algorithm, where the same operations described in (1) and (2) are separately applied to two subsets of nodes: let us define $\mathcal{M}^{\mathcal{S}}(j)$ and $\mathcal{M}^{\mathcal{R}}(j)$ as the two subsets of nodes connected to VN_j and associated, respectively, to structured and random ones of the H matrix. The VN update rule can be rewritten as

$$Q_{j,i} = \lambda_j + \sum_{k \in \mathcal{M}^{\mathcal{S}}(j)/i} R_{k,j} + \sum_{k \in \mathcal{M}^{\mathcal{R}}(j)/i} R_{k,j}. \quad (5)$$

By also defining the following two sums:

$$\begin{aligned} R_j^{\mathcal{R}} &= \sum_{k \in \mathcal{M}^{\mathcal{R}}(j)} R_{k,j}, \\ R_j^{\mathcal{S}} &= \lambda_j + \sum_{k \in \mathcal{M}^{\mathcal{S}}(j)} R_{k,j}, \end{aligned} \quad (6)$$

the variable to check message is computed as

$$Q_{j,i} = R_j^{\mathcal{R}} + R_j^{\mathcal{S}} - R_{i,j}. \quad (7)$$

In Figure 7, a high-level view of the variable node processor operations is given. The purpose here is to better explain which variables are read and written during the execution of the decoding process. The left part of figure shows input and output signals when the VN processor is serving the structured ones ($\text{VNP}(\mathcal{S})$), while the right part refers to the exchanged signals when the same unit is processing random ones ($\text{VNP}(\mathcal{R})$).

The check node processing can be expressed in a similar way. The $\mathcal{N}(i)$ set of nodes connected to $\text{CN } i$ is now split into $\mathcal{N}^{\mathcal{S}}(i)$ and $\mathcal{N}^{\mathcal{R}}(i)$, defined as the two subsets of nodes connected to $\text{CN } i$ and associated, respectively, to structured and random ones. Using this two sets of nodes, we compute the check to variable messages by means of the following expression:

$$R_{i,j} = \Omega(\Omega_{k \in \mathcal{N}^{\mathcal{S}}(i)/j} (Q_{k,i}), \Omega_{k \in \mathcal{N}^{\mathcal{R}}(i)/j} (Q_{k,i})). \quad (8)$$

Two total sums of the incoming messages are defined as

$$\begin{aligned} Q_i^{\mathcal{R}} &= \Omega_{k \in \mathcal{N}^{\mathcal{R}}(i)} (Q_{k,i}), \\ Q_i^{\mathcal{S}} &= \Omega_{k \in \mathcal{N}^{\mathcal{S}}(i)} (Q_{k,i}) \end{aligned} \quad (9)$$

and the check to variable messages are then obtained as

$$R_{i,j} = \Omega(\Omega_{k \in \mathcal{N}^{\mathcal{S}}(i)/j} (Q_{k,i}), Q_i^{\mathcal{R}}) \quad (10)$$

in the case of messages associated to structured ones ($i \in \mathcal{N}^{\mathcal{S}}(i)$), and

$$R_{i,j} = \Omega(\Omega_{k \in \mathcal{N}^{\mathcal{R}}(i)/j} (Q_{k,i}), Q_i^{\mathcal{S}}) \quad (11)$$

in the case of messages associated to random ones ($i \in \mathcal{N}^{\mathcal{R}}(i)$).

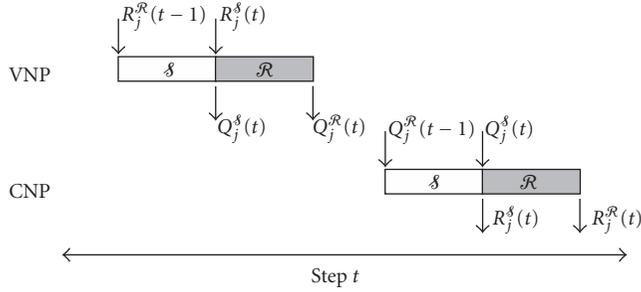


FIGURE 8: Scheduling of operations inside a single iteration of the split decoding algorithm.

The scheduling of the read and write operations for both a VN as well as for the CN is reported in Figure 8. As it can be seen, structured and random parts are always scheduled in sequence.

We call this modified BP algorithm split BP decoding (SBP) or briefly split decoding (SD). The new method is simply an algebraic reorganization of (1) and (2), and the code performance in terms of error correcting capabilities does not decrease when applying (7), (10), and (11). As we will detail in the following section, the key advantage provided by SD is the possibility of independently processing random and structured edges of H : while structured ones follow a modular pattern that strongly simplifies their parallel processing with no collisions in the memory access, the same characteristic does not hold for random ones, which tend to generate collisions when processed in parallel. However, differently from typical random H matrices, the subset of random ones is sparse enough to enable their efficient partitioning into separate memory banks with no collisions in the read or write accesses.

4. DECODING ARCHITECTURE

4.1. Functional description

From an architectural standpoint, the main contribution of our work lies in the different scheduling we use for the decoding process. In fact, as can be observed from Figure 9, the decoder architecture is essentially a partially parallel one very similar to those already proposed in the literature [4–6]. The decoder presented in Figure 9 supports any generic (n, m) LDPC code distributing the $n + m$ node operations over P processing elements (PEs). In particular, the example shown is for $P = 85$ but the same architecture can be used for any value of P . However, in the following analysis we will consider the case of $P = 85$ since for the code designed in Section 2.3 this is the size of the repetitive fixture.

Each PE $_i$ is connected to a memory, called DMEM $_i$. Each PE can write into any memory bank exploiting a crossbar switch, while it can only read from its own memory. These memories are used to store the messages exchanged between PEs during the entire decoding process. Adopting a single crossbar switch solution allows saving of hardware resources without limiting the supported message exchange between

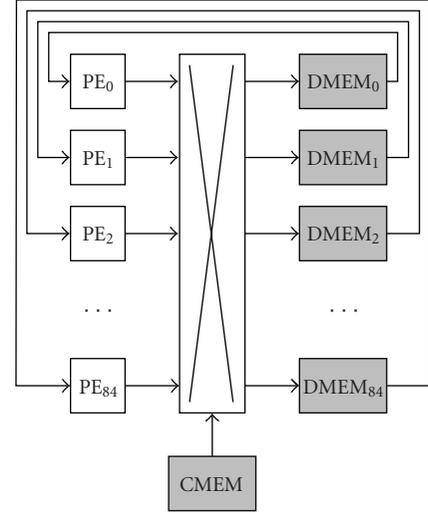


FIGURE 9: General partially parallel architecture.

PEs and memories: in fact, given two generic permutation laws, π_{read} and π_{write} , associated, respectively, to the reading and writing operations of PEs on the memory banks, one of the two laws can always be replaced with a fixed PE to memory coupling, provided that the other law is modified with a new one, obtained as the serial concatenation of π_{read} and π_{write} . We assume the following.

- (i) Each PE is able to be used both as check as well as variable node. Let PE $_i$ be a given processing element of the decoder: this means that through the whole decoding process PE $_i$ will serve n_i variable nodes and m_i check nodes, respectively. If the total workload is evenly split over the PEs' set, $n_i = n/P$ and $m_i = m/P$.
- (ii) Each PE receives messages from one single memory bank (DMEM) through a dedicated connection and sends updated messages to any memory through a $P \times P$ crossbar switch.
- (iii) DMEM is single port memory bank, so that read and write accesses are possible only in different clock cycles.
- (iv) Control values to be applied to the programming input of the crossbar switch are stored in a dedicated memory, CMEM, which is sequentially addressed by a counter.
- (v) The content of CMEM is precomputed offline to resolve memory access conflicts.

Figure 10 represents the data-path architecture of a single PE. Each PE is also reused in time to compute both the structured part as well as the random one. In order to support this feature a second input and some additional logic are required.

Three memory banks are needed to fully support PE operations, namely, LLR MEM, ACC MEM, and S/R MEM. The first one, as the name suggests, is needed to store the information coming from the channel. It is intended to be

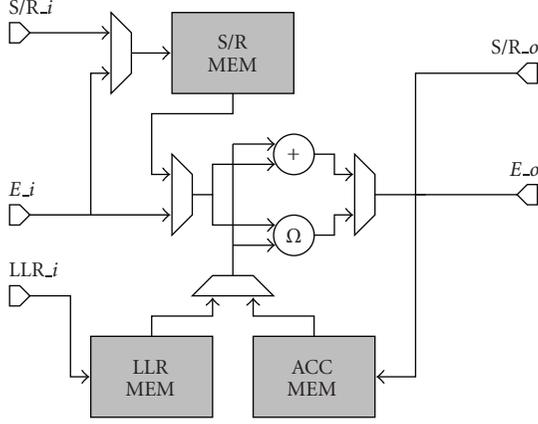


FIGURE 10: Architecture of the processing element dedicated to check and variable node processing.

loaded at the beginning of every new frame to be decoded. ACC MEM is used to implement a flexible accumulator scheme necessary to implement the additions needed both in VN as well as CN. The third memory bank, the so-called S/R memory is useful to allow PE to be reused for structured and random part processing. When the PE is performing the structured part, the S/R memory will hold messages coming from the random one; conversely, when the PE is dedicated to the random part, structured messages will be held into S/R RAM.

When configured to serve as CN processing unit, each PE sequentially receives messages originated by VN (here called generally E_i) from one of the DMEM banks. Partial results of this computation are then stored into the ACC MEM until all the edges for that particular CN have been received. The latency of the whole update procedure for a given CN depends on the degree of the corresponding row in the parity-check matrix. Using the data path depicted in Figure 10 one clock cycle is required for reading each incoming edge as one cycle is needed to write the result back (edge out). The same holds for the VN case, that is, the total latency is equal to twice the column degree.

4.2. Performance analysis

Given a decoder architecture as the one of Figure 9 and a flexible PE as in Figure 10 it is possible to derive a sort of *performance bound* for the decoding latency as well as for the throughput. Given an even workload distribution among the PEs and no collisions in reading message memories, the number of cycles to perform a single iteration can be expressed as

$$\hat{D} = 4 \cdot \frac{E}{P}, \quad (12)$$

where E is the total number of ones in the parity-check matrix, P is the parallelism of the system. The factor of four comes out from the structure of the PE itself. This number assumes that only single-port memories can be used and that layered decoding scheduling is not exploited.

Under these premises the maximum theoretical decoding throughput \hat{T} can be expressed as

$$\hat{T} = \frac{(n-m)f_{CK}}{I \cdot \hat{D}}, \quad (13)$$

where f_{CK} is the clock frequency and I the number of iterations performed. With the proposed 1/2 rate code, assuming a number of iterations equal to ten and a parallelism of $P = 85$ it turns out that $\hat{T} = 0.318 f_{CK}$.

However, in practical situations memory conflicts force to delay simultaneous accesses to the same bank and then to insert stall cycles. We will then introduce a *collisions degradation factor* α able to capture this behavior. We can then express the total number of cycles needed for each iteration in case of collisions as

$$D_{coll} = \alpha \cdot \hat{D}. \quad (14)$$

The resulting throughput of the decoder can be rewritten as

$$T_{coll} = \frac{1}{\alpha} \cdot \hat{T}. \quad (15)$$

It is important to stress how α depends on the specific LDPC code but it is also affected by message scheduling and load partitioning between PE. Partitioning and scheduling techniques can be used to try to minimize α ; however, it is known that good parity-check matrices show little adjacency making partitioning and scheduling benefits very limited.

To evaluate the impact of α we try to partition the associated Tanner graph in order to perform an initial allocation. We use the software Metis [24] freely available on the Internet. Given this allocation we implement a cycle-accurate architectural simulator using the Python language, able to report the total number of cycles needed to perform a single iteration (i.e., D_{coll}). In this way, it has been possible to derive α values for different types of LDPC codes (see Table 3).

The important peculiarity observed is that the α factor for a given LDPC tends to be as large as four. As an example, in the case the proposed code we found $\alpha = 4.28$. This means that the potential parallelism of the architecture given in Figure 9 is largely wasted.

Table 3 summarizes these values for the proposed code (reported as code 1) as well as for other four irregular LDPC codes. In particular, we provide the values of \hat{D} , D_{coll} , and relative throughput figures, evaluated as a function of the clock frequency. To show how collisions are a concern also on different LDPC codes we perform the same partitioning and scheduling steps also on different codes. Code labelled as 2 is directly taken from Professor MacKay website [25], where it can be found as 4986.93i.939. Code 3 is the IEEE 802.11n wireless local area network channel code and exhibits characteristics similar to the proposed one, while code labelled as 4 is one of the IEEE 802.16e 1/2 rate codes. Also in these three cases it is possible to observe how the presence of a significant number of collisions spoils the overall system performance. For the last two codes, however, it is important to consider that they have been specifically

TABLE 3: Cycle and throughput evaluation for different LDPC codes.

Code	n	m	E	\hat{D}	D_{coll}	α	\hat{T}	T_{coll}
1	2040	1020	6885	324	1387	4.28	$0.32 f_{\text{CK}}$	$0.07 f_{\text{CK}}$
2	9972	4986	14958	704	2794	3.97	$0.71 f_{\text{CK}}$	$0.18 f_{\text{CK}}$
3	1944	972	6797	320	1240	4.19	$0.3 f_{\text{CK}}$	$0.07 f_{\text{CK}}$
4	2016	1008	6384	301	1259	4.18	$0.33 f_{\text{CK}}$	$0.08 f_{\text{CK}}$

designed to be decoded avoiding collisions, provided that dual port memories are used. Despite this, we put these code in Table 3 to enhance the relevance of collisions in LDPC decoding as a severe limitation to system throughput.

The SD algorithm described in Section 3 allows to get over this limitation by exploiting the partial structure of the proposed codes. The only significant modification to the PE structure is the need for S/R MEM to store partial VN to CN and N to VN messages, $Q_{i,j}^{\text{S}}$, $Q_{i,j}^{\text{R}}$, $R_{i,j}^{\text{S}}$, and $R_{i,j}^{\text{R}}$.

The complexity increase due to these hardware modifications is negligible and it will be evaluated in Section 4.3. Instead, the reduction of required decoding cycles impacts significantly on the size of CMEM needed to control the crossbar. We derive here for the proposed approach the offered throughput, which depends on the total number of cycles required to complete all read and write memory accesses.

An iteration is divided into the following four subiterations:

- (1) Φ_1 : check node processing for structured ones;
- (2) Φ_2 : check node processing for random ones;
- (3) Φ_3 : variable node processing for structured ones;
- (4) Φ_4 : variable node processing for random ones.

In the first subiteration, variable-to-check messages associated to structured ones of H matrix are read by the PEs, as structured ones are contained into $P \times P$ submatrices concentrated along three lines in H ; the number of cycles required to complete the reading is equal to the number of submatrices. It is worth noting that no conflicts are possible since each submatrix has a single one per row and per column. In the case of the code rate 1/2 matrix given in Figure 2, this number is equal to $D_{\Phi_1, \text{read}} = E_{\text{str}}/P$ being E_{str} the total number of structured edges. Subiteration Φ_1 also needs to write the generated check-to-variable messages: this operation takes the same time as the reading one, $D_{\Phi_1, \text{write}} = D_{\Phi_1, \text{read}}$. Additionally, each PE in phase Φ_1 must also read the partial VN-to-CN messages Q^{R} and update the partial CN-to-VN messages R^{S} : both operations take a number of cycles equal to m/P . Thus, phase Φ_1 takes a total number of cycles equal to

$$D_{\Phi_1} = 2 \cdot \left(\frac{E_{\text{str}} + m}{P} \right). \quad (16)$$

The same value is obtained in the calculation of the total number of cycles needed to complete subiteration Φ_3 (variable node processing for structure ones), with the

exception that the number of cycles needed to read and write CN to VN messages here is n/P , hence (16) becomes

$$D_{\Phi_3} = 2 \cdot \left(\frac{E_{\text{str}} + n}{P} \right). \quad (17)$$

In subiteration Φ_2 , a number of messages equal to the number of the ones in the random part of H matrix must be read and written. Since these ones are distributed in an irregular way, there is no guaranty that they can be handled in parallel P at a time with no collisions. However, very low density of random ones enables their efficient partitioning among the P memories: experiments done with cases proposed in Section 2 and additional random matrices having the same one density show that random ones can almost always be partitioned with no collisions. Under this assumption, the total number of read or write operations is equal to E_{rnd}/P , where E_{rnd} is the number of random ones in H . Subiteration Φ_2 also needs m/P cycles to read the partial VN-to-CN messages Q^{S} and update the partial CN-to-VN messages R^{R} . The total number of cycles for Φ_2 can then be expressed as

$$D_{\Phi_2} = 2 \cdot \left(\frac{E_{\text{rnd}} + m}{P} \right) \quad (18)$$

while Φ_4 cycles can be derived as for the case of Φ_3 , hence

$$D_{\Phi_4} = 2 \cdot \left(\frac{E_{\text{rnd}} + n}{P} \right) \quad (19)$$

combining the four contributions together it turns out that

$$\begin{aligned} D_{\text{sd}} &= 4 \cdot \left(\frac{E_{\text{str}} + E_{\text{rnd}} + n + m}{P} \right) \\ &= 4 \cdot \left(\frac{E + n + m}{P} \right). \end{aligned} \quad (20)$$

We define D_{sd} as the number of cycles needed when the proposed split decoding approach is used. As already stated in (13), the resulting throughput will be

$$T_{\text{sd}} = \frac{(n - m)f_{\text{CK}}}{I \cdot D_{\text{sd}}}. \quad (21)$$

Lastly, we introduce the parameter η as a measure of the total efficiency of the decoding process with respect to the ideal case. For instance, for SD we have

$$\eta_{\text{sd}} = \frac{\hat{D}}{D_{\text{sd}}} = \frac{E}{E + n + m}. \quad (22)$$

TABLE 4: Memory occupation breakdown.

Memory	Instances	Parallelism	Split decoding		Area	Total area
			Words	[Words]		
DMEM	85	8	$\left\lceil 2 \frac{E+m+n}{P} \right\rceil$	256	$12000 \mu\text{m}^2$	1.0 mm^2
CMEM	1	$7 \cdot P = 595$	$D_{\text{sd}}/2$	512	$750 \cdot 10^3 \mu\text{m}^2$	0.75 mm^2
LLR MEM	85	8	$\left\lceil \frac{n}{P} \right\rceil$	32	$2550 \mu\text{m}^2$	0.22 mm^2
ACC MEM	85	8	$\max\left(\left\lceil \frac{n}{P} \right\rceil, \left\lceil \frac{m}{P} \right\rceil\right)$	32	$2550 \mu\text{m}^2$	0.22 mm^2
S/R MEM	85	8	$\max\left(\left\lceil \frac{n}{P} \right\rceil, \left\lceil \frac{m}{P} \right\rceil\right)$	32	$2550 \mu\text{m}^2$	0.22 mm^2
Total						2.36 mm^2
Partition and scheduling (collision)						
Memory	Instances	Parallelism	Words	[Words]	Area	Total area
DMEM	85	8	$\left\lceil 2 \frac{E}{P} \right\rceil$	256	$12000 \mu\text{m}^2$	1.0 mm^2
CMEM	1	$7 \cdot P = 595$	$D_{\text{coll}}/2$	2048	$2.8 \cdot 10^6 \mu\text{m}^2$	2.8 mm^2
LLR MEM	85	8	$\left\lceil \frac{n}{P} \right\rceil$	32	$2550 \mu\text{m}^2$	0.22 mm^2
ACC MEM	85	8	$\max\left(\left\lceil \frac{n}{P} \right\rceil, \left\lceil \frac{m}{P} \right\rceil\right)$	32	$2550 \mu\text{m}^2$	0.22 mm^2
Total						4.24 mm^2

In the case of the proposed code, we have $E = 6885$, $n = 2040$, and $m = 1020$, hence $\eta_{\text{sd}} = 0.69$. It is extremely interesting to compare this figure with the one resulting when only partitioning and scheduling techniques are applied on the same code. In that case, we use for D_{coll} the expression as reported in (14), leading to

$$\eta_{\text{coll}} = \frac{1}{\alpha} \quad (23)$$

and with $\alpha = 4.28$ it turns out that $\eta = 0.23$, nearly three times less than the one obtainable with the proposed method.

If we are interested in the throughput for ten iterations, on the same code it turns out that $T_{\text{coll}} = \eta_{\text{coll}} \hat{T} = 0.074 f_{\text{CK}}$ while SD achieves $T_{\text{sd}} = \eta_{\text{sd}} \hat{T} = 0.22 f_{\text{CK}}$. Thus, the proposed approach achieves a throughput speedup of nearly three for the same decoding architecture and the same clock frequency, as reported in Table 3.

4.3. Synthesis results and performance

The partially parallel architecture previously discussed has been described in VHDL, synthesized and mapped on a $0.13 \mu\text{m}$ CMOS standard cell technology, considering the (2040, 1020) 1/2 rate code.

After logical synthesis and mapping, the maximum combinational delay was $t_{\text{pd}} = 2.5$ nanoseconds which corresponds to a clock frequency of $f_{\text{CK}} = 400$ MHz. As far as the decoding throughput, the split decoding solution is able to achieve 88 Mbps with 10 iterations while the

direct mapping of the same code on the same architecture leads to a throughput of 29.6 Mbps. This means that the proposed solution can achieve the same throughput as a straightforward one using a clock frequency as low as one third with respect to the traditional one. Since power dissipation in CMOS circuits is directly related to the clock frequency, power dissipation of one third or a battery life of three times greater can be achieved with the same throughput.

As far as the area occupation is concerned it is possible to separate two main contributions: area that is dedicated to directly implement logical functions and area which is devoted to memory. The former can also be divided into different contributions, namely, PEs and the crossbar switch. Each PE requires 4.61 equivalent kgates leading to an area occupation of almost 392 kgates. The 85×85 crossbar switch requires 172 equivalent kgates. Hence, a total area occupation of 564 kgates is needed to implement the decoder logic. It is important to remark how this area requirement is independent from the chosen scheduling, thus both SD as well as the traditional partially parallel solution require the same area.

Memory is where things are different for the two approaches: these data are collected in Table 4. As it can be seen, some memories have the same dimensions regardless of the scheduling adopted. What really matters is the size of the memory needed to control the crossbar during an iteration. The number of words required depends directly on the number of write cycles: then each word needs to store a complete 85×85 permutation. For the sake of simplicity,

TABLE 5: Comparisons with state-of-art decoders implementations.

	Previously published architectures			Proposed
	[26]	[27]	[28]	
CN method	3-min	minsum	minsum	Ω
Precision	6 bit	8 bit	6 bit	8 bit
Technology	65 nm	130 nm	90 nm	130 nm
Frequency	400 MHz	83 MHz	109 MHz	400 MHz
Logic gates	520 kgates	420 kgates	380 kgates	564 kgates
Memory bits	500 kbits	106 kbits	100 kbits	544 kbits
Iterations	20	8	20	10
Net throughput	48 MBps	60 MBps	63 MBps	88 MBps
Normalized throughput	960 MBps	480 MBps	1260 MBps	880 MBps
TAR [5]	381	569	1620	321

we decided to store these control signal uncoded using 7 bits to control each crossbar line. This leads to a word length of $85 \times 7 = 595$ bits in either cases. It is also important to note how the memory increase due to S/R MEM is completely negligible when compared to the total area. Summarizing the split decoding approach needs almost 390 equivalent kgates, with respect to more than 700 kgates needed by traditional approach. This results in an area saving of more than one third, bringing also significant possibilities to reduce the overall power dissipation.

To better evaluate the validity of this approach it would be interesting to compare synthesis results to recent LDPC decoder implementations. In Table 5 some implementation results are sketched. Given the particularity of the presented approach both in terms of code design as well as hardware implementation, these comparisons are not straightforward. In order to compare our architecture to similar approaches, we select works that implement IEEE 802.16e LDPC decoder. Throughput figures are obtained considering the largest available 1/2 rate code (i.e., $n = 2304$). Even if this code presents a larger block length than the proposed one, it is important to remark that the total number of edges in this case is $E = 7296$. Comparing this number with the proposed code ($E = 6885$) it turns out that IEEE 802.16e code shows a complexity of 1.06 with respect to the proposed (2040, 1020) code. Under this premises it is then reasonable to consider the two cases almost comparable.

Additionally, it is important to consider how our results have been obtained addressing the (2040, 1020) code: given the flexibility of our partially parallel decoder the same hardware can be exploited to decode also IEEE 802.16e codes. In such a case, obviously, the advantages deriving from SD cannot be exploited anymore.

As can be seen from Table 5, the first two important things to compare are internal data representation and CN implementation: it is worth noting how our architecture represents messages using 8 bits and resorts to the Ω operator as far as the CN is concerned. While this choice tends to produce a larger PE area with respect to minsum approaches, the decoding performances are improved, enabling less decoding iterations to be implemented. In particular, [29] showed how minsum performance tends to be degraded by

fixed-point implementations and in presence of irregular codes. Starting from these considerations we expect our area occupation to be larger than the other ones.

Also the technology used is quite dispersed over different values: while area occupation can be properly compared, this does not hold for delay figures. Our design is able to achieve 2.5 nanoseconds period after logical synthesis on a 130 nm technology node: we expect that resorting to a 65 nm technology asynchronous delay would be less.

As far as the area occupation is concerned, it is important to consider both the logic gates contribution as well as the memory requirements. From Table 5 it is possible to observe how our architecture exhibits a logic gate count similar to others, while memory footprint is where things are different. This difference is mainly due to two factors: the use of lookup table operations inside PEs and the use of a crossbar switch as interconnection fabric. While the former could also be relaxed, moving towards a minsum implementation, the latter cannot be avoided due to more complex code structure with respect to the 802.16e case. In that case, in fact, PEs connections can be implemented using simpler permutations that are obtained from identity matrix cyclical rotations. In our case, on the other hand, we need to support arbitrary permutations to enable collision-free decoding of the random part.

It is also interesting to analyze throughput results. It should be noted how decoding net throughput of considered cases falls between 48 and 63 MBps. Our decoder is able to achieve 88 MBps with 10 decoding iterations. If a throughput of 60 MBps is required, decoding iterations can be increased up to 14, enhancing the correction capabilities. In Table 5 we also include an additional throughput figure called *normalized throughput*: these data are extrapolated from the net throughput multiplied by the total number of decoding iterations. We decided to include also these data to better emphasize the effective throughput sustained by each compared architecture.

Finally, to better assess the throughput-area tradeoff we also report a number called TAR [5]. Under this perspective, it is important to observe how the proposed approach presents a TAR lower than others. However, one should not neglect how TAR is directly affected by technology

node, as already noted earlier. Additionally, our architecture is able to decode codes with highly irregular structure, while the others presented are limited to traditional partially structured codes.

5. CONCLUSIONS

In this paper, we present a novel class of partially structured eIRA codes. We also show how a code of this class can essentially perform equivalently to other state-of-art LDPC codes, while preserving some desirable properties that can be exploited when implementing a decoder. Then we focused on the main issues of implementing a partially parallel decoder architecture suitable for this class of codes. In this framework, we devised an alternative decoding approach, namely, the split decoding, which exhibits remarkable advantages over traditional methods. Following this approach memory requirements can be relaxed of more than one third, leading to significant reductions in power dissipation.

Additionally, split decoding enables also the possibility of achieving higher decoding throughput without any hardware impact. Thanks to this higher efficiency, the clock frequency can be reduced further reducing the total power. Finally, we compare an SD-based architecture with three state-of-art LDPC decoders. From this comparison, it can be noted how the proposed architecture presents similar decoding throughput with a larger area occupation, mainly due to internal data representation and interconnection network. Still it is our opinion that the presented approach is valuable, being able to deal with highly irregular parity-check matrices without sacrificing decoding throughput.

As far as future directions are concerned we feel that split decoding performance could be increased borrowing some ideas from shuffled decoding [30]. In particular, shuffled decoding could enhance the parallelism degree between structured and random edges processing, leading to an increased overall throughput.

ACKNOWLEDGMENT

This work was supported by the European Commission funded Network of Excellence NEWCOM++ under the 7th Framework Programme.

REFERENCES

- [1] R. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.
- [2] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 399–431, 1999.
- [3] A. J. Blanksby and C. J. Howland, "A 690-mW 1-Gb/s 1024-b, rate-1/2 low-density parity-check code decoder," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 3, pp. 404–412, 2002.
- [4] T. Brack, F. Kienle, and N. Wehn, "Disclosing the LDPC code decoder design space," in *Proceedings of the Conference on Design, Automation and Test in Europe (DATE '06)*, vol. 1, pp. 200–205, Munich, Germany, March 2006.
- [5] G. Masera, F. Quaglio, and F. Vacca, "Implementation of a flexible LDPC decoder," *IEEE Transactions on Circuits and Systems II*, vol. 54, no. 6, pp. 542–546, 2007.
- [6] S.-H. Kang and I.-C. Park, "Loosely coupled memory-based encoding architecture for low density parity check codes," *IEEE Transactions on Circuits and Systems I*, vol. 53, no. 5, pp. 1045–1056, 2006.
- [7] F. Kienle, M. J. Thul, and N. Wehn, "Implementation issues of scalable LDPC-decoders," in *Proceedings of the 3rd International Symposium on Turbo-Codes & Related Topics*, pp. 291–294, Brest, France, September 2003.
- [8] A. Tarable, S. Benedetto, and G. Montorsi, "Mapping interleaving laws to parallel turbo and LDPC decoder architectures," *IEEE Transactions on Information Theory*, vol. 50, no. 9, pp. 2002–2009, 2004.
- [9] F. Quaglio, F. Vacca, C. Castellano, A. Tarable, and G. Masera, "Interconnection framework for high-throughput, flexible LDPC decoders," in *Proceedings of the Conference on Design, Automation and Test in Europe (DATE '06)*, vol. 2, pp. 124–129, Munich, Germany, March 2006.
- [10] E. Kim and G. S. Choi, "Diagonal low-density parity-check code for simplified routing in decoder," in *Proceedings of IEEE Workshop on Signal Processing Systems Design and Implementation (SiPS '05)*, vol. 2005, pp. 756–761, Athens, Greece, November 2005.
- [11] L. Dinoi, R. Martini, G. Masera, F. Quaglio, and F. Vacca, "ASIP design for partially structured LDPC codes," *Electronics Letters*, vol. 42, no. 18, pp. 1048–1049, 2006.
- [12] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619–637, 2001.
- [13] L. Ping, W. K. Leung, and N. Phamdo, "Low density parity check codes with semi-random parity check matrix," *Electronics Letters*, vol. 35, no. 1, pp. 38–39, 1999.
- [14] M. Yang, W. E. Ryan, and Y. Li, "Design of efficiently encodable moderate-length high-rate irregular LDPC codes," *IEEE Transactions on Communications*, vol. 52, no. 4, pp. 564–571, 2004.
- [15] Y. Zhang, W. E. Ryan, and Y. Li, "Structured eIRA codes with low floors," in *Proceedings of IEEE International Symposium on Information Theory (ISIT '05)*, vol. 2005, pp. 174–178, Adelaide, Australia, September 2005.
- [16] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 657–670, 2001.
- [17] G. Durisi, L. Dinoi, and S. Benedetto, "eIRA codes for coded modulation systems," in *Proceedings of IEEE International Conference on Communications (ICC '06)*, vol. 3, pp. 1125–1130, Istanbul, Turkey, June 2006.
- [18] X.-Y. Hu, E. Eleftheriou, and D.-M. Arnold, "Progressive edge-growth tanner graphs," in *Proceedings of IEEE Global Communications Conference (GLOBECOM '01)*, vol. 2, pp. 995–1001, San Antonio, Tex, USA, November 2001.
- [19] T. Tian, C. Jones, J. D. Villasenor, and R. D. Wesel, "Construction of irregular LDPC codes with low error floors," in *Proceedings of IEEE International Conference on Communications (ICC '03)*, vol. 5, pp. 3125–3129, Anchorage, Alaska, USA, May 2003.
- [20] A. Ramamoorthy and R. Wesel, "Construction of short block length irregular low-density parity-check codes," in *Proceedings of IEEE International Conference on Communications (ICC '04)*, vol. 1, pp. 410–414, Paris, France, June 2004.

- [21] L. Doini, F. Sottile, and S. Benedetto, "Design of variable-rate irregular LDPC codes with low error floor," in *Proceedings of IEEE International Conference on Communications (ICC '05)*, vol. 1, pp. 647–651, Seoul, Korea, May 2005.
- [22] G. Richter and A. Hof, "On a construction method of irregular LDPC codes without small stopping sets," in *Proceedings of IEEE International Conference on Communications (ICC '06)*, vol. 3, pp. 1119–1124, Istanbul, Turkey, June 2006.
- [23] L. Doini, F. Sottile, and S. Benedetto, "Design of versatile eIRA codes for parallel decoders," to appear in *IEEE Transactions on Communications*.
- [24] <http://glaros.dtc.umn.edu/gkhome/views/metis>.
- [25] D. J. C. MacKay, "Encyclopedia of sparse graph codes," <http://www.inference.phy.cam.ac.uk/mackay/codes/data.html>.
- [26] T. Brack, M. Alles, T. Lehnigk-Emden, et al., "Low complexity LDPC code decoders for next generation standards," in *Proceedings of the Conference on Design, Automation and Test in Europe (DATE '07)*, pp. 331–336, Nice, France, April 2007.
- [27] X.-Y. Shih, C.-Z. Zhan, C.-H. Lin, and A.-Y. Wu, "An 8.29 mm² 52 mW multi-mode LDPC decoder design for mobile WiMAX system in 0.13 μ m CMOS process," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 3, pp. 672–683, 2008.
- [28] C.-H. Liu, S.-W. Yen, C.-L. Chen, et al., "An LDPC decoder chip based on self-routing network for IEEE 802.16e applications," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 3, pp. 684–694, 2008.
- [29] D. Oh and K. K. Parhi, "Performance of quantized min-sum decoding algorithms for irregular LDPC codes," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS '07)*, pp. 2758–2761, New Orleans, La, USA, May 2007.
- [30] J. Zhang and M. P. C. Fossorier, "Shuffled iterative decoding," *IEEE Transactions on Communications*, vol. 53, no. 2, pp. 209–213, 2005.

Research Article

Turbo Decoder for Low-Power Ultrawideband Communication Systems

Esam A. Obiedat and Lei Cao

Department of Electrical Engineering, University of Mississippi, Oxford, MS 38677, USA

Correspondence should be addressed to Esam A. Obiedat, obiedat@ieee.org

Received 1 April 2008; Revised 11 July 2008; Accepted 8 October 2008

Recommended by Marina Mondin

A new method to reduce the computational complexity of the turbo decoding in ultrawideband (UWB) orthogonal frequency division multiplexing (OFDM) system is proposed. Existing stopping techniques for turbo decoding process using constrained decoding assume fixed signal-to-noise ratio (SNR) for all the OFDM symbol bits so they fail to yield an acceptable bit-error rate (BER) performance in multicarrier systems. In this paper, we propose a bit-level stopping technique for turbo decoding process based on the constrained decoding method. In this technique, we combine the cyclic redundancy check (CRC) with an adaptive threshold on the log likelihood ratio (LLR) on each subcarrier to detect for convergence. The threshold is adaptive in the sense that the threshold on the LLR of a bit is determined by the average SNR of the OFDM symbol and the channel gain of the transmission subcarrier. Results show that when the channel state information (CSI) is used to determine the threshold on LLR, the stopping technique can reduce the computational complexity by about 0.5–2.5 equivalent iterations compared to GENIE turbo without degradation in the BER performance.

Copyright © 2008 E. A. Obiedat and L. Cao. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

A leading candidate for fast data transfer is the ultrawideband (UWB), a wireless technology designed for short-range Personal area networks (PANs). High data throughput and low power consumption for distances of less than 10 meters are among the main features of UWB orthogonal frequency division multiplexing (OFDM) systems, which are very applicable to digital home requirements. Due to the Shannon limit approaching performance, turbo codes are expected to play a key role in UWB systems. Turbo codes make it possible to increase data rate without increasing the power of transmission, or they can be used to decrease the amount of power used to transmit at a certain data rate. However, the increase in computational complexity, power consumption, and latency due to the additional computations is big considerations before implementation in UWB systems.

The main challenge in implementing turbo codes in the UWB systems is the consequent complexity consideration in spite of the high bit-error rate (BER) performance and savings in transmission power consumption as pointed out above. The complexity of Log-MAP is approximately four

times that of Viterbis algorithm for a single iteration, which means that to complete 10 full iterations in turbo code, the computational complexity will be around 40 times the computational complexity of the convolutional decoder in addition to increased power consumption and latency due to additional computations. The attractive enhancement in performance and reduction in transmission power for the system with turbo codes makes it important to study the implementation issue in UWB systems in order to reduce the complexity to a reasonable level while maintaining the same performance.

Approaches for reducing the complexity of turbo decoder have appeared and are referred to as early detection or stopping techniques [1–8]. These approaches are based on stopping the iterative process of turbo decoder at a certain point instead of continuing a fixed number of iterations. A criterion is used to determine when the iterative process can be stopped with minimum loss in BER performance. All previously proposed stopping techniques were based on single-carrier systems, where the signal-to-noise ratio is assumed to be constant for all frame bits. These approaches may not be directly implemented for turbo decoder in

multicarrier systems like UWB OFDM, since each bit (or group of bits) undergoes different signal-to-noise ratios (SNRs) due to the different fading parameters of different subcarriers. The resultant different SNRs for each subcarrier after channel equalizations imply different reliabilities, this necessitates different stopping strategies at each subcarrier in turbo decoding.

In this paper, we introduce a new criterion for bit-level stopping for turbo decoders on UWB systems that assign different stopping strategies to each subcarrier based on the average SNR and the instantaneous SNR of that subcarrier. The proposed technique uses constrained decoding method [1, 5] by clamping the bits satisfying the log likelihood ratio (LLR) threshold condition, and stops the decoding process when the cyclic redundancy check (CRC) detects that the whole frame is correctly decoded. The threshold is adaptively determined using channel state information (CSI) values of the UWB channel. The results are compared to the ideal case of the stopping techniques, which is called the GENIE case. In GENIE stopping technique, the decoder is assumed to know all the transmitted bits and stops the decoding process when all the bits are correctly decoded, unless the number of iterations reaches the maximum allowed iterations. Therefore, GENIE case is assumed to have the best BER performance and the least number of iterations that can be achieved by any *frame-based* stopping criterion for a given BER performance.

The remainder of the paper is organized as follows. The quasistatic UWB channel model and general assumptions made for the system model used in the simulations are given in Section 2. In Section 3, we briefly review the current stopping techniques and present simulation results for existing packet-level stopping techniques in UWB channels. The propose stopping technique for turbo decoder is presented in Section 4.

2. SIMULATION ENVIRONMENT

The UWB channel is classified as multipath quasistatic fading channel, meaning that the channel impulse response remains almost constant for a number of transmitted frames and the channel estimation gives almost perfect estimates for the CSI. The channel modeling subgroup of the IEEE 802.15.3 committee proposed a channel model for UWB systems [9] based on the S-V model [10] for arriving multipath components, which are considered to arrive in clusters of rays (paths). The rays have independent uniform phases, and independent Rayleigh amplitudes with variances that decay exponentially with cluster and ray delays. The clusters and the rays within the cluster form Poisson arrival processes with different, but fixed, rates. The clusters are formed by the building superstructure, while the individual rays are formed by objects in the vicinity of both the transmitter and the receiver. According to [9], the channel impulse response for the m th transmitted multiband OFDM symbol is given by

$$h_m(t) = \sum_{c=0}^{C-1} \sum_{r=0}^{R_c-1} \rho_m(c, r) \alpha_m(c, r) \delta(t - T_c^m - \tau_{c,r}^m), \quad (1)$$

where $\alpha_m(c, r)$ and $\rho_m(c, r)$ are the multipath gain and reflection coefficient, respectively, of the r th ray in the c th cluster, and the reflection coefficient has equal probability on “+1” or “-1.” The total number of clusters is C , each contains R_c rays. T_c^m and $\tau_{c,r}^m$ are the excess delay of the c th cluster and the excess delay of the r th ray in the c th cluster, respectively. The values of $\alpha_m(c, r)$, $\rho_m(c, r)$, R_c , T_c^m , and $\tau_{c,r}^m$ are assumed to be constant for each OFDM symbol transmission (quasistatic channel). The corresponding channel frequency response for this realization at the k th subcarrier is given by

$$H_k = \sum_{c=0}^{C-1} \sum_{r=0}^{R_c-1} \rho_m(c, r) \alpha_m(c, r) e^{-j2\pi(f_0+k\Delta f)(T_c+\tau_{c,r})}, \quad (2)$$

where f_0 is the first subcarrier (lowest frequency) of the OFDM symbol, and Δf is the bandwidth of each subcarrier.

The CRC turbo-encoded sequence is first interleaved and converted to modulation symbols and then the latter is OFDM modulated. Frequency hopping is applied to the OFDM symbols before transmitting them through the UWB channel. At the receiver, the channel output is converted back to the frequency domain by fast Fourier transform (FFT) after removing the cyclic prefix. In the frequency domain, if we let $\mathbf{X} = [X_1, X_2, \dots, X_N]^T$ be the binary phase shift keying (BPSK) modulated baseband symbols transmitted in a single OFDM symbol, then the received symbols $\mathbf{Y} = [Y_1, Y_2, \dots, Y_N]^T$ after the FFT stage are

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{Z}, \quad (3)$$

where $\mathbf{Z} = [Z_1, Z_2, \dots, Z_N]^T$ is the Additive white Gaussian noise (AWGN) vector. $\mathbf{H} = \text{Diag}(H_k)$, $k = 1, \dots, N$, where H_k is defined in (2) and N is the number of subcarriers.

The matrix \mathbf{H} components “channel gains” are used to equalize the received signal \mathbf{Y} . The resultant soft channel output BPSK demodulated vector after equalization $\hat{\mathbf{X}} = [\hat{X}_1, \hat{X}_2, \dots, \hat{X}_N]$, where $\hat{X}_i = 4R(E_b/N_0)\text{Re}\{Y_i H_i^*\}$, $i = 1, \dots, N$, is deinterleaved before sending it to the turbo decoder.

For all the simulations in this paper, the channel is assumed to be perfectly estimated and there is no intercarrier interference (ICI) (i.e., the delay spread is less than the cyclic prefix). All interleavers used in the simulation model, the turbo encoder interleaver, and the channel interleaver which proceeds the OFDM modulation, are random interleavers. After adding the CRC sequence for the frame, the 1024 bit frame is turbo encoded with $1/2 (7, 5)_{\text{oc}}$ recursive systematic convolutional (RSC) encoder. For each UWB channel model, CM1 line of sight (LOS) (0–4 m), CM2 nonlinear of sight (NLOS) (0–4 m), CM3 NLOS (4–10 m), and CM4 extreme NLOS, the channel impulse response is obtained randomly according to the channel model characteristics in [11].

3. REVIEW OF THE CURRENT STOPPING TECHNIQUES BASED ON CRC DETECTION

It is noted from simulations of turbo codes that after a certain number of iterations, the improvement in BER performance for any additional iteration is so small that the

decoding process can be stopped with minimum degradation in performance. Considering this, many researchers have sought the best method to detect the stopping point considering three main targets: (1) to maintain the performance degradation (if the stopping method results in a degradation) within certain limits from the performance of turbo decoder with fixed number of iterations; (2) to minimize the number of iterations while keeping in mind the first objective; (3) to assure that the added computation complexity for detecting the early stopping point should not exceed the computation complexity reduced by stopping the iterations.

Generally, stopping techniques can be categorized into three different classes according to the clamping level of bits: frame level, packet level, and bit level. The frame-level stopping technique terminates the iterative process for the entire frame only when the condition for termination has been satisfied. On the other hand, packet and bit-level stopping techniques use constrained decoding by only clamping the bits that are satisfying the criterion condition and continuing on decoding process for remaining bits. There are many examples in the literature of stopping techniques based on frame level, but few methods use the bit- and packet-level terminations. An example of the bit-level stopping technique is the method [2] which implements a trellis slicing algorithm to detect information and codeword symbols as well as state variables during the decoding. However, since the slicing in [2] is based on LLR values, a very large error floor may result.

A generic packet-level method for early detection in iterative decoding processes is introduced in [1], which suggests dividing the full frame into smaller packets and adding a small CRC sequence to each individual packet. The CRC sequences are used at the receiver as an inner code to detect any decoding error in the turbo-decoded packet after each half iteration. If a certain packet is error-free, the decoder stops decoding “constrain” its bits. The bits found by CRC detection stage to be correct are constrained on the trellis in the following decoding stages so that the number of possible paths are reduced.

The method suggested in [1] is an efficient method when implemented for single-carrier systems. It reduces the average computational complexity by reducing the number of paths in trellis as the termination process starts to clamp the correctly decoded packets. However, the constrained decoding methods can lead to error propagation if the clamped bits were not correct. By observing the performance of CRC detection techniques, it was noted that there is a chance of erroneously determining that a CRC protected packet to be correct while it is not (the misdetection in CRC decoder). If a CRC misdetection occurs, some bits will be erroneously clamped; consequently, the number of search paths for the decoding process will be limited and potentially lead to decoding errors for the neighboring bits.

Simulations show that error propagation problem is more serious in multicarrier channels than single-carrier channels. This might be due to the fact that every bit (or group of bits) is transmitted over a different channel with different SNRs, and since the bits with low SNR are more likely to be erroneously clamped than bits with high SNR,

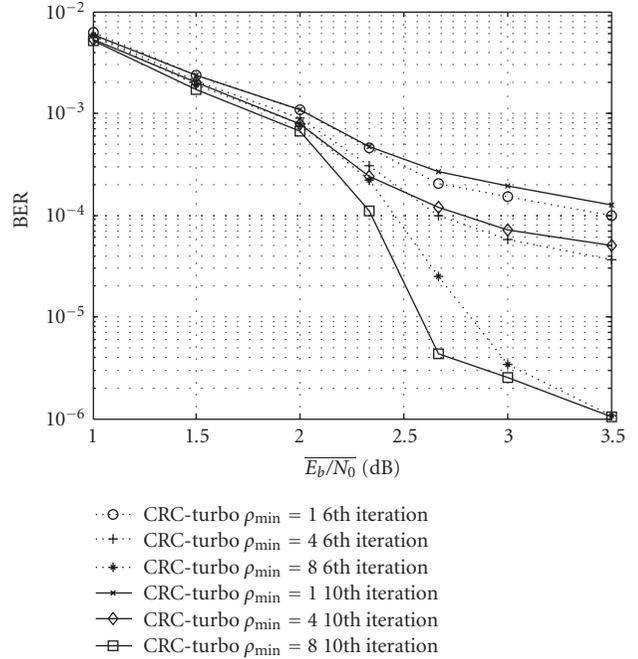


FIGURE 1: BER Performance of CRC-turbo with CM1 channel, 128bit packet (12 bit CRC), start termination after 0.5 iteration, $\rho_{min} = 1, 4, 8$.

the effect of low SNR bits on the performance will dominate over the effect of bits with high SNR, especially for the hard to decode frames. Thus, a good termination method for multicarrier systems (such as UWB OFDM) should take into account the different SNRs for different bits if constrained decoding is implemented.

The error propagation in CRC early stopping techniques can be alleviated by setting the minimum number of CRC detected correct packets (ρ_{min}) required before starting to clamp their bits. Figure 1 shows the BER performance of a turbo decoding in the UWB CM1 channel with CRC-stopping criterion that starts searching for correct packets immediately after the first stage of decoding and starts clamping bits when $\rho_{min} = 1, 4$, and 8, respectively, packets are found to be correct.

4. NEW STOPPING TECHNIQUE

In single-carrier systems, the channel reliability, which can be obtained by measurement of the channel at the receiver input, is given by $L_c = 4R(E_b/N_0)$ which is four times the SNR at this single carrier, where R is the code rate. However, in multicarrier systems such as MB-OFDM, each subcarrier has a different fading coefficient (frequency channel response) and thus will have different SNR value from adjacent subcarriers. Therefore, it is more reasonable to derive the channel reliability for each subcarrier from the SNR of that subcarrier. In our system, we use the following relation to calculate channel reliability values that will be multiplied by the corresponding soft input bits:

$$L_c^i = 4R \frac{E_b}{N_0} |H_i|^2, \quad (4)$$

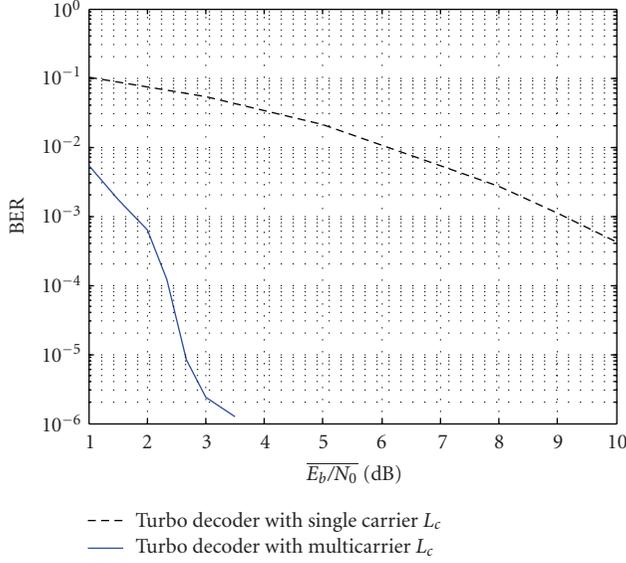


FIGURE 2: BER performance for turbo decoder with single-carrier channel reliability and multicarrier channel reliability obtained from (4).

where the part $(E_b/N_0)|H_i|^2$ represent the instantaneous SNR of the channel, E_b/N_0 is the average SNR for all carriers, and $|H_i|$ is magnitude of the channel frequency response at subcarrier i as shown in (2). As shown in Figure 2, large improvement in BER performance is gained when this relation is used to calculate channel reliability from the CSI values obtained from the channel estimator. In this paper, we show only the results for the turbo decoder using the channel reliability obtained from (4).

In this section, we propose a CRC-based early detection method combined with a bit-level stopping using threshold on LLR. The CRC early detection method serves as an accurate method to determine when all the bits of decoded frame have converged to the correct sequence, and the LLR threshold method works as bit-level stopping for early stages of the decoding process. The bit-level stopping threshold on LLR in our the proposed technique differs from one bit to another and is dynamically updated for each bit from the CSI, unlike the bit-level stopping technique proposed in [2].

For easy-to-decode frames, the LLR output after each decoding stage keeps growing and the hard decision output converges more to the correct sequence. While the change in magnitude of LLR output might be large for easy-to-decode frames, it is small for hard-to-decode frames. The average signal-to-noise ratio has significant effect on the difference between LLR values after each iteration. One way to implement an efficient bit-level early detection is to monitor the magnitudes of the LLR for the decoded bits and clamp the bits that exceed a certain LLR threshold. However, it was noted from simulations that, in general, the hard-decoded outputs and the magnitude of LLR of the bits carried over subcarriers with small SNR are affected by the magnitude of LLR of the neighboring bits with high SNR. Hence, when designing an early detection method that is based on thresholding of LLR, it is crucial to consider

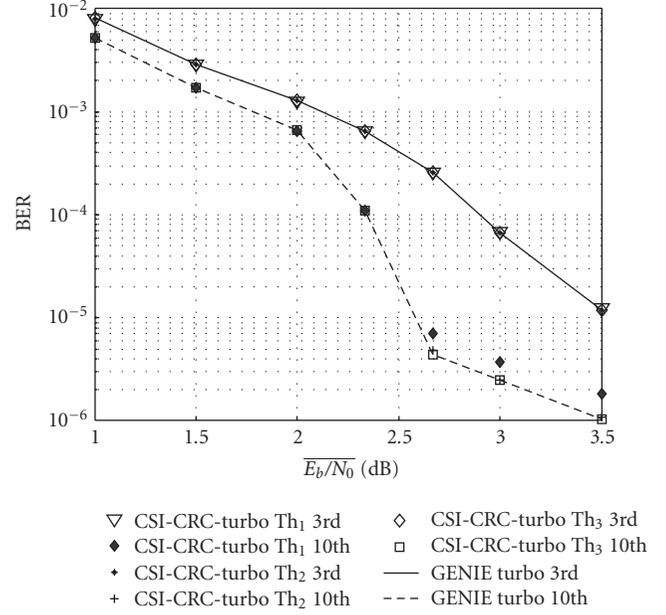


FIGURE 3: BER performance for the proposed method on CM1 channel applying different thresholding strategies: Th_1 , Th_2 , Th_3 displayed together with BER performance of GENIE turbo decoder.

two main factors: the average SNR of the OFDM symbol $(\overline{E_b/N_0})$ and the instantaneous SNR at each subcarrier $(E_b/N_0)_i$. This threshold should increase when the average SNR increases and it should be higher for the bits conveyed on subcarriers with low SNR, so that the bits carried over low SNR subcarriers will not be misdetected. One simple implementation satisfying the above two main factors for this threshold is the following:

$$T_i = L_{\max} \left(1 - \frac{|H_i|^2}{\max_j (|H_j|^2)} \right) + l_{\min}, \quad (5)$$

where L_{\max} is a multiplicative factor and is used to adjust the maximum threshold. H_i is channel state information that is given in (2), and l_{\min} is a constant equal to fixed percent of L_{\max} and is used to adjust the minimum threshold. The value of l_{\min} is chosen experimentally to minimize the loss in BER performance and to maximize the percent of early detected bits. The normalization to the maximum term in the denominator $(\max_j (|H_j|^2))$ can be set to a fixed value to simplify the implementation.

Figure 3 displays BER performance for the turbo decoder with the new stopping technique based on 32 bit CRC detection combined with thresholding on the values of LLR (code rate is $992/2048 \approx 0.484$). The figure shows the results for three arbitrarily selected thresholds by setting $l_{\min} = 0.2L_{\max}$ (experimental value) and choosing different values for the maximum threshold L_{\max} as

- (1) $Th_1: L_{\max} = 1.5 * \overline{E_b/N_0} + 2$,
- (2) $Th_2: L_{\max} = 2.5 * \overline{E_b/N_0} + 2$,
- (3) $Th_3: L_{\max} = 5 * \overline{E_b/N_0} + 5$,

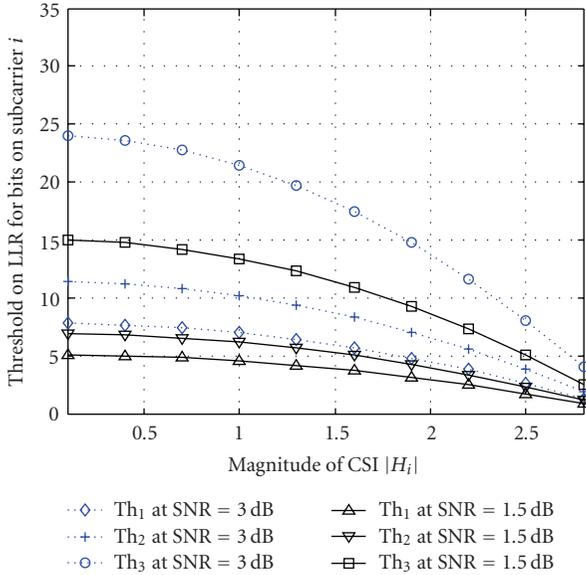


FIGURE 4: The thresholds Th_1 , Th_2 , Th_3 for $\overline{E_b/N_0} = 1.5$ and 3 dB for different subcarriers CSI and $\max(H_i) = 3$ obtained from (5).

where, in this example, we chose linear equations to relate average SNR value with the maximum threshold, however, the maximum threshold for each $\overline{E_b/N_0}$ can be selected differently so that the maximum threshold L_{\max} increases when the average SNR increases. The relation between the value of the CSI for a subcarrier and the corresponding threshold on LLR for the bits carried over this subcarrier, for two values of $\overline{E_b/N_0} = 1.5$ dB and 3 dB, is shown in Figure 4, where the maximum value for CSI is assumed to be 3 dB. The results in Figure 3 show that the BER performance of the CSI-CRC-turbo decoder will have small degradation when the threshold has more tolerance (as the maximum threshold L_{\max} decreases).

At this point, it is important to compare the reduction in computational complexity resulting from the new stopping technique for different thresholding strategies after each decoding stage. Figure 5 displays the average number of bits found to be correct after each decoding stage by the proposed stopping technique applying the three thresholding strategies Th_1 , Th_2 , and Th_3 . In the case where the threshold Th_1 was used, about 47% of the bits on average will be clamped after the first decoding stage (after 0.5 iteration) at $\overline{E_b/N_0} = 3$ dB, and about 89% and 98% of bits on average will be clamped after 1.0 and 1.5 iterations, respectively, which results in a large reduction in computational complexity. Comparing these numbers to the GENIE turbo, at the same SNR, about 2%, 53%, and 88% of bits on average will be decoded by the 0.5, 1.0, 1.5 iterations, respectively. Since there is no bit-level stopping for the case of GENIE turbo (which is the ideal case when the receiver is assumed to know all the received bits sequence and so stops the decoding process when all frame bits are found to be correctly decoded), the curves in Figure 5 for the GENIE case represent the percentage of the total transmitted frames that were completely decoded after each decoding stage. When taking the average percentage of

completely decoded frames after a certain decoding stage, this number is equivalent to the average percentage of frame bits decoded in a bit-level stopping technique after that decoding stage.

Considering this reduction in complexity after each decoding stage, we can compute the effective number of iterations based on the total computations required for each method. For ordinary turbo decoder without stopping techniques applied, the total number of computations required to complete the full N iterations (unconstrained decoding) C_{ud} is expressed by (6), where L is the frame length in bits and C is the total number of computations required to get soft LLR output for single bit in one stage of decoding. For a system applying bit and packet-level early detection, the total number of computations required (constrained decoding) C_{cd} is defined in (7):

$$C_{ud} = 2N \times L \times C, \tag{6}$$

$$C_{cd} = \sum_{i=1}^{2N} \ell_i L \times C, \tag{7}$$

where $0 \leq \ell_i \leq 1$ is the percent of bits decoded (not yet early detected) in the i th decoding stage; it is equal to “0” when all the bits of the frame are considered as correctly decoded, and it is equal to “1” when none of the frame bits are constrained in the trellis decoding. Note that in case $\ell_i = 1$ for all i , then (7) reduces to (6). Equation (7) is used to calculate the average effective number of iterations required to completely decode the bit sequence by dividing it by the total number of computations by the number of computations per decoding stage ($L \times C$) to get the average number of iterations. Figure 6 shows the equivalent average number of iterations for the three thresholds examples shown in Figure 5 compared to the ideal case (GENIE turbo).

From the results for the effective number of iteration in Figure 6 we can infer that the computational complexity of the system with the new stopping technique is reduced by equivalently 2.5 iterations at low SNR and about 0.5 iteration at high SNR for the system with Th_1 thresholding strategy. The main reason why the proposed method is more effective at low SNR, that is, that the stopping technique saves more computations with no degradation in performance, is that the CRC detection technique takes fewer decoding stages to decide that the decoded sequence is correct since there are more errors at low SNR, but the thresholding technique still can detect the bits sent over high SNR subcarriers and clamp these bits, satisfying the LLR condition.

In the following results, we examine the performance of the proposed stopping technique in the other four types of UWB channels presented in [11]. Figure 7 displays the BER performance results for the turbo decoder using CRC detection combined with thresholding on LLR using $L_{\max} = 2 * \overline{E_b/N_0} + 3$ and $l_{\min} = 0.2 * L_{\max}$. The effective number of iterations for these configurations of the stopping technique is shown in Figure 8. Simulation results show that using the new stopping technique gives a BER performance that is very close to the GENIE turbo. While the CRC detection

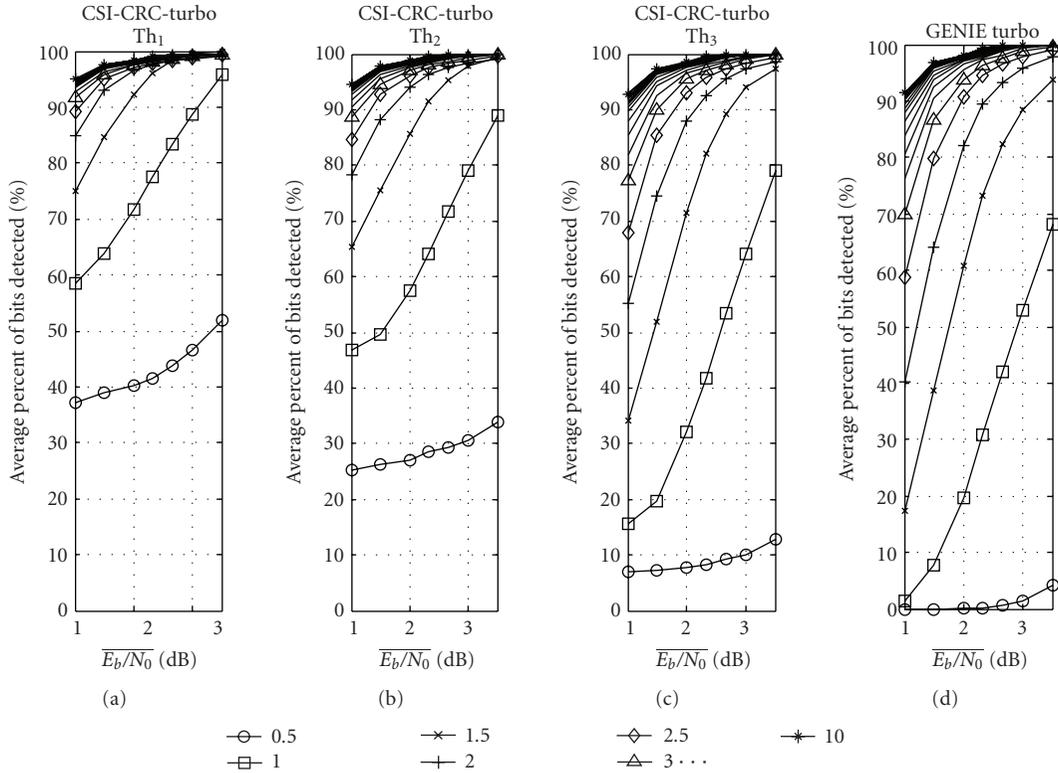


FIGURE 5: Average percent of early detected bits in turbo decoder with CRC detection and thresholding on LLR stopping technique applied for the three thresholds Th_1 , Th_2 , Th_3 , and GENIE.

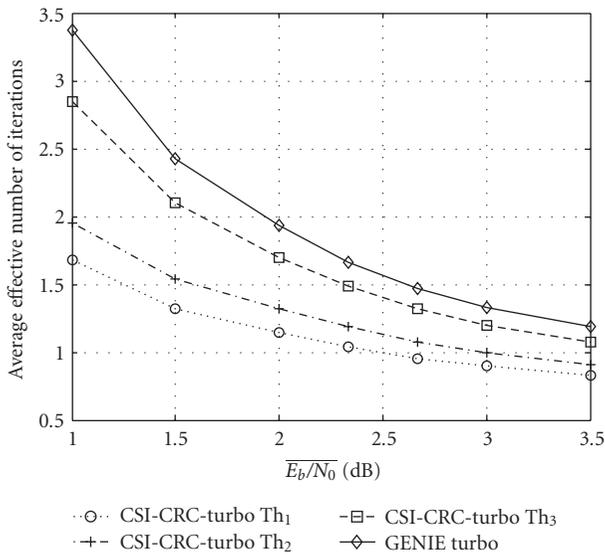


FIGURE 6: Average effective number of iterations for turbo decoder with CRC12 detection combined with thresholding on LLR stopping technique.

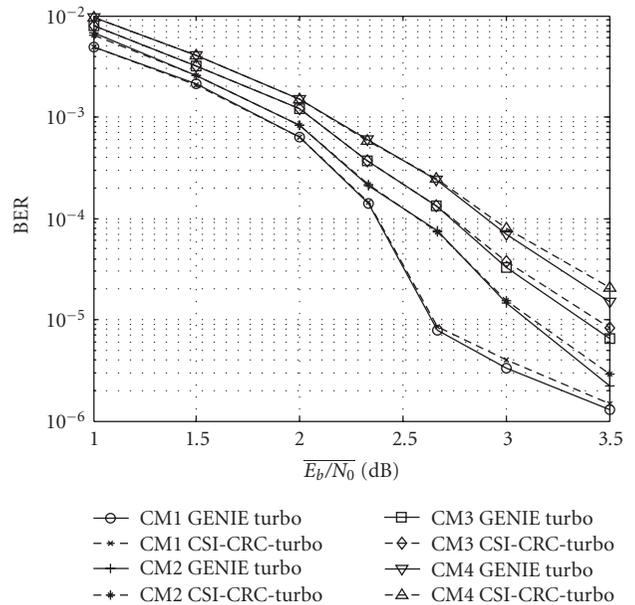


FIGURE 7: The performance of the new stopping technique in the four different types of UWB channels.

gives an accurate method to terminate the decoding process, the dynamic thresholding method helps in gaining large reduction in computational complexity so as to make the turbo codes a good alternative in UWB OFDM system.

5. CONCLUSION

In this paper, we show that the existing early stopping constrained decoding methods for turbo decoder are not

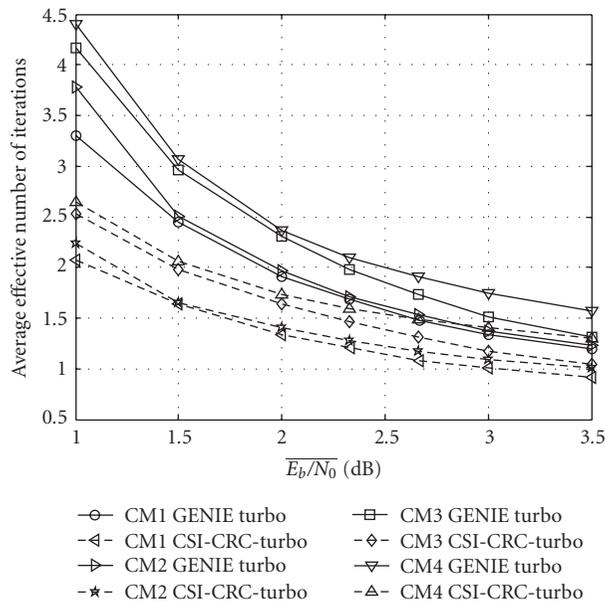


FIGURE 8: The effective number of iterations for the new stopping technique in the four different types of UWB channels used.

appropriate in multicarrier systems with different fading coefficients for each subcarrier since they give equal weights for bits with variable SNR. We present a bit-level stopping technique with adaptive thresholding to reduce the total number of computations and the power consumed by turbo decoder in UWB OFDM systems.

The proposed stopping technique uses CRC detection combined with thresholding technique. It reduces the total computational complexity in the early decoding stages by using a novel technique that compares the LLR outputs with flexible thresholds that are calculated from the CSI values of the UWB channel to take into account the low and high values of SNR for different subcarriers and thus decreasing the average number of iterations. The CRC detection part of this technique is used to determine the accurate stopping point when all the bits of the sequence have converged to the correct sequence.

The results show a large reduction in computational complexity by reducing the average number of iterations to less than one iteration while keeping the BER performance with minimum degradation. This means that if turbo decoding technique is to be implemented in UWB system, only four times or less the decoding complexity of Viterbi decoding is needed to obtain a huge improvement in performance.

NOMENCLATURE

BER: Bit-error rate
 SNR: Signal-to-noise ratio
 BPSK: Binary phase shift keying
 UWB: Ultrawideband
 OFDM: Orthogonal frequency division multiplexing

CRC: Cyclic redundancy check
 LLR: Log likelihood ratio
 CSI: Channel state information
 FFT: Fast Fourier transform
 AWGN: Additive white Gaussian noise
 ICI: Inter-carrier interference
 NLOS: Nonline of sight
 RSC: Recursive systematic convolutional
 ARQ: Automatic repeat request
 PANs: Personal area networks
 LOS: Line of sight.

REFERENCES

- [1] L. Cao, J. Daigle, C. W. Chen, and M. Matalgah, "Complexity reduced turbo decoding with concatenated detection codes," in *Proceedings of the 58th IEEE Vehicular Technology Conference (VTC '03)*, vol. 1, pp. 678–681, Orlando, Fla, USA, October 2003.
- [2] B. J. Frey and F. R. Kschischang, "Early detection and trellis splicing: reduced-complexity iterative decoding," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 153–159, 1998.
- [3] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 429–445, 1996.
- [4] A. Shibutani, H. Suda, and F. Adachi, "Complexity reduction of turbo decoding," in *Proceedings of the 50th IEEE Vehicular Technology Conference (VTC '99)*, vol. 3, pp. 1570–1574, Amsterdam, The Netherlands, 1999.
- [5] H. Chen, L. Cao, and C. W. Chen, "Constrained decoding for turbo-CRC code with high spectral efficient modulation," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC '05)*, vol. 2, pp. 1050–1054, New Orleans, La, USA, March 2005.
- [6] R. Y. Shao, L. Shu, and M. P. C. Fossorier, "Two simple stopping criteria for turbo decoding," *IEEE Transactions on Communications*, vol. 47, no. 8, pp. 1117–1120, 1999.
- [7] E. I. Kalantzis, P. I. Dallas, and B. S. Sharif, "A novel stopping criterion for turbo decoding," in *Proceedings of the 61st IEEE Vehicular Technology Conference (VTC '05)*, vol. 3, pp. 1605–1608, Stockholm, Sweden, May-June 2005.
- [8] B. Kim and H. S. Lee, "Reduction of the number of iterations in turbo decoding using extrinsic information," in *Proceedings of IEEE Region 10 Conference (TENCON '99)*, vol. 1, pp. 494–497, Cheju Island, South Korea, September 1999.
- [9] J. Foerster, "Channel modeling sub-committee report final," IEEE P802.15 Wireless Personal Area Networks, P802.15-02/490r1-SG3a, February 2003.
- [10] A. Saleh and R. Valenzuela, "A statistical model for indoor multipath propagation," *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 2, pp. 128–137, 1987.
- [11] A. F. Molisch, J. R. Foerster, and M. Pendergrass, "Channel models for ultrawideband personal area networks," *IEEE Wireless Communications*, vol. 10, no. 6, pp. 14–21, 2003.

Research Article

System Level Evaluation of Innovative Coded MIMO-OFDM Systems for Broadcasting Digital TV

Y. Nasser, J.-F. H elard, and M. Cruss iere

Institute of Electronics and Telecommunications of Rennes, UMR CNRS, 6164, Rennes, France

Correspondence should be addressed to Y. Nasser, youssef.nasser@ieee.org

Received 28 March 2008; Revised 25 June 2008; Accepted 14 August 2008

Recommended by Marina Mondin

Single-frequency networks (SFNs) for broadcasting digital TV is a topic of theoretical and practical interest for future broadcasting systems. Although progress has been made in the characterization of its description, there are still considerable gaps in its deployment with MIMO technique. The contribution of this paper is multifold. First, we investigate the possibility of applying a space-time (ST) encoder between the antennas of two sites in SFN. Then, we introduce a 3D space-time-space block code for future terrestrial digital TV in SFN architecture. The proposed 3D code is based on a double-layer structure designed for intercell and intracell space time-coded transmissions. Eventually, we propose to adapt a technique called effective exponential signal-to-noise ratio (SNR) mapping (EESM) to predict the bit error rate (BER) at the output of the channel decoder in the MIMO systems. The EESM technique as well as the simulations results will be used to doubly check the efficiency of our 3D code. This efficiency is obtained for equal and unequal received powers whatever is the location of the receiver by adequately combining ST codes. The 3D code is then a very promising candidate for SFN architecture with MIMO transmission.

Copyright   2008 Y. Nasser et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Broadcasting digital TV is currently an area of intensive development and standardisation activities. The terrestrial broadcasting is the most challenging transmission system among the existing radio diffusion systems due to the presence of strong echoes.

Technically, single-frequency networks (SFNs) [1] present great advantages by transmitting lower power at various sites throughout the coverage area. In an SFN, the different antennas transmit the same signal at the same moment on the same carrier frequency. The existing SFN architectures are achieved in a single-input single-output (SISO) system since their deployment is very simple due to the use of one transmitting antenna by site. However, due to the increase of client services demand, it is desirable to deploy SFN with new MIMO techniques which ensure high spectrum efficiency as well as high diversity gain. The MIMO technique combined with the orthogonal frequency division multiplexing (OFDM) technique is pursued as a potential candidate for future generations of terrestrial portable and mobile digital video broadcasting related to DVB-T2 and

DVB-NGH proposals. Actually, one of the main research topics concerns the optimisation of the MIMO-OFDM schemes in order to obtain high-spectrum efficiency for high definition television (HDTV) services. In the literature, there are few studies on the SFN with MIMO transmission. The authors of [2] propose a new SFN model to increase the diversity gain in MIMO SFN architecture. In [3], an array antenna receiver using a maximum ratio combining technique is proposed to improve the system performance of the SFN transmission. This lack of studies on this original idea motivates our work to extend the application of the MIMO-OFDM transmission to the SFN architecture.

The optimisation of the MIMO-OFDM schemes in the SFN is highly desirable to be led in terms of the bit error rate (BER) after channel decoding. However, the optimisation of the MIMO-OFDM systems by simulations is time consuming. Thus, it is very important to accurately abstract the system level BER performance into analytical expression. Moreover, the system level performance abstraction should take into account the different transmission conditions, that is, modulation and coding scheme (MCS), synchronization errors, channel fading, and so forth.

This paper presents a complete study on the optimisation of the MIMO-OFDM schemes for SFN architectures. The optimisation is double checked analytically and by simulations. This work has been carried out within the framework of a new European CELTIC project called *Broadcast for 21st Century* (B21C) project [4] and constitutes an extension of some previous works [5–7]. In this paper, we propose a 3D MIMO-OFDM scheme taking advantage of the particular characteristics of an SFN. More precisely, the contribution of this work is multifold. First of all, we investigate the possibility of applying a space-time block code (STBC) encoder between the antennas of two sites in SFN architecture. Secondly, using an iterative receiver, a generalized framework is proposed for modelling the effect of unbalanced powers received from different transmitting antennas in MIMO-OFDM systems. This is a critical problem in SFN with mobile and portable reception. Another contribution of this work is the proposal of a new 3D space-time-space (STS) block code for SFN environment. The use of a second space dimension in the STS code will be justified as being particularly adapted and efficient in the case of SFN transmission. The proposed code is based on a double-level construction of ST coding resulting from the combination of two coding schemes: the intercell ST coding and the intracell ST coding. Eventually, we propose in this paper to adapt a technique, initially used for OFDM systems, to predict the BER at the output of the channel decoders of the MIMO-OFDM systems. This technique, called exponential effective SNR mapping (EESM) [7], is empiric but has been validated within the 3GPP project for the OFDM study item [8]. It provides a reliable link between link level simulations and system level simulations. In our contribution, we show that the EESM technique is independent of the MIMO scheme and of the power imbalance at the receiving side. It depends only on the modulation and coding scheme (MCS).

This paper is structured as follows. Section 3 describes the architecture of an SFN in which MIMO techniques are used. In Section 4, we present the transmission system model. Section 5 presents the receiving model with iterative receiver. In Section 6, we discuss the construction of different STBC schemes considered in this paper and we describe our proposed 3D code for MIMO-OFDM transmission. In Section 7, we adapt an accurate abstraction of the system level BER performance, initially proposed for the OFDM systems, to the MIMO-OFDM systems. In Section 8, we double check the efficiency of the proposed 3D code using the BER abstraction method described in Section 7. Conclusions are drawn in Section 8.

2. MIMO SYSTEMS IN SFN

In this paper, we propose to apply a MIMO communication scheme between the antennas located in the different sites of an SFN architecture. Such a system could be implemented using M_T transmit antennas (Tx) by site as shown in Figure 1. Without loss of generality, we will consider in our study the transmission behaviour of two neighbouring cells using a total of $(2 \times M_T)$ Tx and M_R receive antennas (Rx).

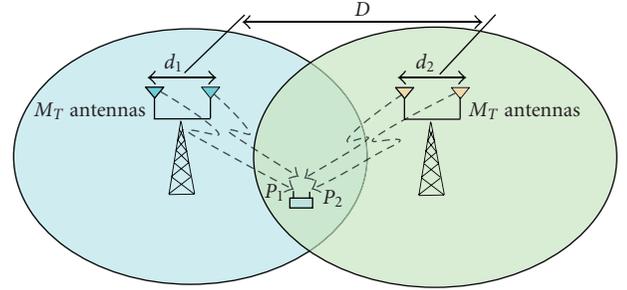


FIGURE 1: SFN with unequal received powers.

The extension of our study to more sites could be adequately adapted.

Classically, in SFN architectures, the different antennas transmit at the same moment the same signal on the same frequency. For the SFN to work properly, the resulting delay spread τ_{\max} of the different received signals must be less than the duration of the guard interval (GI) time inserted at the beginning of each OFDM symbol.

As a starting point, let us assume that each site holds one antenna and that the receiver receives signals from both antennas. In the case of an SFN, the time offset between the signals received from each site antennas could be seen as a superposition of the time offset between transmitters' signals (the signal time delay between the transmitting antennas) and the signal time offset between each transmitter and the receiver. The first offset is generally negligible since the transmitters are synchronized with an ultrastable reference like the global positioning system (GPS). The second offset could be seen as follows. When the mobile terminal (MT) moves within one cell, it receives signal from its own cell antenna but also from the neighbouring cell antenna. Since the MT is not equidistant to both antennas, the signal received from each one will be delayed according to the position of the MT. This results into a delay $\Delta\tau$ between the two signals received from both antennas or equivalently between the channel impulse responses (CIRs) between the transmitters and the receiver. The delays are directly related to the distances between the transmitters and the receiver and thus to the signal strength ratio at the receiver. Assuming an equal transmitted power P_0 at each antenna, the received power from the i th antenna is

$$P_i = \frac{P_0}{d_i^\alpha}, \quad (1)$$

where d_i is the distance between the receiver and the i th transmitter and α is the propagation constant which depends on the transmission environment.

The delay of each CIR between the i th transmitter and the receiver is

$$\tau_i = \frac{d_i}{c}, \quad (2)$$

where c is the light velocity.

Without loss of generality, let us assume that the first transmitter site is the reference site. Substituting d_i from (2)

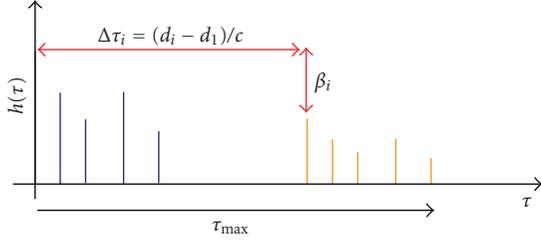


FIGURE 2: SFN with unequal received powers.

in (1), the CIR delay of the i th link (i.e., between the i th transmitter and the receiver) with respect to the reference antenna can be expressed by

$$\Delta\tau_i = \tau_i - \tau_1 = (10^{-\beta_i/10\alpha} - 1) \frac{d_1}{c}, \quad (3)$$

where d_1 is the distance between the reference transmitter (first one) and the receiver; and β_i is the received power difference (expressed in dB) between the signal received from the reference site and the signal received from the i th transmitter. It is given by

$$\beta_i[\text{dB}] = -10 \cdot \alpha \cdot \log_{10} \left(\frac{d_i}{d_1} \right). \quad (4)$$

In the sequel, we will assume that the power received from the reference antenna is equal to 0 dB and the distance d_i is greater than d_1 whatever $i > 1$. It is a real situation where the MT is closer to its own cell antenna than to the other antennas. In this case, β_i is neither than the power attenuation factor between the i th transmitter and the MT. As a consequence, the transmission model becomes equivalent to a system with unbalanced powers received from each site antennas. Figure 2 shows an example of the relation between the power attenuation factor β and the CIR delay of the i th link with respect to the reference antenna signal.

If we now consider that the number of Tx in one site is greater than one (i.e., $M_T > 1$), the choice of an adequate MIMO scheme should then be based on this imbalance. Moreover, it should be adequate for intercell environment, (i.e., between antennas signals of each site) and intracell environment (i.e., between antennas signals in each site). Furthermore, it should be chosen adequately to cope with equal and unequal received powers. This will be the subject of Section 6 where we propose a 3D STS code adapted to such situations.

We note that in this paper we consider independent CIRs with the dominant problem of the SFN architecture, that is, the problem of the CIR delays and the power loss. However, in real situations, other problems like CIRs correlations should be considered also. The reader may refer to [9] and the references therein for more details.

3. TRANSMISSION MODEL

In this section, we describe the transmission model of the double-layer STBC constructed between the antennas of

the different sites. The double layer proposed here has to cope with the equal and unequal received powers. The first layer in our proposed code corresponds to the intercell ST coding while the second corresponds to the intracell ST coding.

Figure 3 depicts the transmitter modules at each site. Information bits b_k are first channel encoded, randomly interleaved, and fed to a quadrature amplitude modulation (QAM) module. We recall that we restrict our study to two sites only and the generalisation could be done in different forms. Therefore, the SFN transmission system involving the two sites (described in Figure 1) could be seen as a double-layer scheme in the space domain. The first layer is seen between the 2 sites separated by D km. The second layer is seen between the antennas separated by d meters within one site. For the first layer, a space time block code (STBC) scheme is applied between the two signals transmitted by each site antennas. In the second layer, we use a second STBC encoder for each subset of M_T signals transmitted from the same site. For the first layer (resp., the second layer), the STBC encoder takes L (resp., M) sets of data complex symbols and transforms them into a $(2, U)$ (resp., (M_T, V)) output matrix according to the STBC scheme. This output is then fed to $2 \times M_T$ OFDM modulators, each using N_c subcarriers. In order to have a fair analysis and comparison between different STBC codes, the signal power at the output of the ST encoder is normalized by $2 \times M_T$.

The double-layer encoding matrix of the proposed code is described by

$$\mathbf{X}^{(1)} = \begin{pmatrix} \mathbf{X}_{11}^{(2)} & \dots & \mathbf{X}_{1U}^{(2)} \\ \mathbf{X}_{21}^{(2)} & \dots & \mathbf{X}_{2U}^{(2)} \end{pmatrix},$$

$$\mathbf{X}_{pq}^{(2)} = \begin{pmatrix} f_{pq,11}(s_1, \dots, s_M) & \dots & f_{pq,1V}(s_1, \dots, s_M) \\ \vdots & \ddots & \vdots \\ f_{pq,M_T1}(s_1, \dots, s_M) & \dots & f_{pq,M_TV}(s_1, \dots, s_M) \end{pmatrix}. \quad (5)$$

In (5), the superscript indicates the layer, $f_{pq,it}(s_1, \dots, s_M)$ is a function of the input complex symbols s_m and depends on the STBC encoder scheme. The subscripts p and q are such that $p = 1, 2$ and $q = 1, \dots, U$. They reflect, respectively, the STBC encoder input size at each layer. The time dimension of the resulting 3D code is equal to $U \times V$ and the resulting coding rate is $R = (L \times M)/(U \times V)$.

In order to simplify the transmission model, the double-layer encoding matrix given in (5) will be represented by $\mathbf{X} = [x_{i,t}]$, where $x_{i,t}$ ($i = 1, \dots, 2 \times M_T$; $t = 1, \dots, U \times V$) is the output of the double-layer STBC encoder on a given subcarrier n . In other words, the layers construction is transparent from the transmission model viewpoint. Moreover, we set $Q = L \times M$ as the number of the complex symbols at the input of the double-layer STBC encoder and we set $T = U \times V$ as the number of the corresponding output symbols. The ST coding rate is then $R = Q/T$.

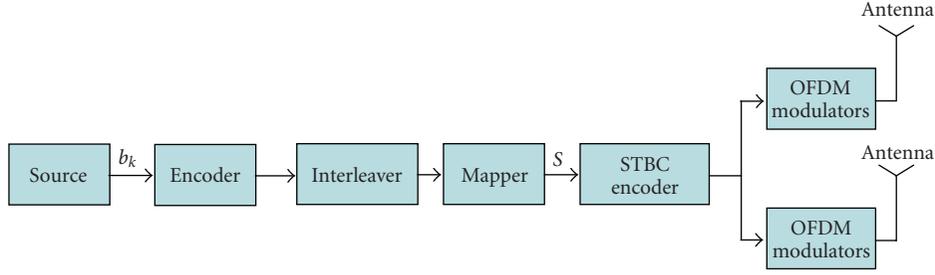


FIGURE 3: MIMO-OFDM transmitter.

4. ITERATIVE STBC RECEIVER

4.1. Receiving model

We assume that the transmitter and the receiver are perfectly synchronised. Moreover, we assume perfect channel state information (CSI) at the receiver. In this paper, the transmission is described in frequency domain for simplicity reasons. However, in real scenario, the signal is transferred to the time domain and cyclic prefix (CP) insertion operations are achieved at the transmitting side. Reciprocal operations are done at the receiving side. The signal received on the subcarrier n by the antenna j is a superposition of the transmitted signal by the different antennas multiplied by the channel coefficients $h_{j,i}[n]$ (“ i ” is the index of the transmitting antenna) to which additive white Gaussian noise (AWGN) is added. It is given by

$$y_{j,t}[n] = \sum_{i=1}^{2M_T} \sqrt{P_i} h_{j,i}[n] x_{i,t}[n] + w_{j,t}[n], \quad (6)$$

where $y_{j,t}[n]$ is the signal received on the n th subcarrier by the j th receiving antenna during the t th OFDM symbol period. $h_{j,i}[n]$ is the frequency channel coefficient assumed to be constant during T symbol durations, $x_{i,t}[n]$ is the signal transmitted by the i th antenna, and $w_{j,t}[n]$ is the additive AWGN with zero mean and variance $N_0/2$. In the sequel, we will drop the subcarrier index n for simplicity. By introducing an equivalent receive matrix $\mathbf{Y} \in \mathbb{C}^{M_R \times T}$ whose elements are the complex received symbols expressed in (6), we can write the received signal on the n th subcarrier on all receiving antennas as

$$\mathbf{Y} = \mathbf{H}\mathbf{P}\mathbf{X} + \mathbf{W}, \quad (7)$$

where \mathbf{H} is the $(M_R, 2M_T)$ channel matrix whose components are the coefficients $h_{j,i}$, \mathbf{P} is a $(2M_T, 2M_T)$ diagonal matrix containing the signal magnitudes $\sqrt{P_i}$, \mathbf{X} is a $(2M_T, T)$ complex matrix containing the transmitted symbols $x_i[t]$. \mathbf{W} is a (M_R, T) complex matrix corresponding to the AWGN.

Let us now describe the transmission link with a general model independently of the ST coding scheme. We separate the real and imaginary parts of the complex symbols input vector $\mathbf{s} \{s_q : q = 1, \dots, Q\}$ of the outputs \mathbf{X} of the double-layer ST encoder as well as those of the channel matrix \mathbf{H} , and the received signal \mathbf{Y} . Let $s_{q,\mathcal{R}}$ and $s_{q,\mathcal{I}}$ be the real

and imaginary parts of s_q . The main parameters of the double code are given by its dispersion matrices \mathbf{U}_q and \mathbf{V}_q corresponding (not equal) to the real and imaginary parts of \mathbf{X} , respectively. With these notations, \mathbf{X} is given by

$$\mathbf{X} = \sum_{q=1}^Q (s_{q,\mathcal{R}} \mathbf{U}_q + j s_{q,\mathcal{I}} \mathbf{V}_q), \quad (8)$$

where \mathbf{U}_q is the dispersion matrix having the same dimensions of \mathbf{X} such that

$$\mathbf{U}_q(m, n) = \begin{cases} 1, & \text{if } \mathbf{X}(m, n) = s_{q,\mathcal{R}}, \\ -1, & \text{if } \mathbf{X}(m, n) = -s_{q,\mathcal{R}}, \\ 0, & \text{elsewhere,} \end{cases} \quad (9)$$

where \mathbf{V}_q could be deduced from (9) by replacing the real part by the imaginary part.

In the sequel, we separate the real and imaginary parts of \mathbf{S} , \mathbf{Y} , and \mathbf{X} , and stack them row-wise in vectors of dimensions $(2Q, 1)$, $(2M_R T, 1)$, and $(4M_T T, 1)$, respectively. We obtain

$$\begin{aligned} \mathbf{s} &= [s_{1,\mathcal{R}}, s_{1,\mathcal{I}}, \dots, s_{Q,\mathcal{R}}, s_{Q,\mathcal{I}}]^T, \\ \mathbf{y} &= [y_{1,\mathcal{R}}, y_{1,\mathcal{I}}, \dots, y_{T,\mathcal{R}}, y_{T,\mathcal{I}}, \dots, y_{M_R T,\mathcal{R}}, y_{M_R T,\mathcal{I}}]^T, \\ \mathbf{x} &= [x_{(1,1),\mathcal{R}}, x_{(1,1),\mathcal{I}}, \dots, x_{(2M_T,T),\mathcal{R}}, x_{(2M_T,T),\mathcal{I}}]^T, \end{aligned} \quad (10)$$

where $[\cdot]^T$ holds for matrix transpose.

Since we use linear ST coding, the vector \mathbf{x} can be written as

$$\mathbf{x} = \mathbf{F} \cdot \mathbf{s}, \quad (11)$$

where \mathbf{F} has the dimensions $(4M_T T, 2Q)$ and is obtained through the dispersion matrices of the real and imaginary parts of \mathbf{s} . It is given by

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_1(1, 1) & \cdots & \cdots & \mathbf{F}_Q(1, 1) \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{F}_1(1, T) & \cdots & \cdots & \mathbf{F}_Q(1, T) \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{F}_1(2M_T, T) & \cdots & \cdots & \mathbf{F}_Q(2M_T, T) \end{bmatrix}, \quad (12)$$

where \mathbf{F} is composed of $2M_T$ blocks of $2T$ rows each, that is, the data transmitted on each antenna is gathered in one block having $2T$ rows and $2Q$ columns according to the ST coding scheme. The different components of \mathbf{F} are given by

$$\mathbf{F}_q(m, t) = \begin{bmatrix} \mathbf{U}_{q,\Re}(m, t) & -\mathbf{V}_{q,\Im}(m, t) \\ \mathbf{U}_{q,\Im}(m, t) & \mathbf{V}_{q,\Re}(m, t) \end{bmatrix}. \quad (13)$$

As we change the formulation of \mathbf{S} , \mathbf{Y} , and \mathbf{X} in (10), it can be shown that vectors \mathbf{x} and \mathbf{y} are related through the matrix

$$\mathbf{G}_{j,i} = \begin{pmatrix} h_{(j,i),\Re} & -h_{(j,i),\Im} & 0 & \cdots & 0 \\ h_{(j,i),\Im} & h_{(j,i),\Re} & 0 & \cdots & 0 \\ 0 & 0 & h_{(j,i),\Re} & -h_{(j,i),\Im} & 0 & \cdots & 0 \\ 0 & 0 & h_{(j,i),\Im} & h_{(j,i),\Re} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \ddots & 0 & 0 & 0 \\ 0 & \cdots & 0 & \ddots & 0 & 0 & 0 \\ 0 & \cdots & 0 & \ddots & 0 & h_{(j,i),\Re} & -h_{(j,i),\Im} \\ 0 & \cdots & 0 & \ddots & 0 & h_{(j,i),\Im} & h_{(j,i),\Re} \end{pmatrix}_{(2T,2T)}. \quad (16)$$

Now, substituting \mathbf{x} from (11) in (14), the relation between \mathbf{y} and \mathbf{s} becomes

$$\mathbf{y} = \mathbf{G}\mathbf{B}\mathbf{F}\mathbf{s} + \mathbf{w} = \mathbf{G}_{\text{eq}}\mathbf{s} + \mathbf{w}, \quad (17)$$

where \mathbf{G}_{eq} is the equivalent channel matrix between \mathbf{s} and \mathbf{y} . It is assumed to be known perfectly at the receiving side.

4.2. STBC detector

The detection problem is to find the transmitted data \mathbf{s} given the vector \mathbf{y} . In the case of orthogonal STBC (OSTBC), the optimal receiver is made of a concatenation of ST decoder and channel decoder modules. In nonorthogonal STBC (NO-STBC) schemes, there is an interelement interference (IEI) at the receiving side. The optimal receiver in this case is based on joint ST and channel-decoding operations. However, such receiver is extremely complex to implement and requires large memory to store the different points of the trellis. Moreover, it could not be implemented reasonably in one chip. Thus, the suboptimal solution proposed here consists of an iterative receiver where the ST detector and channel decoder exchange extrinsic information in an iterative way until the algorithm converges. The iterative detector shown in Figure 4 is composed of a parallel interference canceller (PIC), a demapper which consists in computing the soft information of the transmitted bits, that is, a log likelihood ratio (LLR) computation [10], a soft-input soft-output (SISO) decoder [11], and a soft mapper.

At the first iteration, the demapper takes the estimated symbols $\hat{\mathbf{s}}$, the knowledge of the channel \mathbf{G}_{eq} , and of the noise variance, and computes the LLR values of each of the coded bits transmitted per channel use. The estimated symbols $\hat{\mathbf{s}}$ are

\mathbf{G} of dimensions $(2M_R T, 4M_T T)$ such that

$$\mathbf{y} = \mathbf{G}\mathbf{B}\mathbf{x} + \mathbf{w}. \quad (14)$$

The matrix \mathbf{B} is a $(4M_T T, 4M_T T)$ diagonal matrix whose components are given by

$$B_{i,i} = \sqrt{P_i} \quad 2 \cdot T(p-1) + 1 \leq i \leq 2T \cdot p, \quad (15)$$

$$p = 1, \dots, 2M_T.$$

Matrix \mathbf{G} is composed of blocks $\mathbf{G}_{j,i}$ ($j = 1, \dots, M_R$; $i = 1, \dots, 2M_T$) each having $(2T, 2T)$ elements given by

obtained via minimum mean square error (MMSE) filtering according to

$$\hat{\mathbf{s}}_p^{(1)} = \mathbf{g}_p^{\text{tr}} (\mathbf{G}_{\text{eq}} \cdot \mathbf{G}_{\text{eq}}^{\text{tr}} + \sigma_w^2 \mathbf{I})^{-1} \mathbf{y}, \quad (18)$$

where \mathbf{g}_p^{tr} of dimension $(2M_R T, 1)$ is the p th column of \mathbf{G}_{eq} ($1 \leq p \leq 2Q$). $\hat{\mathbf{s}}_p^{(1)}$ is the estimation of the real part (p odd) or imaginary part (p even) of s_q ($1 \leq q \leq Q$). Once the estimation of the different symbols s_q is achieved by the soft mapper at the first iteration, we use this estimation for the next iterations process.

From the second iteration, we perform PIC operation followed by a simple inverse filtering (instead of MMSE filtering at the first iteration):

$$\hat{\mathbf{y}}_p = \mathbf{y} - \mathbf{G}_{\text{eq},p} \tilde{\mathbf{s}}_p^{(1)}, \quad \hat{\mathbf{s}}_p^{(2)} = \frac{1}{\mathbf{g}_p^{\text{tr}} \mathbf{g}_p} \mathbf{g}_p^{\text{tr}} \hat{\mathbf{y}}_p, \quad (19)$$

where $\mathbf{G}_{\text{eq},p}$ of dimension $(2M_R T, 2Q - 1)$ is the matrix \mathbf{G}_{eq} with its p th column removed, $\tilde{\mathbf{s}}_p^{(1)}$ of dimension $(2Q - 1, 1)$ is the vector $\tilde{\mathbf{s}}$ estimated by the soft mapper with its p th entry removed.

5. 3D STSBC CONSTRUCTION

The aim of this section is to judiciously build the proposed double-layer 3D STS code so that the resulting MIMO scheme behaves efficiently in an SFN context. We then need to choose the adequate ST coding scheme to apply to each layer of our 3D code. In the sequel, we will consider different coding schemes to apply to the different layers. First, we will consider the well-known orthogonal Alamouti ST coding scheme [12] for its robustness and its simplicity. In

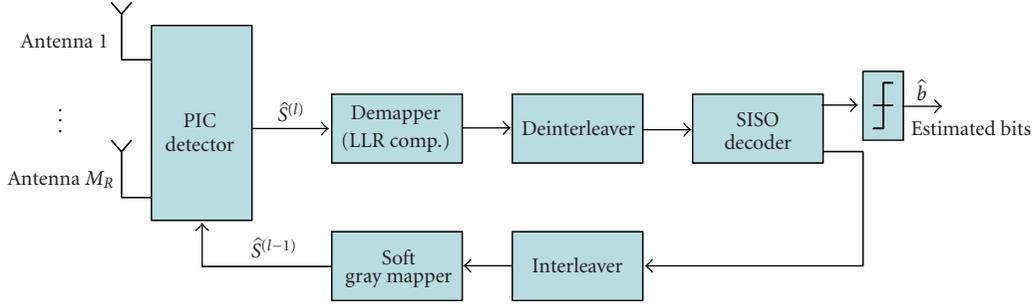


FIGURE 4: Iterative receiver structure.

this case, the maximum likelihood (ML) receiver is simply implemented. This code is described by its dispersion matrix given by

$$\mathbf{X} = \begin{bmatrix} s_1 & s_2 \\ -s_2^* & s_1^* \end{bmatrix}. \quad (20)$$

For NO schemes, we consider in this work the well-known space multiplexing (SM) scheme [13]. SM is designed to maximize the rate by transmitting symbols sequentially on different antennas. Its coding scheme is given by

$$\mathbf{X} = [s_1 \ s_2]^{\text{tr}}. \quad (21)$$

Finally, we consider the full rate and the fully diverse Golden code [14]. The Golden code is designed to maximize the rate such that the diversity gain is preserved for an increased signal constellation size. It is defined by

$$\mathbf{X} = \frac{1}{\sqrt{5}} \begin{bmatrix} \alpha(s_1 + \theta s_2) & \alpha(s_3 + \theta s_4) \\ \mu \bar{\alpha}(s_3 + \bar{\theta} s_4) & \bar{\alpha}(s_1 + \bar{\theta} s_2) \end{bmatrix}, \quad (22)$$

where $\theta = (1 + \sqrt{5})/2$, $\bar{\theta} = 1 - \theta$, $\alpha = 1 + j(1 - \theta)$, $\bar{\alpha} = 1 + j(1 - \bar{\theta})$. To identify the most efficient ST code, the OFDM parameters are derived from those of a DVB-T system (see Table 1). Moreover, we have considered the possibility to extend the size of the constellation size up to 256-QAM. The spectral efficiencies 4 and 6 b/s/Hz are obtained for different ST schemes as shown in Table 2. In all simulations, we assume that two Rx are used by the MT.

In the simulations results given hereafter, we separate the single-layer case and the double-layer case. For NO schemes, we show in [7] that the receiver converges after 3 iterations. This implies an acceptable complexity as compared to the ML detection. This can be observed with Golden code, but also with SM scheme. That is, for NO-STBC schemes, we will present in the sequel the performances after 3 iterations only.

5.1. Single-layer case: inter-cell ST coding

In the case of single-layer reception, we have one antenna by site. Then, the second-layer matrix $\mathbf{X}^{(2)}$ in (5) resumes to one element. The multiple-input component of the MIMO scheme is then only obtained by the single antenna in

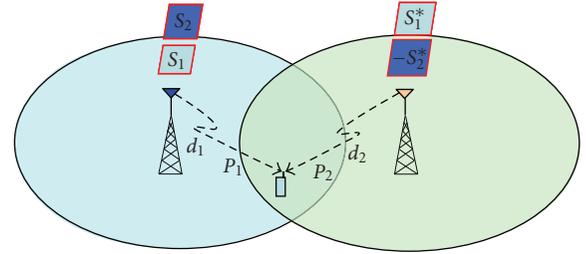
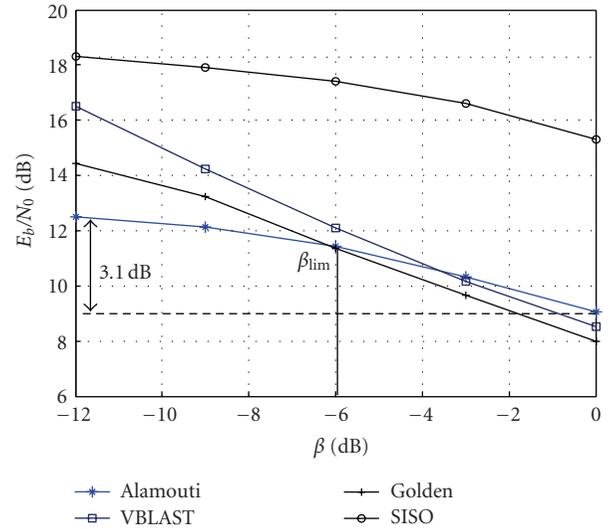


FIGURE 5: Alamouti scheme in SFN environment.

FIGURE 6: Required E_b/N_0 to obtain a BER = 10^{-4} , single-layer case, $\eta = 4$ b/s/Hz.

each site. Due to the mobility, the MT is assumed to occupy different locations and the first-layer ST scheme must be efficient face to unequal received powers. For equal received powers, we assume that the powers of matrix \mathbf{B} in (14) are equal to 0 dB. Figure 5 presents a simple case of MIMO transmission in SFN environment using Alamouti scheme. As it is shown in this figure, the STBC symbols are transmitted through the set of one antenna in each site using Alamouti coding.

TABLE 1: Simulations parameters.

FFT size	8 K
Sampling frequency ($f_s = 1/T_s$)	9.14 MHz
Guard interval (GI) duration	$1024 \times T_s = 112 \mu\text{s}$
Rate R_c of convolutional code	1/2, 2/3, 3/4
Polynomial code generator	(133, 171) _o
Channel estimation	perfect
Constellation	16-QAM, 64-QAM, 256-QAM
Spectral Efficiencies	$\eta = 4$ and 6 b/s/Hz

Figure 6 shows the required E_b/N_0 to obtain a BER equal to 10^{-4} for a spectral efficiency $\eta = 4$ b/s/Hz with a Rayleigh channel model. Since we have one Tx antenna by site, we set $\beta_1 = 0$ dB and we change $\beta = \beta_2$. As expected, this figure shows that the Golden code presents the best performance when the Rx receives the same power from both sites (i.e., $\beta_1 = \beta_2 = 0$ dB). When β_2 decreases, Alamouti scheme is very efficient and presents a maximum loss of only 3 dB in terms of required E_b/N_0 with respect to equal received powers case. Indeed, for very small values of β ,

$$X = 1/\sqrt{5} \begin{pmatrix} \alpha(s_1 + \theta s_2) & \alpha(s_3 + \theta s_4) & \alpha(s_5 + \theta s_6) & \alpha(s_7 + \theta s_8) \\ j\bar{\alpha}(s_3 + \bar{\theta} s_4) & \bar{\alpha}(s_1 + \bar{\theta} s_2) & j\bar{\alpha}(s_7 + \bar{\theta} s_8) & \bar{\alpha}(s_5 + \bar{\theta} s_6) \\ -\alpha^*(s_1^* + \theta^* s_2^*) & -\alpha^*(s_3^* + \theta^* s_4^*) & \alpha^*(s_1^* + \theta^* s_2^*) & \alpha^*(s_3^* + \theta^* s_4^*) \\ j\bar{\alpha}^*(s_7^* + \bar{\theta}^* s_8^*) & -\bar{\alpha}^*(s_5^* + \bar{\theta}^* s_6^*) & -j\bar{\alpha}^*(s_7^* + \bar{\theta}^* s_8^*) & \bar{\alpha}^*(s_5^* + \bar{\theta}^* s_6^*) \end{pmatrix}, \quad (23)$$

where $\theta = (1 + \sqrt{5})/2$, $\bar{\theta} = 1 - \theta$, $\alpha = 1 + j(1 - \theta)$, $\bar{\alpha} = 1 + j(1 - \bar{\theta})$.

Since the distance d between the transmitting antennas in one site is negligible with respect to the distance D (Figure 1), the power attenuation factors in the case of our 3D code are such that $\beta_1 = \beta_2 = 0$ dB and $\beta = \beta_3 = \beta_4$. Figure 7 presents an overview of the proposed 3D STSBC. Since the received powers from each antenna in the same site are equal, we apply the Golden code between the two signals transmitted in a given site. However, we apply the Alamouti code between the signals transmitted by the different sites' antennas.

Figure 8 shows the results in terms of required E_b/N_0 to obtain a BER equal to 10^{-4} for different values of β and 3 STBC schemes, that is, our proposed 3D code scheme, the single-layer Alamouti scheme, and the single-layer Golden scheme. The results obtained in this figure assume that the transmission is achieved through the COST 207 TU-6 channel model [15]. The value β in this figure corresponds to β_2 for the single-layer case and to $\beta = \beta_3 = \beta_4$ for our 3D code. We assume that the MT is moving with a velocity of 10 km/h and the distance d_1 of the reference antenna is equal to 5 km. The CIRs between different transmitters and the MT are delayed according to (3). Figure 8 shows that the proposed scheme presents the best performance whatever the spectral efficiency and the factor β are. Indeed, it is optimized for SFN systems and unbalanced received powers. For $\beta = -12$ dB, the proposed 3D code offers a gain equal to 1.5 dB (resp., 3.1 dB) with respect to the Alamouti scheme

the transmission scenario becomes equivalent to a scenario with one transmitting antenna. In this figure, the value $\beta_{\text{lim}} = -6.2$ dB presents the power imbalance limit where the Alamouti and the Golden code schemes have the same performance at a BER = 10^{-4} . It is also straightforward to note that the SISO transmission in existing SFN presents the worst results when it is compared to the distributed MIMO technique.

5.2. Double-layer case

Considering the whole double-layer space domain construction, one ST coding scheme has to be assigned to each layer of the proposed system. The resulting 3D STS code should be efficient for both environments in SFN architectures. In this paper, we restrict our study to $M_T = 2$ Tx by site. We propose to construct the first layer with Alamouti scheme, since it is the most resistant for the unequal received powers case. In a complementary way, we propose to construct the second layer with the Golden code since it offers the best results in the case of equal received powers. After combination of the two space layers with time dimension, (5) yields

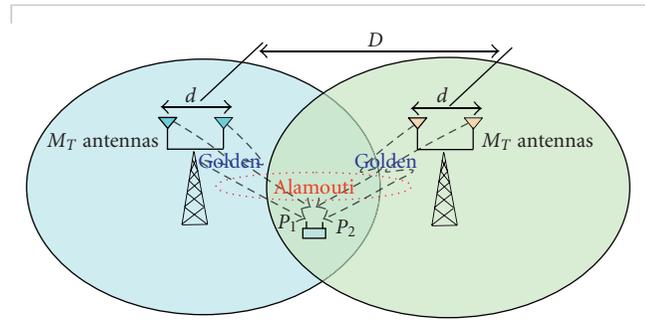


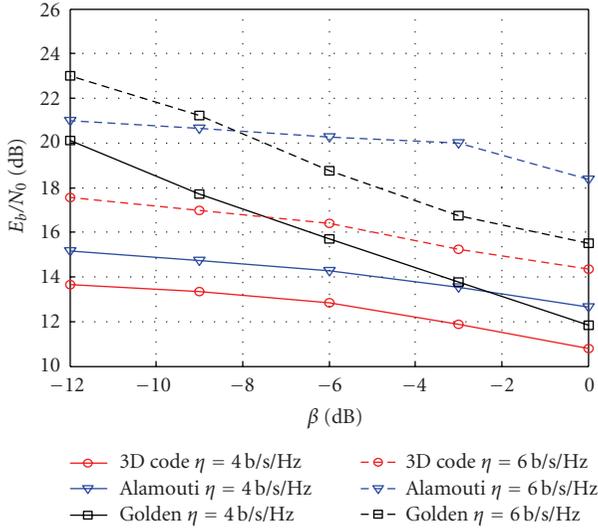
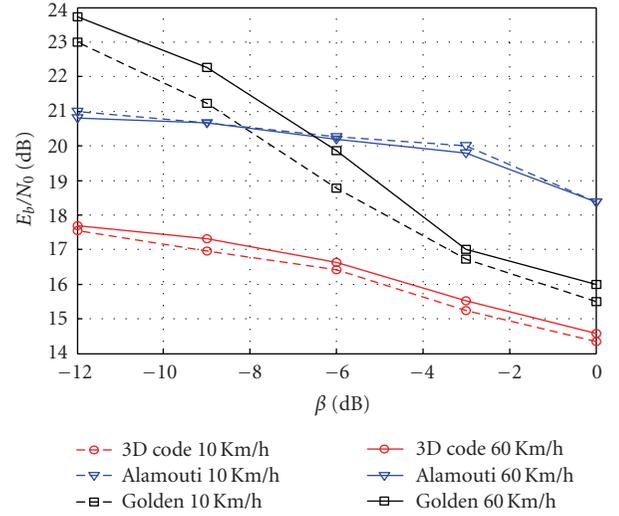
FIGURE 7: 3D STS scheme in SFN environment.

for a spectral efficiency $\eta = 4$ b/s/Hz (resp. $\eta = 6$ b/s/Hz). This gain is greater when it is compared to the Golden code. The maximum loss of our code due to unbalanced received powers is equal to 3 dB in terms of E_b/N_0 . This means that it leads to a powerful code for SFN systems.

Figure 9 evaluates the robustness of the different schemes to the two values of MT velocity, that is, 10 km/h and 60 km/h. We show in this figure that the Alamouti scheme is very robust to the MT velocity. The degradation of the Golden code might reach 1 dB in terms of required E_b/N_0 at a BER = 10^{-4} . Our 3D code presents an intermediate behaviour. Its degradation due to the MT velocity, that is, to Doppler effect is about 0.2 dB only.

TABLE 2: Different MIMO schemes and efficiencies.

Spectral efficiency	ST scheme	ST rate R	Constellation	R_c
$\eta = 4$ [bit/Sec/Hz]	Alamouti	1	64-QAM	2/3
	SM	2	16-QAM	1/2
	Golden	2	16-QAM	1/2
	3D code	2	16-QAM	1/2
$\eta = 6$ [bit/Sec/Hz]	Alamouti	1	256-QAM	3/4
	SM	2	64-QAM	1/2
	Golden	2	64-QAM	1/2
	3D code	2	64-QAM	1/2

FIGURE 8: Required E_b/N_0 to obtain a BER = 10^{-4} , double-layer case, $\eta = 4$ b/s/Hz, $\eta = 6$ b/s/Hz, TU-6 channel.FIGURE 9: Required E_b/N_0 to obtain a BER = 10^{-4} , $\eta = 6$ b/s/Hz, TU-6 channel, different values of MT velocity.

5.3. Complexity considerations

As we have shown, the 3D code outperforms the other MIMO schemes in different reception scenarios. Let us now compare the different MIMO schemes in terms of complexity implementation. At this stage, different complexity points could be evaluated. First, at the transmission side, the implementation of the Alamouti and the Golden code schemes between different sites in SFN architecture does not increase the complexity when it is compared to that of the SISO case. Indeed, we just need to synchronize the transmission from both sites as it should be already done with SFN in the SISO case. This task can be ensured by an ultrastable reference like the GPS. However, for the 3D code, an additional front-end RF should be used at each site. At the receiving side, the iterative receiver used for NO schemes like the SM scheme or the Golden code is the same of that used for the 3D code. Moreover, when compared to the ML detection, we had showed in [7] that the iterative receiver converges after 3 iterations only. However, the ML detection complexity increases with the MIMO architecture size and the modulation order [16]. Thus, the proposed 3D code complexity is of the same order of the NO codes complexity.

6. ANALYTICAL SYSTEM LEVEL EVALUATION

In the previous section, we have proposed a new 3D STSBC for MIMO-OFDM systems in SFN architecture. Using the system level simulations, we have showed that this new ST code is very efficient to cope with equal and unequal received powers. However, explicit bit level simulation of each MT in every cell of the SFN would be forbidding time consuming. The problem becomes more noticeable when MIMO-OFDM techniques are used in SFN architectures. As a consequence, it is desirable to evaluate the system level performance in terms of BER without achieving system simulation. Thus, the practical need of an accurate abstraction of the system level simulation into analytical evaluation highly motivates our work to achieve an analytical BER expression of the MIMO-OFDM systems using iterative receiver.

In some studies, it has been shown that the BER at the output of the channel decoder is directly related to the SINR at the output of the detector [8, 17, 18]. In the OFDM system, [8] proposes a new technique called effective exponential SINR mapping (EESM) to evaluate the BER. The technique is based on the computation of an effective SINR derived from the different values of the estimated SINRs on each

subcarrier. The authors of [19] propose a new method to adapt the EESM technique to the SM transmission and an ML receiver. Their method, however, could not be suited for our work since an iterative receiver is used and, as it will be shown, the SNR expression is not computed for each layer.

In this paper, we propose to adapt the EESM technique to the MIMO-OFDM systems using the iterative receiver. The first step in our work consists in computing the SINRs expressions at the output of the detector. The second step is to establish an accurate relationship between the different SINRs and the coded BER through the adapted EESM technique. We note that the SINR expressions and the predicted BER given hereafter are not specified for a given antenna, that is, we do not separate between the different antennas like in [19]. However, it is possible to apply our methods for each antenna received signal.

6.1. SINR evaluation

Without loss of generality, we assume in the sequel that we are interested by the p th symbol. Using the vector-matrix notation of Sections 4 and 5, the estimated received symbol at the first iteration in (18) could be written in an equivalent form as

$$\begin{aligned}\hat{s}_p &= I_0^{(1)} + I_1^{(1)} + I_2^{(1)}, \\ I_0^{(1)} &= \mathbf{g}_p^{\text{tr}} (\mathbf{G}_{\text{eq}} \cdot \mathbf{G}_{\text{eq}}^{\text{tr}} + \sigma_w^2 \mathbf{I})^{-1} \mathbf{g}_p s_p, \\ I_1^{(1)} &= \sum_{\substack{q=1 \\ q \neq p}}^{2Q} \mathbf{g}_p^{\text{tr}} (\mathbf{G}_{\text{eq}} \cdot \mathbf{G}_{\text{eq}}^{\text{tr}} + \sigma_w^2 \mathbf{I})^{-1} \mathbf{g}_q s_q, \\ I_2^{(1)} &= \mathbf{g}_p^{\text{tr}} (\mathbf{G}_{\text{eq}} \cdot \mathbf{G}_{\text{eq}}^{\text{tr}} + \sigma_w^2 \mathbf{I})^{-1} \mathbf{w}.\end{aligned}\quad (24)$$

In (24), $I_0^{(1)}$ is the useful received signal, $I_1^{(1)}$ is the IEI due to the nonorthogonality of the considered STBC. We can easily verify that it is equal to zero for O-STBC schemes. $I_2^{(1)}$ is the coloured noise. The superscript ℓ in the signal expressions $I^{(\ell)}$ indicates the number of iteration in the iterative process.

The complex transmitted data symbols are assumed i.i.d. having zero mean and unit variance (the variance of the real and imaginary parts is equal to 1/2). Due to this distribution, the SINR expression can be deduced from (24) by

$$\text{SINR} = \frac{E\{|I_0^{(1)}|^2\}}{E\{|I_1^{(1)}|^2\} + E\{|I_2^{(1)}|^2\}}. \quad (25)$$

The expectations values in (25) over the random data symbols are given by

$$\begin{aligned}E\{|I_0^{(1)}|^2\} &= \frac{1}{2} |\mathbf{g}_p^{\text{tr}} (\mathbf{G}_{\text{eq}} \cdot \mathbf{G}_{\text{eq}}^{\text{tr}} + \sigma_w^2 \mathbf{I})^{-1} \mathbf{g}_p|^2, \\ E\{|I_1^{(1)}|^2\} &= \frac{1}{2} \sum_{\substack{q=1 \\ q \neq p}}^{2Q} |\mathbf{g}_p^{\text{tr}} (\mathbf{G}_{\text{eq}} \cdot \mathbf{G}_{\text{eq}}^{\text{tr}} + \sigma_w^2 \mathbf{I})^{-1} \mathbf{g}_q|^2, \\ E\{|I_2^{(1)}|^2\} &= \frac{N_0}{2} |\mathbf{g}_p^{\text{tr}} (\mathbf{G}_{\text{eq}} \cdot \mathbf{G}_{\text{eq}}^{\text{tr}} + \sigma_w^2 \mathbf{I})^{-1}|^2.\end{aligned}\quad (26)$$

At the second iteration, the estimated symbol expressed in (24) becomes more complex. It is obtained using (17) and (18) in (19) by

$$\hat{s}_p^{(2)} = I_0^{(2)} + I_1^{(2)} + I_2^{(2)}, \quad (27)$$

where

$$\begin{aligned}I_0^{(2)} &= s_p, \\ I_1^{(2)} &= \sum_{\substack{q=1 \\ q \neq p}}^{2Q} \frac{1}{\mathbf{g}_p^{\text{tr}} \mathbf{g}_p} \mathbf{g}_p^{\text{tr}} [\mathbf{g}_q s_q - (\mathbf{G}_{\text{eq}} \cdot \mathbf{G}_{\text{eq}}^{\text{tr}} + \sigma_w^2 \mathbf{I})^{-1} \mathbf{g}_q \hat{s}_q^{(1)}], \\ I_2^{(2)} &= \frac{1}{\mathbf{g}_p^{\text{tr}} \mathbf{g}_p} \mathbf{g}_p^{\text{tr}} \mathbf{w}.\end{aligned}\quad (28)$$

For next iterations, it is clear from (27) that the expressions of the estimated received symbol as well as the estimated SINR become more complex. Therefore, some manipulations should be considered to give an analytical expression of the SINR.

Based on the structure of the iterative receiver, we already know that the outputs of the soft Gray mapper are complex symbols which belong to the set of constellation points. Let $I_t^{(2)} = I_1^{(2)} + I_2^{(2)} = s_p^{(2)} - I_0^{(2)}$ be the total interference power at the second iteration. Then, two cases can be presented at this stage.

(I) If the estimated symbol $\hat{s}_p^{(1)}$ at the output of the Gray mapper is equal to the transmitted symbol s_p , the useful signal $I_0^{(2)}$ in (28) is such that $I_0^{(2)} = s_p = \hat{s}_p^{(1)}$ and the total interference signal at the second iteration becomes

$$I_t^{(2)} = I_1^{(2)} + I_2^{(2)} = s_p^{(2)} - s_p^{(1)}. \quad (29)$$

Since $I_1^{(2)}$ and $I_2^{(2)}$ are independent and the complex outputs of the Gray mapper are zero mean with unit variance, the estimated SINR at the second iteration is

$$\text{SINR}^{(2)} = \frac{E\{|I_0^{(2)}|^2\}}{E\{|I_1^{(2)} + I_2^{(2)}|^2\}} = \frac{1}{2 \cdot E\{|\hat{s}_p^{(2)} - \hat{s}_p^{(1)}|^2\}}, \quad (30)$$

where $\hat{s}_p^{(2)}$ is estimated through (19) and $\hat{s}_p^{(1)}$ is the output of the soft Gray mapper at the first iteration.

(II) If the estimated symbol $\hat{s}_p^{(1)}$ at the first iteration is different from the transmitted symbol s_p , the difference between the received signals at the first two successive iterations yields by substituting $I_0^{(2)}$ from (28) in (27):

$$\hat{s}_p^{(2)} - \hat{s}_p^{(1)} = s_p + I_1^{(2)} + I_2^{(2)} - \hat{s}_p^{(1)}. \quad (31)$$

Since $\hat{s}_p^{(1)}$ is different from s_p in this case, and the different transmitted symbols are i.i.d., we can verify due to the expectation operation that

$$\begin{aligned} \text{SINR}^{(2)} &= \frac{E\{|I_0^{(2)}|^2\}}{E\{|I_1^{(2)} + I_2^{(2)}|^2\}} \\ &= \frac{E\{|I_0^{(2)}|^2\}}{E\{|I_1^{(2)} + I_2^{(2)} + s_p - \hat{s}_p^{(1)}|^2\}} \\ &= \frac{E\{|I_0|^2\}}{E\{|\hat{s}_p^{(2)} - \hat{s}_p^{(1)}|^2\}} \\ &= \frac{1}{2 \cdot E\{|\hat{s}_p^{(2)} - \hat{s}_p^{(1)}|^2\}}. \end{aligned} \quad (32)$$

It is clear from the last term of (30) and (32) that the SINR expression at the second iteration is simpler than that of (28). In this case, only the estimated symbols at each iteration are used for SINR estimation, that is, we do not have to compute complex expressions. Also, we can show that (30) and (32) could be generalized for successive iterations. In the next section, we will exploit our theoretical SINR model through BER measurements at the output of the channel decoder.

6.2. BER evaluation with EESM technique

In order to evaluate the BER at the output of the channel decoder, we propose in this section to adapt the EESM technique to the MIMO-OFDM context. At the first step, we will develop analytically the EESM technique. At the second step, we will present its application in the OFDM system. Then, we will adapt it to the MIMO-OFDM context using the SINR expressions computed at the previous subsection.

Let J denote the packet size in complex data symbols. In general, the data symbols in the packets are transmitted over different resource elements (e.g., subcarriers) and, therefore, they may experience different propagation and interference conditions. Thus, the data symbols may have different SINR values. Let SINR be the vector of J instantaneous SINR received at the output of the detector. The problem of determining an accurate BER prediction method comes back to looking for a relationship such that

$$P_e = f(\text{SINR}), \quad (33)$$

where P_e denotes the bit error probability (BEP) and f is the prediction function, which should be invariant with respect to the fading realization and to the multipath channel model, and should be applicable to different MCSs in a soft way, that is, by changing the values of some generic parameters [18]. In the context of AWGN channel, the SINR becomes SNR and it remains constant over the packet. In this context, a direct relationship ξ exists between the SNR and the error probability:

$$P_{e,WGN} = \xi(\text{SNR}). \quad (34)$$

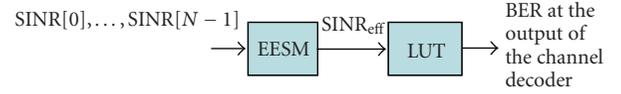


FIGURE 10: BER prediction through EESM.

The function ξ is called the mapping function. It is obtained through theoretical analysis or system level simulation with AWGN channel. In the general context of a fading channel, where the SINR varies, the function f in (33) can be written exactly as a compound function of the AWGN function ξ and a compression function r [8]:

$$\begin{aligned} P_e &= \xi \circ r(\text{SINR}) = \xi(\text{SINR}_{\text{eff}}) \\ &\text{with } \text{SINR}_{\text{eff}} = r(\text{SINR}). \end{aligned} \quad (35)$$

The function r is referred to as the compression function since its role is to compress the vector SINR of J components into one scalar SINR_{eff} . The scalar SINR_{eff} is called the *effective SINR* and it is defined as the SINR which would yield the same error probability in an equivalent AWGN channel as the associated vector SINR in a fading channel. By writing (35), we have merely turned the problem of determining the evaluation function f into the problem of determining the compression function r .

In an OFDM system, it was concluded that the key issue to accurately determine the appropriate BER after channel decoding is to use the effective SINR in combination with AWGN curves. The work in [8] proposes the EESM technique which is based on the Chernoff Union bound [18] to find the effective SINR. The key EESM technique expression relevant to an OFDM system is given by

$$\text{SINR}_{\text{eff}} = -\lambda \ln \left(\frac{1}{N} \sum_{n=0}^{N-1} \exp \left(-\frac{\text{SINR}[n]}{\lambda} \right) \right). \quad (36)$$

$\text{SINR}[n]$ is the SINR obtained over the n th sub-carrier and λ is a unique parameter which must be estimated from the system level simulations for each MCS. It is estimated once by preliminary simulation for each MCS. When the SINR_{eff} is computed, it will be used for BER prediction at the output of the channel decoder with a simple lookup table (LUT) as shown in Figure 10. This LUT gives the BER at the output of the channel decoder as a function of the SNR for a Gaussian channel. It is computed analytically or by simulations. The uniqueness of λ for each MCS is derived from the fact that the effective SINR must fulfil the approximate relation

$$\text{BER}(\text{SINR}) = \text{BER}_{\text{AWGN}}(\text{SINR}_{\text{eff}}), \quad (37)$$

where BER_{AWGN} is the BEP for the AWGN channel which depends only on the MCSs.

In our study, the EESM technique must be adapted to the MIMO-OFDM system. Indeed, the estimated received symbol at each subcarrier is a superposition of different symbols transmitted by the different antennas on that subcarrier. Therefore, the EESM technique will be applied on the set of Q symbols transmitted on the M_T antennas

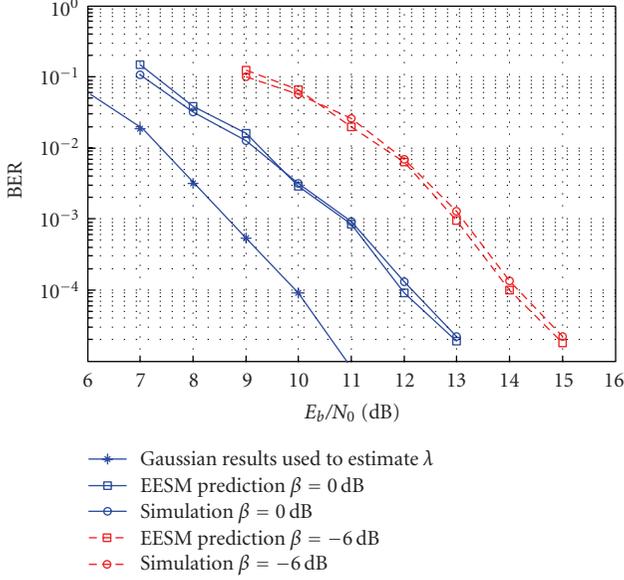


FIGURE 11: Validation of EESM technique, Alamouti scheme, $\eta = 6$ b/s/Hz, $\lambda = 22.6$, TU-6 channel.

during T OFDM symbols. The effective SINR is, therefore, computed through

$$\text{SINR}_{\text{eff}} = -\lambda \ln \left(\frac{1}{NQ} \sum_{q=1}^Q \sum_{n=1}^N \exp \left(-\frac{\text{SINR}_q[n]}{\lambda} \right) \right). \quad (38)$$

Using the effective SINR of (38), we are now able to evaluate the BER using the LUT as shown in Figure 10.

7. APPLICATION OF THE EESM TECHNIQUE TO THE PROPOSED 3D STS CODE

In this section, we validate through the EESM technique and the SINR analysis the efficiency of the proposed 3D STS code. The considered simulation parameters are the same of those given in Table 1. The parameter λ is estimated using the AWGN channel model. Its estimation is done as follows. For a given channel model and a given MCS, we estimate the different SINRs at the output of the detector. Therefore, we select a value of λ and we compute the BER at the output of the channel decoder according to (38) and Figure 10. The accurate value of λ is that which ensures minimum error between the predicted and the simulated BERs. Once the value of λ is decided, it does not change with the channel or the MIMO scheme. We will show by simulations that this value depends only on the spectral efficiency.

The results given in this section are obtained with the COST 207 TU-6 channel model. The AWGN results used to estimate the parameter λ are obtained using Alamouti scheme since NO schemes are not efficient with AWGN channel. They will be plotted in the same figure of those obtained by EESM technique or by simulations, with TU-6 channel. First of all, we will validate our SINR analysis and EESM technique on the Alamouti and the Golden code schemes. Then, it will be suited by the validation of the

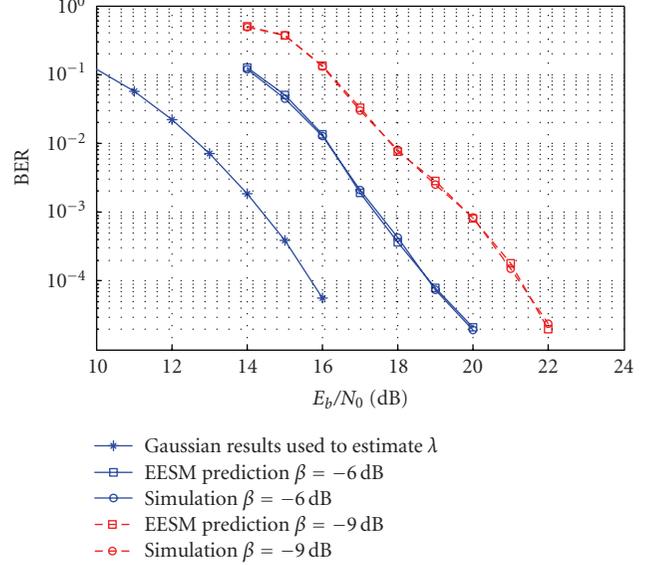


FIGURE 12: Validation of EESM technique, Golden code scheme, $\eta = 6$ b/s/Hz, $\lambda = 22.6$, TU-6 channel.

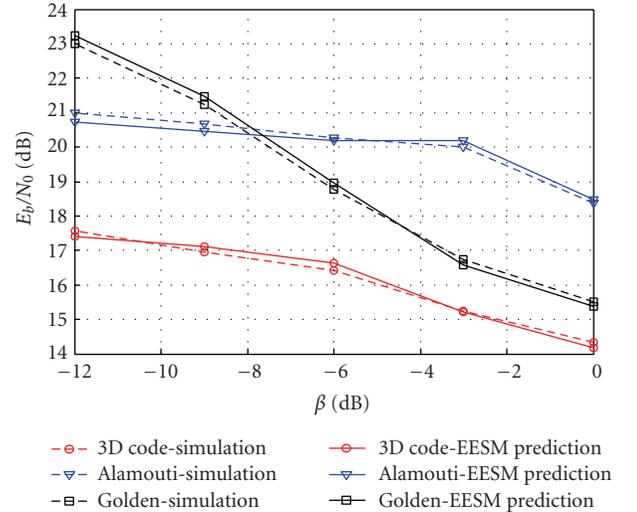


FIGURE 13: Validation of EESM technique, $\eta = 6$ b/s/Hz, $\lambda = 22.6$, TU-6 channel.

efficiency of our proposed code. In all figures, the results are given with mobile velocity equal to 10 km/h.

Figure 11 compares the BER obtained by simulations and the BER obtained with the EESM technique for the Alamouti scheme, considering a spectral efficiency $\eta = 6$ b/s/Hz and different values of transmitted powers. These figures show the accuracy of the proposed technique based on the SINR analytical expression. Moreover, they show that the parameter λ is constant ($\lambda = 22.6$ for $\eta = 6$) and it is independent of the power imbalance but depends on the MCSs or equivalently on the spectral efficiency. The parameter λ is obtained by simulations. It is computed once for a given MCS.

Figure 12 compares the BER obtained by simulations and the BER evaluated with the EESM technique for the Golden code scheme, a spectral efficiency $\eta = 6$ b/s/Hz and different values of transmitted powers. Since, compared with the parameters used in Figure 11, the spectral efficiency does not change, the parameter $\lambda = 22.6$ gives an accurate BER prediction and validates our analytical expressions and prediction method. Again, we show that this parameter is independent of the power imbalance.

In Figure 13, we give the same kind of results of those given in Figure 8 by using the EESM technique. Figure 13 validates our proposed prediction method for all considered STBC schemes in the SFN architecture. Again, it shows that parameter λ is independent of the power imbalance and of the STBC scheme. Moreover, it shows again the superiority of the proposed 3D STSBC whatever the power imbalance factor β .

8. CONCLUSION

In this paper, a new 3D STSBC has been presented for MIMO transmission in SFN architecture including two transmitting sites. The proposed 3D STSBC is based on a double-layer structure defined for intercell and intracell situations by adequately combining the Alamouti code and the Golden code schemes. We showed that our proposed 3D STS scheme is very efficient to cope with equal and unequal received powers in SFN scenarios whatever the receiver position is.

Moreover, we have proposed an analytical SINR evaluation of MIMO-OFDM systems using an iterative receiver as well as an adaptation of the EESM technique to efficiently evaluate the BER at the output of the channel decoder. Using the EESM technique and the analytical evaluation, we have showed again the superiority of the proposed 3D code. It is then a very promising candidate for the broadcasting of the future terrestrial digital TV in SFN architectures.

ACKNOWLEDGMENT

The authors would like to thank the European CELTIC Project "B21C" for its support to this work.

REFERENCES

- [1] A. Mattsson, "Single frequency networks in DTV," *IEEE Transactions on Broadcasting*, vol. 51, no. 4, pp. 413–422, 2005.
- [2] L. Zhang, L. Gui, Y. Qiao, and W. Zhang, "Obtaining diversity gain for DTV by using MIMO structure in SFN," *IEEE Transactions on Broadcasting*, vol. 50, no. 1, pp. 83–90, 2004.
- [3] Y. Kanbe, M. Itami, K. Itoh, and A. Hamid Aghvami, "Reception of OFDM signals with an array antenna in a SFN environment," in *Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '02)*, vol. 3, pp. 1310–1315, Lisbon, Portugal, September 2002.
- [4] <http://www.celtic-initiative.org/Projects/B21C>.
- [5] M.-A. Khalighi, J.-F. Hélar, and S. Bourennane, "Contrasting orthogonal and non-orthogonal space-time schemes for perfectly-known and estimated MIMO channels," in *Proceedings of the 10th IEEE Singapore International Conference on Communications Systems (ICCS '06)*, Singapore, October 2006.
- [6] Y. Nasser, J.-F. Hélar, and M. Crussiere, "Robustness of MIMO-OFDM schemes for future digital TV to carrier frequency offset," in *Proceedings of IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB '08)*, pp. 1–4, Las Vegas, Nev, USA, March-April 2008.
- [7] Y. Nasser, J.-F. Hélar, M. Crussiere, and O. Pasquero, "Efficient MIMO-OFDM schemes for future terrestrial digital TV with unequal received powers," in *Proceedings of IEEE International Conference on Communications (ICC '08)*, pp. 2021–2027, Beijing, China, May 2008.
- [8] 3GPP TSG-RAN-1, "TR 25.892: feasibility study for OFDM for UTRAN enhancement," version 1.1.0, March 2004.
- [9] T. A. Lamahewa, R. A. Kennedy, T. D. Abhayapala, and T. Betlehem, "MIMO channel correlation in general scattering environments," in *Proceedings of the 7th Australian Communications Theory Workshop (AUSCTW '06)*, pp. 93–98, Perth, Australia, February 2006.
- [10] F. Tosato and P. Bisaglia, "Simplified soft-output demapper for binary interleaved COFDM with application to HIPER-LAN/2," in *Proceedings of IEEE International Conference on Communications (ICC '02)*, vol. 2, pp. 664–668, New York, NY, USA, April-May 2002.
- [11] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," in *Proceedings of IEEE Global Telecommunications Conference and Exhibition (GLOBECOM '89)*, vol. 3, pp. 1680–1686, Dallas, Tex, USA, November 1989.
- [12] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 8, pp. 1451–1458, 1998.
- [13] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs Technical Journal*, vol. 1, no. 2, pp. 41–59, 1996.
- [14] J.-C. Belfiore, G. Rekaya, and E. Viterbo, "The golden code: a 2×2 full-rate space-time code with nonvanishing determinants," *IEEE Transactions on Information Theory*, vol. 51, no. 4, pp. 1432–1436, 2005.
- [15] COST 207 Report, "Digital land mobile radio communications, Commission of European Communities, Directorate General, Telecommunications Information Industries and Innovation," Luxemburg, 1989.
- [16] M. Rupp, G. Gritsch, and H. Weinrichter, "Approximate ML detection for MIMO systems with very low complexity," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04)*, vol. 4, pp. 809–812, Montreal, Canada, May 2004.
- [17] R. Ratasuk, A. Ghosh, and B. Classon, "Quasi-static method for predicting link-level performance," in *Proceedings of the 55th IEEE Vehicular Technology Conference (VTC '02)*, vol. 3, pp. 1298–1302, Birmingham, Ala, USA, May 2002.
- [18] Ericsson, "System level evaluation of OFDM- further considerations," TSG-RAN WG1 #35, November 2003, R1-031303, Lisbon, Portugal.
- [19] H. Liu, L. Cia, H. Yang, and D. Li, "EESM based link error prediction for adaptive MIMO-OFDM system".

Research Article

Stationary Interference Cancellation in Upstream Coordinated DSL Using a Turbo-MMSE Receiver

Issam Wahibi,^{1,2} Meryem Ouzzif,¹ Jérôme Le Masson,¹ and Samir Saoudi²

¹Orange Laboratories, 2 avenue Pierre Marzin, 22307 Lannion, France

²Département Signal et Communications, INSTITUT Telecom, TELECOM Bretagne, Technopôle Brest-Iroise, CS 83818, 29238 Brest Cedex 3, France

Correspondence should be addressed to Meryem Ouzzif, meryem.ouzzif@francetelecom.com

Received 1 April 2008; Accepted 19 July 2008

Recommended by Marina Mondin

We investigate the scenario of an upstream coordinated DSL transmission in presence of spatial-correlated noise. Joint signal processing helps mitigate this noise and reduce internal interference effects between coordinated users. We propose to couple noise whitening with a mean-squared error iterative receiver in order to approach the matched filter bound of the DSL coordinated system. The convergence of the iterative scheme in this scenario is predicted using EXIT charts under realistic transmission conditions.

Copyright © 2008 Issam Wahibi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Faced with the mounting need of residential and professional customers to benefit from new high data-rate multimedia services, access network operators are improving their offers thanks to diverse high-speed technologies. Very high speed digital subscriber line (VDSL2) allows tens of megabits per second to be sent to customers over copper lines for short distances. This technique, associated with an optical termination situated close to the customer premises, can supply this need while saving on infrastructure costs. In this configuration, far-end crosstalk (FEXT) coupling between copper lines can represent a major performance limitation.

Crosstalk, in the case of a coordinated DSL transmission, can be separated into two categories. The first, called *in-domain crosstalk*, comes from the vectored system whereas the second category of crosstalk, called *out-of-domain crosstalk*, originates from outside the coordinated system when two or more access network operators share the same binder [1]. When the DSL systems transmit continuously, the crosstalk noise can be considered stationary. Several techniques have been proposed for in-domain crosstalk cancellation in the upstream link. In the literature, Ginis and Cioffi use a zero forcing-successive interference cancellation (ZF-SIC) structure for FEXT cancellation [2], while Cen-

drillon et al. show that a linear zero-forcing (ZF) canceller achieves near optimal performance [3].

Out-of-domain crosstalk or equivalently external interference presents a spatial correlation on the receiver side. A whitening filter based on the Cholesky decomposition is applied to mitigate its impact and is followed by a successive interference cancellation (SIC) structure to help reduce the inherent interference of the equivalent channel [1]. A power allocation method that mitigates the external crosstalk effect in vectored DSL systems has been proposed in [4].

In the presence of strong external interference, the previously presented interference vectored cancellation schemes perform far from the optimum. In this paper, we propose to carry out after the whitening filter an iterative processing based on the minimum mean-squared error (MMSE) criterion using a priori information [5–7]. The iterative detection of multiple-input multiple-output (MIMO) signals using the MMSE detector is presented in [8].

The remainder of the paper is organized as follows. The system model is described in Section 2. Maximum throughput upper bound for coordinated DSL systems in presence of external interference is derived in Section 3. The ZF-SIC and the linear iterative MMSE detectors are examined in Section 4. Numerical results are analyzed in Section 5 and conclusions are drawn in Section 6.

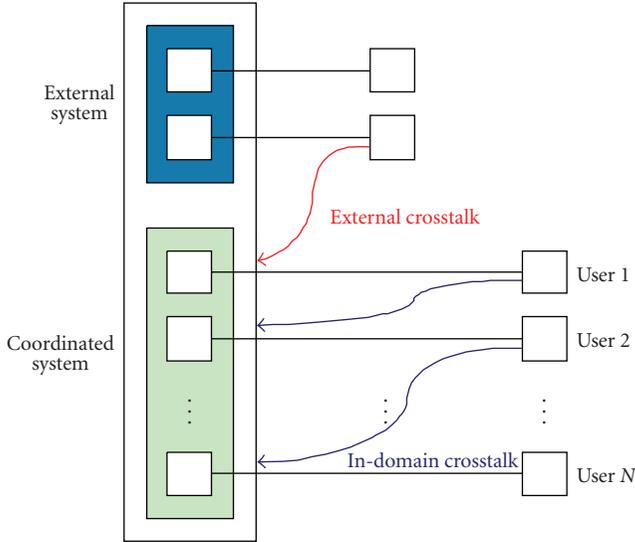


FIGURE 1: External crosstalk environment.

2. SYSTEM MODEL

A coordinated DSL system is depicted in Figure 1. N coordinated lines are colocated in the same Optical Network Unit (ONU). Because of crosstalk between pairs of the same binder, the coordinated signals interfere with each other and undergo interference from L different external sources. The N coordinated lines can benefit from a joint signal processing at the receiver side in the upstream direction.

Such as the recently adopted standard VDSL2, the transmission is based on a discrete multitone (DMT) modulation applied to data that were first coded and interleaved before being mapped to complex QAM symbols. According to the known DMT modulation principle, the constellation size is determined on each tone as a function of its signal-to-noise ratio. It is assumed that the cyclic prefix length exceeds the maximum delay of the channel and that the coordinated transmissions are perfectly synchronized. Thus, the transmission can be modelled in the frequency domain. For each tone k , the received signal \mathbf{y}^k is therefore written as

$$\mathbf{y}^k = \mathbf{H}^k \mathbf{x}^k + \mathbf{A}^k \mathbf{z}^k + \mathbf{w}^k, \quad k = 1, \dots, K, \quad (1)$$

where K is the number of tones. The vector $\mathbf{x}^k = [x_1^k, \dots, x_N^k]^T$ contains the complex symbols transmitted on tone k for the N coordinated users. The vector $\mathbf{z}^k = [z_1^k, \dots, z_L^k]^T$ contains the frequency components of the external transmitted signals on tone k . \mathbf{w}^k is the vector of additive white Gaussian noise (AWGN) elements on each line. If the transmit power spectral density (PSD) on each coordinated line i on tone k is denoted by $\sigma_{x_i^k}^2$, the autocorrelation of the transmitted spatially uncorrelated signal is a diagonal matrix $\mathbf{D}_{\mathbf{x}^k} = \mathcal{E}\{\mathbf{x}^k \mathbf{x}^{kH}\} = \text{diag}(\sigma_{x_1^k}^2, \dots, \sigma_{x_N^k}^2)$. In a similar way, the external sources are spatially uncorrelated and their autocorrelation matrix is $\mathbf{D}_{\mathbf{z}^k} = \mathcal{E}\{\mathbf{z}^k \mathbf{z}^{kH}\} = \text{diag}(\sigma_{z_1^k}^2, \dots, \sigma_{z_L^k}^2)$. The white noise PSD is assumed equal for the different coordinated lines. Therefore, the noise

autocorrelation matrix is $\mathcal{E}\{\mathbf{w}^k \mathbf{w}^{kH}\} = \sigma_{\mathbf{w}^k}^2 \mathbf{I}_N$. The matrix \mathbf{H}^k is the frequency response of the coordinated MIMO channel on tone k . The diagonal element $h_{i,i}^k$ of matrix \mathbf{H}^k is the attenuation coefficient of the line i and the off-diagonal element $h_{i,j}^k$ is the FEXT coupling coefficient between transmitter j and receiver i on tone k . Thanks to the physical properties of the cable, \mathbf{H}^k is diagonally dominant. This property is inherent to the fact that the direct channel is stronger than the crosstalk coupling channels which implies diagonal elements greater than the off-diagonal elements in the channel matrix. The $N \times L$ matrix \mathbf{A}^k contains the coupling coefficients between the external sources and the coordinated lines in the frequency domain. For reasons of clarity, the tone index is dropped in the following sections.

3. NOISE MITIGATION AND MAXIMUM THROUGHPUT UPPER BOUND

The resulting noise term $\mathbf{n} = \mathbf{A}\mathbf{z} + \mathbf{w}$ in (1) is spatially correlated. Its different components cannot be directly known by the receiver but its covariance matrix $R_{\mathbf{nn}}$ can be measured. It is written as follows:

$$R_{\mathbf{nn}} = E\{\mathbf{nn}^H\} = \mathbf{A}\mathbf{D}_{\mathbf{z}}\mathbf{A}^H + \sigma_{\mathbf{w}}^2 \mathbf{I}_N. \quad (2)$$

In this section, an upper bound of the multiuser DSL system maximum throughput is derived when noise whitening is carried out.

3.1. Noise whitening

Noise mitigation is obtained through linear filtering of the received signal in the frequency domain. The noise part of the resulting whitened signal has a diagonal correlation matrix. The computation of the whitening filter can be carried out using different methods among which two are described. The first method is based on the Cholesky decomposition of the noise correlation matrix $R_{\mathbf{nn}}$. The whitening filter can be expressed by

$$\mathbf{W} = \mathbf{G}^{-1}, \quad (3)$$

where $R_{\mathbf{nn}} = \mathbf{G}\mathbf{G}^H$ and \mathbf{G} is a lower triangular matrix. The upper triangular structure of the whitening filter allows successive noise cancelling to be carried out when the channel matrix \mathbf{H} is diagonal as proposed in [1].

The second method is based on the inverse square root of the noise covariance matrix $R_{\mathbf{nn}}$ given by

$$\mathbf{W} = (R_{\mathbf{nn}})^{-1/2}. \quad (4)$$

Eldar and Oppenheim show in [9] that this solution minimizes the mean-squared error between the original and whitened data. Therefore, it is used as a whitening solution with the iterative receiver described in the next section. In order to avoid matrix inversion, the computation of the noise whitening filter could be implemented iteratively as suggested in [10]. The received signal after whitening is expressed by

$$\check{\mathbf{y}} = \mathbf{W}\mathbf{y} = \check{\mathbf{H}}\mathbf{x} + \check{\mathbf{n}}. \quad (5)$$

Noise $\check{\mathbf{n}}$ has a covariance matrix $R_{\check{\mathbf{n}}} = \mathbf{I}_N$ which overcomes the noise correlation due to external noise in DSL coordinated systems. Unfortunately, the equivalent channel matrix $\check{\mathbf{H}}$ loses the diagonal dominance property useful for linear crosstalk cancellation.

3.2. Matched filter bound (MFB)

In order to determine the matched filter bound of the multiuser coordinated DSL receiver, a single user is assumed to be transmitting. At the receiver side, the signal is received on the direct line and on the other lines belonging to the coordinated system through crosstalk between lines. Thanks to whitening at the reception, the noise is uncorrelated and has the same power level on the different lines. This system is therefore equivalent to a channel with N diversity branches. It is shown in [11] that the maximum ratio combiner (MRC) is the optimum receiver for such a transmission scheme. The MRC linearly combines the individually received branch signals so as to maximize the signal-to-noise ratio for the considered user. The MRC output signal for user i can be written as

$$y_i^{\text{MFB}} = \sum_{j=1}^N |\check{h}_{j,i}|^2 x_i + \sum_{j=1}^N \check{h}_{j,i}^* \check{n}_j. \quad (6)$$

With the assumption that the different users of the coordinated system do not interfere with each other, the maximum throughput of user i on a given tone is expressed as

$$C_i^{\text{MFB}} = \log_2 \left(1 + \frac{1}{\Gamma} \sum_{j=1}^N |\check{h}_{j,i}|^2 \sigma_{x_i}^2 \right), \quad (7)$$

where Γ is the signal-to-noise ratio gap to capacity.

In practice, all the coordinated users transmit simultaneously and interfere with each other. These interferences have to be dealt with using signal processing techniques that can improve the system performance upper bounded by the maximum throughput (7).

4. MULTIUSER DETECTION

After whitening as described in Section 3, the different transmitted signals have to be estimated from the whitened signal $\check{\mathbf{y}}$ expressed by (5). In such a scenario, the linear processing for the coordinated upstream DSL system proposed in [3] is no longer efficient since the equivalent channel matrix $\check{\mathbf{H}}$ is not diagonally dominant. In this section, we present two multiuser detection methods that will be investigated in this transmission configuration. The first method uses successive interference cancellation based on ZF criterion as suggested in [2] for vectored DSL systems. The second method consists of an iterative MMSE receiver as proposed for wireless MIMO systems in [8].

4.1. ZF-based SIC

The successive interference canceller investigated for the multiuser detection is based on the QR decomposition of the

equivalent channel matrix after whitening as follows:

$$\check{\mathbf{H}} = \mathbf{Q}\mathbf{R}, \quad (8)$$

where \mathbf{Q} is a unitary matrix and \mathbf{R} is upper triangular. The whitened vector $\check{\mathbf{y}}$ is multiplied by matrix \mathbf{Q}^H which results in the following output:

$$\bar{\mathbf{y}} = \mathbf{Q}^H \check{\mathbf{y}} = \mathbf{R}\mathbf{x} + \bar{\mathbf{n}}, \quad (9)$$

where $\bar{\mathbf{n}} = \mathbf{Q}^H \check{\mathbf{n}}$ and $R_{\bar{\mathbf{n}}} = \mathbf{I}_N$. Since matrix \mathbf{R} is upper triangular, successive interference cancellation can be carried out as follows:

$$\hat{x}_i = \frac{1}{r_{i,i}} \left(\bar{y}_i - \sum_{j=i+1}^N r_{i,j} \text{dec}(\hat{x}_j) \right), \quad i = N, \dots, 1, \quad (10)$$

where $\text{dec}(\cdot)$ denotes symbol decision that might include channel decoding. When symbol x_i is being estimated, it is assumed that the decisions about symbols \hat{x}_j with $j = i + 1, \dots, N$ were error-free. Therefore, the signal-to-noise ratio for user i is expressed as

$$\text{SNR}_i = |r_{i,i}|^2 \sigma_{x_i}^2. \quad (11)$$

Since matrix \mathbf{Q} is unitary, the norm of each column vector i of matrix $\check{\mathbf{H}}$ is equal to its dual column vector i norm from matrix \mathbf{R} . Therefore, the previous expression can be rewritten as

$$\text{SNR}_i = \left(\sum_{j=1}^N |\check{h}_{j,i}|^2 - \sum_{k=1}^{i-1} |r_{k,i}|^2 \right) \sigma_{x_i}^2. \quad (12)$$

The maximum throughput attained by the ZF-based SIC for user i on one given tone is

$$C_i^{\text{ZF-SIC}} = \log_2 \left(1 + \frac{1}{\Gamma} \left(\sum_{j=1}^N |\check{h}_{j,i}|^2 - \sum_{k=1}^{i-1} |r_{k,i}|^2 \right) \sigma_{x_i}^2 \right). \quad (13)$$

4.2. Iterative MMSE receiver

The iterative receiver presented here in the context of multiuser DSL transmissions is based on MMSE detection with a priori information. Such a receiver was introduced for turbo-equalization purposes in [6, 12] and then adapted for MIMO systems [8]. Information exchanged between the MMSE receiver and the soft-input soft-output (SISO) channel decoder is represented by log-likelihood ratios (LLRs) [13] denoted by $\mathcal{L}(\cdot)$. The LLR on a binary value b is defined by

$$\mathcal{L}(b) = \ln \left(\frac{P\{b=1\}}{P\{b=0\}} \right). \quad (14)$$

As depicted in Figure 2, the MMSE detection provides soft symbols $\tilde{\mathbf{x}}$ using the channel output $\check{\mathbf{y}}$ and a priori information in the form of soft symbols $\hat{\mathbf{x}}$. During each iteration and for each user, enhanced LLR values \mathcal{L}_a^d are calculated from the previous ones \mathcal{L}_e^d by the channel-decoding stage.

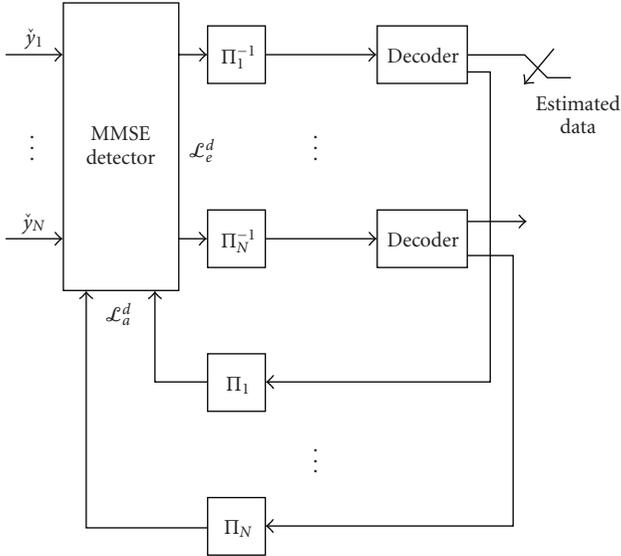


FIGURE 2: Iterative MIMO receiver.

The decoder requires soft inputs represented by the LLR values of each binary element. It is therefore necessary to convert complex symbols coming from the MMSE detector to LLR values on their binary elements. The output of the MMSE detector can be expressed by

$$\tilde{x} = g \cdot x + \eta, \quad (15)$$

where g is a bias given by (23) and η is noise with a variance of σ_η^2 . The transmitted symbols x are chosen in QAM-constellations \mathcal{X} . The conditional probability $P\{\tilde{x}|x\}$ is given by

$$\begin{aligned} \mathcal{L}(b_i|\tilde{x}) &= \ln \left(\frac{P\{b_i = 1|\tilde{x}\}}{P\{b_i = 0|\tilde{x}\}} \right), \\ \mathcal{L}(b_i|\tilde{x}) &= \ln \left(\frac{\sum_{x \in \mathcal{X}_1^i} P\{x|\tilde{x}\}}{\sum_{x \in \mathcal{X}_0^i} P\{x|\tilde{x}\}} \right), \\ \mathcal{L}(b_i|\tilde{x}) &= \ln \left(\frac{\sum_{x \in \mathcal{X}_1^i} P\{\tilde{x}|x\} P\{x\}}{\sum_{x \in \mathcal{X}_0^i} P\{\tilde{x}|x\} P\{x\}} \right), \end{aligned} \quad (16)$$

where \mathcal{X}_b^i are subsets of constellation \mathcal{X} whose bit of index i is equal to b .

In the case of demodulation without a priori information, all symbols are equiprobables. Therefore

$$\begin{aligned} \mathcal{L}(b_i|\tilde{x}) &= \ln \left(\frac{\sum_{x \in \mathcal{X}_1^i} P\{\tilde{x}|x\}}{\sum_{x \in \mathcal{X}_0^i} P\{\tilde{x}|x\}} \right), \\ \mathcal{L}(b_i|\tilde{x}) &= \ln \left(\frac{\sum_{x \in \mathcal{X}_1^i} \exp(-|\tilde{x} - g \cdot x|^2 / \sigma_\eta^2)}{\sum_{x \in \mathcal{X}_0^i} \exp(-|\tilde{x} - g \cdot x|^2 / \sigma_\eta^2)} \right). \end{aligned} \quad (17)$$

The decoder delivers LLR values for the decoded binary elements. This information has to be translated to its equivalent complex symbols in order to be fed back to the

MMSE detector. The LLR-to-symbol mapping is performed according to

$$\begin{aligned} \hat{x} &= E\{x|\mathcal{L}_a^d\}, \\ \hat{x} &= \sum_{s \in \mathcal{X}} s P\{x = s|\mathcal{L}_a^d\}, \end{aligned} \quad (18)$$

where \mathcal{L}_a^d is the a priori LLR value for each binary element contained in the transmitted symbol x .

The sets of interleavers $\{\Pi_1 \cdots \Pi_N\}$ and $\{\Pi_1^{-1} \cdots \Pi_N^{-1}\}$ are, respectively, used to arrange the LLR values in the correct order before channel decoding and soft mapping.

The output of the MMSE receiver with a priori information can be expressed for each user i of the system by [14]

$$\tilde{x}_i = \lambda_i \mathbf{p}_i^H (\check{\mathbf{y}} - \check{\mathbf{H}}\hat{\mathbf{x}} + \hat{x}_i \check{\mathbf{h}}_i), \quad (19)$$

where $\check{\mathbf{h}}_i$ is the i th column of the equivalent channel matrix $\check{\mathbf{H}}$. Equation (19) corresponds to interference cancellation followed by MMSE equalization where

$$\mathbf{p}_i = (\check{\mathbf{H}}\mathbf{V}\check{\mathbf{H}}^H + \mathbf{I}_N)^{-1} \check{\mathbf{y}}_i \quad (20)$$

is a vector containing the equalization coefficients. It takes into account the reliability of the a priori information by means of the matrix $\mathbf{V} = \text{diag}(\hat{\gamma}_1^2, \dots, \hat{\gamma}_N^2)$ with

$$\hat{\gamma}_i^2 = \sum_{s \in \mathcal{X}_i} |s|^2 P\{x = s|\mathcal{L}_a\} - |\hat{x}_i|^2. \quad (21)$$

The coefficients λ_i are computed as follows:

$$\lambda_i = \frac{\sigma_{x_i}^2}{1 + (\sigma_{x_i}^2 - \hat{\gamma}_i^2)\beta_i}, \quad \beta_i = \mathbf{p}_i^H \check{\mathbf{h}}_i. \quad (22)$$

The bias g_i and the variance σ_η^2 defined in (15) can be expressed by

$$\begin{aligned} g_i &= \lambda_i \beta_i, \\ \sigma_\eta^2 &= \text{var}(\tilde{x}_i - g_i \hat{x}_i). \end{aligned} \quad (23)$$

At the first iteration in the MMSE receiver, there is no a priori information and $\hat{\gamma}_i^2 = \sigma_{x_i}^2$. This results in the classical MMSE receiver. In case of perfect a priori information, the interference in the received signal is completely removed and as matrix \mathbf{V} becomes null, the receiver acts as the MRC described in Section 3.2. Then, from (7) and considering a convergence of the iterative process, the maximum throughput of the system with the iterative MMSE receiver on a given tone is expressed as

$$C_i^{\text{MMSE}} = \log_2 \left(1 + \frac{1}{\Gamma} \sum_{j=1}^N |\check{h}_{j,i}|^2 \sigma_{x_i}^2 \right). \quad (24)$$

5. NUMERICAL RESULTS

This section is dedicated first to the analysis of the maximum throughput of the different detection schemes presented in Section 4. In a second step, the iterative MMSE receiver behavior is examined in realistic DSL transmission conditions.

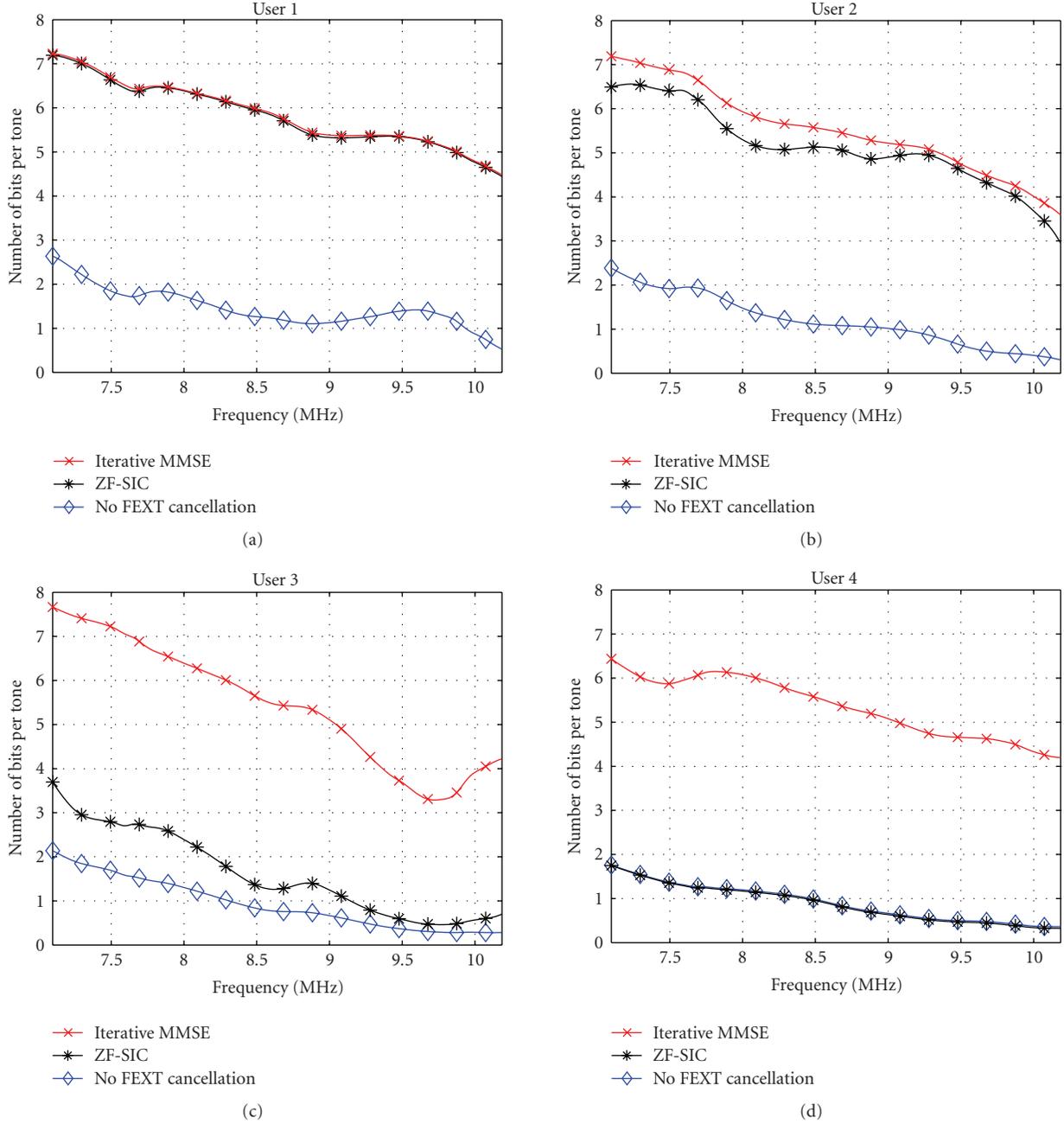


FIGURE 3: Iterative MMSE and ZF-SIC bit allocation in the coordinated system.

5.1. Performance evaluation

Derived maximum throughput expressions for the ZF-based SIC and iterative MMSE receiver in (13) and (24), respectively, help compare the expected performance of these structures. For this purpose, a scenario of four-coordinated users and two external crosstalk interferers as depicted in Figure 1 is considered. Transmission parameters mimic the VDSL2 setup and are reported in Table 1.

Performance results in terms of allocated bits per tone for each user are drawn in Figure 3. Maximum throughput of the uncoordinated upstream transmission are represented by the

diamond curves. The attained maximum throughput by the ZF-SIC scheme is represented by the starred curves and the iterative MMSE performance after convergence is depicted by the crossed curves. The difference in throughput between the coordinated system users is due to the channel structure and the position of the copper pairs in the binder. Moreover, the throughputs decrease with frequency because of the channel attenuation. ZF-SIC performance of user 1 attains the iterative MMSE maximum throughput (24) which has been shown to be equal to the MFB, $C_1^{MMSE} = C_1^{ZF-SIC}$. Conversely, user 4 does not benefit from the multiuser processing in the ZF-SIC scheme, and its performance is

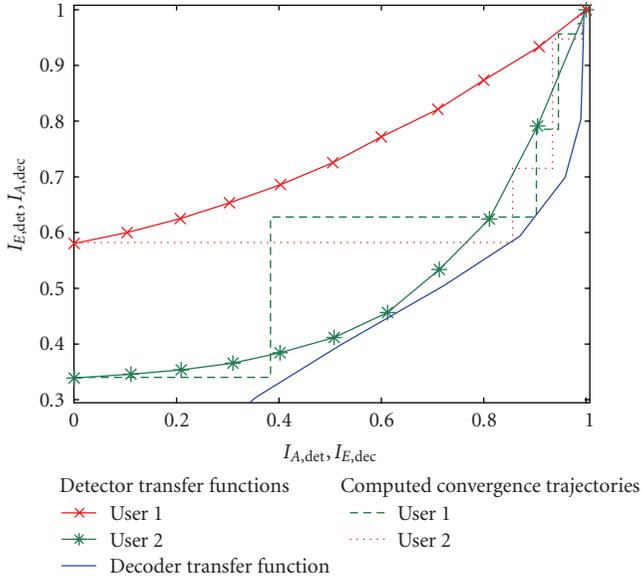


FIGURE 4: EXIT chart for two-coordinated users and one external interferer.

equivalent to single-user transmission performance. On the opposite, all users in the iterative MMSE receiver benefit equally from the multiuser processing to approach the MFB.

5.2. Iterative MMSE convergence analysis

The previous results in terms of maximum throughputs assume the convergence of the MMSE iterative processing that reaches the matched filter bound. A practical approach for the iterative MMSE convergence analysis is to use extrinsic information transfer (EXIT) charts introduced by ten Brink in [15]. This allows the exchange of information to be displayed and the transfer characteristics of an iterative process are represented on a chart.

The mutual information between a binary element $x \in \{-1, +1\}$ and its weighted information \mathcal{L} is given by

$$I(\mathcal{L}, x) = \frac{1}{2} \sum_{x \in \{-1, +1\}} \int_{-\infty}^{+\infty} f_{\mathcal{L}}(l|x) \log_2 \left(\frac{2f_{\mathcal{L}}(l|x)}{f_{\mathcal{L}}(l+1) + f_{\mathcal{L}}(l-1)} \right) dl. \quad (25)$$

It is characterized by the probability density function $f_{\mathcal{L}}(\mathcal{L}|x)$. This quantity can be approached by a histogram of LLR values (14) which are assumed to follow a Gaussian distribution.

Each block in the iterative MMSE receiver is characterized by a transfer function linking its input and output mutual information [15]. The mutual information transfer function for the detector is plotted with its a priori input $I_{A,det}$ on the abscissa axis and its extrinsic output $I_{E,det}$ on the ordinate axis. The decoder component transfer function is plotted with its a priori input $I_{A,dec}$ on the ordinate axis and its extrinsic output $I_{E,dec}$ on the abscissa axis. The

TABLE 1: System setup.

Coordinated line length	600 m
External line length	75 m
Frequency band	[7.09–10.18] MHz
Transmitted signal PSD	$\mathcal{E}_x = -60$ dBm/Hz
External interferer PSD	$\mathcal{E}_z = -60$ dBm/Hz
White Gaussian noise PSD	$\mathcal{E}_w = -140$ dBm/Hz
Transmission gap	9.8 dB

decoder is common to all coordinated users, whereas each user has its own detector transfer function. In the sequel, the convolutional used code is $G(D) = ((1 + D + D^2), (1 + D^2))$ for which a maximum a posteriori MAP decoding algorithm is implemented. Data coding and decoding are carried out in the frequency domain; and each codeword corresponds to a DMT block. Interleaving helps decorrelate data between the detector and the decoder.

Figure 4 shows an EXIT chart for two-coordinated receiving users with one external crosstalk interferer. The mutual information transfer functions are computed using blocks of one hundred DMT symbols. The same parameters setup reported in Table 1 except for the frequency band that is chosen within [7.3–8.9] MHz is considered. This frequency band results in a bit allocation ranging between 1 and 6 bits per tone.

In an EXIT chart representation, a decoding trajectory will take the form of a stair-shaped curve. The trajectory goes between one of the detector mutual information transfer functions and the decoder transfer function along the different iterations. In the case of two-coordinated lines and one external line, we can prove that the output of the MMSE detector of user 1 depends only on the input of the MMSE detector of user 2 and vice versa, as shown in the appendix. Consequently, the first user trajectory intersects, respectively, the first detector transfer function for the odd iterations and the second detector transfer function for the even iterations. The second user trajectory behaves the opposite way. This progress of the trajectories' iterations is illustrated in Figure 4. The difference between the MMSE detector transfer functions for both users is caused by the structure of the equivalent channel matrix $\tilde{\mathbf{H}}$. The departing point is low for user 2 because the equivalent channel matrix off-diagonal element $\tilde{h}_{2,1}$ magnitude is of the same order as the direct path $\tilde{h}_{2,2}$ magnitude, whereas the off-diagonal element $\tilde{h}_{1,2}$ magnitude is small compared to the direct path $\tilde{h}_{1,1}$ magnitude.

The same scenario with four-coordinated users and two external crosstalk interferers considered in Section 5.1 is examined regarding convergence of the iterative MMSE process for the setup given in Table 1. The associated EXIT chart is depicted in Figure 5. The resulting EXIT chart and the position of the detector mutual information transfer

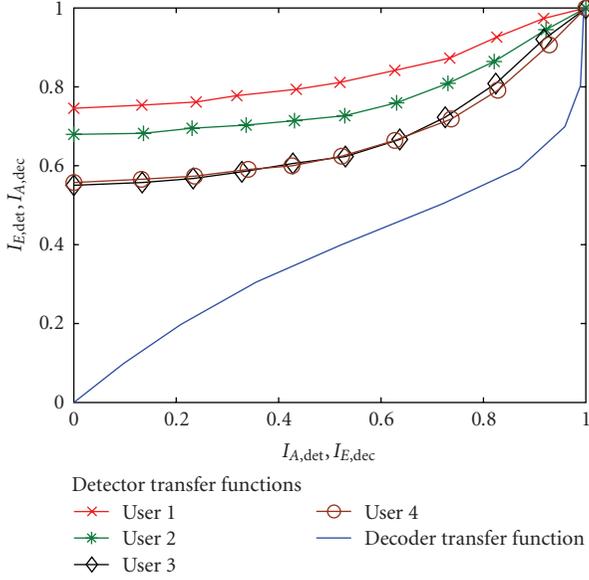


FIGURE 5: Convergence of the iterative system for four-coordinated users with two external crosstalk interferers.

functions indicate that the system converges within few iterations and therefore the matched filter bound performance will be attained.

6. CONCLUSION

In this paper, the combination of external crosstalk whitening and iterative MMSE processing is examined for upstream coordinated DSL systems. The iterative MMSE receiver achieves the MFB performance under the assumption of perfect a priori information. Unlike the ZF-SIC, all users in the iterative MMSE receiver benefit equally from the multiuser processing to approach the MFB. EXIT charts are used to analyze the behavior of this scheme for upstream coordinated DSL systems in the presence of correlated external noise. Convergence of the iterative MMSE receiver for coordinated DSL systems is obtained in realistic conditions.

APPENDIX

Lemma A.1. *In the MIMO system with an iterative MMSE detector for two-coordinated users and without a priori information fed to the soft demapper, the output information for user 1 depends only on the input information for user 2.*

Proof. The output signal for user 1 with interference cancellation can be written as

$$\begin{aligned}\tilde{\mathbf{x}}_1 &= \lambda_1 \mathbf{P}_1^H (\mathbf{y} - \check{\mathbf{H}}\hat{\mathbf{x}} + \hat{\mathbf{x}}_1 \check{\mathbf{h}}_1), \\ \tilde{\mathbf{x}}_1 &= \lambda_1 \mathbf{P}_1^H (\mathbf{y} + \hat{\mathbf{x}}_2 \check{\mathbf{h}}_2).\end{aligned}\quad (\text{A.1})$$

Developing the matrix computation of $\lambda_1 \mathbf{P}_1^H$, we show that

$$\begin{aligned}\lambda_1 \mathbf{P}_1^H &= \frac{\sigma_x^2}{A\nu_2^2\sigma_x^2 + B\sigma_n^2\sigma_x^2 + C\nu_2^2\sigma_n^2 + \sigma_n^4} \\ &\times \begin{bmatrix} (\check{h}_{11}^* |\check{h}_{22}|^2 - \check{h}_{21}^* \check{h}_{12} \check{h}_{22}^*) \nu_2^2 + \check{h}_{11}^* \sigma_n^2 \\ (\check{h}_{21}^* |\check{h}_{12}|^2 - \check{h}_{11}^* \check{h}_{22} \check{h}_{12}^*) \nu_2^2 + \check{h}_{21}^* \sigma_n^2 \end{bmatrix}^T\end{aligned}\quad (\text{A.2})$$

with

$$\begin{aligned}A &= |\check{h}_{11}|^2 |\check{h}_{22}|^2 + |\check{h}_{21}|^2 |\check{h}_{12}|^2 \\ &\quad - \check{h}_{11} \check{h}_{21}^* \check{h}_{12} \check{h}_{22} - \check{h}_{21} \check{h}_{11}^* \check{h}_{22} \check{h}_{12}, \\ B &= |\check{h}_{11}|^2 + |\check{h}_{21}|^2, \\ C &= |\check{h}_{22}|^2 + |\check{h}_{12}|^2.\end{aligned}\quad (\text{A.3})$$

Therefore, $\tilde{\mathbf{x}}_1$ depends only on the reliability of user 2. \square

REFERENCES

- [1] G. Ginis and C.-N. Peng, "Alien crosstalk cancellation for multipair digital subscriber line systems," *EURASIP Journal on Applied Signal Processing*, vol. 2006, Article ID 16828, 12 pages, 2006.
- [2] G. Ginis and J. M. Cioffi, "Vectored transmission for digital subscriber line systems," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 5, pp. 1085–1104, 2002.
- [3] R. Cendrillon, G. Ginis, E. van den Bogaert, and M. Moonen, "A near-optimal linear crosstalk canceler for upstream VDSL," *IEEE Transactions on Signal Processing*, vol. 54, no. 8, pp. 3136–3146, 2006.
- [4] V. Le Nir, M. Moonen, and J. Verlinden, "Optimal power allocation under per-modem total power and spectral mask constraints in xDSL vector channels with alien crosstalk," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '07)*, vol. 3, pp. 357–360, Honolulu, Hawaii, USA, April 2007.
- [5] X. Wang and H. V. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Transactions on Communications*, vol. 47, no. 7, pp. 1046–1061, 1999.
- [6] M. Tüchler, A. C. Singer, and R. Koetter, "Minimum mean squared error equalization using a priori information," *IEEE Transactions on Signal Processing*, vol. 50, no. 3, pp. 673–683, 2002.
- [7] J. Le Masson, C. Langlais, and C. Berrou, "Linear precoding with low complexity MMSE turbo-equalization and application to the wireless LAN system," in *Proceedings of the IEEE International Conference on Communications (ICC '05)*, vol. 4, pp. 2352–2356, Seoul, Korea, May 2005.
- [8] M. Witzke, S. Bärö, F. Schreckenbach, and J. Hagenauer, "Iterative detection of MIMO signals with linear detectors," in *Proceedings of the 36th Asilomar Conference on Signals, Systems and Computers (ACSSC '02)*, vol. 1, pp. 289–293, Pacific Grove, Calif, USA, November 2002.
- [9] Y. C. Eldar and A. V. Oppenheim, "MMSE whitening and subspace whitening," *IEEE Transactions on Information Theory*, vol. 49, no. 7, pp. 1846–1851, 2003.

- [10] S. Venkatesan, L. Mailaender, and J. Salz, "An iterative algorithm for computing a spatial whitening filter," in *Proceedings of the 5th IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC '04)*, pp. 338–342, Lisbon, Portugal, July 2004.
- [11] J. G. Proakis, *Digital Communications*, McGraw-Hill, New York, NY, USA, 1995.
- [12] A. Glavieux, C. Laot, and J. Labat, "Turbo equalization over a frequency selective channel," in *Proceedings of the International Symposium on Turbo Codes and Related Topics (ISTC '97)*, pp. 96–102, Brest, France, September 1997.
- [13] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 429–445, 1996.
- [14] C. Laot, R. Le Bidan, and D. Leroux, "Low-complexity MMSE turbo equalization: a possible solution for EDGE," *IEEE Transactions on Wireless Communications*, vol. 4, no. 3, pp. 965–974, 2005.
- [15] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Transactions on Communications*, vol. 49, no. 10, pp. 1727–1737, 2001.

Research Article

Iterative Mean Removal Superimposed Training for SISO and MIMO Channel Estimation

O. Longoria-Gandara,¹ R. Parra-Michel,¹ M. Bazdresch,² and A. G. Orozco-Lugo³

¹ Department of Electrical Engineering, CINVESTAV-IPN, Apartado Postal 31-438, Plaza La Luna, Guadalajara, 44550 JAL, Mexico

² Department of Electronics, Systems and Computer Science, ITESO, Per. Sur M. Gomez Morin 8585, Tlaquepaque, 45604 JAL, Mexico

³ Department of Electrical Engineering, CINVESTAV-IPN, Apartado Postal 14-740, 07000 Mexico, DF, Mexico

Correspondence should be addressed to O. Longoria-Gandara, olongo@gdl.cinvestav.mx

Received 2 April 2008; Revised 1 August 2008; Accepted 22 September 2008

Recommended by Fred Daneshgaran

This contribution describes a novel iterative radio channel estimation algorithm based on superimposed training (ST) estimation technique. The proposed algorithm draws an analogy with the data dependent ST (DDST) algorithm, that is, extracts the cycling mean of the data, but in this case at the receiver's end. We first demonstrate that this mean removal ST (MRST) applied to estimate a single-input single-output (SISO) wideband channel results in similar bit error rate (BER) performance in comparison with other iterative techniques, but with less complexity. Subsequently, we jointly use the MRST and Alamouti coding to obtain an estimate of the multiple-input multiple-output (MIMO) narrowband radio channel. The impact of imperfect channel on the BER performance is evidenced by a comparison between the MRST method and the best iterative techniques found in the literature. The proposed algorithm shows a good tradeoff performance between complexity, channel estimation error, and noise immunity.

Copyright © 2008 O. Longoria-Gandara et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

One of the most widely used approaches to channel estimation is to employ pilot assisted transmission (PAT), where a known training sequence, also referred as pilot, is inserted at each block of transmitted data [1]. Using the knowledge of the training symbols and the corresponding received signal, the channel estimator block at the receiver is able to make an estimate of the channel impulse response (CIR). However, these training pilots, using time division multiplexing (TDM) scheme, consume valuable bandwidth resulting in the reduction of the data rate.

There are two well-known channel estimation techniques to avoid the bandwidth waste of PAT schemes: superimposed training (ST) [2, 3] and data dependent ST (DDST) [4]. These techniques are based on the arithmetic addition (superimposed) of a training sequence to the information data. Both schemes provide a simple (unsophisticated) channel estimation processes; they differ only in that the cyclic mean of the transmitted data of the DDST scheme is superimposed into the transmitted sequence in similar way that the training signal.

Although DDST outperforms ST [4] in terms of channel estimation error, it is worth mentioning that the decoding of data under DDST is of iterative nature because it needs to extract the data-dependent distortion. Considering this, DDST with DDD removal (henceforth we will refer to this scheme as DDST-DDD removal) gives a similar performance as TDM-based channel estimation, but with fewer bandwidth losses. There are however some drawbacks in trade; DDST technique introduces a delay in the transmitted data in order to calculate the cyclic mean. It also assigns less power to the data signal, and hence the use of high-order symbol constellations has repercussions on the data decoding process [5].

The last constraints lead to research on iterative implementations of ST as in [6–10], starting from the SISO radio channel. These works are based on the use of the decoded data to eliminate the distortion introduced by the received data in the channel estimation process. The first approach uses ST in combination with a traditional least squares channel estimate (LSST) and was developed in [6]. In [6, 7] it is clearly shown that in terms of the channel estimation mean square error (MSE), LSST converges to

a fully trained system for high SNR in two iterations, thereby outperforming both conventional ST and DDST. The great disadvantage of LSST scheme is the computational burden of $\mathcal{O}(NM^2)$ for one iteration, where N is the block length and M is the order of the CIR. In [11], it is demonstrated that DDST-DDD achieves similar performance than LSST but with considerably less complexity.

Alternative iterative procedures of both ST and DDST methods were introduced in [7]. The first one, IST, uses the equalized symbols, obtained via ST, to improve the channel estimate in an iterative way but with less complexity than LSST [6]. The second one, LSDDST employs the LSST iterative approach but based on DDST instead of ST. LSDDST scheme, has the same computational burden as LSST, that is, $\mathcal{O}(NM^2)$, but it converges faster to the fully trained system. In terms of bit error rate (BER), LSDDST scheme yields almost the same BER than LSST.

From the previous works, we are now able to establish this contribution in the context of low-complexity iterative algorithms using ST that shows good BER performance. We introduce a new iterative mean removal ST (MRST) proposal and compare its performance with the previous and most relevant works.

This MRST yields similar performance to DDST-DDD removal and IST but with less complexity when they are compared with LSDDST and LSST. Because the iterative channel estimation methods depend on and work jointly with the equalization stage, we present the results using two equalizers widely used in communication systems: the minimum mean square error (MMSE) equalizer and the maximum likelihood sequence estimation (MLSE) equalizer. The inclusion of both techniques is helpful to accentuate some particularities of the channel estimation methods used.

Additionally, we extend the results of SISO to MIMO case, and study the performance of training-based flat block-fading MIMO channel estimation. Three training-based channel estimators are considered (TDM, DDST, and MRST), which offer different tradeoffs in terms of performance. We analyze the error performance of the MRST method based on the traditional least squares (LS) method and obtain the corresponding MSE. The proposed MRST estimator for the MIMO case is illustrated using an orthogonal space-time block coder (OSTBC) with two transmit and two receive antennas, that is, Alamouti 2×2 space-time coding.

This paper is organized as follows. Section 2 deals with the new iterative ST approach in the framework for ST/DDST schemes for SISO systems. The performance analysis of the MRST for MIMO systems is obtained in Section 3. In Sections 4 and 5, the simulation results and performance comparison are given for SISO and MIMO, respectively. Finally, the conclusions are set down in Section 6.

2. MEAN REMOVAL ST FOR SISO SYSTEMS

2.1. SISO system model

Assuming a frequency-selective channel, Figure 1 depicts the discrete-time baseband block diagram of a digital com-

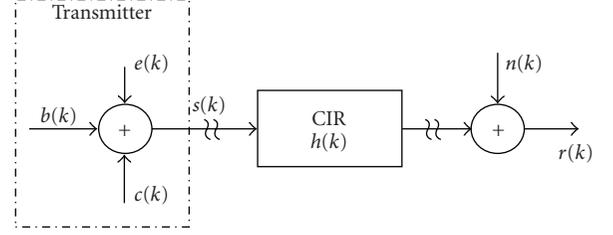


FIGURE 1: SISO digital communication system model.

munication system, where the channel input signal $\{s(k)\}$ represents a succession of information blocks of length N , and is given by

$$s(k) = b(k) + c(k) + e(k), \quad (1)$$

taking into account that the block of interest is k indexed and $k = 0, 1, \dots, N - 1$. The sequence $\{b(k)\}$ represents the data with zero mean and variance $E\{|b(k)|^2\} = \sigma_b^2$, $\{c(k)\}$ is a deterministic periodic training sequence with period P and power $\sigma_c^2 = (1/P)\sum_{k=0}^{P-1}|c(k)|^2$, and $\{e(k)\}$ refers to a data-dependent sequence, with period P , obtained by periodically repeating N_P times the signal

$$e(j) = -\left(\frac{1}{N_P}\right) \sum_{i=0}^{N_P-1} b(iP + j), \quad j = 0, \dots, P - 1, \quad (2)$$

where $N_P = N/P$.

For the ST case, $e(k) = 0$ and the sequences $\{c(k)\}$, $\{e(k)\}$ are arithmetically added in a superimposed way according to [4], before transmission. Then, the received signal, assuming that exact synchronization and DC-offset are provided, can be observed at the receiver as

$$r(k) = \sum_{l=0}^{M-1} h(l)s(k-l) + n(k), \quad (3)$$

where $h(k)$ is the impulse response of order $M - 1$ (i.e., $h(0) \neq 0$ and $h(M - 1) \neq 0$), and $\{n(k)\}$ is the complex Gaussian random noise with zero-mean, white, uncorrelated to $\{b(k)\}$ and independent real and imaginary parts with variance $\sigma_n^2 = N_0/2$ per dimension.

Assuming that the channel is quasistatic, that is, the channel is time-invariant during the information block received, and the exact channel order ($M - 1$) is known in advance, then the strong constraint ($P \geq 2M + 1$) showed in [11], with exact synchronization and DC-offset provided, can be relaxed to $P = M$. We used this assumption for the mathematical analysis of the system. Despite this relaxation, all simulations were carried out considering the strong constraint.

2.2. Performance analysis

It is clear that DDST method must compute the sequence $\{e(k)\}$ from the data block that will be transmitted; consequently this data processing has an impact on the total delay

of every transmitted block. On the other hand, the sequence $\{s(k)\}$ has a wider dynamic range in comparison with the sequence obtained with ST method. This results in a higher peak-to-average power ratio (PAPR) of the preamplifier communication building block. Another implication using DDST is the fact that the sequence $\{b(k)\}$ will have less power in comparison again with ST and then less noise immunity. For these reasons, iterative ST schemes are very attractive.

We developed the MRST scheme starting from the hypothesis that if we could obtain an estimate of the signal $\{e(k)\}$ (i.e., a cyclic mean of sequence $\{b(k)\}$) at the receiver side, then we would achieve the performance of DDST in terms of channel error estimate MSE but with more power assigned to the data sequence $\{b(k)\}$, having an impact on a better performance in terms of BER.

Estimating the cycling mean of period P as in [3], we can write

$$\tilde{y}(j) = \frac{1}{N_P} \sum_{i=0}^{N_P-1} r(iP + j), \quad (4)$$

with $j = 0, 1, \dots, P - 1$.

Combining (3) and (4) yields

$$\tilde{y}(j) = \sum_{m=0}^{M-1} h(m) \{ \tilde{b}(j-m) + e(j-m) + c(j-m) \} + \tilde{n}(j), \quad (5)$$

where

$$\tilde{b}(k) = \frac{1}{N_P} \sum_{i=0}^{N_P-1} b(iP + j), \quad \tilde{n}(k) = \frac{1}{N_P} \sum_{i=0}^{N_P-1} \tilde{n}(iP + j). \quad (6)$$

From (5), it follows that in matrix form

$$(\mathbf{C} + \tilde{\mathbf{B}} + \mathbf{E})\mathbf{h} = \tilde{\mathbf{y}} - \tilde{\mathbf{n}}, \quad (7)$$

where \mathbf{C} and \mathbf{E} are $P \times P$ circulant matrices with first columns $[c(0) c(1) \dots c(P-1)]^T$ and $[e(0) e(1) \dots e(P-1)]^T$, respectively, and $\mathbf{h} = [h(0) h(1) \dots h(P-1)]$. The $P \times 1$ column vectors $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{n}}$ have similar expression to \mathbf{h} . Assuming a cyclic prefix of length P (as was done in [4, 7]), $\tilde{\mathbf{B}}$ is circulant matrix with first column $[\tilde{b}(0) \tilde{b}(1) \dots \tilde{b}(P-1)]^T$.

For the ST case (i.e., when $\mathbf{E} = \mathbf{0}$ in (7)) we have $\tilde{\mathbf{y}} = (\mathbf{C} + \tilde{\mathbf{B}})\mathbf{h} + \tilde{\mathbf{n}}$, and using the channel estimate $\mathbf{C}^{-1}\tilde{\mathbf{y}}$ from [3] then

$$\hat{\mathbf{h}}_{\text{ST}} = \mathbf{C}^{-1}\tilde{\mathbf{y}} = \mathbf{h} + \mathbf{C}^{-1}\tilde{\mathbf{B}}\mathbf{h} + \mathbf{C}^{-1}\tilde{\mathbf{n}}. \quad (8)$$

For the DDST case $\mathbf{E} = -\tilde{\mathbf{B}}$ in (7), so we have the channel estimate [4] as

$$\hat{\mathbf{h}}_{\text{DDST}} = \mathbf{C}^{-1}\tilde{\mathbf{y}} = \mathbf{h} + \mathbf{C}^{-1}\tilde{\mathbf{n}}. \quad (9)$$

Clearly, the difference between these two channel estimation schemes is the factor $\mathbf{C}^{-1}\tilde{\mathbf{B}}\mathbf{h}$.

Now using the hypothesis early mentioned (i.e., $\mathbf{E} = -\tilde{\mathbf{B}} \simeq -\tilde{\mathbf{B}}$ with ST scheme), we proceed to make the channel estimation for the MRST scheme, then (7) becomes

$$\tilde{\mathbf{y}} = (\mathbf{C} + \tilde{\mathbf{B}} - \hat{\tilde{\mathbf{B}}})\mathbf{h} + \tilde{\mathbf{n}}, \quad (10)$$

and multiplying (10) by \mathbf{C}^{-1} we obtain

$$\hat{\mathbf{h}}_{\text{MRST}} = \mathbf{C}^{-1}\tilde{\mathbf{y}} = \mathbf{C}^{-1}\{(\mathbf{C} + \tilde{\mathbf{B}} - \hat{\tilde{\mathbf{B}}})\mathbf{h} + \tilde{\mathbf{n}}\}, \quad (11)$$

and finally

$$\hat{\mathbf{h}}_{\text{MRST}} = \mathbf{h} + \mathbf{C}^{-1}(\tilde{\mathbf{B}} - \hat{\tilde{\mathbf{B}}})\mathbf{h} + \mathbf{C}^{-1}\tilde{\mathbf{n}}. \quad (12)$$

From (12) it follows that if $\hat{\tilde{\mathbf{B}}} = \tilde{\mathbf{B}}$ then $\hat{\mathbf{h}}_{\text{MRST}} = \hat{\mathbf{h}}_{\text{DDST}}$. In order to make a good estimate ($\hat{\tilde{\mathbf{B}}}$) of $\tilde{\mathbf{B}}$, the following steps are made.

- (1) Use (8) to have an initial channel estimate as plain ST and make $\tilde{\mathbf{y}}_{\text{old}} = \tilde{\mathbf{y}}$.
- (2) Use the channel estimated to obtain the equalized symbols and employ a hard decision detector.
- (3) Use the hard-decision symbols detected to calculate $\hat{\tilde{\mathbf{B}}}$.
- (4) Remove $\hat{\tilde{\mathbf{B}}}$ from the received signal to obtain a new $\tilde{\mathbf{y}}$ according to

$$\tilde{\mathbf{y}}_{\text{new}} = \tilde{\mathbf{y}}_{\text{old}} - \hat{\tilde{\mathbf{B}}}\hat{\mathbf{h}}, \quad (13)$$

where $\tilde{\mathbf{y}}_{\text{new}}$ is a column vector $P \times 1$.

- (5) Use (11) with $\tilde{\mathbf{y}} = \tilde{\mathbf{y}}_{\text{new}}$ and update the channel estimate $\hat{\mathbf{h}}_{\text{MRST}}$.
- (6) Go to step 2 and repeat as need it.

Defining the $\text{MSE}(\hat{\mathbf{h}}) := E\{\sum_{k=0}^{M-1} |\hat{h}(k) - h(k)|^2\}$, then from [3, 4], and having a better estimation ($\hat{\tilde{\mathbf{B}}}$) of $\tilde{\mathbf{B}}$, it follows that $\hat{\mathbf{h}}_{\text{MRST}} \simeq \hat{\mathbf{h}}_{\text{DDST}}$, then

$$\text{MSE}(\hat{\mathbf{h}}_{\text{MRST}}) \simeq \frac{\sigma_n^2}{N_P \sigma_c^2}. \quad (14)$$

From (9) and (11), DDST and MRST, respectively, we note the fact that \mathbf{C}^{-1} is calculated only once beforehand. The IST scheme, however, uses the estimate ($\hat{\tilde{\mathbf{B}}}$) of $\tilde{\mathbf{B}}$ to calculate $(\mathbf{C} + \hat{\tilde{\mathbf{B}}})^{-1}$ (see [7, (12)]) for every iteration and information block received. This action implies an additional complexity to the computational burden.

On the other hand, as we will see in Section 4, the estimate $\hat{\tilde{\mathbf{B}}}$ used to estimate the channel in IST and MRST methods depends on what type of equalizer is used. Conversely, the scheme DDST-DDD removal cannot take advantage of MLSE equalizer, because this equalizer delivers hard-decision symbols, and DDST-DDD removal uses the equalized symbols previous to hard-decision procedure.

Table 1 summarizes the computational burden, the computation of inverse matrix (\mathbf{C}^{-1}), and the MSE performance approach. The complexity of an algorithm is a quite important metric when the algorithm is to be implemented in HW, as this is the case we are very concern of this value. As for computational burden metric, we choose the number

TABLE 1: Synoptic Table of the Computational burden, \mathbf{C}^{-1} Computation and the Channel Estimate MSE.

Procedure	Computational burden	\mathbf{C}^{-1}	MSE($\hat{\mathbf{h}}$) approach
DDST-DDD removal [11]	$\mathcal{O}(\mathcal{Q}N)$ 1 iteration	Pre-calculated	DDST ([4])
IST scheme in [7]	$\mathcal{O}(3\mathcal{Q}N)$ 2 iterations	$(\mathbf{C} + \hat{\mathbf{B}})^{-1}$ every iteration	DDST ([4])
MRST	$\mathcal{O}(3\mathcal{Q}N)$ 2 iterations	Pre-calculated	DDST ([4])

$\mathcal{O}(\cdot)$ notation expresses complexity, \mathcal{Q} , and N denote the equalizer length and the block length, respectively.

of iterations as function of the transmitted block length N and the equalizer length \mathcal{Q} , to be precise, and the number of coefficients in the equalizer.

Although DDST has the least computational complexity at the receiver side, it needs to calculate the data-dependent sequence $\{e(k)\}$, that implies additional complexity and time delay. Furthermore, it assigns less transmission power to the data signal.

3. MEAN REMOVAL ST FOR MIMO SYSTEMS

3.1. MIMO system model

We consider a wireless MIMO communication link with n_T transmit and n_R receive antennas, operating in a Rayleigh flat-fading environment. The fading coefficient $h_{i,j}$ is the complex path gain from transmit antenna i to receive antenna j . We assume that the coefficients are independently complex circular symmetric Gaussian with unit variance, and then $\mathbf{H} = [h_{i,j}] \in \mathbb{C}^{n_T \times n_R}$. The expression for the received symbols can be expressed as

$$\mathbf{r} = \mathbf{x}\mathbf{H} + \mathbf{n}, \quad (15)$$

where \mathbf{r} is the $1 \times n_R$ received signal vector, \mathbf{x} is the $1 \times n_T$ transmitted signal vector, and \mathbf{n} is an $1 \times n_R$ vector of additive noise terms, assuming that noise is spatially and temporally white Gaussian with zero mean and independent real and imaginary parts with variance $\sigma_n^2 = N_0/2$ per dimension.

Let us assume the block transmission scheme with the block length N at times $k = 1, \dots, N$, and we also assume that the channel matrix \mathbf{H} remains constant within a block of N symbols, that is, the block length is much small than the channel coherence time. Under these assumptions, the channel, within one block, can be written as

$$\mathbf{R} = \mathbf{X}\mathbf{H} + \mathbf{N}, \quad (16)$$

where

$$\mathbf{R} \triangleq \begin{bmatrix} \mathbf{r}(1) \\ \mathbf{r}(2) \\ \vdots \\ \mathbf{r}(N) \end{bmatrix}, \quad \mathbf{X} \triangleq \begin{bmatrix} \mathbf{x}(1) \\ \mathbf{x}(2) \\ \vdots \\ \mathbf{x}(N) \end{bmatrix}, \quad \mathbf{N} \triangleq \begin{bmatrix} \mathbf{n}(1) \\ \mathbf{n}(2) \\ \vdots \\ \mathbf{n}(N) \end{bmatrix} \quad (17)$$

are the matrices of the received signal, transmitted signals, and noise, respectively [12].

Let us denote the set of complex information symbols prior to space-time encoding as $\{s_\lambda \mid s_\lambda \in \mathcal{U}_\lambda, \lambda = 1, 2, \dots, \Lambda\}$, where each \mathcal{U}_λ denotes a set of signal constellation points. The symbols $s_1, s_2, \dots, s_\Lambda$ are zero-mean

mutually uncorrelated random variables. Let us introduce the vector

$$\mathbf{s} \triangleq [s_1 \ s_2 \ \dots \ s_\Lambda]^T, \quad (18)$$

where $(\cdot)^T$ denotes transpose. Note that $\mathbf{s} \in \mathcal{S}$, where $\mathcal{S} = \{\mathbf{s}^{(1)} \ \mathbf{s}^{(2)} \ \dots \ \mathbf{s}^{(L)}\}$ is the set of all possible symbol vectors and L is the cardinality of this set. The $N \times n_T$ complex matrix-valued function $\mathbf{X}(\mathbf{s})$ is called OSTBC [13] if it satisfies that

- (1) all the entries of $\mathbf{X}(\mathbf{s})$ are linear functions of the Λ complex variables $s_1, s_2, \dots, s_\Lambda$ and their complex conjugates;
- (2) for any arbitrary \mathbf{s} in $\mathbb{C}^{\Lambda \times 1}$,

$$\mathbf{X}^H(\mathbf{s})\mathbf{X}(\mathbf{s}) = \|\mathbf{s}\|^2 \mathbf{I}_{n_T}, \quad (19)$$

where \mathbf{I}_{n_T} is the identity $n_T \times n_T$ matrix, $\|\cdot\|$ is the Euclidean norm, and $(\cdot)^H$ denotes Hermitian transpose.

3.2. Performance analysis

In order to estimate the channel matrix \mathbf{H} , it should be emphasized that in any statistical expectation below, the matrix \mathbf{H} is treated as random; at the same time, any estimator of \mathbf{H} is supposed to obtain an estimate of a particular realization of this random matrix that corresponds to the current block of the received data.

In the conventional ST estimation technique for MIMO systems [9], a known training $N \times n_T$ matrix \mathbf{C} , is added arithmetically to the data $N \times n_T$ matrix \mathbf{B} during every block transmitted. In this way, the transmitted signal matrix \mathbf{X} can be expressed as $\mathbf{X} = \mathbf{B} + \mathbf{C}$. It is clear that the total transmitted power is distributed between the data and the training signals, that is, $P_x = P_b + P_c$.

The ST system is depicted in Figure 2, where a signal matrix \mathbf{X} is transmitted over the radio MIMO channel with the channel matrix \mathbf{H} , and distorted with the noise matrix \mathbf{N} .

Based on the received signal \mathbf{R} , the MRST delivers an estimate of the MIMO channel matrix, denoted as $\hat{\mathbf{H}}$; subsequently the decoder obtains an estimate of \mathbf{X} , denoted as $\hat{\mathbf{X}}$, and an estimate of the mean of the data, $\hat{\mathbf{b}}$, which can be used by the channel estimator block, in an iterative way, to provide the decoder with a better estimate of \mathbf{H} .

Now, the task of this channel estimation algorithm is to recover the channel matrix \mathbf{H} based on the knowledge of \mathbf{R} and \mathbf{C} . Assuming flat-frequency channel conditions then all the row vectors of the training matrix \mathbf{C} can be equal. Hence, a time-domain estimator based on the synchronized

averaging of the received signal can be implemented. The time-average of the signal \mathbf{R} is given by

$$\begin{aligned} \mathbf{v} &= E\{\mathbf{R}\} = \left(\frac{1}{N}\right)\mathbf{u}\mathbf{R} \\ &= \left[\left(\frac{1}{N}\right)(\mathbf{u}\mathbf{B} + \mathbf{u}\mathbf{C})\right]\mathbf{H} + \left(\frac{1}{N}\right)\mathbf{u}\mathbf{N} \\ &= [\tilde{\mathbf{b}} + \mathbf{c}]\mathbf{H} + \tilde{\mathbf{n}}, \end{aligned} \quad (20)$$

where \mathbf{v} is a $1 \times n_R$ column vector, \mathbf{u} is the unit $1 \times N$ vector, \mathbf{c} is the $1 \times n_T$ training vector repeated N times and $\tilde{\mathbf{b}}$, $\tilde{\mathbf{n}}$ represent the time-average of the data matrix \mathbf{B} and noise matrix \mathbf{N} , respectively, for every block transmitted.

Using the LS approach [14, 15], an estimate of \mathbf{H} can be obtained as

$$\hat{\mathbf{H}}_{\text{LS}} = (\mathbf{c}^H\mathbf{c})^{-1}\mathbf{c}^H\mathbf{v} = \mathbf{c}^\dagger\mathbf{v}, \quad (21)$$

where $(\mathbf{c}^H\mathbf{c})^{-1}\mathbf{c}^H$ is the pseudoinverse of \mathbf{c} .

We will use the following transmitted training power constraint

$$\mathbf{c}^H\mathbf{c} = P_c\mathbf{I}, \quad (22)$$

where \mathbf{I} is the $n_T \times n_T$ identity matrix; now the channel estimate is

$$\hat{\mathbf{H}}_{\text{LS}} = \mathbf{H} + \left(\frac{1}{P_c}\right)\mathbf{c}^H(\tilde{\mathbf{b}}\mathbf{H} + \tilde{\mathbf{n}}). \quad (23)$$

It is clear that the average $\tilde{\mathbf{b}}$ over the data signal \mathbf{B} represents an extra term for the channel estimate. This average is exploited by the DDST method at the transmitter or by the MRST method at the receiver.

In order to obtain the MSE achieved by MRST method, we use the performance analysis of the DDST method, explicitly we use $\tilde{\mathbf{b}} = (1/N)\mathbf{u}\mathbf{B}$ and define the perturbation matrix $\mathbf{E} = -\mathbf{u}^T\tilde{\mathbf{b}}$ that will be arithmetically added to the data signal \mathbf{B} every transmitted block. Hence, the transmitted signal is $\mathbf{X} = \mathbf{B} + \mathbf{E} + \mathbf{C}$ and the corresponding received signal is $\mathbf{R} = (\mathbf{B} + \mathbf{E} + \mathbf{C})\mathbf{H} + \mathbf{N}$, then from (21) we obtain

$$\hat{\mathbf{H}}_{\text{LS}} = \mathbf{H} + \left(\frac{1}{P_c}\right)\mathbf{c}^H\tilde{\mathbf{n}}. \quad (24)$$

Considering (22) under optimal training and $E\{\text{tr}(\mathbf{N}^H\mathbf{N})\} = \sigma_n^2 n_R$, the MSE for DDST [9] is given by

$$\text{MSE}(\hat{\mathbf{H}}_{\text{DDST}}) = \frac{\sigma_n^2 n_T n_R}{NP_c}. \quad (25)$$

Instead to take away the contribution of $\tilde{\mathbf{b}}$ at the transmitter, the MRST uses the plain ST and removes $\tilde{\mathbf{b}}$ in an iterative manner at the receiver. From (21), the channel estimation error can be expressed as

$$\begin{aligned} \hat{\mathbf{H}}_{\text{MRST}} &= \mathbf{c}^\dagger([\tilde{\mathbf{b}} + \mathbf{c}]\mathbf{H} - \hat{\mathbf{b}}\hat{\mathbf{H}} + \tilde{\mathbf{n}}) \\ &= \mathbf{H} + \left(\frac{1}{P_c}\right)\mathbf{c}^H(\tilde{\mathbf{b}}\mathbf{H} - \hat{\mathbf{b}}\hat{\mathbf{H}}) + \left(\frac{1}{P_c}\right)\mathbf{c}^H\tilde{\mathbf{n}}. \end{aligned} \quad (26)$$

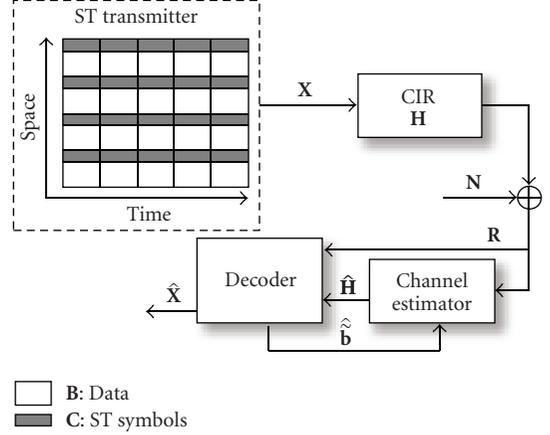


FIGURE 2: MIMO digital communication system model.

It follows that if $\hat{\mathbf{H}}\tilde{\mathbf{b}} \approx \mathbf{H}\tilde{\mathbf{b}}$, then $\hat{\mathbf{H}}_{\text{MRST}} \approx \hat{\mathbf{H}}_{\text{DDST}}$, therefore the $\text{MSE}(\hat{\mathbf{H}}_{\text{MRST}}) \approx \text{MSE}(\hat{\mathbf{H}}_{\text{DDST}})$ as it will be corroborated by the simulation results in Section 5. With respect to the iterative procedure to obtain a better estimate of $\tilde{\mathbf{b}}$, we follow the steps explained for the SISO case.

3.3. MRST with Alamouti space-time coding

Let us use MRST estimation method with the \mathcal{Q} 2-OSTBC $n_T = 2$, $n_R = 2$ system (Alamouti coding scheme). Let us assume N to be even, and

$$\mathbf{X} = \begin{bmatrix} s_1 & s_2 \\ -s_2^* & s_1^* \\ s_3 & s_4 \\ -s_4^* & s_3^* \\ \vdots & \vdots \\ s_{N-1} & s_N \\ -s_N^* & s_{N-1}^* \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} b_1 & b_2 \\ -b_2^* & b_1^* \\ b_3 & b_4 \\ -b_4^* & b_3^* \\ \vdots & \vdots \\ b_{N-1} & b_N \\ -b_N^* & b_{N-1}^* \end{bmatrix}, \quad (27)$$

$$\mathbf{C} = \begin{bmatrix} c_1 & c_2 \\ -c_2^* & c_1^* \\ c_1 & c_2 \\ -c_2^* & c_1^* \\ \vdots & \vdots \\ c_1 & c_2 \\ -c_2^* & c_1^* \end{bmatrix}$$

are the matrices of the transmitted signal, data signal, and training signal, respectively, with $(\cdot)^*$ denoting complex conjugation. The matrix's rows and columns indicate the transmission time and transmit antenna, respectively. Assuming flat-fading scenario, no cyclic prefix is required, and the signal training selection can be done choosing two symbols c_1 , c_2 and exploiting the property of orthogonally achieved with the OSTBC system, that is, the two column vectors of the matrix \mathbf{C} are orthogonal.

The estimate of the cycling mean in flat-fading scenario can be done obtaining the mean of the receive matrix \mathbf{R} , denoted as $\tilde{\mathbf{R}}$. Each element of $\tilde{\mathbf{R}}$ is given by

$$\tilde{\mathbf{R}}_{i,j} = \frac{1}{N/2} \sum_{k=1}^{N/2} \mathbf{R}_{i+2(k-1),j}, \quad (28)$$

where the indexes i, j correspond to Alamouti's emission time ($i = 1, 2$) and transmit antenna ($j = 1, 2$) respectively for this particular block coding transmission.

Let us estimate the channel matrix \mathbf{H} using (28) and assuming a noisy free scenario (without loss of generality), hence, we have

$$\mathbf{A} \text{vec}\{\mathbf{H}\} = \text{vec}\{\tilde{\mathbf{R}}\},$$

$$\text{with } \mathbf{A} = \begin{bmatrix} (c_1 - \hat{b}_1) & (c_2 - \hat{b}_2) & 0 & 0 \\ 0 & 0 & (c_1 - \hat{b}_1) & (c_2 - \hat{b}_2) \\ -(c_2 - \hat{b}_2)^* & (c_1 - \hat{b}_1)^* & 0 & 0 \\ 0 & 0 & -(c_2 - \hat{b}_2)^* & (c_1 - \hat{b}_1)^* \end{bmatrix}, \quad (29)$$

and $\text{vec}\{\cdot\}$ is the vectorization operator stacking all rows of a matrix on top of each other. Therefore, the channel estimate is given by

$$\text{vec}\{\hat{\mathbf{H}}\} = \mathbf{A}^{-1} \text{vec}\{\tilde{\mathbf{R}}\}. \quad (30)$$

Because matrix \mathbf{A} is unitary, then (30) can be rewritten as

$$\text{vec}\{\hat{\mathbf{H}}\} = \mathbf{A}^H \text{vec}\{\tilde{\mathbf{R}}\}, \quad (31)$$

and finally,

$$\hat{\mathbf{H}} = [\text{vec}\{\mathbf{I}_{n_R}\}^T \otimes \mathbf{I}_{n_T}] [\mathbf{I}_{n_R} \otimes \text{vec}\{\hat{\mathbf{H}}\}], \quad (32)$$

where \mathbf{I}_{n_T} and \mathbf{I}_{n_R} are the $n_T \times n_T$ and $n_R \times n_R$ identity matrix, respectively, and \otimes denotes the Kronecker product. The matrix $\hat{\mathbf{H}}$ represents the new MIMO channel estimate used in the iterative procedure to get a new estimate of the mean of the data-bearing $\hat{\mathbf{b}}$.

The key idea of this implementation is the way to remove the mean at the receiver. Instead of subtracts the mean of the data, from the received matrix \mathbf{R} , we incorporate the mean estimate in the matrix \mathbf{A} and use the Alamouti decoding procedure to estimate the channel matrix.

4. SIMULATION AND RESULTS FOR SISO SYSTEMS

Equalization is a well-known technique used to combat intersymbol interference (ISI) whereby the receiver attempts to compensate for the effects of the channel of the transmitted symbols. An equalizer attempts to determine the

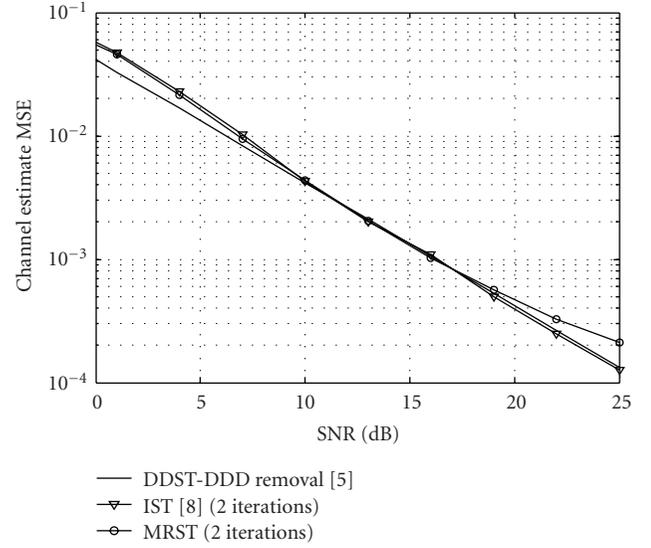


FIGURE 3: Channel estimate MSE with MMSE equalizer.

transmitted data from the received distorted symbols using an estimate of the channel that caused the distortions. In this contribution, we consider two types of equalizers widely used in communications systems: the MMSE and MLSE equalizers. The MMSE has lesser complexity and performance in terms of BER than the MLSE equalizer that is optimal for ISI.

4.1. SISO system using MMSE equalizer

We considered a time-invariant random three-tap frequency-selective Rayleigh fading channel $h(k)$ with $h(0) \neq 0$ and $h(2) \neq 0$. The channel coefficients were complex Gaussian, i.i.d. with unit variance, rescaled to achieve unitary mean energy. The sequence $\{b(k)\} \in \{-1, +1\}$ is an R.V with uniform p.d.f. and variance σ_b^2 . The parameters σ_b^2 and σ_c^2 are chosen for ST and DDST independently such that $\sigma_{b+e}^2 + \sigma_c^2 = 1$. The training to information power ratio $\sigma_c^2 / \sigma_{b+e}^2$ was set to -6.9798 dB with $P = 7$. The block length is fixed to $N = 420$ and a cyclic prefix of length P added at the beginning of each block in both ST and DDST methods. The channel estimated is used to design an MMSE equalizer of length 11 and equalization delay $d = 7$. All simulations that were run until 1000 blocks with errors were found.

Figure 3 shows the channel estimate MSE obtained from the three iterative procedures considered (a) DDST with DDD removal exposed in [11], (b) IST scheme introduced in [7], and (c) MRST method presented here. It can be observed that IST and MRST have a significant approach to DDST (from 10 dB to 20 dB) just after two iterations.

Figure 4 shows the BER performance comparison where for 2 iterations, all schemes show similar behavior for low and medium SNR levels; DDST exhibits a slight advantage for high SNRs because it achieves a better channel estimate at these levels as is shown in Figure 3.

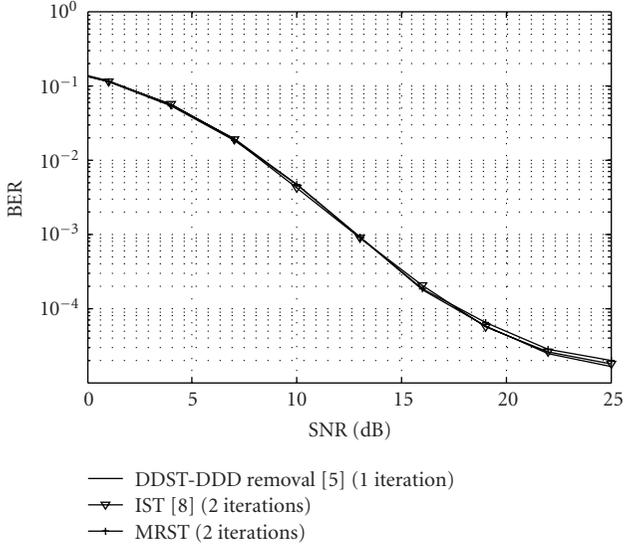


FIGURE 4: BER performance of BPSK signals with MMSE equalizer.

4.2. SISO system using MLSE equalizer

In order to get a better estimate ($\hat{\mathbf{B}}$) of $\tilde{\mathbf{B}}$ for both IST and MRST schemes, we use an MLSE equalizer with traceback length of 11. To be more specific, this equalizer comprises a Viterbi algorithm, which finds the most likely data sequence transmitted. Although, to perform close to ideal MLSE, the equalizer requires traceback lengths of the order of 5-6 times the ISI span, we chose the traceback length of 11, that is, the same number of taps of the MMSE equalizer used in Section 4.1.

Due to the fact that DDST-DDD removal [11] works with the equalizer output before proceeding with a new hard-decision process, that is, equalization and detection stages are carried out separately, it cannot exploit the benefits of the MLSE equalizer. In order to have similar conditions for the DDST method in the decoding procedure, firstly we used the MMSE equalizer, secondly we removed the data distortion, and finally we used the MLSE equalizer to obtain the data symbols. Figure 5 shows the BER performance of these three methods and the conventional ST method.

Clearly, we observe the benefits of use IST or MRST coupling with an MLSE equalizer, where both schemes outperform DDST-DDD removal when it only uses the MLSE equalizer. However, there is no noticeable performance difference when DDST uses the concatenation of the MMSE and MLSE equalizers.

Figure 6 shows the block error rate (BLER) performance. BLER is the statistical measurement of the ratio of the number of blocks with error received to the total number of blocks transmitted and it is part of the performance requirements of 3GPP test. We observe that there is no noticeable performance difference between the MRST and IST with one iteration, and DDST-DDD removal using the equalizer concatenation and removing the data distortion.

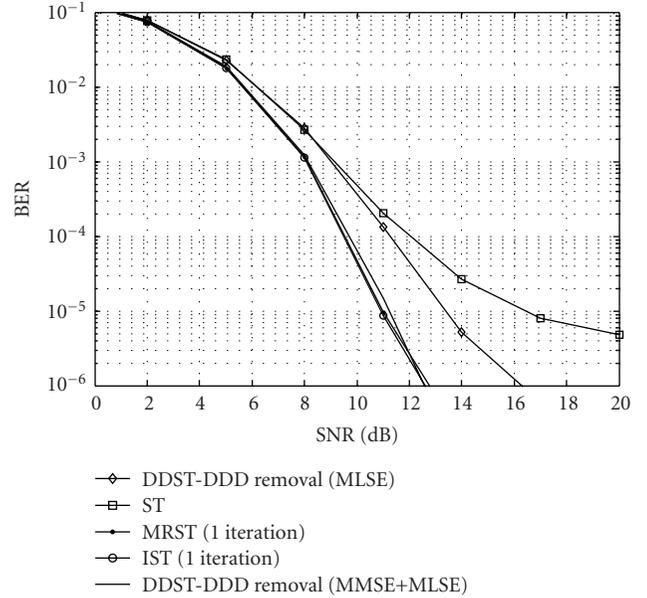


FIGURE 5: BER performance of BPSK signals with MLSE equalizer.

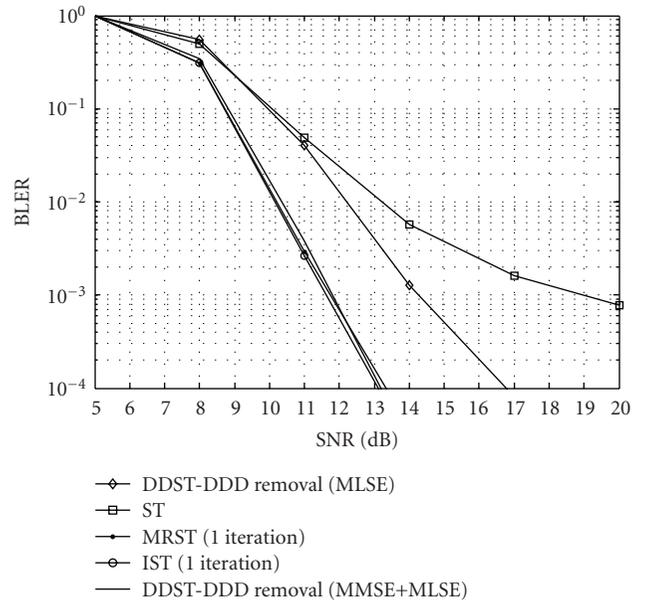


FIGURE 6: BLER performance of BPSK signals with MLSE equalizer using the three iterative methods and ST.

5. SIMULATION AND RESULTS FOR MIMO SYSTEMS

In what follows, we illustrate the performance of the proposed scheme on MIMO systems working in a flat-fading scenario. We use the \mathcal{G}_2 -OSTBC system, $n_T = n_R = 2$ transmit and receive antennas, respectively, with ideally uncorrelated elements. The block length is fixed to $N = 256$ symbols, and all simulations were run until 1000 blocks were in error. The BER is represented as a function of the average SNR, where $N_0 = E_s n_T / \text{SNR}$, and E_s is the average energy per symbol.

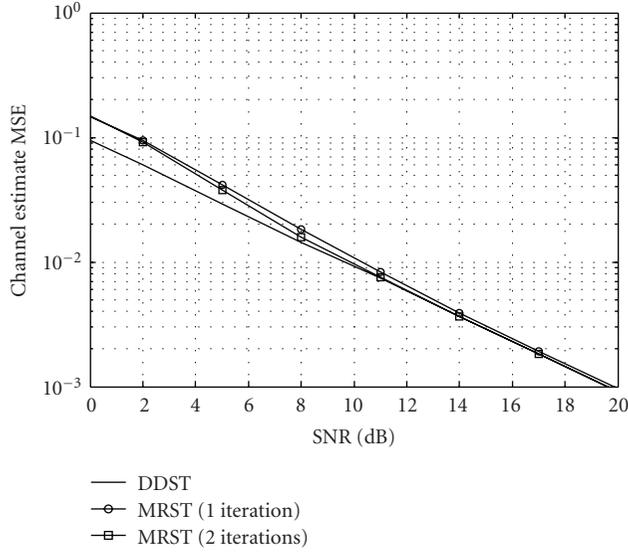
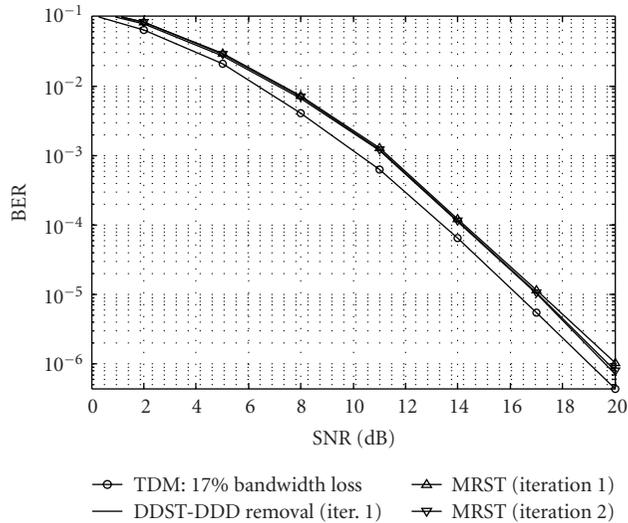
FIGURE 7: Channel estimate MSE of $\mathcal{G}2$ -OSTBC 2×2 .

FIGURE 8: BER comparisons for various channel estimation techniques.

A QPSK symbol constellation is used, and the power transmitted for each antenna is normalized to one, that is, $P_x = 1$ [Watt].

Figure 7 depicts the channel estimate MSE comparisons for DDST and the MRST techniques with one and two iterations; both of them use $P_c = 17\%$ of the power transmitted P_x , that correspond to an approximate upper bound of the region with the best BER shown in Figure 10. Making an analysis of this performance, we can realize that the proposed scheme starts to approach the DDST performance from 10 [dB]. The MRST plot for the second iteration reaches the best approach. Note that this benefit comes at the price of decoding complexity.

Figure 8 shows the BER performance comparison between TDM with 17% bandwidth loss, DDST with DDD

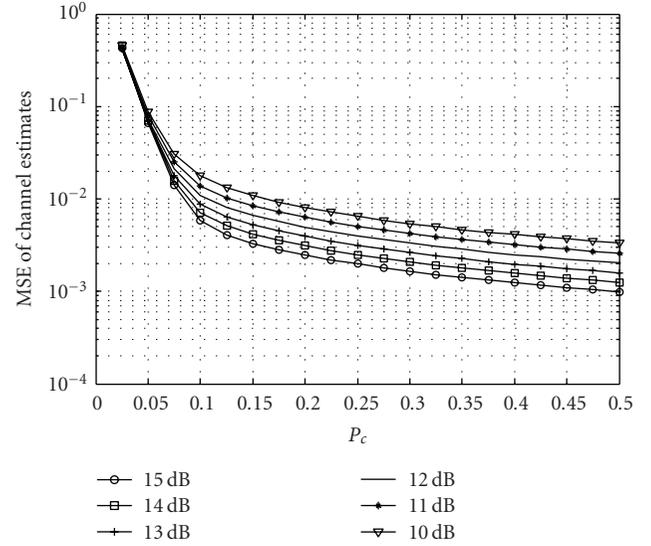


FIGURE 9: MSE of channel estimates versus power fraction for training signal.

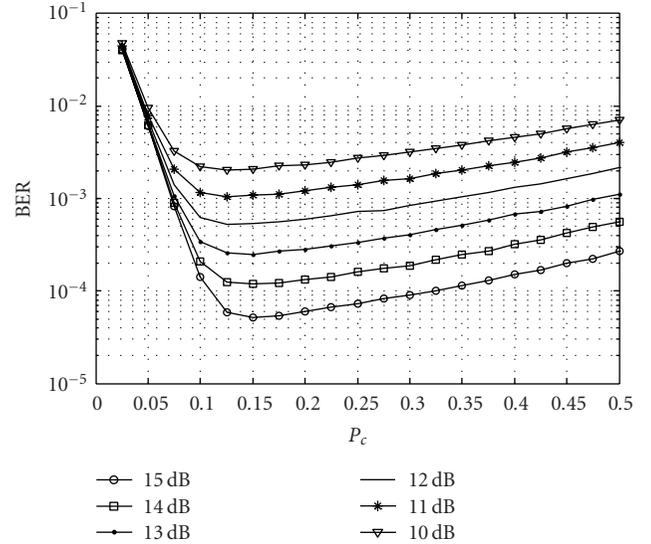


FIGURE 10: BER versus power fraction for training signal.

removal and the MRST algorithm. We observe that DDST and MRST plots with one or two iterations practically achieved the same performance. TDM-based channel estimation attained the best BER performance at the expenses of 17% bandwidth loss.

Figure 9 depicts the MSE of MRST versus P_c when SNR = 15 dB. For example, with $P_c = 0.125$, we have that $MSE(P_c) = 4[0.0316/(0.125 \times 256)] = 0.004031$. A better MSE performance is directly proportional to the training power used.

Figure 10 plots the BER of MRST versus P_c from SNR = 10 dB to 15 dB. Note that the minimum is achieved in the range where $0.1 < P_c \leq 0.17$ approximately. Because the BER achievement is the key and critical factor in digital communication systems, we worked in the region with the

best BER in the selection of the training power assigned. This selection is only a guide or suggestion for this type of radio propagation conditions at these particular SNR levels. Similar training power range was chosen for DDST in [9], but using a spatial multiplexing system. A more detailed analysis of the training power allocation problem is shown in [16]. Particularly, the simulations showed in Figures 9 and 10 were run until 10 000 blocks were in error.

6. CONCLUSION

low-complexity iterative superimposed training schemes that work jointly with equalization stage can offer similar or better performance than the iterative symbol-by-symbol detection DDST with DDD removal. The MRST introduced in this work effectively compensates the data dependent distortion that ST is unable to deal with, but at the receiver side. This leads to communication systems that perform similar to DDST but without suffering its drawbacks. Furthermore, MRST can be successfully applied to both SISO and MIMO systems, and in cooperation with the most widely used equalizers. Particularly, for the SISO case, MRST shows similar performance like the previous proposed method IST, but it avoids the inverse matrix computation $(\mathbf{C} + \hat{\mathbf{B}})^{-1}$ that every iteration IST does. Although the results of both iterative show insignificant gaps in the performance of BER and MSE, MRST is preferred by its less complexity hardware implementation.

For the MIMO case, the performance results are closely similar, that is, the performance of DDST is attained with the proposed method. Additionally, iterative ST methods can still be applied to time-varying channels, while DDST based system does not. This is true because the concept of cyclic mean in DDST is meaningless due to the fact that every symbol of the block transmitted is distorted in different way by the channel.

ACKNOWLEDGMENTS

The authors thank the reviewers for their careful reading of the manuscript and their constructive comments. This work was supported by INTEL research grants DCIT2006 and CERMIMO2008, and CONACYT research grants 47909-Y and 84559-Y.

REFERENCES

- [1] L. Tong, B. M. Sadler, and M. Dong, "Pilot-assisted wireless transmissions: general model, design criteria, and signal processing," *IEEE Signal Processing Magazine*, vol. 21, no. 6, pp. 12–25, 2004.
- [2] B. Farhang-Boroujeny, "Pilot-based channel identification: proposal for semi-blind identification of communication channels," *Electronics Letters*, vol. 31, no. 13, pp. 1044–1046, 1995.
- [3] A. G. Orozco-Lugo, M. M. Lara, and D. C. McLernon, "Channel estimation using implicit training," *IEEE Transactions on Signal Processing*, vol. 52, no. 1, pp. 240–254, 2004.
- [4] M. Ghogho, D. C. McLernon, E. Alameda-Hernandez, and A. Swami, "Channel estimation and symbol detection for block transmission using data-dependent superimposed training," *IEEE Signal Processing Letters*, vol. 12, no. 3, pp. 226–229, 2005.
- [5] T. Whitworth, M. Ghogho, and D. C. McLernon, "Data identifiability for data-dependent superimposed training," in *Proceedings of the IEEE International Conference on Communications (ICC '07)*, pp. 2545–2550, Glasgow, Scotland, June 2007.
- [6] X. Meng and J. K. Tugnait, "Semi-blind channel estimation and detection using superimposed training," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04)*, vol. 4, pp. 417–420, Montreal, Canada, May 2004.
- [7] S. M. Moosvi, D. C. McLernon, E. Alameda-Hernandez, A. G. Orozco-Lugo, and M. M. Lara, "A low complexity iterative channel estimation and equalisation scheme for (data dependent) superimposed training," in *Proceedings of the 14th European Signal Processing Conference (EUSIPCO '06)*, Florence, Italy, September 2006.
- [8] M. Ghogho, D. C. McLernon, E. Alameda-Hernandez, and A. Swami, "SISO and MIMO channel estimation and symbol detection using data-dependent superimposed training," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, vol. 3, pp. 461–464, Philadelphia, Pa, USA, March 2005.
- [9] M. Ghogho and A. Swami, "Channel estimation for MIMO systems using data-dependent superimposed training," in *Proceedings of the 42nd Annual Allerton Conference on Communication, Control and Computing*, Monticello, Ill, USA, September-October 2004.
- [10] S. He, J. K. Tugnait, and X. Meng, "On superimposed training for MIMO channel estimation and symbol detection," *IEEE Transactions on Signal Processing*, vol. 55, no. 6, part 2, pp. 3007–3021, 2007.
- [11] E. Alameda-Hernandez, D. C. McLernon, A. G. Orozco-Lugo, M. M. Lara, and M. Ghogho, "Frame/training sequence synchronization and DC-offset removal for (data-dependent) superimposed training based channel estimation," *IEEE Transactions on Signal Processing*, vol. 55, no. 6, part 1, pp. 2557–2569, 2007.
- [12] A. B. Gershman and N. D. Sidiropoulos, *Space-Time Processing for MIMO Communications*, John Wiley & Sons, New York, NY, USA, 2005.
- [13] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time block codes from orthogonal designs," *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1456–1467, 1999.
- [14] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, vol. 1 of *Prentice Hall Signal Processing Series*, Prentice Hall, Upper Saddle River, NJ, USA, 1993.
- [15] M. Biguesh and A. B. Gershman, "Training-based MIMO channel estimation: a study of estimator tradeoffs and optimal training signals," *IEEE Transactions on Signal Processing*, vol. 54, no. 3, pp. 884–893, 2006.
- [16] J. K. Tugnait and X. Meng, "On superimposed training for channel estimation: performance analysis, training power allocation, and frame synchronization," *IEEE Transactions on Signal Processing*, vol. 54, no. 2, pp. 752–765, 2006.

Research Article

Block Iterative/Adaptive Frequency-Domain Channel Estimation for Cyclic-Prefixed Single-Carrier Broadband Wireless Systems

Jong-Seob Baek¹ and Jong-Soo Seo²

¹Channel Laboratory, Samsung Electronics Co. Ltd., Suwon 443-742, South Korea

²Department of Electrical and Electronic Engineering, Yonsei University, Seoul 120-749, South Korea

Correspondence should be addressed to Jong-Seob Baek, blackgachi@yonsei.ac.kr

Received 27 April 2008; Revised 13 July 2008; Accepted 11 September 2008

Recommended by Fred Daneshgaran

This paper presents a new block iterative/adaptive frequency-domain channel estimation scheme, in which a channel frequency response (CFR) is estimated iteratively by the proposed weighted element-wise block adaptive frequency-domain channel estimation (WEB-CE) scheme using the soft information obtained by a soft-input soft-output (SISO) decoder. In the WEB-CE, an equalizer coefficient is calculated by minimizing a weighted conditional squared-norm of the a posteriori error vector with respect to its correction term. Simulation results verify the superiority of the WEB-CE in a time-varying typical urban (TU) channel.

Copyright © 2008 J.-S. Baek and J.-S. Seo. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Cyclic-prefixed single-carrier frequency-domain equalization (SC-FDE) has received enormous attention in recent years because of its efficient implementation and low peak-to-average power ratio (PAPR) characteristics over broadband wireless channels. On the other hand, it is noticed that most works have been performed with the assumption that a channel frequency response (CFR) is perfectly known to the receiver [1, 2]. In practice, this assumption is not valid since a real channel is unknown and time-varying. This, it highlights the need for the precise estimation of CFR. In [3, 4], an adaptive channel estimation scheme using the hard information from the decision was presented. In [5], an iterative (nonadaptive) channel estimation scheme using the soft information obtained from a soft-input soft-output (SISO) decoder was presented. Recently, an iterative/adaptive scheme which performs a channel estimation iteratively by employing an adaptive algorithm using the soft information was proposed in [6]. This work concludes that the iterative/adaptive approach is adequate to support a good channel tracking performance over time-varying channels. However, the scheme has not been applied to the SC-

FDE. In this correspondence, an iterative/adaptive channel estimation scheme for the SC-FDE is studied. Moreover, a new block-type channel estimation scheme is studied in order to provide a better channel tracking performance.

In this paper, we propose a block iterative/adaptive CFR estimation scheme, in which a weighted element-wise block adaptive frequency-domain channel estimation (WEB-CE) using the soft information obtained from the SISO decoder is presented. It is found that the WEB-CE has a flexibility over the element-wise block length as compared to an approximated recursive least square (RLS)-CE of [6] by applying a weighted conditional least-square (LS) criterion formulated with the *a posteriori error* vector [7]. Moreover, mean square error (MSE) of the WEB-CE with respect to the element-wise block length is analyzed. Simulation results show that the WEB-CE yields good performance in a typical urban (TU) channel as the iteration number and element-wise block length increase.

The paper is organized as follows. The next section describes the system model. In Section 3, the derivation procedure and property of the proposed CFR estimation scheme are discussed. Simulation results are discussed in Section 4, and conclusions are drawn in Section 5.

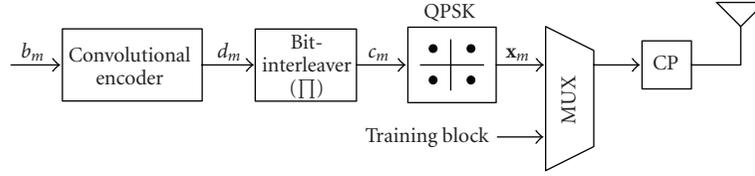


FIGURE 1: Block diagram of SC-FDE transmitter.

Notation

$E\{\cdot\}$ and $(\cdot)^T$ denote an expectation and a transpose operator, respectively. $(\cdot)^*$ denotes complex conjugates of a symbol, vector, and matrix. \mathbf{I}_N , $\mathbf{0}_N$, and $\text{diag}(\cdot)$ denote the identity matrix of size N , zero matrix of size N , and diagonal matrix, respectively.

2. SC-FDE SYSTEM MODEL

2.1. SC-FDE transmitter

The transmitter of SC-FDE is shown in Figure 1. The blocks of data bits $b_m \in \{0, 1\}$ are encoded by the convolutional encoder. The code bits d_m are interleaved by an interleaver $\Pi(\cdot)$, $c_m = \Pi(d_m)$. QPSK mapper maps a pair of input bits to a symbol x_m from the symbol alphabet \mathcal{S} . (In this paper, a constant-amplitude PSK constellation is assumed.) Then, a block of signals $\mathbf{x}_n = [x_{n,0} \cdots x_{n,N-2} x_{n,N-1}]^T$ is structured with block length of N and a cyclic-prefix (CP) is inserted between blocks to prevent an interblock interference (IBI) at block time n . A training block (TB) that has block length of N is inserted every data frame that consists of data blocks in order to initialize the channel coefficients and track the channel variations.

2.2. SC-FDE receiver

Suppose that an overall channel impulse response (CIR) containing the combined effects of transmit and receive filters, multipath fading and sampling is $\mathbf{h}_n = [h_{n,0} \cdots h_{n,L-2} h_{n,L-1}]^T$, where L is the total length of the CIR, at a block time n , and it is constant within one block. It is also supposed that the CP length is equal or larger than $L - 1$. At the receiver, discarding the CP, the time-domain received block \mathbf{y}_n can be expressed in a matrix form as

$$\mathbf{y}_n = \mathbf{H}_n^c \mathbf{x}_n + \mathbf{w}_n, \quad (1)$$

where $\mathbf{w}_n = [w_{n,0} \cdots w_{n,N-2} w_{n,N-1}]^T$ is an additive white noise vector, and \mathbf{H}_n^c is a circular matrix. It is assumed that $E\{\mathbf{x}_n \mathbf{x}_n^*\} = \sigma_x^2 \mathbf{I}_N$, $E\{\mathbf{x}_n \mathbf{w}_n^*\} = \mathbf{0}_N$, and $E\{\mathbf{w}_n \mathbf{w}_n^*\} = \sigma_w^2 \mathbf{I}_N$, where σ_x^2 and σ_w^2 denote the signal power and noise variance, respectively. Then, the discrete-time Fourier transform (DFT) matrix $\mathbf{Q} = (1/\sqrt{N})e^{-j2\pi il/N}$ (for $0 \leq i, l \leq N - 1$) yields the frequency-domain block as follows:

$$\mathbf{Y}_n = \mathbf{H}_n \mathbf{X}_n + \mathbf{W}_n, \quad (2)$$

where $\mathbf{Y}_n = \mathbf{Q} \mathbf{y}_n = [Y_n[0] \cdots Y_n[N-2] Y_n[N-1]]^T$, $\mathbf{X}_n = \mathbf{Q} \mathbf{x}_n = [X_n[0] \cdots X_n[N-2] X_n[N-1]]^T$, $\mathbf{W}_n = \mathbf{Q} \mathbf{w}_n = [W_n[0] \cdots W_n[N-2] W_n[N-1]]^T$, and $\mathbf{H}_n = \mathbf{Q} \mathbf{H}_n^c \mathbf{Q}^* = \text{diag}(H_n[0], \dots, H_n[N-2], H_n[N-1])$, where $H_n[i]$ represents the channel frequency response at the i th frequency bin.

From (2), the frequency-domain estimate of the transmitted data block, $\hat{\mathbf{X}}_n = [\hat{X}_n[0] \cdots \hat{X}_n[N-2] \hat{X}_n[N-1]]^T$, is performed by employing FDE at a block time n . In this paper, a zero-forcing FDE (ZF-FDE) is considered. (Note that the minimum-mean-square error (MMSE) equalization can be used in place of the ZF equalization.) The received data at the l th frequency bin is described as

$$\hat{X}_n[l] = \frac{Y_n[l]}{H_n[l]}, \quad 0 \leq l \leq N - 1. \quad (3)$$

After the inverse DFT (IDFT) operation, an extrinsic log-likelihood ratio (LLR) of a coded bit is computed [8]. The extrinsic LLR value is de-interleaved and fed to an SISO decoder. Then, the SISO decoder outputs the a priori LLR, which is interleaved to compute the soft information, that is, a priori mean vector $E\{\mathbf{x}_n\} = [\bar{x}_{n,0} \cdots \bar{x}_{n,N-2} \bar{x}_{n,N-1}]^T$, in which $\bar{x}_{n,i}$ is defined as [8]

$$\bar{x}_{n,i} = \sum_{s \in \mathcal{S}} s \cdot P(x_{n,i} = s), \quad (4)$$

where s denotes a symbol \mathcal{S} . Correspondingly, the DFT of the soft information such as the a priori mean vector is given by

$$\begin{aligned} \bar{\mathbf{X}}_n &= \mathbf{Q} E\{\mathbf{x}_n\} \\ &= [\bar{X}_n[0] \cdots \bar{X}_n[N-2] \bar{X}_n[N-1]]^T. \end{aligned} \quad (5)$$

The soft information is used to estimate $H_n[l]$ in (3).

3. ITERATIVE/ADAPTIVE CFR ESTIMATION SCHEME

3.1. WEB-CE using the soft information

The coefficient update algorithm of the WEB-CE at l th frequency bin is expressed as [7]

$$\hat{H}_n[l] = \hat{H}_{n-1}[l] + \Delta \hat{H}_n[l], \quad (6)$$

where $\Delta\hat{H}_n[l]$ is the correction term of $\hat{H}_{n-1}[l]$. From (2), we define $1 \times B$ input vectors $\hat{H}_n[l]$ at a block time n as follows:

$$\begin{aligned} \mathbf{Y}_n[l] &= [Y_n[l] \ Y_{n-1}[l] \ \cdots \ Y_{n-B+1}[l]], \\ \mathbf{X}_n[l] &= [X_n[l] \ X_{n-1}[l] \ \cdots \ X_{n-B+1}[l]], \\ \mathbf{W}_n[l] &= [W_n[l] \ W_{n-1}[l] \ \cdots \ W_{n-B+1}[l]], \end{aligned} \quad (7)$$

where B is the element-wise block length of the WEB-CE. In (7), it is noticed that $\mathbf{X}_n[l]$ may actually not be known to the receiver, thus $\mathbf{X}_n[l]$ is alternatively defined as [6]

$$\mathbf{X}_n[l] = \bar{\mathbf{X}}_n[l] + \mathbf{V}_n[l], \quad (8)$$

where $\bar{\mathbf{X}}_n[l] = [\bar{X}_n[l] \ \bar{X}_{n-1}[l] \ \cdots \ \bar{X}_{n-B+1}[l]]$ and $\mathbf{V}_n[l] = [V_n[l] \ V_{n-1}[l] \ \cdots \ V_{n-B+1}[l]]$, whose element $V_{n-k}[l]$ has zero mean and variance given by

$$E\{V_{n-k}[l]V_{n-k}^*[l]\} = E\{X_{n-k}[l]X_{n-k}^*[l]\} - \bar{X}_{n-k}[l]\bar{X}_{n-k}^*[l]. \quad (9)$$

In addition, it is assumed that $V_{n-k}[l]$ is independent of $\bar{X}_{n-k}[l]$. Furthermore, we define $1 \times B$ *a priori error* vector $\mathbf{e}_n[l]$ and a *posteriori error* vector $\bar{\mathbf{e}}_n[l]$ as follows [7]:

$$\begin{aligned} \mathbf{e}_n[l] &= \mathbf{Y}_n[l] - \hat{H}_{n-1}[l]\mathbf{X}_n[l], \\ \bar{\mathbf{e}}_n[l] &= \mathbf{e}_n[l] - \Delta\hat{H}_n[l]\mathbf{X}_n[l]. \end{aligned} \quad (10)$$

From (7)–(10), a *posteriori error* vector-based weighted conditional LS criterion is defined as [6, 7]

$$J_n[l] = E\{\bar{\mathbf{e}}_n[l]\mathbf{F}\bar{\mathbf{e}}_n^*[l] \mid \mathbf{Y}_n[l], \hat{H}_{n-1}[l], \mathbf{X}_n[l]\}, \quad (11)$$

where a squared norm of $\bar{\mathbf{e}}_n[l]$ is attenuated geometrically by $B \times B$ diagonal matrix $\mathbf{F} = \text{diag}(1, \dots, \beta^{B-1})$ with the forgetting factor β that satisfies $0 < \beta \leq 1$. Using (8) and (10), (11) can be rewritten as

$$\begin{aligned} J_n[l] &= E\left\{\bar{\mathbf{e}}_n[l]\mathbf{F}\bar{\mathbf{e}}_n^*[l] \mid \underbrace{\mathbf{e}_n[l], \Delta\hat{H}_n[l], \mathbf{V}_n[l], \bar{\mathbf{X}}_n[l]}_{G_n}\right\} \\ &= E\{\mathbf{e}_n[l]\mathbf{F}\mathbf{e}_n^*[l] \mid G_n\} \\ &\quad + E\{\Delta\hat{H}_n[l]\mathbf{X}_n[l]\mathbf{F}\mathbf{X}_n^*[l]\Delta\hat{H}_n^*[l] \mid G_n\} \\ &\quad - E\{\Delta\hat{H}_n[l]\mathbf{X}_n[l]\mathbf{F}\mathbf{e}_n^*[l] \mid G_n\} \\ &\quad - E\{\mathbf{e}_n[l]\mathbf{F}\mathbf{X}_n^*[l]\Delta\hat{H}_n^*[l] \mid G_n\}. \end{aligned} \quad (12)$$

In (12), using (8), the second term is computed as

$$\begin{aligned} &E\{\Delta\hat{H}_n[l]\mathbf{X}_n[l]\mathbf{F}\mathbf{X}_n^*[l]\Delta\hat{H}_n^*[l] \mid G_n\} \\ &= \Delta\hat{H}_n[l]E\left\{\underbrace{(\bar{\mathbf{X}}_n[l] + \mathbf{V}_n[l])\mathbf{F}(\bar{\mathbf{X}}_n^*[l] + \mathbf{V}_n^*[l])}_{P_n^s}\right\}\Delta\hat{H}_n^*[l] \\ &= \Delta\hat{H}_n[l]P_n^s\Delta\hat{H}_n^*[l]. \end{aligned} \quad (13)$$

Using (9), P_n^s can be expressed as

$$\begin{aligned} P_n^s &= \bar{\mathbf{X}}_n[l]\mathbf{F}\bar{\mathbf{X}}_n^*[l] + E\{\mathbf{V}_n[l]\mathbf{F}\mathbf{V}_n^*[l]\} \\ &= \sum_{k=0}^{B-1} \beta^k (\bar{X}_{n-k}[l]\bar{X}_{n-k}^*[l] + E\{V_{n-k}[l]V_{n-k}^*[l]\}) \\ &= \sum_{k=0}^{B-1} \beta^k \mathbf{q}_l E\{\mathbf{x}_{n-k}\mathbf{x}_{n-k}^*\} \mathbf{q}_l^* \\ &= \sigma_x^2 \left(\frac{1 - \beta^B}{1 - \beta} \right), \end{aligned} \quad (14)$$

where \mathbf{q}_l denotes the l th row vector of \mathbf{Q} . In (12), using (8) and (10), the third term is given by

$$\begin{aligned} &E\{\Delta\hat{H}_n[l]\mathbf{X}_n[l]\mathbf{F}\mathbf{e}_n^*[l] \mid G_n\} \\ &= \Delta\hat{H}_n[l](\bar{\mathbf{X}}_n[l]\mathbf{F}\mathbf{Y}_n^*[l] + E\{\mathbf{V}_n[l]\mathbf{F}\mathbf{Y}_n^*[l]\} - P_n^s\hat{H}_{n-1}^*[l]). \end{aligned} \quad (15)$$

Furthermore, assuming $H_n \simeq \hat{H}_{n-1}$ [6], $E\{\mathbf{V}_n[l]\mathbf{F}\mathbf{Y}_n^*[l]\}$ in (15) can be expressed as

$$\begin{aligned} E\{\mathbf{V}_n[l]\mathbf{F}\mathbf{Y}_n^*[l]\} &= E\{\mathbf{V}_n[l]\mathbf{F}(\mathbf{X}_n^*[l]H_n^*[l] + \mathbf{W}_n^*[l])\} \\ &= E\{\mathbf{V}_n[l]\mathbf{F}\mathbf{V}_n^*[l]\}\hat{H}_{n-1}^*[l]. \end{aligned} \quad (16)$$

Then, (15) can be rewritten as

$$\begin{aligned} &E\{\Delta\hat{H}_n[l]\mathbf{X}_n[l]\mathbf{F}\mathbf{e}_n^*[l] \mid G_n\} \\ &= \Delta\hat{H}_n[l](\bar{\mathbf{X}}_n[l]\mathbf{F}\mathbf{Y}_n^*[l] - \bar{\mathbf{X}}_n[l]\mathbf{F}\bar{\mathbf{X}}_n^*[l]\hat{H}_{n-1}^*[l]). \end{aligned} \quad (17)$$

The computation of the fourth term in (12) can be done similarly as that of the third term as follows:

$$\begin{aligned} &E\{\mathbf{e}_n[l]\mathbf{F}\mathbf{X}_n^*[l]\Delta\hat{H}_n^*[l] \mid G_n\} \\ &= (\mathbf{Y}_n[l]\mathbf{F}\bar{\mathbf{X}}_n^*[l] - \hat{H}_{n-1}[l]\bar{\mathbf{X}}_n[l]\mathbf{F}\bar{\mathbf{X}}_n^*[l])\Delta\hat{H}_n^*[l]. \end{aligned} \quad (18)$$

Finally, substituting the results of (13), (14), (17), and (18) into (12), and then minimizing $J_n[l]$ with respect to $\Delta\hat{H}_n^*[l]$, $\Delta\hat{H}_n$ in (6) is computed:

$$\begin{aligned} \frac{\partial J_n[l]}{\partial \Delta\hat{H}_n^*[l]} &= \Delta\hat{H}_n[l]P_n^s - \mathbf{Y}_n[l]\mathbf{F}\bar{\mathbf{X}}_n^*[l] - \hat{H}_{n-1}[l]\bar{\mathbf{X}}_n[l]\mathbf{F}\bar{\mathbf{X}}_n^*[l] \\ &= \Delta\hat{H}_n[l]P_n^s - (\mathbf{Y}_n[l] - \hat{H}_{n-1}[l]\bar{\mathbf{X}}_n[l])\mathbf{F}\bar{\mathbf{X}}_n^*[l] \\ &= \Delta\hat{H}_n[l]P_n^s - \mathbf{e}_n^s[l]\mathbf{F}\bar{\mathbf{X}}_n^*[l] \\ &= 0, \end{aligned} \quad (19)$$

where $1 \times B$ *a priori soft error* vector $\mathbf{e}_n^s[l]$ is defined as

$$\mathbf{e}_n^s[l] = \mathbf{Y}_n[l] - \hat{H}_{n-1}[l]\bar{\mathbf{X}}_n[l]. \quad (20)$$

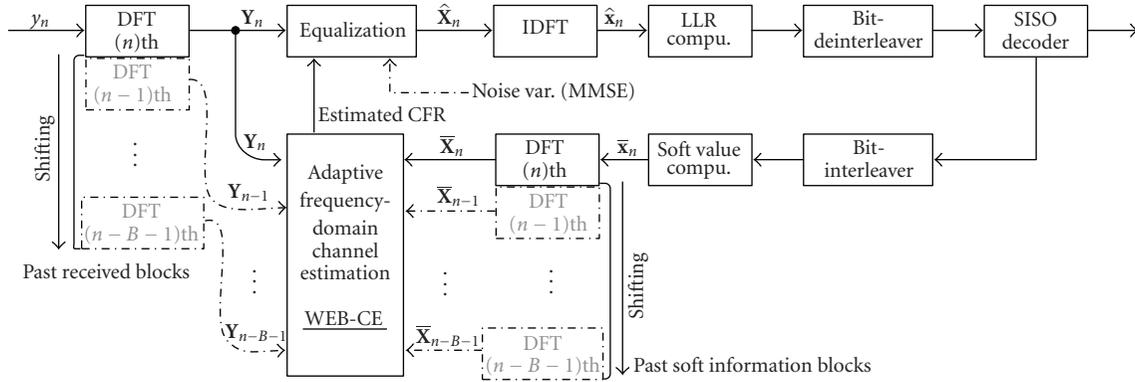


FIGURE 2: Iterative receive structure employing the WEB-CE when the iteration number is greater than one. Dotted blocks correspond to registers (or buffers) preserving the past information.

TABLE 1: Computational complexity of approximated RLS-CE and WEB-CE algorithms.

Algorithm	Overall number of complex Mul./Add.
Approximated RLS-CE	$N\log_2 N + 3N$ (Mul.) $2N\log_2 N + 2N$ (Add.)
WEB-CE	$N\log_2 N + 3BN$ (Mul.) $2N\log_2 N + 2BN$ (Add.)

From (19), $\Delta\hat{H}_n[l]$ is given by

$$\Delta\hat{H}_n[l] = \mathbf{e}_n^s[l] \mathbf{F} \bar{\mathbf{X}}_n^*[l] (P_n^s)^{-1}. \quad (21)$$

Consequently, substituting (21) into (6), we can formulate WEB-CE algorithm as follows:

$$\hat{H}_n[l] = \hat{H}_{n-1}[l] + \mathbf{e}_n^s[l] \mathbf{F} \bar{\mathbf{X}}_n^*[l] (P_n^s)^{-1}. \quad (22)$$

The overall iterative receive structure employing the WEB-CE is depicted in Figure 2.

3.2. Properties of WEB-CE

(1) *Flexibility*: In (14), (21), and (22), assuming $B = 1$, it is found that the WEB-CE is similar to the approximated RLS-CE (or least mean square (LMS)-CE) of [6] defined as

$$\hat{H}_n^{\text{RLS}}[l] = \hat{H}_{n-1}^{\text{RLS}}[l] + e_n^{\text{RLS},s}[l] \bar{X}_n^*[l] (P_n^{\text{RLS},s})^{-1}, \quad (23)$$

where $(P_n^{\text{RLS},s})^{-1} \approx (1 - \beta)/\sigma_x^2$, and $e_n^{\text{RLS},s}[l] = Y_n[l] - \hat{H}_{n-1}^{\text{RLS}}[l] \bar{X}_n[l]$. As a result, we can see that the WEB-CE has flexibility over the block length B as compared to the approximated RLS-CE.

(2) *Computational complexity*: The computational complexity of the WEB-CE and the approximated RLS-CE are compared with respect to a complex operator. We only focus on the required computation number of the channel estimators, that is, (22) and (23). It is assumed that the DFT operator requires $(N/2)\log_2 N$ complex multiplications (Mul.) and $N\log_2 N$ complex additions (Add.) per input

symbols, that is, \bar{X}_n and Y_n . It is noticed in Figure 2 that the dotted DFT blocks would not be considered in computing the computational complexity because these correspond to the register (or buffer). In the WEB-CE, the following steps are necessary for each block length of B .

- (i) $\mathbf{e}_n^s[l]$ in (22) requires B complex multiplications and B complex additions, respectively, when $Y_n[l]$ and $\bar{X}_n[l]$ are given. Considering the block length of N , this step requires $(N\log_2 N + BN)$ complex multiplications and $(2N\log_2 N + BN)$ complex additions.
- (ii) (22) requires $2B$ complex multiplications and B complex additions, respectively, where the term $(P_n^s)^{-1}$ is a constant from (14). Therefore, this step requires $2BN$ complex multiplications and BN complex additions.

The computational complexity of the approximated RLS-CE can be computed identically. As a result, the overall computational complexities of the WEB-CE and the approximated RLS-CE are shown in Table 1, which indicates that the computational load involved in the WEB-CE is linearly proportional to B .

(3) *MSE analysis*: The MSE $\varepsilon_n[l]$ of the WEB-CE with respect to B is analyzed in the Appendix. From (A.7), the MSE is given by

$$\varepsilon_n[l] = \sigma_w^2 \frac{(1 - \beta)(1 + \beta^B)}{(1 + \beta)(1 - \beta^B)}. \quad (24)$$

When $\sigma_w^2 = 0.1, 0.0316$ and 0.01 , the MSE (dB) curves versus β and B is depicted in Figure 3, where it is found that the MSE decreases exponentially as B increases for a given β .

4. SIMULATION RESULTS

We evaluate the performance of SC-FDE employing WEB-CE in the slow time-varying channels [3]. In the simulation, a recursive systematic convolutional encoder with a generator $G = [7 \ 5]$ and BCJR SISO decoding algorithm were employed. S-random interleaver with a construction $S = 0.5\sqrt{0.5 \cdot I}$ was used, where $I = 1024$ is a block size of the

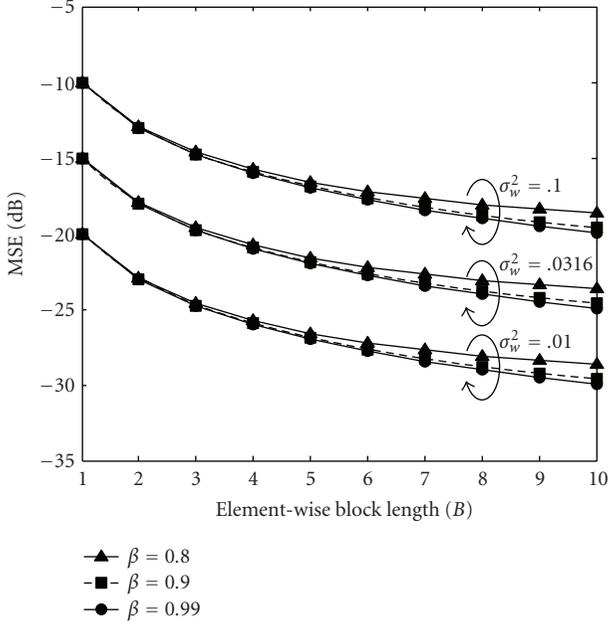


FIGURE 3: The MSE (dB) curves versus β and B when $\sigma_w^2 = 0.1, 0.0316$ and 0.01 .

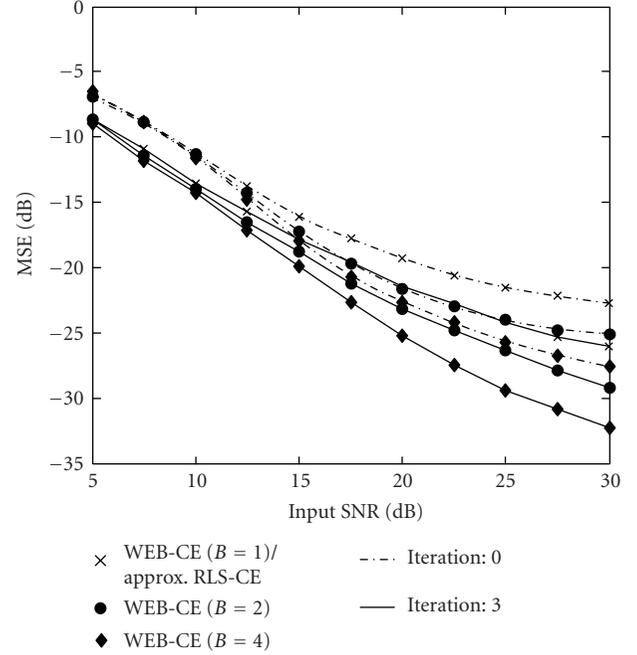


FIGURE 5: MSE performance of the WEB-CE ($B = 1$)/approx. RLS-CE, and WEB-CE ($B = 2, 4$) when $f_d = 5$ Hz.

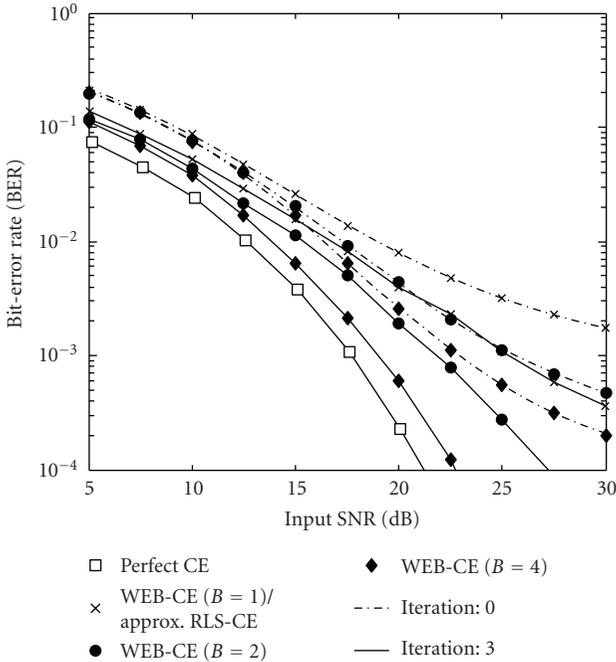


FIGURE 4: BER performance of ZF-FDE with the WEB-CE ($B = 1$)/approx. RLS-CE, and WEB-CE ($B = 2, 4$) when $f_d = 5$ Hz.

interleaver [8]. One block consists of 64 symbols ($N = 64$), where the symbol duration is $T_s = 3.69 \mu\text{s}$. A normalized root mean square (rms) delay spread $\tau = 0.2886$ and radio channel memory of 3 (sample delay) are assumed for the TU channel as in Jake’s model [7]. The CP length of 3 was set to eliminate the IBI. A data frame consists of 16 data blocks and

one TS block. In the simulations, the bit error rate (BER) was measured to evaluate the performance of the ZF-FDE with respect to B of the WEB-CE. In addition, the MSE of channel estimate, that is, MSE between the perfect CFR and the estimated CFR by the WEB-CE, was also measured.

Figure 4 depicts the BER curve of the ZF-FDE employing the WEB-CE ($B = 1, 2, 4$), where the iteration number is 0 and 3. In this figure, the maximum Doppler shift was set to $f_d = 5$ Hz and a parameter $\beta = 0.85$ was chosen through a simulation in order to allow a good tracking for the WEB-CE. In order to provide a reference BER, a perfect channel condition was also considered. Here, the approximated RLS-CE could be regarded equivalently as the WEB-CE ($B = 1$). It is readily shown that the performance of the ZF-FDE is improved as the iteration number increases. In particular, it is noticed that the WEB-CE ($B = 2, 4$) yields a significant improvement of performance as compared to the WEB-CE ($B = 1$), which is caused by the gain obtained from B (see Figure 3). It is also noticed that the error propagation due to the inaccurate past soft information [7] can be reduced as the iteration number increases.

Figure 5 depicts the corresponding MSE curve of Figure 4. It is shown that the WEB-CE ($B = 2, 4$) yields better MSE performance than the WEB-CE ($B = 1$).

Figures 6 and 7 depict the BER and MSE curves when $f_d = 10$ Hz, where the simulation parameters are the same as in Figures 4 and 5. From these figures, we can also see that the WEB-CE yields superior performance as the iteration number and B increase in the presence of a relatively fast fading. Furthermore, it is noticed from [9] that the TS blocks insertion, more often and using smaller N , can help improve the BER performance as the Doppler shift increases.

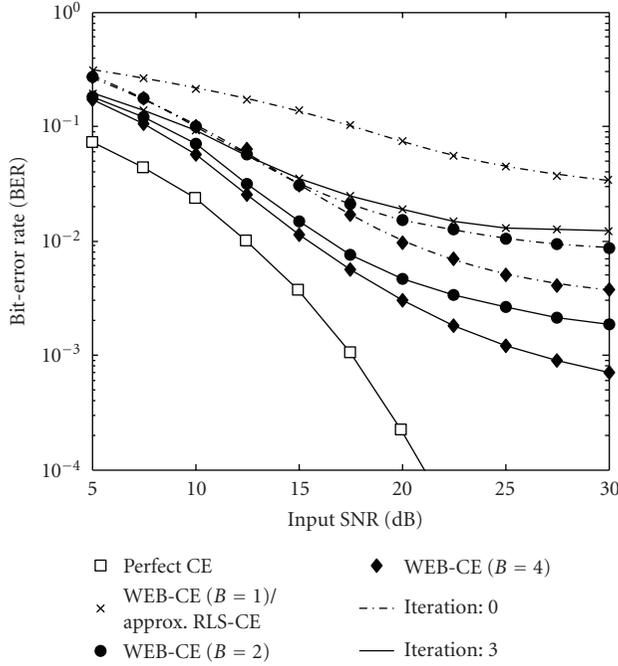


FIGURE 6: BER performance of ZF-FDE with the WEB-CE ($B = 1$)/approx. RLS-CE, and WEB-CE ($B = 2, 4$) when $f_d = 10$ Hz.

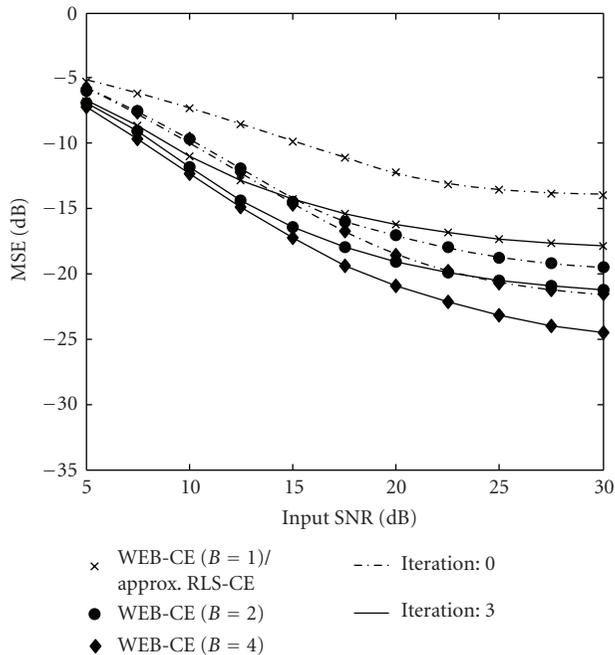


FIGURE 7: MSE performance of the WEB-CE ($B = 1$)/approx. RLS-CE, and WEB-CE ($B = 2, 4$) when $f_d = 10$ Hz.

Figure 8 depicts the performance comparison of the ZF-FDE with WEB-CE ($B = 4$) and the MMSE-FDE with WEB-CE ($B = 4$), where it is assumed that a noise variance is known to the receiver. Figure 8 shows that the MMSE-FDE provides the lower BER performance than the ZF-FDE.

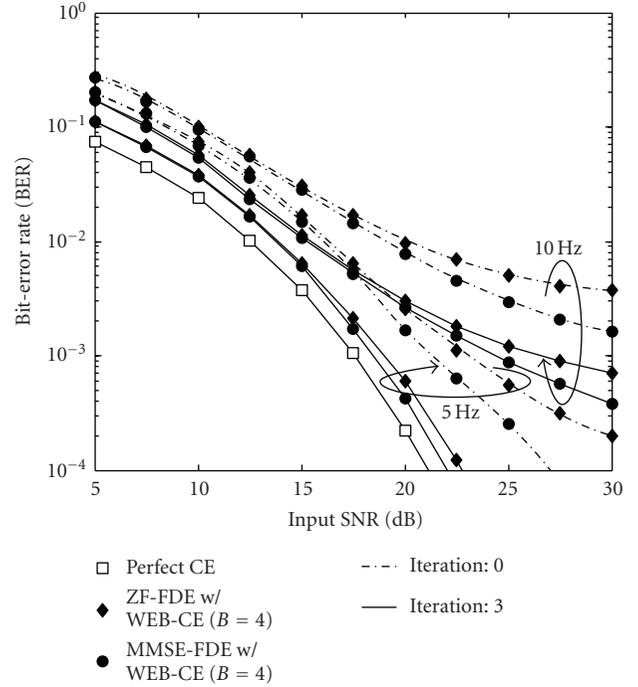


FIGURE 8: BER performance comparison of ZF-FDE w/WEB-CE ($B = 4$) and MMSE-FDE w/WEB-CE ($B = 4$) when $f_d = 5$ and 10 Hz.

5. CONCLUSIONS

In this paper, we proposed the novel block iterative/adaptive frequency-domain channel estimation named as the WEB-CE that utilizes the soft information in the time-varying channel. In the WEB-CE, the correction term of the coefficient was calculated by minimizing the weighted conditional *a posteriori* error vector-based LS criterion at each block iteration. It was found that the MSE of the WEB-CE decreases as B increases, which gives flexibility as compared to the approximated RLS-CE. From the simulation results, it was shown that the WEB-CE ($B > 1$) would be a good choice for the channel estimation scheme of the SC-FDE receiver in time-varying channels.

APPENDIX

MSE ANALYSIS OF THE WEB-CE

In (22), assuming that the channel is stationary, \mathbf{X}_n is perfectly known to the receiver and the WEB-CE is on the steady state, we analyze the MSE $\varepsilon_n[l]$ defined as

$$\begin{aligned} \varepsilon_n[l] &= E \left\{ \left(\hat{H}_n^*[l] - H_n^*[l] \right) \underbrace{\left(\hat{H}_n[l] - H_n[l] \right)}_{\Delta H_n} \right\} \\ &= E \{ \Delta H_n^* \Delta H_n \}, \end{aligned} \quad (\text{A.1})$$

where $\hat{H}_n[l]$ is defined as

$$\hat{H}_n[l] = \hat{H}_{n-1}[l] + \mathbf{e}_n[l] \mathbf{F} \mathbf{X}_n^*[l] P_n^{-1} \quad (\text{A.2})$$

with $P_n = \mathbf{X}_n[l] \mathbf{F} \mathbf{X}_n^*[l] = \sum_k \beta^k \mathbf{q}_l \mathbf{x}_{n-k} \mathbf{x}_{n-k}^* \mathbf{q}_l^*$, and $\mathbf{e}_n[l] = \mathbf{Y}_n[l] - \hat{H}_{n-1}[l] \mathbf{X}_n[l]$. Premultiplying both sides of (A.2) by P_n and using $\mathbf{e}_n[l]$, (A.2) can be rewritten as

$$\hat{H}_n[l] = \mathbf{Y}_n[l] \mathbf{F} \mathbf{X}_n^*[l] P_n^{-1}. \quad (\text{A.3})$$

Substituting $\mathbf{Y}_n[l] = H_n[l] \mathbf{X}_n[l] + \mathbf{W}_n[l]$ into (A.3), ΔH_n is expressed as

$$\Delta H_n = \mathbf{W}_n[l] \mathbf{F} \mathbf{X}_n^*[l] P_n^{-1}. \quad (\text{A.4})$$

Correspondingly, the MSE $E\{\Delta H_n^* \Delta H_n\}$ is computed as

$$E\{\Delta H_n^* \Delta H_n\} = \sigma_w^2 P_n^{-1} \underbrace{E\{\mathbf{X}_n[l] \mathbf{F}^2 \mathbf{X}_n^*[l]\}}_{P'_n} P_n^{-1}. \quad (\text{A.5})$$

Here, assuming $E\{\mathbf{x}_{n-k} \mathbf{x}_{n-k}^*\} = \mathbf{I}_N$ for all k in (1), P'_n and P_n are given by

$$\begin{aligned} P'_n &= \frac{(1 - \beta^{2B})}{(1 - \beta^2)}, \\ P_n &= \frac{(1 - \beta^B)}{(1 - \beta)}. \end{aligned} \quad (\text{A.6})$$

Finally, substituting (A.6) into (A.5), the MSE of the WEB-CE is calculated as

$$\varepsilon_n[l] = \sigma_w^2 \frac{(1 - \beta)(1 + \beta^B)}{(1 + \beta)(1 - \beta^B)}. \quad (\text{A.7})$$

ACKNOWLEDGMENTS

This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute for Information Technology Advancement) (IITA-2008-(C1090-0801-0011)).

REFERENCES

- [1] D. Falconer, S. L. Ariyavisitakul, A. Benyamin-Seeyar, and B. Eidson, "Frequency domain equalization for single-carrier broadband wireless systems," *IEEE Communications Magazine*, vol. 40, no. 4, pp. 58–66, 2002.
- [2] N. Al-Dhahir, "Single-carrier frequency-domain equalization for space-time block-coded transmissions over frequency-selective fading channels," *IEEE Communications Letters*, vol. 5, no. 7, pp. 304–306, 2001.
- [3] J. Coon, M. Sandell, M. Beach, and J. McGeehan, "Channel and noise variance estimation and tracking algorithms for unique-word based single-carrier systems," *IEEE Transactions on Wireless Communications*, vol. 5, no. 6, pp. 1488–1496, 2006.
- [4] M. Morelli, L. Sanguinetti, and U. Mengali, "Channel estimation for adaptive frequency-domain equalization," *IEEE Transactions on Wireless Communications*, vol. 4, no. 5, pp. 2508–2518, 2005.
- [5] M. Sandell, C. Luschi, P. Strauch, and R. Yan, "Iterative channel estimation using soft decision feedback," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '98)*, vol. 6, pp. 3728–3733, Sydney, Australia, November 1998.
- [6] R. Otnes and M. Tüchler, "Iterative channel estimation for turbo equalization of time-varying frequency-selective channels," *IEEE Transactions on Wireless Communications*, vol. 3, no. 6, pp. 1918–1923, 2004.
- [7] J.-S. Baek and J.-S. Seo, "A weighted element-wise block adaptive frequency-domain equalization in frequency-selective time-varying channels," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '07)*, vol. 3, pp. 1313–1316, Honolulu, Hawaii, USA, April 2007.
- [8] M. Tüchler and J. Hagenauer, "Linear time and frequency domain turbo equalization," in *Proceedings of the 53rd IEEE Vehicular Technology Conference (VTS '01)*, vol. 2, pp. 1449–1453, Rhodes, Greece, May 2001.
- [9] J.-S. Baek and J.-S. Seo, "A weighted STBC-block adaptive frequency domain equalization for single-carrier systems in frequency-selective time-varying channels," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC '07)*, pp. 1456–1460, Kowloon, Hong Kong, March 2007.

Research Article

Real-Time Communications in Large-Scale Wireless Networks

Liang Song and Dimitrios Hatzinakos

*Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto,
10 King's College Road, Toronto, ON, Canada M5S 3G4*

Correspondence should be addressed to Liang Song, lsong@ieee.org

Received 25 March 2008; Revised 13 July 2008; Accepted 17 September 2008

Recommended by Massimiliano Laddomada

There is an emerging need for realizing real-time quality of service (QoS) over multihop wireless communications in large-scale wireless networks. The applications can include wireless mesh infrastructure for broadband Internet access supporting multimedia services, visual sensor networks for surveillance, and disaster-relief networks. However, a number of challenges still exist as revealed by recent works, where the dataflow QoS performance such as throughput and end-to-end delay can degrade fast with the number of wireless hops. We propose to use large-scale cognitive networking methods to resolve the wireless multihop challenges. By the cognitive-networking concept, data packets travel along opportunistically available paths in the network with opportunistically available spectrum in every hop. Reliable end-to-end communications can be achieved for real-time services, where we show that (1) dataflow throughput can be independent of any number of wireless hops, (2) end-to-end delay and delay variance increase linearly with the number of wireless hops, and (3) delay variance decreases to zero with higher network density. These results are supported by analysis, simulations, and experiments.

Copyright © 2008 L. Song and D. Hatzinakos. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Large-scale wireless networks, for example, mobile ad hoc networks, wireless mesh networks, as well as wireless sensor networks, have been receiving significant attention in the past few years. Recently, there has been growing interest in realizing real-time quality of service (QoS) over multihop wireless transmissions in large-scale wireless networks. For example, the driving commercial applications include wireless mesh infrastructure for broadband Internet access [1], supporting Voice over Internet Protocols (VoIP) or Multimedia over IP, and wireless sensor network with real-time event detection/reporting, especially in video surveillance, that is, visual sensor networks [2].

Compared to the classical many-to-one (starlike) networks, such as cellular phone networks and WiFi- (IEEE 802.11) [3] enabled wireless local area networks (WLANs), a wireless mesh-based infrastructure for broadband access has the advantage of much lower capital and operational costs and can achieve higher service coverage. As one example, municipal WiFi mesh networks are rolling out in a number of major cities, providing wireless web surfing and email

services to general public. Considering the popularity of Internet voice and video, further developments for supporting VoIP within the wireless mesh infrastructure are highly demanded. On the other hand, visual sensor networks appear to be one of the killer applications for wireless sensor networks, since a lot of major cities have installed public-administrated surveillance cameras for various purposes. As of now, those cameras are normally attached by cables. By adopting wireless connections, there will be significant cost reduction in the installation, which can enable much denser deployment. Moreover, related applications also include the disaster-relief wireless networks, with ad hoc formation, where the real-time information delivery can be of life and death importance.

However, many challenges still exist in realizing real-time QoS over multihop wireless networks. It has been well known that dataflow throughput, end-to-end delay, and delay jitter can degrade fast when the number of wireless hops increases [4, 5]. The limitation can be primarily introduced by dynamic network-resource availabilities including both spectrum bandwidth and wireless nodes/radios. Specifically, random spectrum availability is based on wireless fading

and interferences prevailing in unlicensed bands (e.g., ISM bands); and random radio availability is due to dynamic traffic load (congestions) and other factors such as radio failure. In traditional wireless networking, the media access control (MAC) layer sets up “logic wired links” over wireless media, which assume predetermined spectrum availability. On top of it, the network layer runs ad hoc routing protocols which assume predetermined radio availability and network topology. It is difficult to have effective QoS models in the traditional network protocol stack [6], when both spectrum bandwidth and radio availabilities cannot be predetermined in large-scale wireless networks.

In order to resolve the wireless multihop limitation, there have been proposals to install multiple radios in one single wireless node [4] and design network routing protocols based on wireless link status, as well as other combined metrics [5]. Although evidence shows that improvements could be obtained, these approaches do not address real-time QoS. Moreover, existing approaches can generally suffer from the lack of scalability. Typically, the complexity of existing protocols can grow fast with the number of wireless nodes neighboring each other (higher network density), since there are a large number of links to be managed and/or tracked on every node. Other research work that investigates real-time QoS in wireless networks could be based on inappropriate assumptions for large-scale wireless networks, such as the assumption on fixed link bandwidth [7].

In this paper, we apply the methods of large-scale cognitive networking [8] to resolve the wireless multihop limitation. The concept of cognitive networking opportunistically utilizes network resources including both spectrum bandwidth and wireless node/radio availability to realize reliable communications in large-scale wireless networks. The adopted network architecture is the Embedded Wireless Interconnect (EWI) [8–11], where wireless linkage is redefined introducing *abstract wireless links*. By definition, abstract wireless links can be arbitrary functional abstractions of mutual cooperations among proximity wireless nodes, where the categories can include broadcast, unicast, multicast/anycast, and data aggregation, among others. Therefore, *wireless link modules* are designed as the units for realizing different abstract wireless links at individual wireless nodes. The architecture of EWI is then developed upon two layers, where the bottom wireless link layer supplies a library of wireless link modules; and the upper system layer can organize the modules for achieving effective application programming interface (API).

Conforming to the cognitive networking concept, both the operating spectrum and the participating nodes of an abstract wireless link is opportunistically decided based on their instantaneous availabilities. By the unicast module design in the cognitive networking, data packets can take opportunistically available paths in the network from source to destination, with opportunistically available spectrum in every hop. In comparison, dynamic changes in spectrum and radio availability can create bottlenecks on the routing path of traditional wireless networking, where such bottlenecks are all resolved by the proposed methods. Therefore, reliable

end-to-end communications can be achieved, by the opportunistic utilization of network resources. The presented unicast module design is also implemented by cognitive-radio prototyping [12]. In the performance analysis, real-time QoS metrics including throughput, expected end-to-end delay, and delay variance, are studied. The following propositions are obtained.

- (i) All the investigated real-time QoS metrics improve with larger network scale, and the complexity on individual nodes remains constant independent of any network scale. It is also shown that the performance dynamics, as indicated by end-to-end delay variance, can diminish to zero with higher network density.
- (ii) Reliable communications can be achieved over a large number of wireless hops. In principle, the throughput can be independent of the source-to-destination distance; while the expected end-to-end delay, and the delay variance, increase linearly with the source-to-destination distance.
- (iii) Considering the resource constraints, such as energy consumption or network capacity, we identify that radio transmitting power can be an effective control knob for traffic prioritization in dealing with the tradeoff between achievable QoS performance and resource consumptions.

Therefore, the two major contributions of the paper are the following: (1) the first design & implementation of real-time communications for large-scale wireless networks; (2) the QoS performance analysis of the designed real-time unicast module. In what follows, related works are surveyed in Section 2; the unicast wireless link module design and implementation are described in Section 3; the performance analysis of real-time QoS metrics is presented in Section 4; the simulation and experiment results are reported in Section 5 and 6, respectively; the conclusion is given in Section 7.

2. RELATED WORK

Related works in cognitive networking and the EWI network architecture have previously appeared in [1, 8–12]. As a pilot architectural reference model, EWI was first introduced in the application-specific studies in wireless sensor networks [10, 11]. In [9], recent research works in the cross-layer design of wireless sensor networks are surveyed, where it is suggested that EWI can be a unified design architecture. The concept of large-scale cognitive wireless networks has been presented in [8], where EWI was adopted as the network architecture. In [1, 12], the EWI is further utilized to construct a ubiquitous wireless mesh infrastructure for broadband Internet access.

The unicast design can also be related to existing research works in opportunistic routing [13–17]. In principle, opportunistic routing schemes deal with the codesign of routing and MAC protocols. However, previous research works usually deal with single-hop performance metrics, while

the contribution of this paper deals with end-to-end QoS for real-time communications. Moreover, contributions are made in large-scale cognitive networking, with a report of experiment results.

In [13], the protocol selects the next hop relay node by a slotted ACK (acknowledge) mechanism. Having successfully received a data packet, the node calculates a priority level, which is inversely proportional to the distance between the node and the packet destination. The node with the highest priority is then selected distributively as the next hop relay node. In [14, 15], a technique named GeRaF was proposed, where the node forwarding region is divided into smaller sections with different priorities, as decided by the forwarding distances. The performance analysis of GeRaF included the single-hop energy consumption and latency tradeoff, as well as the multihop count number. The study in [16] investigated the architectural aspects of opportunistic routing, where the performance of the component design is analyzed based on single-hop metrics and light-traffic scenario. In [17], the authors discussed the opportunistic relay node selection in a two-hop transmission scenario. The relay-node priority criterion is based on the gain of source-to-relay channels and relay-to-destination channels.

3. MODULE DESIGN AND IMPLEMENTATION

A unicast wireless link is an abstraction of the proximity wireless nodes cooperation for unicasting. Since unicasting usually involves the information delivery from source to destination over multiple hops, the unicast module design decouples the end-to-end network behavior into proximity wireless nodes cooperation. In the following, some elements of the module design and implementation are presented. The module state-diagram is then shown.

3.1. Network address

Network address is related to the context, subject to a “cost of delivery” criteria. Let d and s denote the network addresses of the destination node and the source node, respectively. Given the destination d of a data packet at one wireless node n , a local parameter $c_{n,d}$ is assumed obtainable, which indicates the approximate or average cost of delivering the packet to the destination from the node n independent of dynamic changes. In large-scale wireless networks, the cost of delivery $c_{n,d}$ is usually a function of the distance from one local node n to the destination d .

In location-centric networks (e.g., [10]), where wireless nodes are aware of their own locations, for example, by global position systems (GPS) or triangulation estimations, the network address of one node n is solely decided by the node n location coordinates \mathbf{L}_n . Given the destination coordinates \mathbf{L}_d of a packet, $c_{n,d} = |\mathbf{L}_n - \mathbf{L}_d|$ can be readily calculated in geometry, defined as the specific distance.

In data-centric networks (e.g., [18]), the network address can be decided by the application specific data. The cost of delivery $c_{n,d}$ is the application data gradient, which can be assumed as a monotonically increasing function of the distance between n and d .

In a data-collecting or fusion network, for example, wireless sensor networks, the sink (data collector) can broadcast a number of identity advertisement packets, which is thereon flooded in the network, by broadcasting. Every node can count the average smallest number of hops from the sink, on receiving the advertisement packets. The count number can be used as $c_{n,d}$ for one node n . A similar approach appeared in [19] yet in a more general scenario, where the logic address, that is, a vector of the estimated distance or hop number to a group of anchors, is maintained for every wireless node. New nodes joining in the network can estimate their own logic address by acquiring the address of neighboring nodes.

Although other types of address can be applicable, we use the location address in this paper for analysis, simulations, and experiments.

3.2. Radio implementation

The terminology of cognitive radio was first suggested in [20] as an ideal-omnipotent radio which can take all the parameters into consideration for user-centric communications. Cognitive radio was later reviewed in [21, 22] as the radio with dynamic spectrum access. In [8], two propositions are further suggested for the cognitive radio for large-scale wireless networks.

- (i) The radio can opportunistically sense the spectrum resource, so that the selected spectrum usage will not be interfering with other on-going wireless communications.
- (ii) The radio can opportunistically poll one or more other proximity radios onto the selected spectrum, so as to realize certain types of local cooperations.

The above two propositions can extend the concept of pure cognitive radio to cognitive networks, which implement both dynamic spectrum access and dynamic radio access. In the network architecture EW1, a wireless node can initiate an abstract wireless link, that is, certain types of local cooperations among a number of proximity wireless nodes. Hence, both the set of wireless nodes and the operating spectrum are opportunistically decided for an abstract wireless link. Further, the initiated link will not be interfering with other wireless communications, in accordance with the first cognitive proposition.

In the test-bed development of the paper, we prototyped an experimental tone-based cognitive radio [8, 12]. Specifically, the radio can access a group of predetermined data channels, where every data channel is also associated with two distinctive frequency tones, that is, one sensing tone and one polling tone. The sensing and polling tones are at distinctive frequency different from the data channel. The radio hardware is therefore composed of two transceivers, which are the tone transceiver and the data transceiver, respectively. On initiating an abstract wireless link, the radio senses for an available channel, with the vacant data channel and the vacant sensing/polling tones. It then broadcasts the polling tone associated to the selected channel, to poll its

surrounding nodes. On detecting the rising edge of the polling tone, the surrounding nodes can decide to join in the initiated abstract wireless link based on their autonomous availability. On joining in an abstract wireless link, the radio of the surrounding nodes also broadcasts the sensing tone. As such, both sensing and polling tones protect wireless link modules from spectrum interferences. We note that the use of frequency (busy) tones has also appeared in numerous works of MAC protocols [23, 24]. However, here we have the first implementation over multiple channels, and such implementation is in the context of large-scale cognitive wireless networks.

The unicast wireless link module is programmed on the cognitive radio prototype (shown in Figure 1). The experiment hardware is a stack of two radio boards (TELOSB [25]). The radio board is composed of one IEEE 802.15.4 [26] compatible transceiver, Chipcon CC2420, one TI MSP430 Microcontroller, and the program interface. The top radio board is utilized as the tone transceiver, while the bottom radio board is utilized as the data transceiver. The two are wired up by the digital interface shown in Figure 1. Three independent data channels in the 2.4 GHz band are allotted to the radio platform, while the associated tones are also in the band of 2.4 GHz. The unicast module is programmed in the firmware of TI MSP430.

3.3. Packet relaying process

Under an initiated wireless link, the packet relaying process picks one available relay node and sends the data information from the source or a previous relay node to the new relay node.

Four types of packets are used in the packet relaying process, which are RTS, CTS, DATA, and ACK, respectively. RTS, CTS, and ACK are control packets, while DATA is the data information unit. Let node n_p denote the relay node at the hop p , the next hop relay node n_{p+1} is found by the following procedure, which incurs local wireless nodes cooperation in proximity, that is, a unicast wireless link.

First, the node n_p broadcasts a RTS, including the module type (unicast), the destination address d , and the self-address n_p . Upon receiving the RTS at one node n , if the condition $c_{n,d} < c_{n_p,d}$ is satisfied, the node n initializes a timer, with the timeout period as $T_d(n_p, n, d)$. $T_d(n_p, n, d)$ is inversely proportional to $c_{n_p,d} - c_{n,d}$, which is determined by the specific physical radio technology, for example,

$$T_d(n_p, n, d) = \frac{C_0}{c_{n_p,d} - c_{n,d}} + \text{SIFS}, \quad (1)$$

where C_0 is a constant, and $n \neq d$. SIFS is the smallest (minimal) inter frame space (delay constant), which is composed of the module processing time and the transceiver RX/TX switch time. In our current test-bed implementation, $T_d(n_p, n, d)$ is also slotted according to the minimal carrier-sensing time. Note that the condition $c_{n,d} < c_{n_p,d}$ is enforced, so that the cost of delivery to the destination is always descending which prevents any loop.

The node n then backs off and monitors the data channel for the period $T_d(n_p, n, d)$. If the data channel is free during

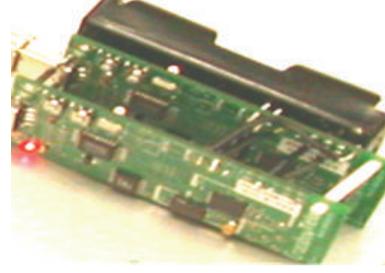


FIGURE 1: Cognitive radio prototype.

that period, the node n replies with a CTS, declaring itself as the next hop relay node n_{p+1} . Otherwise, the node n quits the unicast module. As such, the node with the smallest cost $c_{n,d}$ should be elected as the next hop relay node n_{p+1} . After having received the CTS, the node n_p transmits the DATA packet, and after receiving the DATA packet, the node n_{p+1} replies an ACK, completing the round of relaying. In the case that more than one node send out their CTS simultaneously, the transmissions will collide on some specific bits, for example, the node address, where the CTS packets are different. Since this collision can be detected by certain mechanism, such as cyclic redundancy check (CRC), at n_p , the node n_p can use an appended procedure to differentiate one node as n_{p+1} , either before or after the DATA transmission.

The described mechanism is illustrated in Figure 2. In the last hop, the destination d is always assigned with the minimal delay, that is, $T_d(n_p, d, d) = \text{SIFS}$.

3.4. Module state diagram

The state diagram of a single unicast module in the wireless link layer is shown in Figure 3, which can give a summary of the implementation. The definitions of the states and the associated transferring branches can be self-explainable, in accordance with previous descriptions.

The wireless link layer stays in the IDLE state, when no module is involved. It can initiate a unicast module, for example, on receiving the command from the system layer, so as to send certain amount of information (i.e., contained in one DATA packet) to a specified destination. In the IDLE state, the wireless link layer also monitors the set of polling tones, which are associated to the predetermined group of data channels. On detecting the rising edge of one polling tone, it sends one module request to the system layer. Upon approval, the wireless link layer listens to the RTS packet on the data channel, which is associated with the detected polling tone. Otherwise, it goes back to the IDLE state. Furthermore, after having received one DATA packet, the unicast module automatically relays it, if the current node is not the destination.

Table 1 shows an example set of primitive functions on the interface between the system layer and the wireless link layer, as related to the unicast module. The parameter *Priority* in Table 1 can be used for specifying the traffic class, which will be explored further in Section 4.5.3.

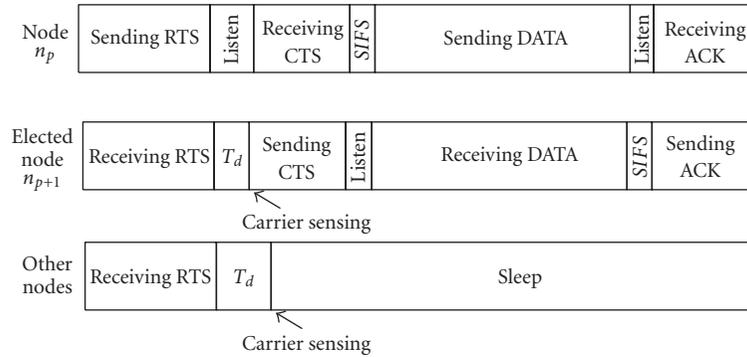


FIGURE 2: Packet relaying mechanism.

3.5. Discussion

By opportunistically dealing with network resources including both spectrum bandwidth and radio availability, data packets take opportunistically available paths with opportunistically available spectrum in every hop. Since the unicast module operates opportunistically in proximity wireless nodes, the end-to-end performance of real-time QoS, can be statistically assured, adding up the sequential proximity operations. This discussion can provide an intuitive explanation of the performance analysis results in Section 4.

Specifically, random spectrum availability is handled by the cognitive radio implementation, where a source or relay node tries to find an available data channel for initiating the unicast wireless link. Important aspects of random node/radio availability are as follows: the node deployment, mobility, and traffic congestion. In general, it indicates that the node n_p , that is, the p th hop relay, would be uncertain about the availability of the next hop relay node, before the RTS probing. The RTS/CTS exchange in the module design opportunistically finds the next hop relay, in a group of available candidates.

In traditional computer networks, dynamic traffic load is limited by network congestion control. The network layer drops overflowed packets by queueing management, which is then detected by the transport layer. Such methods have been known to incur problems in wireless networks [27]. By the real-time unicast module, the paradigm of the classical “network queueing” could be transferred to “queueing in network.” In an individual relay node, a DATA packet is forwarded automatically without lengthy buffering in one local queue. Sequential DATA packets nominally take opportunistically decided paths from the source to the destination, which are buffered in the network, instead of any predetermined nodes.

Therefore, it can be intuitive to conceive that the QoS performance metrics should improve with larger network scale, since the network density contributes to the diversity (or more radio resource) that can be opportunistically exploited. This agrees with other theoretical results, for example, in [28], where it is stated that summed network transport capacity should increase with network scale, on the order of $\Theta(\sqrt{N})$, where N is the number of nodes.

Moreover, in the proposed method, any node does not need to consider or know every other possible node as a candidate for the next hop. In the formation of abstract wireless linkage, the participating nodes are decided by their autonomous availability. The unicast module design also assures that there would only be one RTS response in most cases by the carrier-sensing mechanism. Therefore, the module complexity at individual nodes remains constant, independent of the network scale.

4. PERFORMANCE ANALYSIS OF REAL-TIME QoS

The performance analysis mathematically quantifies the major propositions in the paper. The results can also provide useful network planning guidance for deployments.

4.1. Objectives and approaches

In supporting real-time QoS, the metrics of throughput, expected end-to-end delay, and delay variance, are studied for the unicast wireless link module. Two types of resource consumptions are considered, which are network capacity and network energy consumption. Therefore, we centrally explore the following three questions. (1) How do the QoS metrics change with network scale? (2) How do the QoS metrics change with the source-to-destination distance? (3) How to control the resource allocation, so as to achieve the QoS requirements?

Given the source s and the destination d of a unicast dataflow, let $l = |\mathbf{L}_s - \mathbf{L}_d|$ denote the source-to-destination distance. Analytical results are obtained under the asymptotic condition of long-range unicasting, $l \rightarrow \infty$, where the justification can be that long-range unicasting presents the worst-case scenario of the network performance under practical interests. On the other hand, concise analytical formulation of the QoS metrics can be obtained under this long-range asymptotic condition, which can provide analytical insights to previous questions.

4.2. Analytical models

In the performance analysis, models about node distribution, wireless channel, power consumption, bandwidth availability, and the IDLE probability are adopted.

TABLE 1: Exemplary primitive functions related to the unicast module.

Function	Description	Direction
SendUnicast($DA, DATA, Priority$)	Command of sending the unicast DATA to the destination address DA;	From system to wireless link
StatusUnicast(ϕ -Status)	Return the command status of sendUnicast($\cdot \cdot \cdot$), for example, Success or Fail;	From wireless link to system
IndicateUnicast($SA, DATA, Priority$)	Indicate the received unicast DATA from the source address SA;	From wireless link to system
ModuleRequest()	Request for listening to the RTS, that is, defined in the unicast module;	From wireless link to system
ModuleResponse(ϕ -Status)	Return the request status of moduleRequest(), that is, Permitted or Denied;	From system to wireless link
GetState(ϕ -State)	Get the current state.	From system to wireless link

4.2.1. Node distribution model

The node distribution is modeled as a two-dimensional (2D) Poisson process [29], with the node density λ . That is, given an area \mathbf{A} of the size $|\mathbf{A}|$ in the field, the number of nodes in the area follows Poisson distribution with the parameter $\lambda|\mathbf{A}|$. The Poisson modeling can be typical for random node distribution and/or random mobility.

4.2.2. Wireless channel model

Given an arbitrary transmitting node n , and a receiving node m , the successful transmission probability, in general, is a function of the transmitting power P_t , the radio data rate R , and the distance $\zeta = |\mathbf{L}_m - \mathbf{L}_n|$. We denote this function as $f(P_t, R, \zeta)$.

For example, if small-scale Rayleigh fading [30] is assumed, the channel model can be given by

$$f(P_t, R, \zeta) = \text{Prob}\left(\frac{P_t \cdot G \cdot \xi}{N_0 \cdot R \cdot \zeta^\alpha} \geq B\right) = e^{-(N_0 \cdot R \cdot \zeta^\alpha \cdot B)/(P_t \cdot G)}. \quad (2)$$

In (2), G is a propagation-loss constant; N_0 is the receiver-noise power spectrum density; α is the path loss component in wireless channel [30]; B is a threshold constant representing the receiver sensitivity; and ξ is a unit-mean exponential random variable. This model will also be utilized to obtain the numerical results in Section 5.

4.2.3. Power consumption model

Referring to the state diagram in Figure 3, the node power consumption at the IDLE state is denoted by P_I , that is, the low-power monitoring of the polling tones; the power consumption at the states of “Send DATA”, “Send RTS”, “Send CTS”, and “Send ACK”, is modeled by

$$P_S + (1 + \beta) \cdot P_I, \quad (3)$$

where P_S is the transmitter circuits power consumption; β is a constant decided by the RF power amplifier efficiency [31]; and P_t is the transmitting power as defined perviously. The node power consumption in other states or time is modeled by a constant P_R , which denotes the receiving (or idle listening) power consumption.

4.2.4. Bandwidth availability and the IDLE probability

When initiating a unicast module, we assume that the node can always find an available channel by the cognitive radio without significant delay. The IDLE probability is the probability that a given node is in the IDLE state (shown in Figure 3), and can be engaged in a unicast wireless link being initiated by other nodes around it. We assume that there is a fixed (or lower bounded) IDLE probability “ ρ ”, in the analysis.

With limited network capacity and arbitrary traffic load, these two assumptions need to be ensured by an appropriate call admission control (CAC) mechanism. On the other hand, given analogous experiences of the popular video streaming on peer-to-peer overlay networks, it could also be appropriate to assume no CAC mechanism, while the traffic volume demanding can always fall below the planned (or unplanned) network capacity. In large-scale wireless networks, the capacity could be of relatively cheap resources, given abundant wireless nodes and the use of unlicensed band, while the traffic load is not necessarily related to the number of wireless nodes. These considerations give the optimization formulations under different types of resource consumptions in Section 4.5.

4.2.5. Other parameters and notations

The packet lengths (bit) of RTS, CTS, DATA, and ACK are denoted by L_R , L_C , L_D , and L_A , respectively. T_S denotes the time delay in the “channel sensing” state in Figure 3, and T_B denotes the time delay in the “Backoff T_d ” state. We also use the notations $E(\cdot)$ and $\sigma^2(\cdot)$ to denote the mean and the variance of a random variable.

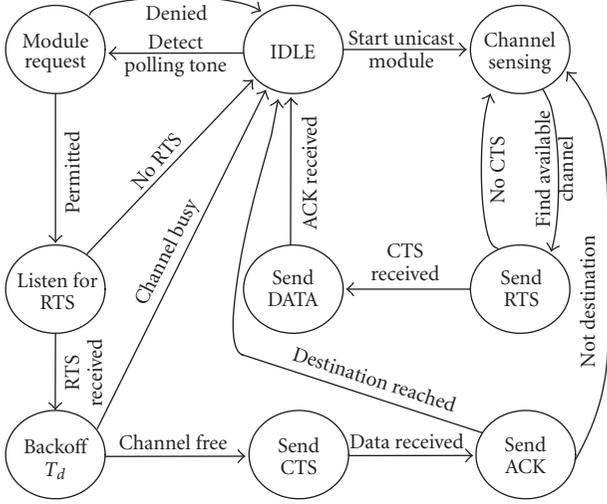


FIGURE 3: Wireless link layer state diagram.

4.3. Iterative performance analysis

“Iterative performance” can be similar to “single-hop performance.” However, with the notation of “iterative performance,” we also consider the scenario where the next hop relay node is not found in the transferring branch from “Send RTS” to “channel sensing” of Figure 3. The reason resides in the random node availability, for example, no potential relay candidate is in the IDLE states. Due to this difference, the terminology “iteration” instead of “hop” is used in the following analysis, and “iterative delay” indicates the time delay of an iteration, whereas “iterative energy consumption” indicates the network energy consumption in one iteration.

Particularly, consider that there are totally I iterations from the source s to the destination d . Let $\{n_i \mid 1 \leq i \leq I\}$ denote the relevant set of nodes, which initiates the unicast wireless link. Obviously, there is $n_1 = s$, and n_I sends the DATA packet directly to the destination d . The iteration number I is therefore a random variable.

4.3.1. Iterative delay

Let τ_i denote the iterative delay at an iteration i . (Other possible models are also discussed in Appendix C, which contribute to the same end-to-end performance analysis results.) If the DATA transmission is completed in the iteration, τ_i can be calculated by the formula $(L_R + L_C + L_D + L_A)/R + T_S + T_B$. Otherwise, if the DATA transmission is uncompleted since no relay is found, τ_i can be given by $L_R/R + T_S + T_B$. Obviously, for the last iteration ($i = I$), the DATA transmission is completed.

When $i < I$, the probability of uncompleted DATA transmission depends on the relay candidates availability on the forwarding plane. Due to the Poisson node distribution model with the density λ , and the node IDLE probability ρ , the distribution of relay candidates conforms to an inhomogeneous Poisson process with the density function

$\lambda \cdot \rho \cdot f(P_t, R, \zeta)$ on the forwarding plane, where ζ indicates the distance to the node n_i . Therefore, the “uncompleted” probability is decided by the “equal-zero” probability of a Poisson random variable with the mean $\lambda \cdot \rho \cdot \Omega$, which is given by $e^{-\lambda \rho \Omega}$. Ω is always positive, and is given by the integration of the channel model $f(\cdot, \cdot, \cdot)$ on the forwarding plane:

$$\Omega = \int_0^\infty \int_{-\infty}^\infty f(P_t, R, \sqrt{x^2 + y^2}) dx dy. \quad (4)$$

According to the above analysis, $\{\tau_i \mid 1 \leq i < I\}$ can be modeled as a set of random variables with identical distributions, where the expectation $E(\tau_i)$ is

$$E(\tau_i) = \frac{L_R}{R} + T_S + T_B + \frac{L_C + L_D + L_A}{R} \cdot (1 - e^{-\lambda \rho \Omega}). \quad (5)$$

4.3.2. Iterative energy consumption

Let ϵ_i denote the iterative energy consumption at an iteration i . (Other possible models are also discussed in Appendix C, which contribute to the same end-to-end performance analysis results.) If the DATA transmission is not completed in the iteration, the energy consumption of the node n_i is $(T_S + T_B) \cdot P_R + L_R/R \cdot [P_S + (1 + \beta)P_t]$, according to the described models. The energy consumption of one relay candidate is given by $T_S \cdot P_I + L_R/R \cdot P_R$. The number of the relay candidates around n_i , which receive the RTS, is a Poisson random variable, with the mean value $\lambda \cdot \rho \cdot \Omega$. Here Ω can be still given by (4), but is integrated on the backward plane, that is, $\int_{-\infty}^0 \int_{-\infty}^\infty f(P_t, R, \sqrt{x^2 + y^2}) dx dy$, instead of the forwarding plane. Therefore, the expected iterative energy consumption under uncompleted DATA transmission is the summation: $(T_S + T_B) \cdot P_R + L_R/R \cdot [P_S + (1 + \beta)P_t] + \lambda \cdot \rho \cdot \Omega \cdot (T_S \cdot P_I + L_R/R \cdot P_R)$.

If the DATA transmission is completed in the iteration, the energy consumption of the node n_i is $(T_S + T_B) \cdot P_R + (L_R + L_D)/R \cdot [P_S + (1 + \beta)P_t] + (L_C + L_A)/R \cdot P_R$. The energy consumption of one relay candidate, which is not elected as the next node n_{i+1} , is still $T_S \cdot P_I + L_R/R \cdot P_R$. The number of the relay candidates is a Poisson random variable with the mean value $2\lambda \cdot \rho \cdot \Omega$, including both the forward and the backward planes. On the node n_{i+1} , the additional energy consumption is $L_D/R \cdot P_R + (L_C + L_A)/R \cdot [P_S + (1 + \beta)P_t]$. Therefore, the expected iterative network energy consumption under completed DATA transmission is $(T_S + T_B + (L_C + L_D + L_A)/R) \cdot P_R + (L_R + L_C + L_D + L_A)/R \cdot [P_S + (1 + \beta)P_t] + 2\lambda \cdot \rho \cdot \Omega \cdot (T_S \cdot P_I + L_R/R \cdot P_R)$.

Since the DATA transmission is always completed in the last iteration, ϵ_I is directly obtained from the previous discussion. When $i < I$, similar to the previous analysis of τ_i , the probability of uncompleted DATA transmission is decided by $e^{-\lambda \rho \Omega}$. Therefore, $\{\epsilon_i \mid 1 \leq i < I\}$ can be modeled

as a set of random variables with identical distributions, where the expectation $E(\epsilon_i)$ is

$$\begin{aligned}
E(\epsilon_i) &= (T_S + T_B) \cdot P_R + \frac{L_R}{R} \cdot [P_S + (1 + \beta)P_t] \\
&\quad + \lambda \cdot \rho \cdot \Omega \cdot \left(T_S \cdot P_t + \frac{L_R}{R} \cdot P_R \right) + (1 - e^{-\lambda \rho \Omega}) \\
&\quad \cdot \left\{ \frac{L_C + L_D + L_A}{R} \cdot [P_R + P_S + (1 + \beta)P_t] \right. \\
&\quad \left. + \lambda \cdot \rho \cdot \Omega \cdot \left(T_S \cdot P_t + \frac{L_R}{R} \cdot P_R \right) \right\}. \tag{6}
\end{aligned}$$

4.3.3. Number of iterations

Given all the parameters, the total iteration number I is a random variable decided by the source-to-destination distance l . It is shown in Appendix A that the mean value $E(I)$ and the variance $\sigma^2(I)$ are

$$E(I) = \frac{l}{E(\Lambda)} + o(\sqrt{l}), \tag{7}$$

$$\sigma^2(I) = \frac{l \cdot \sigma^2(\Lambda)}{E(\Lambda)^3} + o(l), \tag{8}$$

respectively.

In the above, the random variable Λ generally indicates the forwarding distance of one single iteration. Λ is defined with the pdf (probability density function) $p_\Lambda(x)$:

$$p_\Lambda(x) = \begin{cases} \lambda \rho \cdot e^{-\lambda \rho \int_x^\infty g(u) du} \cdot g(x), & x > 0, \\ e^{-\lambda \rho \Omega} \cdot \delta(0), & x = 0, \\ 0, & x < 0, \end{cases} \tag{9}$$

where

$$g(u) = \int_{-\infty}^{\infty} f(P_t, R, \sqrt{u^2 + v^2}) dv. \tag{10}$$

Proved in Appendix B, we point out that the expectation $E(\Lambda)$ is an increasing function of the node density λ , but is upper bounded, as decided by the channel model $f(\cdot, \cdot, \cdot)$. Furthermore, the variance $\sigma^2(\Lambda)$ decreases to zero with higher λ . Given (7) and (8), both $E(I)$ and $\sigma^2(I)$ can decrease with higher node density λ . These relations can also be intuitively understood, which will be exploited for interpreting the end-to-end performance analysis results next.

4.4. End-to-end performance analysis

Analytical results are presented for the maximal throughput Φ , the expected end-to-end delay Y , the end-to-end delay variance Θ , and the expected network energy consumption per DATA packet Ξ , of a dataflow from the source s to the destination d , under the long-range asymptotic condition $l \rightarrow \infty$. The discussion shows how these metrics change with the source-to-destination distance l , as well as the network density λ .

4.4.1. Throughput

The maximal throughput Φ is decided by how many DATA packets can be sent out from the source s to the destination d in a period of time, subject to the condition that this time period is much larger than the packet traveling time. According to previous analysis in Section 4.3.1, the probability of the completed DATA transmission in an iteration is $1 - e^{-\lambda \rho \Omega}$. Therefore, the expected number of iterations initiated by s for one DATA packet is $1/(1 - e^{-\lambda \rho \Omega})$, and the associated time expectation is $(L_R/R + T_S + T_B)/(1 - e^{-\lambda \rho \Omega}) + (L_C + L_D + L_A)/R$.

Given the DATA packet length L_D , the maximal throughput is thus obtained as

$$\Phi = \frac{L_D \cdot R}{(L_R + (T_S + T_B) \cdot R)/(1 - e^{-\lambda \rho \Omega}) + L_C + L_D + L_A}. \tag{11}$$

It is shown that the maximal throughput Φ can be independent of the source-to-destination distance l . It also increases monotonically with the node density λ , approaching the limit decided by the radio data rate R and the overhead ratio.

4.4.2. Expected end-to-end delay

The end-to-end delay is given by $\sum_{i=1}^I \tau_i$. The expected end-to-end delay Y is

$$\begin{aligned}
Y &\stackrel{(a)}{=} E\left(\sum_{i=1}^I \tau_i\right) \\
&\stackrel{(b)}{=} [E(I) - 1] \cdot E(\tau_1) + \tau_1 \\
&\stackrel{(c)}{=} l \cdot \frac{L_R + (T_S + T_B) \cdot R + (L_C + L_D + L_A) \cdot (1 - e^{-\lambda \rho \Omega})}{R \cdot E(\Lambda)} \\
&\quad + o(\sqrt{l}), \tag{12}
\end{aligned}$$

where (a) is given by the definition; (b) is due to the fact that $\{\tau_i \mid 1 \leq i < I\}$ are of identical probability distribution [32]; and (c) is obtained directly from (5) and (7).

Therefore, the expected end-to-end delay Y increases linearly with the source-to-destination distance l . It also decreases with the node density λ , approaching a bounded limit.

4.4.3. Delay variance

Let I_c denote the number of the iterations with completed DATA transmission, that is, the number of wireless hops, and let I_u denote the number of the iterations with uncompleted DATA transmission. I_c and I_u are independent random

variables, and $I = I_c + I_u$. The delay variance Θ is calculated by the following:

$$\begin{aligned}
\Theta & \stackrel{(a)}{=} \sigma^2 \left(\sum_{i=1}^I \tau_i \right) \\
& \stackrel{(b)}{=} \sigma^2(I_c) \cdot \left(\frac{L_R + L_C + L_D + L_A}{R} + T_S + T_B \right)^2 \\
& \quad + \sigma^2(I_u) \cdot \left(\frac{L_R}{R} + T_S + T_B \right)^2 \\
& \stackrel{(c)}{=} \sigma^2(I_c) \cdot \left[\left(\frac{L_C + L_D + L_A}{R} \right)^2 + 2 \cdot \left(T_S + T_B + \frac{L_R}{R} \right) \right. \\
& \quad \left. \cdot \frac{L_C + L_D + L_A}{R} \right] + \sigma^2(I_u) \cdot \left(\frac{L_R}{R} + T_S + T_B \right)^2 \\
& \stackrel{(d)}{=} I \cdot \frac{\sigma^2(\Lambda)}{E(\Lambda)^3} \cdot \left\{ \left(1 - e^{-\lambda\rho\Omega} \right)^2 \right. \\
& \quad \cdot \left[\left(\frac{L_C + L_D + L_A}{R} \right)^2 + 2 \cdot \left(T_S + T_B + \frac{L_R}{R} \right) \right. \\
& \quad \left. \left. \cdot \frac{L_C + L_D + L_A}{R} \right] + \left(\frac{L_R}{R} + T_S + T_B \right)^2 \right\} \\
& \quad - I \cdot e^{-\lambda\rho\Omega} \cdot \frac{1 - e^{-\lambda\rho\Omega}}{E(\Lambda)} \cdot \left[\left(\frac{L_C + L_D + L_A}{R} \right)^2 + 2 \cdot \left(T_S + T_B + \frac{L_R}{R} \right) \right. \\
& \quad \left. \left. \cdot \frac{L_C + L_D + L_A}{R} \right] + o(I), \tag{13}
\end{aligned}$$

where (a) is given by the definition; (b) is due to the analysis in Section 4.3.1, that is, the iterative delay under completed and uncompleted transmissions respectively; (c) is due to the fact that $\sigma^2(I) = \sigma^2(I_c) + \sigma^2(I_u)$; (d) is obtained from the result of $\sigma^2(I)$ in (8), as well as the $\sigma^2(I_c)$ derived in (A.4) of Appendix A.

Therefore, the end-to-end delay variance Θ increases linearly with the source-to-destination distance l . Θ also diminishes to zero with the network density λ , since the component $\sigma^2(\Lambda)$ also decreases to zero with higher λ .

4.4.4. Network energy consumption

The expected network energy consumption for one DATA packet Ξ is

$$\begin{aligned}
\Xi & \stackrel{(a)}{=} E \left(\sum_{i=1}^I \epsilon_i \right) \\
& \stackrel{(b)}{=} [E(I) - 1] \cdot E(\epsilon_1) + \epsilon_I \tag{14} \\
& \stackrel{(c)}{=} l \cdot \frac{E(\epsilon_1)}{E(\Lambda)} + o(\sqrt{l}),
\end{aligned}$$

where (a) and (b) are given by the definition and the fact that $\{\epsilon_i \mid 1 \leq i < I\}$ are of identical probability distributions [32];

(c) is obtained directly from (7), and the component ϵ_1 can be replaced by the result in (6). Therefore, Ξ also increases linearly with the source-to-destination distance l .

4.5. Optimizations under QoS constraints

Consider that the dataflow has certain QoS requirements on throughput, expected end-to-end delay, and delay variance, for example, $\Phi \geq \Phi_{th}$, $\Upsilon \leq \Upsilon_{th}$, and $\Theta \leq \Theta_{th}$. Here, Φ_{th} , Υ_{th} , and Θ_{th} are the threshold constants representing those QoS constraints, respectively. The optimizations aim to minimize the resource consumption, subject to these QoS constraints.

Two types of resource consumptions, including network energy consumption and network capacity, are considered here, which lead to different optimization formulations. As pointed out in Section 4.2.4, these two formulations can be representing different networking scenarios. In capacity limited networks, every node can have cabled power supply. The number of nodes is therefore limited by engineering costs, and the possible use of long-range radio also limits the available bandwidth. On the other hand, in energy limited networks, the nodes can be powered by energy scavenging, for example, solar cells. Therefore, the number of wireless nodes can be large, which can also use short-range radio with vast unlicensed bandwidth. These contribute to large network capacity. Other works with similar considerations have been discussed in [33].

4.5.1. Capacity limited networks

The objective of the optimization is to minimize the transmitting power P_t , so as to limit the occupied geographic area of an unicast link. In accordance with the results in [28], this approach can save the use of wireless network capacity. Therefore, the optimization formulation is

$$\begin{aligned}
& \min P_t \\
& \text{subject to: } \Phi \geq \Phi_{th}; \quad \Upsilon \leq \Upsilon_{th}; \quad \Theta \leq \Theta_{th}. \tag{15}
\end{aligned}$$

4.5.2. Energy limited networks

The objective of the optimization is to minimize the expected network energy consumption per DATA packet Ξ :

$$\begin{aligned}
& \min \Xi \\
& \text{subject to: } \Phi \geq \Phi_{th}; \quad \Upsilon \leq \Upsilon_{th}; \quad \Theta \leq \Theta_{th}. \tag{16}
\end{aligned}$$

4.5.3. Solutions and traffic priority

In order to solve the optimizations in (15) and (16), the transmitting power P_t can be an ideal control knob (optimization variable), for the resource allocation. Both (15) and (16) can be found as convex optimization problems over P_t , where efficient solutions are guaranteed [34].

Specifically, as shown in (11), Φ monotonically increases with P_t , since Ω (defined in (4)) monotonically increases with P_t , provided by any realistic channel model $f(\cdot, \cdot, \cdot)$. As shown in (12), Υ monotonically decreases with P_t , since $E(\Lambda)$ monotonically increases with P_t (proved in Appendix B).

Θ also monotonically decreases with P_t in (13), since $\sigma^2(\Lambda)$ monotonically decreases with P_t (proved in Appendix B). Moreover, as shown in (14), Ξ can be verified as a convex function of P_t , where both $E(\epsilon_1)$ and $E(\Lambda)$ monotonically increase with P_t . Due to the described monotonicity, the convexity of the optimizations in (15) and (16) can then be obtained, over the control knob P_t .

Therefore, we suggest that the transmitting power P_t can provide an effective control knob for the tradeoff between the resource consumptions and the real-time QoS requirements. In fact, different levels of P_t can be configured for different classes of traffics, for example, voice, video (streaming), or data, which then decide the unicasting “priority”.

Further investigation may also include the radio data rate R , so as to generate the problem of joint power and rate control. Unfortunately, the convexity over R cannot be obtained for a general channel propagation model $f(\cdot, \cdot, \cdot)$. Future studies may address the problem on some specific models.

4.6. Summary

In summary, the analytical results indicate that all the investigated real-time QoS metrics can improve with larger network scale. In particular, the maximal throughput Φ increases with the network density λ to a predetermined bound, while the expected end-to-end delay Y decreases with λ , approaching a predetermined bound, as decided by other related parameters. On the other hand, the delay variance Θ falls to zero with higher network density λ , which indicates that the network performance can be made arbitrarily stable, simply by dropping more nodes in the network. For supporting long-range communications over a relatively large number of wireless hops, the analytical results show that the maximal throughput Φ can be independent of the source-to-destination distance, while both the expected delay Y and the delay variance Θ increase linearly with the source-to-destination distance. Furthermore, it is identified that the radio transmitting power P_t can be an ideal control knob for the tradeoff between the real-time QoS requirements and the resource consumptions, for example, energy consumption or network capacity.

We consider that all these propositions could be intuitively conceivable for large-scale wireless networks. Since the analytical results are obtained under the long-range asymptotic condition $l \rightarrow \infty$, they may present the worst-case scenario analysis for practical considerations. In the next section, the simulation results are also provided, showing how these metrics converge.

5. SIMULATION RESULTS

Some fixed parameters in the simulations are listed in Table 2. In particular, the radio transmission parameters are conforming to the developed test-bed, while others, for example, power consumption and channel model parameters, are typical for indoor short-range radios. For example,

TABLE 2: Parameters in the simulations.

Parameter	Value	Unit
ρ	0.2	N/A
R	250 K	bits/sec
L_R	120	bit
L_C, L_A	96	bit
L_D	1024	bit
T_S	250	μ sec
T_B	750	μ sec
P_S, P_R	1	mW
P_t	100	μ W
β	3	N/A
G	-40	dB
N_0	-150	dBm/Hz
B	10	dB
α	4	N/A

under these parameters, the radio range is about 10–15 meters, when the transmitting power P_t is 1 mW.

Given the source and the destination nodes, separated by the distance l , random Poisson points are generated, for every iteration. The generation of the Poisson points conforms to the density $\lambda \cdot \rho$ on the 2D plane, which represent the available wireless nodes. 1000 DATA packets are then “sent out” consecutively from the source to the destination, according to the described real-time unicast module. The end-to-end delay instance of every DATA packet is then recorded, from which the simulated end-to-end delay expectation and the simulated delay variance are estimated by standard statistical methods. The simulated throughput is also obtained, by dividing the data bulk size $1000 \cdot L_D$ with the time consumption of sending out the 1000 DATA packets at the source. In the following figures, we also compare the simulation results to the analytical results side-by-side, where the lower-order terms, for example, $o(\sqrt{l})$ or $o(l)$, are neglected in plotting the analytical results.

Figures 4, 5, and 6 show how the QoS metrics change with the source-to-destination distance l , where the node density λ is fixed at $0.1/\text{m}^2$. Then Figures 7, 8, and 9 show how the QoS metrics change with the network density λ , where the source-to-destination l is fixed at 160 m. The number of wireless hops in the simulations can be up to 20. All the obtained simulation results match well with the analytical results, as well as the main propositions given in Section 4.6. Especially, Figures 4, 5, and 6 show that close-matching also exists in short-range unicasting, although the analytical results were obtained under long-range asymptotic approximation.

Moreover, Figure 10 shows how the expected network energy consumption per DATA packet changes with the transmitting power P_t , which is observed as a convex function. It is also observed in Figure 10 that the expected network energy consumption increases linearly with l , and it decreases with λ in the simulation. These simulation results conform to previous performance analysis.

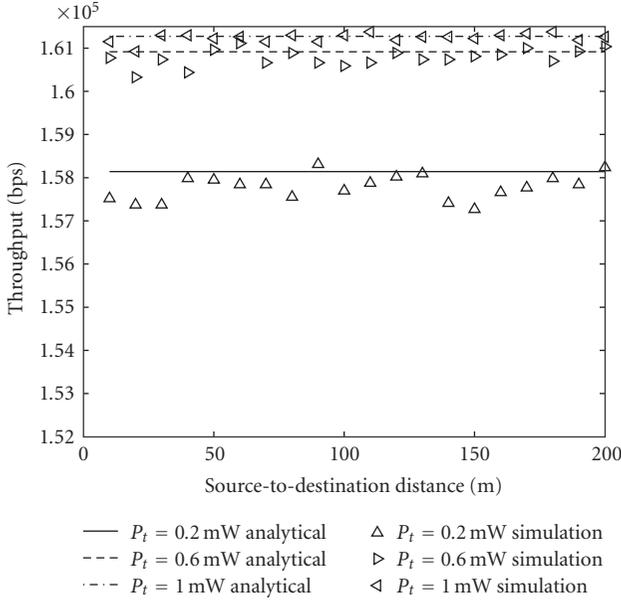


FIGURE 4: Throughput versus source-to-destination distance l ($\lambda = 0.1/\text{m}^2$).

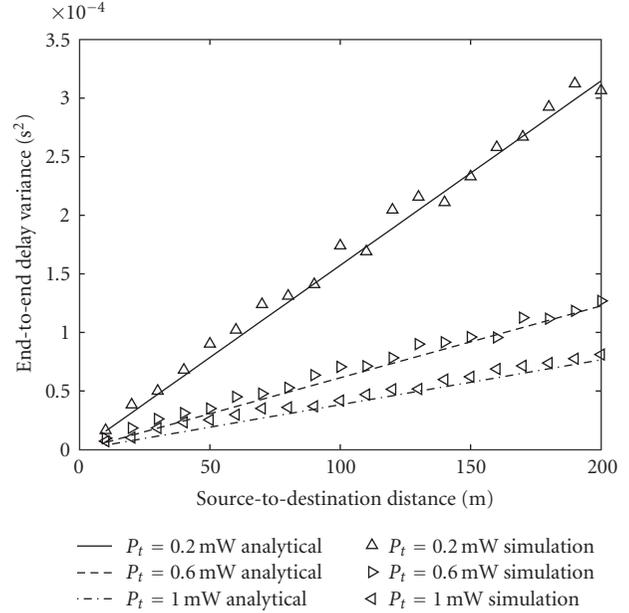


FIGURE 6: Delay variance versus source-to-destination distance l ($\lambda = 0.1/\text{m}^2$).

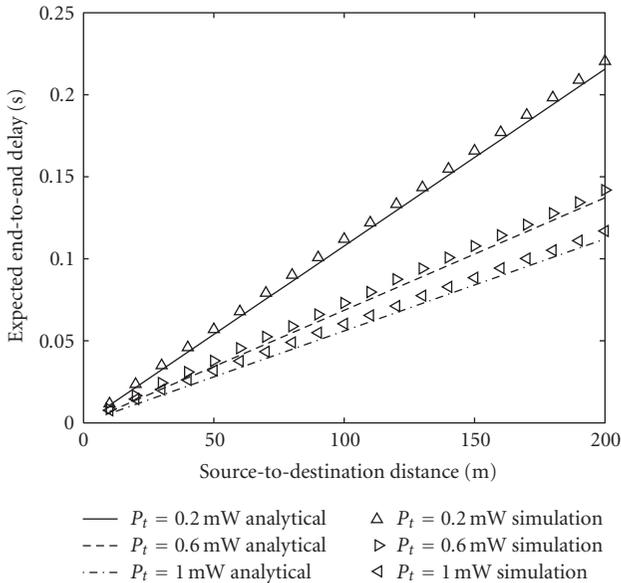


FIGURE 5: Expected delay versus source-to-destination distance l ($\lambda = 0.1/\text{m}^2$).

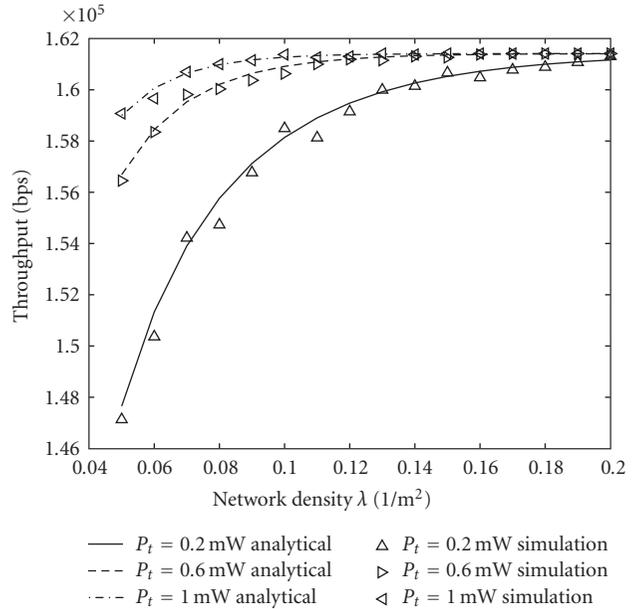


FIGURE 7: Throughput versus network density λ ($l = 160$ m).

6. EXPERIMENT RESULTS

In the test-bed (Figure 11), the source and the destination nodes are connected to a laptop computer for the purpose of collecting experiment records. The wireless nodes are based on the cognitive radio prototype (Figure 1), with a stack of two radio boards. The cognitive radio can opportunistically access 3 data channels in 2.4GHz band, as described in Section 3.2, where each channel has an air-interface data rate 250 kbps as defined in IEEE 802.15.4 [26].

The real-time unicast module is implemented in the cognitive radio prototype, with preconfigured address. The packet length parameters, that is, L_R, L_C, L_D, L_A , are about the same as those listed in Table 2. The node transmitting power is however fixed at -15 dBm. The experiment results on throughput, end-to-end delay expectation, and delay variance, are obtained by the following methods. (1) Throughput is calculated at the source, where the laptop computer forwards 1000 DATA packets to the source node.

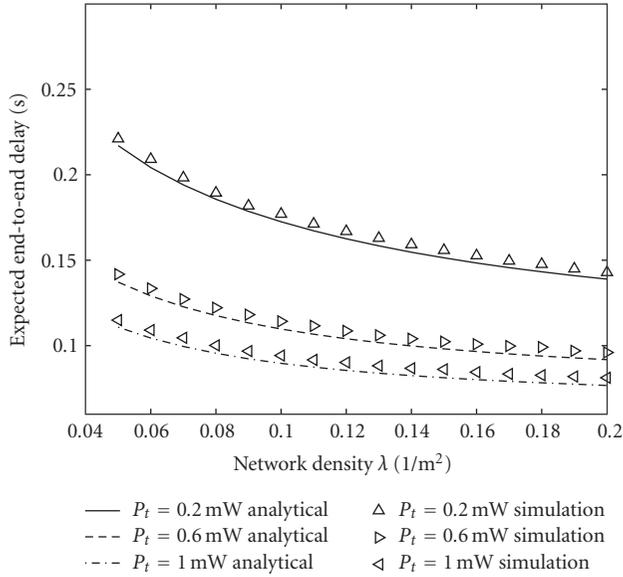


FIGURE 8: Expected delay versus network density λ ($l = 160$ m).

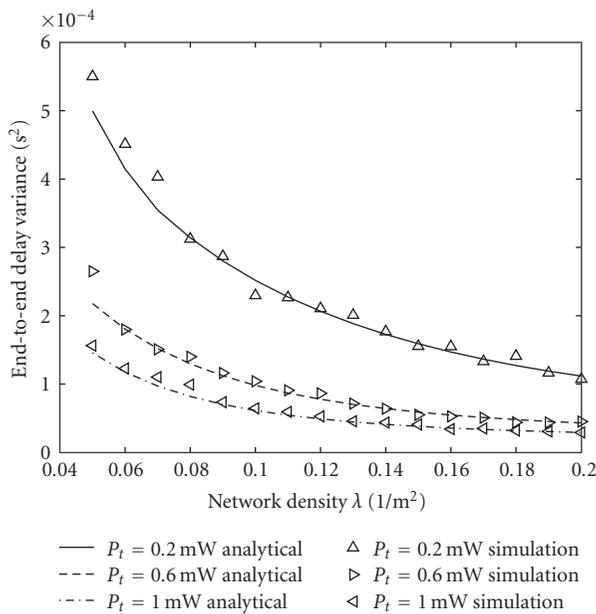


FIGURE 9: Delay variance versus network density λ ($l = 160$ m).

Since the source node accepts a new DATA packet from the computer only after the previous one has been sent out, the computer can count the total time expenditure in sending the 1000 DATA packets. The measured throughput is obtained by dividing the bulk size with the time expenditure. (2) On receiving a DATA packet, the destination also forwards the DATA packet to the laptop computer, where the end-to-end delay instance of every DATA packet is calculated at the computer by the difference between the packet departure and arrival time. The measured end-to-end delay expectation and

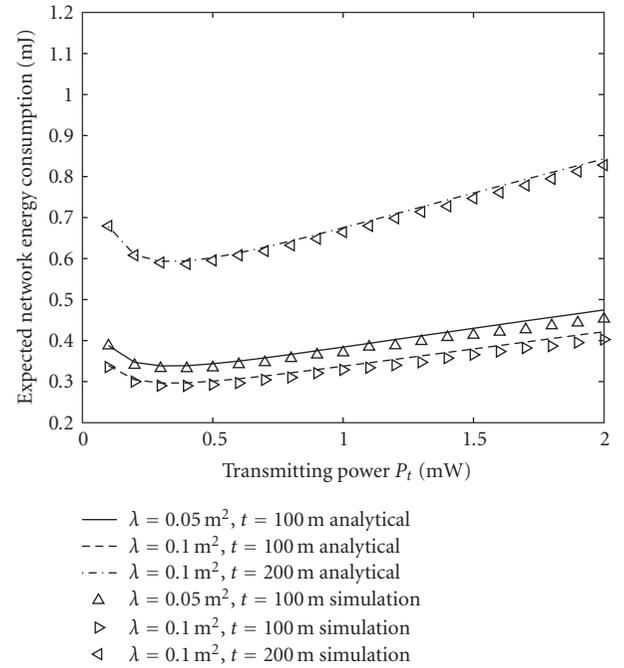


FIGURE 10: Network energy consumption versus transmitting power P_t .

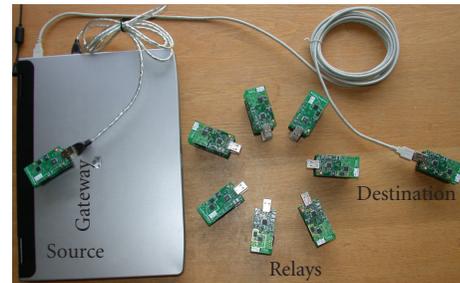


FIGURE 11: A snapshot of the test-bed.

the delay variance are then obtained by standard statistical methods. Since the operating system running at the laptop computer (Windows XP Professional Edition) is by no means real-time, we tried to minimize the computer running threads in the experiments.

The experiments were conducted in a standard office environment, where external interferences in the utilized 2.4 GHz band can be intensive, for example, from adjacent WLAN hot spots. Two sets of experiments are illustrated in Figure 12. In Set-I, the nodes are placed in a straight line, such that the four configurations correspond approximately to one-hop to four-hop communications. For example, in the second configuration of Set-I, the node placement is such that the chance of direct transmission from s to d is small. In Set-II, more relay nodes are added to every hop being illustrated in Figure 12. The purpose of the two-set experiments is to compare how the designed unicast module can improve the test-bed QoS performance. In fact,

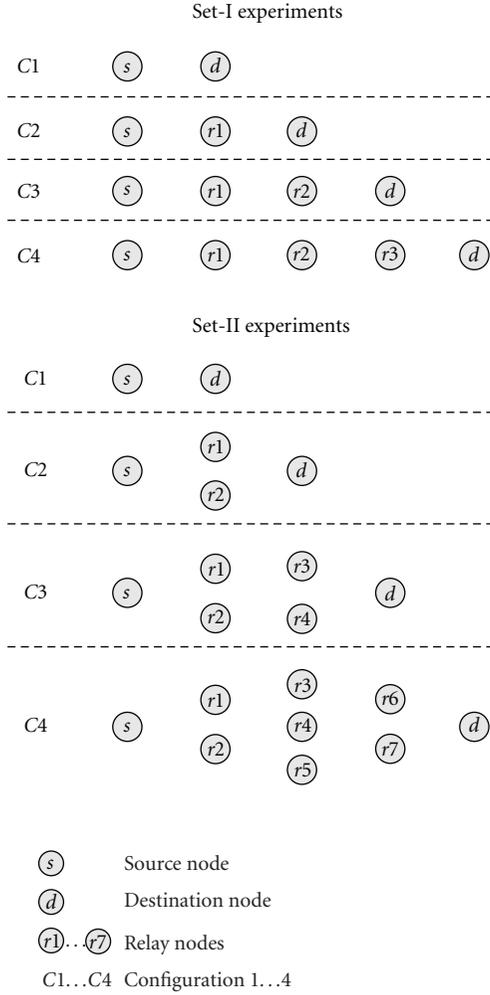


FIGURE 12: Illustration of the two sets of experiments.

Set-I experiments can be comparable to traditional wireless networking where a routing path is predetermined with best available spectrum in every hop. However, in Set-II experiments, data is forwarded from *s* to *d* opportunistically without any predetermined path.

Figures 13, 14, and 15 show the measurement results of the two-set experiments on throughput, expected end-to-end delay, and delay variance, respectively. Evidently, the results of Set-II are strictly better than those of Set-I, beyond the first configuration. In Figure 13, the throughput of Set-I is reduced by more than 50% from the first to the second configuration, that is, from one-hop to two-hop, due to the impact of half-duplex radios. A comparable reduction of throughput is not observed in Set-II. In general, the throughput of Set-II changes smoothly with different configurations (number of wireless hops), where the limited reduction can be due to the limited number of nodes. In Figures 14 and 15, it is further observed that the expected end-to-end delay and the delay variance increase linearly with the number of hops (configuration index) in Set-II experiments; whereas they tend to increase much faster in Set-I experiments.

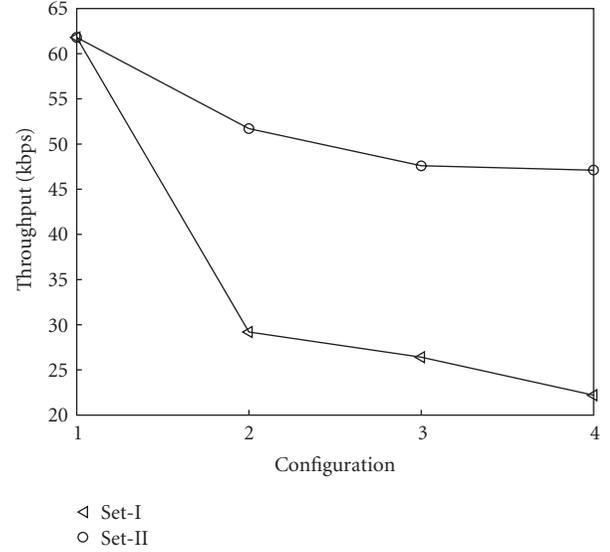


FIGURE 13: Measurement results on the throughput.

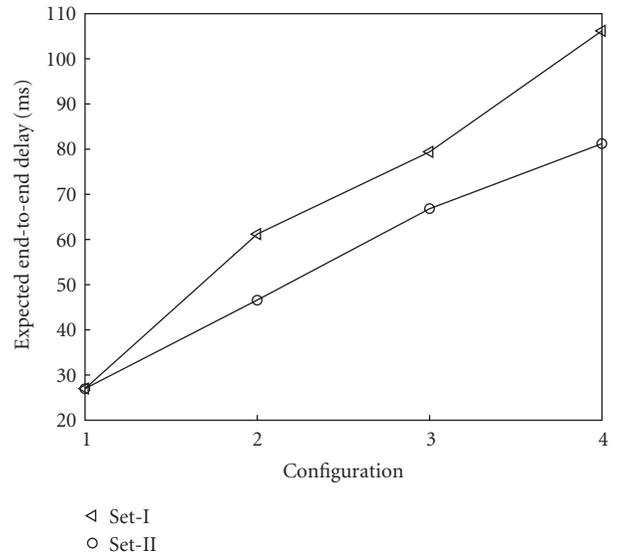


FIGURE 14: Measurement results on the expected end-to-end packet delay.

7. CONCLUSIONS

In the paper, we have implemented a real-time unicast module in the context of large-scale cognitive wireless networking. By opportunistically using both spectrum bandwidth and wireless nodes/radio availability, the most significant result is that the network performance can improve with larger network scale. Specifically, it is shown that the performance dynamics, that is, indicated by dataflow end-to-end delay variance, can fall to zero with higher network density. Furthermore, additional results show that dataflow throughput can be independent of the number of wireless hops or the source-to-destination distance, whereas end-to-end delay expectation and delay variance increase

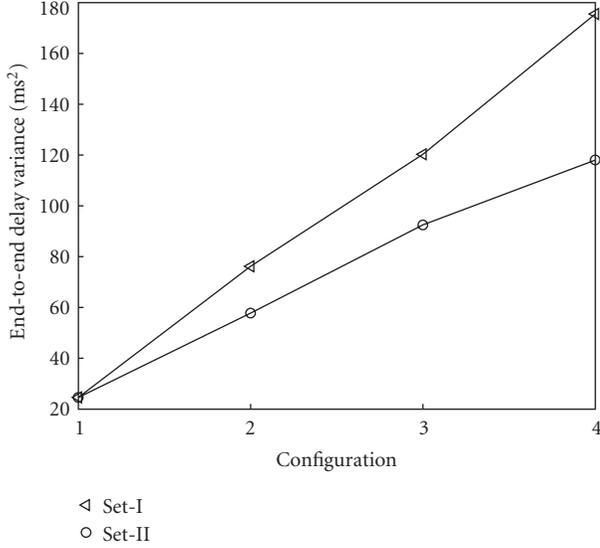


FIGURE 15: Measurement results on the packet delay variance.

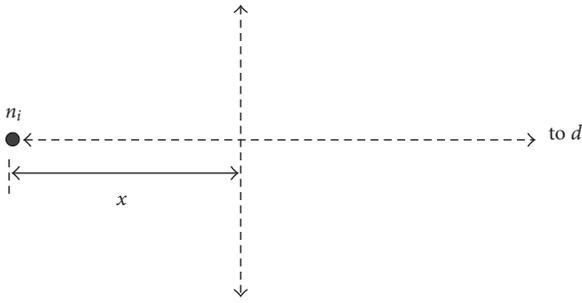


FIGURE 16: Calculation of $p_{\Lambda}(x)$.

linearly. The obtained analytical, simulation, and experiment results support these propositions. It is also identified that node transmitting power can be an effective control knob, for the tradeoff between the real-time QoS performance metrics and resource consumptions. We suggest that the presented research could contribute to the development of diverse commercial and scientific applications with large-scale wireless networking.

APPENDICES

A. CALCULATIONS OF THE ITERATION NUMBER I

At one iteration i , let X_i denote the distance advance to the destination d , that is,

$$X_i = \begin{cases} |\mathbf{L}_{n_i} - \mathbf{L}_d| - |\mathbf{L}_{n_{i+1}} - \mathbf{L}_d|, & 1 \leq i < I, \\ |\mathbf{L}_{n_i} - \mathbf{L}_d|, & i = I. \end{cases} \quad (\text{A.1})$$

Furthermore, let $p_{X_i}(x)$ denote the pdf of the positive random variable X_i . Given $1 \leq i < I$, $p_{X_i}(x)$ is decided by the distance between the node n_i and the destination d , $|\mathbf{L}_{n_i} - \mathbf{L}_d|$. Under the condition that $|\mathbf{L}_{n_i} - \mathbf{L}_d| \rightarrow \infty$, the probability of $X_i < x$, that is, the cumulative distribution function

(cdf), is given by the probability that there is no available relay candidates to the right of the vertical line shown in Figure 16. Particularly, the relay candidates distribution is an inhomogeneous Poisson process with the density function $\lambda \cdot \rho \cdot f(P_t, R, \zeta)$, where ζ is the distance to n_i . Therefore, for $x \geq 0$, the cdf of X_i is obtained:

$$\text{Prob}(X_i \leq x) = e^{-\lambda \rho \cdot \int_x^{\infty} \int_{-\infty}^{\infty} f(P_t, R, \sqrt{u^2 + v^2}) du dv}. \quad (\text{A.2})$$

The pdf of X_i , $p_{X_i}(x)$, can be calculated by taking the differentiation of the cdf function in (A.2), which equals to the $p_{\Lambda}(x)$ defined in (9).

Therefore, $p_{X_i}(x) \rightarrow p_{\Lambda}(x)$, that is, X_i converges to Λ in distribution, when the distance $|\mathbf{L}_{n_i} - \mathbf{L}_d| \rightarrow \infty$. In obtaining the statistics of the iteration number I , we divide the source-to-destination distance l into two segments, with the lengths $l - \sqrt[3]{l}$ and $\sqrt[3]{l}$, respectively. Let I_1 denote the first passage iteration of first segment, which is $I_1 = \min\{i \mid |\mathbf{L}_{n_i} - \mathbf{L}_d| < \sqrt[3]{l}\}$, and let I_2 denote the rest number of iterations in the second segment. Given l , I_1 and I_2 are independent variables, and $I = I_1 + I_2$. Therefore, we obtain $E(I) = E(I_1) + E(I_2)$, and $\sigma^2(I) = \sigma^2(I_1) + \sigma^2(I_2)$.

According to the above definitions, under the asymptotic condition $l \rightarrow \infty$, the set of random variables $\{X_i \mid 1 \leq i \leq I_1\}$ converge to the random variable Λ in distribution. Therefore, I_1 is virtually the first passage time of a positive-drift stopped random walk beyond the level $l - \sqrt[3]{l}$, where the random walk steps are i.i.d. (The proposition is mathematically rigorous under the uniform integrability [35] of $\{|X_i|^r \mid 1 \leq i \leq I_1, r \geq 1\}$, which can be verified for realistic wireless propagation models $f(\cdot, \cdot, \cdot)$) according to the distribution $p_{\Lambda}(x)$. Due to the results of stopped random walks [36, Theorem 9.1, page 94], there are

$$E(I_1) = \frac{l}{E(\Lambda)} + o(\sqrt{l}), \quad \sigma^2(I_1) = \frac{l \cdot \sigma^2(\Lambda)}{E(\Lambda)^3} + o(l) \quad (\text{A.3})$$

Now, if we assume that $E(I) = C_1 \cdot l^{c_1} + o(l^{c_1})$, and $\sigma^2(I) = C_2 \cdot l^{c_2} + o(l^{c_2})$, where C_1, C_2, c_1, c_2 are constants, there will be $E(I_2) = C_1 \cdot l^{c_1/3} + o(l^{c_1/3})$, and $\sigma^2(I_2) = C_2 \cdot l^{c_2/3} + o(l^{c_2/3})$, since the difference between I and I_2 is only by the distances l and $\sqrt[3]{l}$ as defined. Due to the statistics of I_1 in (A.3), it is easy to obtain $C_1 = 1/E(\Lambda)$, $C_2 = \sigma^2(\Lambda)/E(\Lambda)^3$, and $c_1 = c_2 = 1$. Therefore, the results on the iteration number I , as given in (7) and (8), are obtained directly.

We further consider I_c , which denotes the number of iterations with completed DATA transmissions, that is, the number of wireless hops. Corresponding to the random variable Λ in the previous analysis, the asymptotic step length Λ_c under the prior condition of completed DATA transmission is decided by the pdf, $p_{\Lambda_c}(x) = p_{\Lambda}(x)/(1 - e^{-\lambda \rho \Omega})$, since the probability of an iteration with uncompleted DATA transmission is $e^{-\lambda \rho \Omega}$. Therefore, simply by replacing Λ with Λ_c in the previous analysis, the statistics of I_c can be obtained.

In particular, $E(I_c) = l/E(\Lambda_c) + o(\sqrt{l}) = l \cdot (1 - e^{-\lambda\rho\Omega})/E(\Lambda) + o(\sqrt{l})$, and

$$\begin{aligned}\sigma^2(I_c) &= l \cdot \frac{\sigma^2(\Lambda_c)}{E(\Lambda_c)^3} + o(l) \\ &= l \cdot \left(\frac{E(\Lambda_c^2)}{E(\Lambda_c)^3} - \frac{1}{E(\Lambda_c)} \right) + o(l) \\ &= l \cdot \left(\frac{E(\Lambda^2)}{E(\Lambda)^3} \cdot (1 - e^{-\lambda\rho\Omega})^2 - \frac{1 - e^{-\lambda\rho\Omega}}{E(\Lambda)} \right) + o(l) \\ &= l \cdot \left[\frac{\sigma^2(\Lambda)}{E(\Lambda)^3} \cdot (1 - e^{-\lambda\rho\Omega})^2 - \frac{e^{-\lambda\rho\Omega}(1 - e^{-\lambda\rho\Omega})}{E(\Lambda)} \right] + o(l).\end{aligned}\tag{A.4}$$

B. EXPRESSIONS OF $E(\Lambda)$ AND $\sigma^2(\Lambda)$

Due to the pdf $p_\Lambda(x)$ in (9), we obtain

$$\begin{aligned}E(\Lambda) &= \lim_{C \rightarrow \infty} \int_0^C x \cdot de^{-\lambda\rho \int_x^\infty g(u)du} \\ &= \lim_{C \rightarrow \infty} x \cdot e^{-\lambda\rho \int_x^\infty g(u)du} \Big|_{x=0}^C - \int_0^C e^{-\lambda\rho \int_x^\infty g(u)du} dx \\ &= \lim_{C \rightarrow \infty} C - \int_0^C e^{-\lambda\rho \int_x^\infty g(u)du} dx,\end{aligned}\tag{B.1}$$

and similarly

$$\begin{aligned}E(\Lambda^2) &= \lim_{C \rightarrow \infty} \int_0^C x^2 \cdot de^{-\lambda\rho \int_x^\infty g(u)du} \\ &= \lim_{C \rightarrow \infty} C^2 - 2 \int_0^C x \cdot e^{-\lambda\rho \int_x^\infty g(u)du} dx.\end{aligned}\tag{B.2}$$

Therefore,

$$\begin{aligned}\sigma^2(\Lambda) &= E(\Lambda^2) - E(\Lambda)^2 \\ &= \lim_{C \rightarrow \infty} 2C \cdot \int_0^C e^{-\lambda\rho \int_x^\infty g(u)du} dx \\ &\quad - \left(\int_0^C e^{-\lambda\rho \int_x^\infty g(u)du} dx \right)^2 \\ &\quad - 2 \int_0^C x \cdot e^{-\lambda\rho \int_x^\infty g(u)du} dx.\end{aligned}\tag{B.3}$$

From (B.1) and (B.3), it is shown that $E(\Lambda)$ increases with λ , approaching a limit decided by the characteristics of $g(u)$, and $\sigma^2(\Lambda)$ decreases with λ , approaching zero. Since $g(u)$, as defined in (10), is a monotonically increasing function of P_t , it is also shown that $E(\Lambda)$ increases monotonically with P_t , and $\sigma^2(\Lambda)$ decreases monotonically with P_t .

C. MODELS OF THE ITERATIVE DELAY AND ENERGY CONSUMPTION

We point out that more accurate models of the iterative delay τ_i ($1 \leq i < I$) and the iterative energy consumption ϵ_i ($1 \leq i < I$) can include the distance $|\mathbf{L}_{n_i} - \mathbf{L}_d|$. Under such models, τ_i and ϵ_i will converge to the results in Sections 4.3.1 and 4.3.2, respectively, when $|\mathbf{L}_{n_i} - \mathbf{L}_d| \rightarrow \infty$. With similar discussion in Appendix A, this will contribute to the same end-to-end analytical results, under the asymptotic condition $l \rightarrow \infty$.

ACKNOWLEDGMENTS

The authors would like to acknowledge the support from Natural Sciences and Engineering Research Council of Canada (NSERC) and Ontario Research Commercialization Program (ORCP).

REFERENCES

- [1] L. Song, "Wireless mesh infrastructure supporting broadband internet with multimedia services," in *Proceedings of the 4th IEEE International Conference on Circuits and Systems for Communications (ICCSC '08)*, pp. 792–796, Shanghai, China, May 2008.
- [2] D. Kundur, C.-Y. Lin, and C.-S. Lu, "Visual sensor networks," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, Article ID 21515, 3 pages, 2007.
- [3] ANSI/IEEE Standard for Local and Metropolitan Area Networks Part 11 (802.11), "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," revised 2003.
- [4] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MOBICOM '04)*, pp. 114–128, Philadelphia, PA, USA, September 2004.
- [5] Y. Zhu, H. Liu, M. Wu, D. Li, and S. Mathur, "Implementation experience of a prototype for video streaming over wireless mesh networks," in *Proceedings of the 4th Annual IEEE Consumer Communications and Networking Conference (CCNC '07)*, pp. 23–28, Las Vegas, Nev, USA, January 2007.
- [6] A. Kumar, D. Manjunath, and J. Kuri, *Communication Networking: An Analytical Approach*, Morgan Kaufmann, San Francisco, Calif, USA, 2004.
- [7] P. Djukic and S. Valaee, "Link scheduling for minimum delay in spatial re-use TDMA," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 28–36, Anchorage, Alaska, USA, May 2007.
- [8] L. Song, "Cognitive networks: standardizing the large scale wireless systems," in *Proceedings of the 5th IEEE Consumer Communications and Networking Conference (CCNC '08)*, pp. 988–992, Las Vegas, Nev, USA, January 2008.
- [9] L. Song and D. Hatzinakos, "Embedded wireless interconnect for sensor networks: concept and example," in *Proceedings of the 4th Annual IEEE Consumer Communications and Networking Conference (CCNC '07)*, pp. 850–854, Las Vegas, Nev, USA, January 2007.
- [10] L. Song and D. Hatzinakos, "A cross-layer architecture of wireless sensor networks for target tracking," *IEEE/ACM Transactions on Networking*, vol. 15, no. 1, pp. 145–158, 2007.

- [11] L. Song and D. Hatzinakos, "Architecture of wireless sensor networks with mobile sinks: sparsely deployed sensors," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 4, pp. 1826–1836, 2007.
- [12] L. Song, D. Hatzinakos, and X. Wang, "Wireless mesh infrastructure for ubiquitous voice and video," in *Proceedings of the 5th IEEE Consumer Communications and Networking Conference (CCNC '08)*, pp. 1267–1268, Las Vegas, Nev, USA, January 2008.
- [13] S. Biswas and R. Morris, "Opportunistic routing in multi hop wireless networks," in *Proceedings of the 2nd ACM Workshop on Hot Topics in Networks (HotNets '03)*, Cambridge, Mass, USA, November 2003.
- [14] M. Zorzi and R. R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: energy and latency performance," *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 349–365, 2003.
- [15] M. Zorzi and R. R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: multihop performance," *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 337–348, 2003.
- [16] R. C. Shah, S. Wietholter, A. Wölisz, and J. M. Rabaey, "Modeling and analysis of opportunistic routing in low traffic scenarios," in *Proceedings of the 3rd International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WIOPT '05)*, pp. 294–304, Trentino, Italy, April 2005.
- [17] A. Bletsas, A. Khisti, D. P. Reed, and A. Lippman, "A simple cooperative diversity method based on network path selection," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 3, pp. 659–672, 2006.
- [18] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 2–16, 2003.
- [19] Q. Cao and T. Abdelzaher, "Scalable logical coordinates framework for routing in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 2, no. 4, pp. 557–593, 2006.
- [20] J. Mitola III and G. Q. Maguire Jr., "Cognitive radio: making software radios more personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, 1999.
- [21] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 201–220, 2005.
- [22] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: a survey," *Computer Networks*, vol. 50, no. 13, pp. 2127–2159, 2006.
- [23] Z. J. Haas and J. Deng, "Dual busy tone multiple access (DBTMA)-a multiple access control scheme for ad hoc networks," *IEEE Transactions on Communications*, vol. 50, no. 6, pp. 975–985, 2002.
- [24] F. A. Tobagi and L. Kleinrock, "Packet switching in radio channels: part II. The hidden terminal problem in carrier sense multiple-access and the busy-tone solution," *IEEE Transactions on Communications*, vol. 23, no. 12, pp. 1417–1433, 1975.
- [25] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 364–369, Los Angeles, Calif, USA, April 2005.
- [26] IEEE Standard for Local and Metropolitan Area Networks Part 15.4 (802.15.4), "Wireless medium access control (MAC) and physical (PHY) layer specifications for low-rate wireless personal area networks (LR-WPANs)," 2003.
- [27] R. Abdelmoumen, M. Malli, and C. Barakat, "Analysis of TCP latency over wireless links supporting FEC/ARQ-SR for error recovery," in *Proceedings of IEEE International Conference on Communications (ICC '04)*, vol. 7, pp. 3994–3998, Paris, France, June 2004.
- [28] P. Gupta and P. R. Kumar, "Towards an information theory of large networks: an achievable rate region," *IEEE Transactions on Information Theory*, vol. 49, no. 8, pp. 1877–1894, 2003.
- [29] J. F. C. Kingman, *Poisson Processes*, Clarendon Press, Oxford, UK, 1993.
- [30] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall, Upper Saddle River, NJ, USA, 1996.
- [31] B. Otis, Y. H. Chee, and J. Rabaey, "A 400 μ W Rx, 1.6 mW Tx super-regenerative transceiver for wireless sensor networks," in *Proceedings of the 42nd IEEE International Solid-State Circuits Conference (ISSCC '05)*, vol. 1, pp. 396–606, San Francisco, Calif, USA, February 2005.
- [32] S. M. Ross, *Stochastic Processes*, John Wiley & Sons, New York, NY, USA, 1983.
- [33] S. Verdú, "Spectral efficiency in the wideband regime," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1319–1343, 2002.
- [34] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.
- [35] P. Billingsley, *Convergence of Probability Measures*, John Wiley & Sons, New York, NY, USA, 1968.
- [36] A. Gut, *Stopped Random Walks: Limit Theorems and Applications*, Springer, New York, NY, USA, 1988.