electronics

MDPI

Article

Labeling-Based Recipient Identification with Low-Order Modulation

Maciej Krasicki

Institute of Radiocommunications, Poznan University of Technology, Polanka 3, 61-131 Poznan, Poland; maciej.krasicki@put.poznan.pl

Abstract: Labeling-Based Recipient Identification (LABRID) brings the possibility of representing the destination station address in a Bit-Interleaved Coded Modulation with Iterative Decoding (BICM-ID) system by a signal labeling rule. Low-order modulations, such as BPSK or QPSK, pose a general problem for BICM-ID due to a limited convergence of iterative decoding. In the context of LABRID, they have one more drawback—a small number of different labeling rules in general; the number of the optimal ones, which exhibit the maximal asymptotic coding gain, is reduced even further. Meanwhile, LABRID needs a sizable collection of different optimal labeling rules to serve many users in large wireless networks. In this paper, the author suggests the use of hypercube BPSK or QPSK labeling to overcome all these challenges. By means of the Reactive Tabu Search (RTS) algorithm, more than 1500 equivalent optimal hypercube labeling rules are found. Analytical error bounds of the system are developed and supported by simulation experiments. Then, the focus is moved to the criterion to determine the frame destination at the LABRID receiver; a simple threshold-based method is proposed to keep the incorrect decision probability below 10^{-5}. Finally, it is shown that LABRID outperforms a reference BICM-ID system in terms of computational complexity.

Keywords: coded modulation; diversity; iterative decoding; labeling map

check for updates

Citation: Krasicki, M. Labeling-Based Recipient Identification with Low-Order Modulation. *Electronics* **2024**, *13*, 425. https://doi.org/10.3390/electronics13020425

Academic Editors: Lin Wang, Pingping Chen, Qiwang Chen and Xiangming Cai

Received: 6 December 2023
Revised: 4 January 2024
Accepted: 16 January 2024
Published: 19 January 2024

Copyright: © 2024 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

1.1. LABRID at a Glance

BICM-ID [1] is a variation of BICM [2,3], in which the receivers, similar to turbo decoders, can gradually improve the reliability of data estimates thanks to the iterative decoding procedure. In greater detail, the demapper and channel decoder exchange their bitwise log-likelihood beliefs on codeword bits with each other. For the process to be successful, a specific signal labeling map (or labeling, for brevity) must be used for mapping the interleaved codeword bits onto constellation points at the transmitter—it should exhibit a high asymptotic coding gain.

LABRID has been invented as a method for physical-layer addressing in wireless communications without the explicit sending of any address. Instead, the recipient ID is reflected by a unique signal labeling map. LABRID draws its strength from the observation that the iterative process at the BICM-ID receiver does not converge unless the labeling map used at the transmitter side suits the one assigned to the considered demapper. Any foreign frame (i.e., a frame destined for another recipient), can therefore be easily discriminated against due to its lack of convergence decoding, which can be detected using simple threshold-based methods—in [4] it was proposed to measure the absolute extrinsic log-likelihood ratio (LLR) mean at the demapper output, related to the codeword bits, in the last executed decoding iteration (however, one can take into consideration different measures available at the receiver, like the mutual information estimates). Hence, the decision to reject foreign frames is made in the physical layer with only marginal computational cost of obtaining a mean. For the frames identified as foreign, there is no need to run a decoding pass in the last iteration. In contrast, for conservative Medium Access Control (MAC) addressing, the likelihoods of data bits are indispensable for both desired and foreign

frames, and every decoded data frame is transferred to the MAC layer, where it undergoes integrity checking, after which the destination address might finally be read.

Along with reasonable computational payload reduction, LABRID offers a small improvement in the data rate, as there is no need to spend any bit to carry the recipient address. However, to be fair, this might be relevant only for very short data frames, which, in turn, are not preferred for BICM-ID systems due to the deficit of decoding convergence. This deficit is caused by the fact that the a priori LLRs on subsequent codeword bits are not sufficiently independent from the extrinsic decisions made about them in the previous iteration.

In Ref. [5], it has been shown that the LABRID receiver can effectively work in a "heterogeneous" environment, in which the signals of different modulation order are transmitted. In addition, the LABRID receiver demonstrated its capability of rejecting legacy BICM frames, in which the typical Gray labeling was used.

1.2. LABRID in Comparison with Other Techniques

LABRID has emerged from BICM(-ID), like many other novel transmission concepts. Therefore, for the reader's better orientation, let us consider LABRID against the background of a few competing techniques.

There are some BICM-ID enhancements, like signal-space diversity (SSD) [1] or Constellation Shaping (CS) [6,7], which aim to increase the asymptotic coding gain over that of the original BICM-ID or move the position of the turbo cliff. Actually, LABRID does not assume either signal component interleaving (as for SSD) or re-designing the signal constellation (as in the case of CS). Instead, the regular 16-QAM constellation of equiprobable points, and the error probability measures are the same as for the original BICM-ID.

LABRID can be wrongly seen as a flavor of the popular Index Modulation (IM) [8,9], where the so-called Virtual Bits determine the resources selected to transmit subsequent portions of data. The resources considered to be used with IM are, e.g., RF chains, signaling intervals, OFDM subcarriers, or even I-Q signal components. The recipient address, represented by a labeling map according to the LABRID concept, seems to be similar to Virtual Bits, which are not explicitly transmitted according to IM techniques. But note the significant difference: LABRID does not determine any selection of available resources according to the given recipient ID—all resources are still in use. What actually undergoes any manipulation in the case of LABRID is the way of bit mapping onto constellation points, with no impact on the system diversity order.

The result of LABRID use seems to be comparable to the application of scramble coding, known, e.g., from 3G telecommunication systems. In fact, the bit stream scrambled with one scrambling code is weakly correlated with that using another scrambling code. However, LABRID has nothing in common with spectrum spreading—it does not have an impact on the shape of the signal spectrum.

1.3. Research Problem Formulation

LABRID requires a sizable number of different labelings to serve a reasonable number of users. In addition, it is not enough to take into account any set of labeling maps for a given modulation order. Not only should the good labelings yield a high asymptotic coding gain over the (noniterative) Gray-labeled BICM, but also they must be not similar to each other (e.g., exchanging two constellation points' labels with each other cannot be considered a significant difference). Therefore, the author in [4] focused on a set of 768 $M16^a$-compliant 16-QAM labelings, all of which exhibit identical high-rank asymptotic coding gain. The results presented in [4] are entirely satisfactory: none of the users are discriminated against in terms of the iterative decoding results, and the receivers can correctly assess the destination of every single frame, judging by the observed convergence of the iterative process. In addition to 16-QAM, LABRID has also been studied along with 64-QAM BICM-ID with a similar, positive result [10].

For any BICM-ID system, a factor that affects the effectiveness of iterative decoding is the average distance between constellation points that differ in exactly one bit position [11]. According to widespread Gray labeling, the Hamming distance between the labels associated with neighboring constellation points is 1, which is the poorest case from the BICM-ID point of view. In general, the number of possible labelings (i.e., the ways in which binary labels can be assigned to the constellation points) is the factorial of modulation order. For better clarification: the number of candidate labelings that promise good iterative decoding performance is even smaller. Note that for low-order modulation the limit is very strict; there are only two different labelings for BPSK and $4! = 24$ for QPSK. For BPSK, bit 0 can be assigned to the constellation point $+1$ and bit 1 to -1 or vice versa. None of such two labelings can achieve any benefit from iterative decoding, as both are of the Gray type. In the case of QPSK, any labeling can be of either Gray or natural type. The Gray labelings, just like for higher-order modulations, are not worth being considered as candidate BICM-ID labelings. The natural labelings exhibit only a little gain over Gray when applied in a BICM-ID system [12]. In this light, it seems as if BICM-ID was useless in the case of low-order modulations.

Another problem appears if one aims to incorporate LABRID into QPSK BICM-ID; there are only 12 equivalent natural labelings (recall the two groups of QPSK labelings, mentioned above)—not enough for any practical use of LABRID. For BPSK the situation is even worse, as there are only two different labelings at all, none accurate for iterative decoding purposes. The question then arises whether LABRID can be used together with modulations having modulation order below 16.

The solution to the problem of poor BICM-ID performance in the case of low-order modulations was developed a couple of years ago; the so-called hypercube (or multidimensional) labeling. It consists in mapping binary tuples onto a series of symbols of the given modulation, thereby increasing channel diversity. It does not affect the data rate. In particular, for the BPSK hypercube labeling, there is still one bipolar symbol to be transmitted within one modulation period, but the values of subsequent symbols no longer depend solely on individual codeword bits. In [12], the authors developed the optimal QPSK 4-cube labeling, i.e., an assignment of 4-bit labels to pairs of QPSK symbols, whose elements are to be transmitted within two subsequent modulation periods (note that the real and imaginary components of a QPSK symbol constitute separate hypercube dimensions). Then, in [13] they considered the optimal 4-cube and 6-cube labeling for BICM-ID over different channels. In parallel, in [14] a larger variety of hypercubes consisting of QPSK or BPSK symbols was considered for transmission over the AWGN channel. In the current paper it is shown that hypercube labeling brings the possibility to merge the novel LABRID concept with low-order modulations. Indeed, thanks to the use of a general-purpose optimization algorithm, the author found more than 1500 equally good hypercube labelings of QPSK and BPSK symbols—so needed to enable physical layer recipient addressing. The other outcomes of the research are as follows: derivation of the analytical error bound of the considered system, and a proposal of a simple threshold-based criterion for assessing frame destination.

1.4. Paper Structure

The organization of this paper is as follows. Section 2 introduces the system model and describes frame filtering—the method for distinguishing foreign and desired frames. Section 3 covers the theoretical aspects of LABRID BICM-ID error bounds, specifies the requirements that should be met by the optimal hypercube labelings, and delivers exemplary outcomes of the optimization algorithm used to find them. In Section 4, the results of an overall LABRID system simulation experiment are presented and verified by means of the EXIT chart [15,16] and compared with analytical bounds. The efficiency of frame filtering is studied, and the computational cost is delivered therein. Finally, Section 5 concludes the work.

1.5. Notations

Scalars are denoted by italic letters, whereas vectors and matrices are denoted by bold-face letters (bold-face italic in the case of real or complex data type). $\mathbb{R}^N$ and $\mathbb{C}^N$ represent the N-dimensional space of real, and complex numbers, respectively. $\{0,1\}^M$ means the M-dimensional Boolean space. $\Re(\cdot)$ stands for the real part. $\overline{(\cdot)}$ and $(\cdot)^T$ mean complex conjugate and the vector/matrix transpose, respectively. $\mathbf{I}$ represents an identity matrix, while j is the imaginary unit. $E[\cdot]$ denotes the statistical expectation. The expression $\sim\mathcal{N}(0, N_0/2)$ means that the given random variable is distributed normally with mean, 0, and variance, $N_0/2$.

2. LABRID for BICM-ID with Multidimensional Labeling

2.1. System Model

The considered LABRID BICM-ID system model is shown in Figure 1.

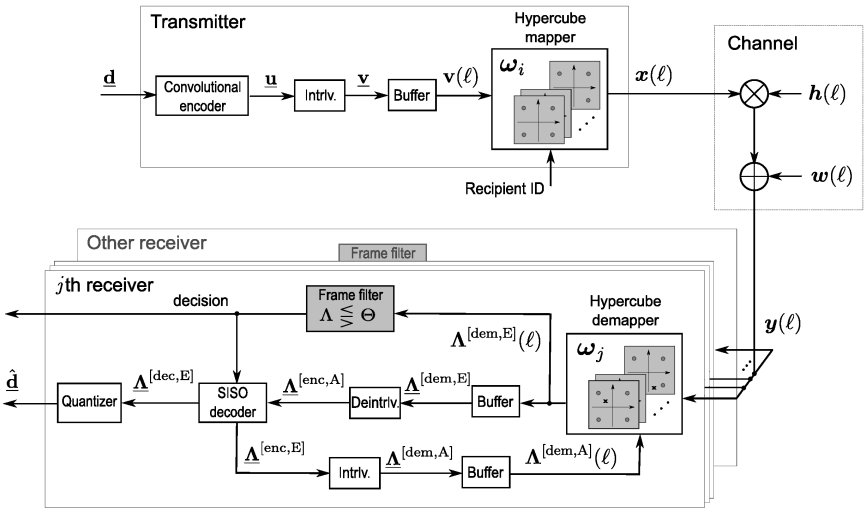


Figure 1. Model of BICM-ID system employing LABRID and multidimensional labeling.

Data frame of length K, denoted by $\underline{\mathbf{d}}$, is encoded by a convolutional encoder employing the short $[5\ 7]_8$ code, which is a common choice for BICM-ID. The resultant codeword $\underline{\mathbf{u}}$ is passed through a pseudo-random bitwise interleaver of depth equal to the codeword size. The interleaved codeword $\underline{\mathbf{v}}$ is then buffered and cut into L binary blocks $\mathbf{v}(\ell)$ of length $M = 4$, where $\ell = 1, \ldots, L$ is the time offset of a given block. Each $\mathbf{v}(\ell)$ block is mapped onto a 4-cube vertex $\boldsymbol{x}(\ell)$ according to a labeling map

$$\boldsymbol{\omega}_i : \begin{cases} \{0,1\}^M \mapsto \mathbb{R}^N & \text{for BPSK modulation } (N = M), \\ \{0,1\}^M \mapsto \mathbb{C}^N & \text{for QPSK modulation } (N = M/2). \end{cases} \tag{1}$$

Depending on the modulation type, the elements of $\boldsymbol{x}(\ell)$ belong to either BPSK signal set: $\chi_{\text{BPSK}} = \{-1, +1\}$ or QPSK signal set: $\chi_{\text{QPSK}} = \{-1{-}1j,\ -1{+}1j,\ +1{-}1j\ , {+}1{+}1j\}$.

Conforming to LABRID principles, the hypercube labeling map, $\boldsymbol{\omega}_i$, is recipient-specific. While running in the real system, the recipient ID, i, is dropped down from the MAC layer to the physical layer. Vertices $\boldsymbol{x}(\ell)$ are transmitted through the uncorrelated Rayleigh fading channel, so that the received vector

$$\boldsymbol{y}(\ell) = \boldsymbol{x}(\ell)\mathbf{I}\boldsymbol{h}^T(\ell) + \boldsymbol{w}^T(\ell). \tag{2}$$

Due to the assumption of coherent detection, the elements of $\boldsymbol{h}(\ell) \in \mathbb{R}^N$ are modeled as real i.i.d. Rayleigh RVs, all having zero mean and unitary variance; the elements of $\boldsymbol{w}(\ell) \in \mathbb{C}^N$ represent the samples of additive white Gaussian noise with zero mean and variance of $N_0/2$ in either real or imaginary dimension (the complex channel model is

valid for either QPSK and BPSK transmitted symbols). Note that the signal reaches many receivers simultaneously; only one of them is the desired frame recipient. With no generality loss, we can assume identical channel realization for each receiver, as the role of a given station might change from one event of frame transmission to another.

At the considered jth receiver, using labeling map $\boldsymbol{\omega}_j$, the hypercube demapper computes the extrinsic LLRs (i.e., log-likelihood ratios) individually per codeword bit m of block $\mathbf{v}(\ell)$ as

$$\Lambda_m^{[\mathrm{dem,E}]}(\ell) = \frac{\sum_{\mathbf{v}:v_m=1} \exp\left(\Gamma(\mathbf{v},\ell) + \boldsymbol{\Lambda}^{[\mathrm{dem,A}]}(\ell)\mathbf{v}^T\right)}{\sum_{\mathbf{v}:v_m=0} \exp\left(\Gamma(\mathbf{v},\ell) + \boldsymbol{\Lambda}^{[\mathrm{dem,A}]}(\ell)\mathbf{v}^T\right)} - \Lambda_m^{[\mathrm{dem,A}]}(\ell), \tag{3}$$

where $\mathbf{v} : v_m = b$ is a hypothetically considered binary block of length M, having $b \in \{0,1\}$ on bit m,

$$\begin{aligned} \Gamma(\mathbf{v},\ell) &= \log P_j(\boldsymbol{y}(\ell)|\mathbf{v},\boldsymbol{h}(\ell)) = \log P(\boldsymbol{y}(\ell)|\boldsymbol{x}(\mathbf{v},\boldsymbol{\omega}_j),\boldsymbol{h}(\ell)) \\ &= M\log\frac{1}{\pi N_0} - \frac{1}{N_0}\left\|\boldsymbol{y}(\ell) - \boldsymbol{x}(\mathbf{v},\boldsymbol{\omega}_j)\mathbf{I}\boldsymbol{h}^T(\ell)\right\|^2, \end{aligned} \tag{4}$$

$\boldsymbol{x}$ is a vertex assigned to $\mathbf{v}$ according to the labeling $\boldsymbol{\omega}_j$. $\|\cdot\|$ is the norm in $\mathbb{C}^N$ induced by the inner product

$$\langle \boldsymbol{a}|\boldsymbol{b}\rangle \stackrel{\mathrm{def}}{=} \sum_{n=1}^{N} a_n\bar{b}_n = \sum_{n=1}^{N}\left(a_n^I + ja_n^Q\right)\left(b_n^I - jb_n^Q\right) = \sum_{n=1}^{N}\left[a_n^I b_n^I + a_n^Q b_n^Q - j\cdot\left(a_n^I b_n^Q - a_n^Q b_n^I\right)\right], \tag{5}$$

so that $\|\boldsymbol{a}\|^2 = \sum_{n=1}^{N}|a_n|^2$. The a priori demapper information, $\boldsymbol{\Lambda}^{[\mathrm{dem,A}]}(\ell) = [\Lambda_m^{[\mathrm{dem,A}]}(\ell)]_{1\le m\le M}$ is gleaned from the Soft-Input Soft-Output (SISO) decoder [17].

The individual extrinsic demapper LLRs are merged into vectors

$$\boldsymbol{\Lambda}^{[\mathrm{dem,E}]}(\ell) = [\Lambda_m^{[\mathrm{dem,E}]}(\ell)]_{1\le m\le M}\ , \tag{6}$$

related to respective binary blocks $\mathbf{v}(\ell)$, and then into vector

$$\underline{\boldsymbol{\Lambda}}^{[\mathrm{dem,E}]} = [\boldsymbol{\Lambda}^{[\mathrm{dem,E}]}(\ell)]_{1\le\ell\le L}\ , \tag{7}$$

including the extrinsic demapper LLRs for all bits of the codeword. Then the elements of $\underline{\boldsymbol{\Lambda}}^{[\mathrm{dem,E}]}$ are deinterleaved to constitute $\underline{\boldsymbol{\Lambda}}^{[\mathrm{enc,A}]}$, i.e., the input information of the SISO decoder. Taking into account the code rate of $1/n_C$, there are n_C LLRs related to the codeword bits per each kth trellis section: $\Lambda_1^{[\mathrm{enc,A}]}(k),\ldots,\Lambda_{n_C}^{[\mathrm{enc,A}]}(k)$, which constitute a vector $\boldsymbol{\Lambda}^{[\mathrm{enc,A}]}(k)$ such that

$$\underline{\boldsymbol{\Lambda}}^{[\mathrm{enc,A}]} = \left[\boldsymbol{\Lambda}^{[\mathrm{enc,A}]}(k)\right]_{1\le k\le K}\ . \tag{8}$$

With no generality loss, let us assume that the decoder operates according to the max* principle. It can represent the accurate optimal log-MAP routine, simplified approach with the use of look-up tables as corrections, or the basic max-log-MAP, deriving its simplicity from the Jacobi logarithm approximation [18].

The decoding process in each iteration consists of four phases [17]:

2.1.1. Calculation of Edge Transition Cost

The transition cost is obtained for each e edge in every kth trellis segment as

$$\gamma(\mathsf{e},k) = [u_1(\mathsf{e})\ldots u_{n_C}(\mathsf{e})]\left[\boldsymbol{\Lambda}^{[\mathrm{enc,A}]}(k)\right]^T, \tag{9}$$

where $u_1(\mathsf{e})\ldots u_{n_C}(\mathsf{e})$ are the subsequent codeword bits assigned to the trellis edge e.

2.1.2. Calculation of State Metrics in the Forward Trellis Pass

The forward metric of state s in the k trellis segment is calculated recursively as

$$\alpha(\mathsf{s},k) = \max_{\mathsf{e}:\mathsf{s}^E(\mathsf{e})=\mathsf{s}}{}^{*}\Big\{\alpha(\mathsf{s}^S(\mathsf{e}),k-1)+\gamma(\mathsf{e},k)\Big\}, k>1, \tag{10}$$

where $\mathsf{s}^S(\mathsf{e})$ and $\mathsf{s}^E(\mathsf{e})$ are the origin and destination state of edge e, respectively. To break the recursion loop, it is assumed that the encoder starts the encoding process in the all-zero state.

2.1.3. Calculation of State Metrics in the Backward Trellis Pass

Similarly to (10), one can obtain the metric of state s in the kth trellis segment as

$$\beta(\mathsf{s},k) = \max_{\mathsf{e}:\mathsf{s}^S(\mathsf{e})=\mathsf{s}}{}^{*}\Big\{\beta(\mathsf{s}^E(\mathsf{e}),k+1)+\gamma(\mathsf{e},k)\Big\}, k<K. \tag{11}$$

Since the encoder appends a respective number of tail bits to the data frame, the backward trellis pass starts at the all-zero state.

2.1.4. Generation of LLRs

For all but the last decoding iteration, the decoder generates the extrinsic codeword-related LLRs as

$$\begin{aligned}\Lambda^{[\mathrm{enc,E}]}_{p=1,\ldots,n_C}(k) = &\max_{\mathsf{e}:u_p(\mathsf{e})=1}{}^{*}\Big\{\alpha(\mathsf{s}^S(e),k-1)+\gamma(\mathsf{e},k)+\beta(\mathsf{s}^E(e),k)\Big\}\\ &-\max_{\mathsf{e}:u_p(\mathsf{e})=0}{}^{*}\Big\{\alpha(\mathsf{s}^S(e),k-1)+\gamma(\mathsf{e},k)+\beta(\mathsf{s}^E(e),k)\Big\}-\Lambda^{[\mathrm{enc,A}]}_p(k).\end{aligned} \tag{12}$$

They are the elements of vector $\mathbf{\Lambda}^{[\mathrm{enc,E}]}(k)$ such that

$$\underline{\mathbf{\Lambda}}^{[\mathrm{enc,E}]} = \Big[\mathbf{\Lambda}^{[\mathrm{enc,E}]}(k)\Big]_{1\le k\le K}. \tag{13}$$

Passed through the interleaver, $\underline{\mathbf{\Lambda}}^{[\mathrm{enc,E}]}$ becomes a priori demapper information,

$$\underline{\mathbf{\Lambda}}^{[\mathrm{dem,A}]} = \Big[\mathbf{\Lambda}^{[\mathrm{dem,A}]}(\ell)\Big]_{1\le \ell\le L} \tag{14}$$

in the next iteration.

In the last decoding iteration, the decoder delivers the extrinsic LLRs associated with the data bits (one bit per one trellis segment):

$$\begin{aligned}\Lambda^{[\mathrm{dec,E}]}(k) = &\max_{\mathsf{e}:d(\mathsf{e})=1}{}^{*}\Big\{\alpha(\mathsf{s}^S(e),k-1)+\gamma(\mathsf{e},k)+\beta(\mathsf{s}^E(e),k)\Big\}\\ &-\max_{\mathsf{e}:d(\mathsf{e})=0}{}^{*}\Big\{\alpha(\mathsf{s}^S(e),k-1)+\gamma(\mathsf{e},k)+\beta(\mathsf{s}^E(e),k)\Big\},\end{aligned} \tag{15}$$

where $d(\mathsf{e})$ is the data bit assigned to the trellis edge e. Subsequent LLRs from (15) constitute vector

$$\underline{\mathbf{\Lambda}}^{[\mathrm{dec,E}]} = \Big[\Lambda^{[\mathrm{dec,E}]}(k)\Big]_{1\le k\le K},$$

which is, finally, passed through a quantizer to obtain an estimated data frame, $\underline{\hat{\mathbf{d}}}$.

Note that in the LABRID receiver, the last-iteration decoder pass is not run for any foreign frame, which significantly reduces the computational payload.

2.2. Frame Filtering

LABRID receivers are equipped with a frame filter, necessary to reject the frames which are not actually destined for them. The frame filter makes use of the author's observation

that the iterative decoding process in any BICM-ID receiver can converge only if both conditions are met:

- E_b/N_0 is high enough,
- the demapper labeling map $(\boldsymbol{\omega}_j)$ is identical to the map used at the transmitter $(\boldsymbol{\omega}_i)$ [4].

Convergence means that the mutual information $\mathcal{I}$ between the codeword and the extrinsic demapper or decoder LLRs increases within subsequent decoding iterations; the LLRs become more and more reliable from one iteration to another. Fortunately, there is no need to compute mutual information explicitly, as the reliability of any LLR strictly corresponds to its magnitude [19]. Consequently, the LABRID receiver computes the absolute extrinsic demapper LLR mean per frame, denoted by Λ, hereinafter, and compares it against a threshold value Θ, empirically specified. The main reason for choosing the values of the extrinsic demapper LLRs as the frame filtering criterion instead of the extrinsic decoder LLRs is that it might save the time necessary for decoder's operation in the last iteration when the frame is classified as foreign. Obviously, there is a nonzero probability that a frame that is actually destined for the considered receiver (called a desired frame, hereinafter) is wrongly classified as a foreign frame (i.e., directed to the other receiver). Such an error event will be termed misdetection. Another kind of error (called false alarm) appears if the receiver makes the decision to acquire a foreign frame.

With the aim of formalizing the probability of both types of errors, mentioned above, let $p_{\Lambda|\boldsymbol{\omega}}(\lambda)$ be the pdf of the value of Λ at the considered E_b/N_0, given the labeling map used on the transmitter side. The misdetection event at the jth receiver appears if the labeling used on the transmitter side is identical to labeling $\boldsymbol{\omega}_j$ used by the considered receiver and Λ falls below the threshold at the same time. Thus, the respective probability of misdetection error (or Misdetection Rate, MR) at the jth receiver reads

$$\mathrm{MR}_j(\Theta) = \int_0^{\Theta} p_{\Lambda|\boldsymbol{\omega}}(\lambda|\boldsymbol{\omega}_j)\mathrm{d}\lambda. \tag{16}$$

By averaging MR_j over all possible labeling maps, one obtains

$$\mathrm{MR}(\Theta) = E_j\big[\mathrm{MR}_j(\Theta)\big] = \sum_j P(\boldsymbol{\omega}_j) \int_0^{\Theta} p_{\Lambda|\boldsymbol{\omega}}(\lambda|\boldsymbol{\omega}_j)\mathrm{d}\lambda, \tag{17}$$

which seems to be a more valuable measure of frame filtering efficacy. However, all the optimal labelings—by definition—exhibit exactly the same features, so the MR calculated for any receiver j gives the value of the general MR. In other words, the conditional Λ pdfs are identical for all equivalent labelings. Consequently, to calculate MR, one can assume any optimal labeling per frame, necessarily the same at both sides of the system, which is represented by condition $\boldsymbol{\omega}_{i=j}$ in the final MR definition:

$$\mathrm{MR}(\Theta) = \int_0^{\Theta} p_{\Lambda|\boldsymbol{\omega}}(\lambda|\boldsymbol{\omega}_{i=j})\mathrm{d}\lambda. \tag{18}$$

By analogy to MR, let us define False Alarm Rate (FAR) as

$$\mathrm{FAR}(\Theta) = \int_{\Theta}^{\infty} p_{\Lambda|\boldsymbol{\omega}}(\lambda|\boldsymbol{\omega}_{i\neq j})\mathrm{d}\lambda, \tag{19}$$

where the condition is that the receiver labeling is different from that used on the transmitter side, though both are equivalent optimal labelings.

The definitions of MR and FAR will be used in the experimental part (Section 4.4) to determine the optimal threshold values per the assumed values of E_b/N_0.

3. Optimal Hypercube Labeling Set

3.1. Error-Free Feedback BER Bound

BICM-ID achieves the Error-Free Feedback (EF) state having performed a sufficient number of iterations, unless E_b/N_0 is too low for the iterative process to converge. Consider the binary blocks $\mathbf{v}$ and $\mathbf{v}^{\neg m}$, which are identical except for bit m. The results of $\mathbf{v}$ and $\mathbf{v}^{\neg m}$ mapping into the M-cube are the $\boldsymbol{x} = [x_1, \ldots, x_n, \ldots, x_N]$ and $\boldsymbol{x}^{\neg m} = \left[x_{1,}^{\neg m}, \ldots, x_n^{\neg m}, \ldots, x_N^{\neg m}\right]$ vertices, respectively. The elements of $\boldsymbol{x}$ and $\boldsymbol{x}^{\neg m}$ are regular constellation points of BPSK (in such a case, $M = N$) or QPSK ($M = 2N$), which means that a binary vector of length $M = 4$ can be mapped onto a sequence of 4 BPSK signals or 2 QPSK signals. Note the same hypercube size (or number of dimensions) in both cases. Let $\boldsymbol{x}$ be the actually transmitted symbol. For the EF case in BICM-ID, the demapper's decision on the mth bit of block $\mathbf{v}$ mapped onto $\boldsymbol{x}$ is incorrect if the distance of the received vertex $\boldsymbol{y}$ from $\boldsymbol{x}$ is longer than that from competing vertex $\boldsymbol{x}^{\neg m}$ (all other competing vertices can be expurgated, since all remaining bits of $\mathbf{v}$ are assumed to be ideally estimated [1]). Accordingly, let us consider the metric difference $\Delta \equiv \Delta(\boldsymbol{x}, \boldsymbol{x}^{\neg m})$ between $\boldsymbol{x}$ and $\boldsymbol{x}^{\neg m}$ when $\boldsymbol{y}$ is received, given the current channel state $\boldsymbol{h}$ and noise vector $\boldsymbol{w}$. Unlike for the regular BICM-ID system from [1], the metric difference reflects hypercube labeling (for brevity, let $\boldsymbol{x}^{\neg m}$ be replaced by $\boldsymbol{z}$ when considering a given $(\boldsymbol{x}, \boldsymbol{x}^{\neg m})$ pair):

$$\begin{aligned}\Delta &= \left\|\boldsymbol{y} - \boldsymbol{z}\mathbf{I}\boldsymbol{h}^T\right\|^2 - \left\|\boldsymbol{y} - \boldsymbol{x}\mathbf{I}\boldsymbol{h}^T\right\|^2 = \left\|\boldsymbol{x}\mathbf{I}\boldsymbol{h}^T + \boldsymbol{w}^T - \boldsymbol{z}\mathbf{I}\boldsymbol{h}^T\right\|^2 - \left\|\boldsymbol{x}\mathbf{I}\boldsymbol{h}^T + \boldsymbol{w}^T - \boldsymbol{x}\mathbf{I}\boldsymbol{h}^T\right\|^2 \\ &= \left\|\boldsymbol{x}\mathbf{I}\boldsymbol{h}^T + \boldsymbol{w}^T - \boldsymbol{z}\mathbf{I}\boldsymbol{h}^T\right\|^2 - \left\|\boldsymbol{x}\mathbf{I}\boldsymbol{h}^T + \boldsymbol{w}^T - \boldsymbol{x}\mathbf{I}\boldsymbol{h}^T\right\|^2 = \left\|(\boldsymbol{x} - \boldsymbol{z})\mathbf{I}\boldsymbol{h}^T + \boldsymbol{w}^T\right\|^2 - \left\|\boldsymbol{w}^T\right\|^2 \\ &= \langle(\boldsymbol{x} - \boldsymbol{z})\mathbf{I}\boldsymbol{h}^T + \boldsymbol{w}^T | (\boldsymbol{x} - \boldsymbol{z})\mathbf{I}\boldsymbol{h}^T + \boldsymbol{w}^T\rangle - \left\|\boldsymbol{w}^T\right\|^2. \end{aligned} \tag{20}$$

Repeated use of the following features of the inner product:

$$\langle \boldsymbol{a} + \boldsymbol{b} | \boldsymbol{c}\rangle = \langle \boldsymbol{a} | \boldsymbol{c}\rangle + \langle \boldsymbol{b} | \boldsymbol{c}\rangle \tag{21}$$

and

$$\langle \boldsymbol{a} | \boldsymbol{b}\rangle = \overline{\langle \boldsymbol{b} | \boldsymbol{a}\rangle} \tag{22}$$

gives

$$\begin{aligned}\Delta &= \langle(\boldsymbol{x} - \boldsymbol{z})\mathbf{I}\boldsymbol{h}^T | (\boldsymbol{x} - \boldsymbol{z})\mathbf{I}\boldsymbol{h}^T\rangle + \langle(\boldsymbol{x} - \boldsymbol{z})\mathbf{I}\boldsymbol{h}^T | \boldsymbol{w}^T\rangle \\ &+ \overline{\langle(\boldsymbol{x} - \boldsymbol{z})\mathbf{I}\boldsymbol{h}^T | \boldsymbol{w}^T\rangle} = \left\|(\boldsymbol{x} - \boldsymbol{z})\mathbf{I}\boldsymbol{h}^T\right\|^2 + 2\Re\left[\langle(\boldsymbol{x} - \boldsymbol{z})\mathbf{I}\boldsymbol{h}^T | \boldsymbol{w}^T\rangle\right]. \end{aligned} \tag{23}$$

Since

$$\left\|(\boldsymbol{x} - \boldsymbol{z})\mathbf{I}\boldsymbol{h}^T\right\|^2 = \sum_{n=1}^{N} |(x_n - z_n)h_n|^2 = \sum_{n=1}^{N} h_n^2 |x_n - z_n|^2 \tag{24}$$

and

$$\Re\left[\langle(\boldsymbol{x} - \boldsymbol{z})\mathbf{I}\boldsymbol{h}^T | \boldsymbol{w}^T\rangle\right] = \textstyle\sum_{n=1}^{N} h_n \cdot \left[w_n^I \cdot \left(x_n^I - z_n^I\right) + w_n^Q \cdot \left(x_n^Q - z_n^Q\right)\right], \tag{25}$$

it can be observed that

$$\Delta = \textstyle\sum_{n=1}^{N} \Delta_n, \tag{26}$$

where

$$\Delta_n \equiv \Delta_n(\boldsymbol{x}, \boldsymbol{z}) = h_n^2 |x_n - z_n|^2 + 2h_n \cdot \left[w_n^I \cdot \left(x_n^I - z_n^I\right) + w_n^Q \cdot \left(x_n^Q - z_n^Q\right)\right]. \tag{27}$$

Note that (25) is valid only in the case of coherent detection, which allows for the assumption that $\boldsymbol{h} \in \mathbb{R}^N$.

Given h_n, metric difference Δ_n is a linear combination of two i.i.d. RVs: $w_n^I, w_n^Q \sim \mathcal{N}(0, N_0/2)$, and—as such—is the Gaussian RV, too. From $E[w_n^I] = E[w_n^Q] = 0$ it can immediately be inferred that the mean of Δ_n is

$$\mu_{\Delta_n} = E_{w_n^I, w_n^Q}[\Delta_n] = h_n^2 |x_n - z_n|^2. \tag{28}$$

Accordingly, the variance of Δ_n yields

$$\sigma_{\Delta_n}^2 = E_{w_n^I, w_n^Q}\left[\Delta_n^2\right] - \mu_{\Delta_n}^2, \tag{29}$$

where

$$\begin{aligned} E_{w_n^I, w_n^Q}\left[\Delta_n^2\right] =& \mu_{\Delta_n}^2 + 2h_n^2|x_n - z_n|^2 \cdot 2E_{w_n^I, w_n^Q}\left[h_n \cdot \left[w_n^I \cdot \left(x_n^I - z_n^I\right) + w_n^Q \cdot \left(x_n^Q - z_n^Q\right)\right]\right] \\ &+ E_{w_n^I, w_n^Q}\left[2h_n \cdot \left[w_n^I \cdot \left(x_n^I - z_n^I\right) + w_n^Q \cdot \left(x_n^Q - z_n^Q\right)\right]\right]^2 \end{aligned} \tag{30}$$

Since the real and imaginary noise components are mutually independent,

$$\begin{aligned} E_{w_n^I, w_n^Q}\left[\Delta_n^2\right] &= \mu_{\Delta_n}^2 + 4h_n^3|x_n - z_n|^2 \left[\left(x_n^I - z_n^I\right)\cancelto{0}{E\left[w_n^I\right]} + \left(x_n^Q - z_n^Q\right)\cancelto{0}{E\left[w_n^Q\right]}\right] \\ &+ 4h_n^2 E_{w_n^I, w_n^Q}\left[w_n^I \cdot \left(x_n^I - z_n^I\right) + w_n^Q \cdot \left(x_n^Q - z_n^Q\right)\right]^2 \\ &= \mu_{\Delta_n}^2 + 4h_n^2 E_{w_n^I, w_n^Q}\left[\left(w_n^I\right)^2\left|x_n^I - z_n^I\right|^2 + 2w_n^I w_n^Q \cdot \left(x_n^I - z_n^I\right) \cdot \left(x_n^Q - z_n^Q\right) + \left(w_n^Q\right)^2\left|x_n^Q - z_n^Q\right|^2\right] \\ &= \mu_{\Delta_n}^2 + 4h_n^2 \cdot \left[E\left[w_n^I\right]^2\left|x_n^I - z_n^I\right|^2 + E\left[w_n^Q\right]^2\left|x_n^Q - z_n^Q\right|^2\right]. \end{aligned} \tag{31}$$

In this position, to obtain $E\left[w_n^I\right]^2$ and $E\left[w_n^Q\right]^2$ it is enough to take into account that by definition $\sigma_{w_n^{I/Q}}^2 = E\left[w_n^{I/Q}\right]^2 - E^2\left[w_n^{I/Q}\right] = N_0/2$. Knowing that $E\left[w_n^{I/Q}\right] = 0$, one obtains $E\left[w_n^{I/Q}\right]^2 = N_0/2$, and then

$$E\left[\Delta_n^2\right] = \mu_{\Delta_n}^2 + 4h_n^2 \frac{N_0}{2}\left(\left|x_n^I - z_n^I\right|^2 + \left|x_n^Q - z_n^Q\right|^2\right) = \mu_{\Delta_n}^2 + 2N_0 h_n^2 |x_n - z_n|^2. \tag{32}$$

Finally, it is enough to apply the result of (32) to (29) to obtain

$$\sigma_{\Delta_n}^2 = 2N_0 h_n^2 |x_n - z_n|^2. \tag{33}$$

For the future use let us specify the bilateral Laplace transform of the pdf of the metric difference Δ from (20):

$$\Phi_\Delta(s) = \int_{-\infty}^{\infty} e^{-s\delta} f_\Delta(\delta) \mathrm{d}\delta = E_{\boldsymbol{h},\boldsymbol{w}}\left[e^{-s\Delta}\right]. \tag{34}$$

For the given h_n, since $\Delta = \sum_{n=1}^{N} \Delta_n$, all Δ_n terms are i.i.d. Gaussian RVs, so

$$\Phi_\Delta(s) = E_{\boldsymbol{h},\boldsymbol{w}}\left[e^{-s\sum_n \Delta_n}\right] = E_{\boldsymbol{h}}\left[\prod_{n=1}^{N} E_{w_n^I, w_n^Q}\left[e^{-s\Delta_n}\right]\right]. \tag{35}$$

Since $E[e^{-sa}] = \exp(-s\mu_a + s^2\sigma_a^2/2)$ for any Gaussian RV a, it is also valid for every Δ_n:

$$E_{w_n^I, w_n^Q}\left[e^{-s\Delta_n}\right] = \exp\left(-s\mu_{\Delta_n} + \frac{s^2\sigma_{\Delta_n}^2}{2}\right). \tag{36}$$

Taking into account that the individual elements of $\boldsymbol{h}$ are i.i.d. RVs,

$$\Phi_\Delta(s) = \prod_{n=1}^{N} E_{h_n}\left[\exp\left\{h_n^2 s \cdot (N_0 s - 1)|x_n - z_n|^2\right\}\right]. \tag{37}$$

In (37), one can recognize the moment generating function (MGF) of a normalized Rayleigh-distributed RV $R := h_n^2$, which yields

$$E\left[e^{aR^2}\right] = \frac{1}{1-a}. \tag{38}$$

The convergence criterion of MGF is $\Re\{a\} < 1$. Accordingly, by assigning

$$a := s \cdot (N_0 s - 1)|x_n - z_n|^2, \tag{39}$$

one finally arrives at

$$\Phi_\Delta(s) = \prod_{n=1}^{N} \frac{1}{1 - s \cdot (N_0 s - 1)|x_n - z_n|^2}. \tag{40}$$

The obtained Laplace transform (40) has $2N$ purely real poles, i.e., there is an

$$\left(\epsilon_n^- = \frac{1-\sqrt{1+4N_0/|x_n-z_n|^2}}{2N_0} < 0,\ \epsilon_n^+ = \frac{1+\sqrt{1+4N_0/|x_n-z_n|^2}}{2N_0} > 0\right)$$

pair for each n, and it can converge in the region where

$$\Re\{s\} \in (\max(\epsilon_1^-, \ldots, \epsilon_N^-), \min(\epsilon_1^+, \ldots, \epsilon_N^+)).$$

Note that the restriction of Φ_Δ to that interval is a real, positive and convex function with the minimum at $s = 1/(2N_0)$ for any $(\boldsymbol{x}, \boldsymbol{z})$ pair.

Now, let us study the overall system performance. Consider d subsequent bits of the codeword: $u_{k+1}, \ldots, u_{k+d}$, each equally likely to be 0 or 1 ($0 \le k \le K - d$). After interleaving, they are mapped onto random positions $m(\iota_1), \ldots, m(\iota_d)$ of the labels of randomly selected vertices $\boldsymbol{x}(\iota_1), \ldots, \boldsymbol{x}(\iota_d)$ constituting a sequence $\underline{\underline{\boldsymbol{x}}}$. That is, bit u_{k+1} is mapped onto position $m(\iota_1)$ of the label of vertex $\boldsymbol{x}(\iota_1)$, and so on. Then, let $\underline{\underline{\boldsymbol{z}}}$ be a sequence of vertices $\boldsymbol{z}(\iota_1), \ldots, \boldsymbol{z}(\iota_d)$, whose labels are identical to those of $\underline{\underline{\boldsymbol{x}}}$ except for the positions $m(\iota_1), \ldots, m(\iota_d)$, respectively. For the given codeword, the error event of Hamming distance d occurs when the cumulative metric difference

$$\underline{\underline{\Delta}} = \Delta(\boldsymbol{x}(\iota_1), \boldsymbol{z}(\iota_1)) + \cdots + \Delta(\boldsymbol{x}(\iota_d), \boldsymbol{z}(\iota_d)) \le 0.$$

Having said this, recall the EF BER bound for a BICM-ID system employing a rate-k_c/n_c convolutional code [1,2]:

$$P_b \le \frac{1}{k_c} \sum_{d=d_{\text{free}}}^{\infty} B_d f_{\text{EF}}(d, \boldsymbol{\omega}, \boldsymbol{\chi}), \tag{41}$$

where B_d is the total input weight of the error events at Hamming distance d, d_{free} is the free distance of the code, and $\boldsymbol{\chi}$ is the set of all hypercubes. The pairwise error probability (PEP)

$$f_{\text{EF}}(d, \boldsymbol{\omega}, \boldsymbol{\chi}) = E_{\boldsymbol{x}(\iota_1),\ldots,\boldsymbol{x}(\iota_d),m(\iota_1),\ldots,m(\iota_d)}(P(\underline{\underline{\Delta}} \le 0)). \tag{42}$$

A general formula for $P(\underline{\underline{\Delta}} \leq 0)$, derived in [20], is

$$P(\underline{\underline{\Delta}} \leq 0) = \frac{1}{2\pi j} \int_{\epsilon - j\infty}^{\epsilon + j\infty} \Phi_{\underline{\underline{\Delta}}}(s) \frac{\mathrm{d}s}{s}, \tag{43}$$

where

$$\Phi_{\underline{\underline{\Delta}}}(s) = \int_{-\infty}^{\infty} e^{-s\delta} f_{\underline{\underline{\Delta}}}(\delta) \mathrm{d}\delta = E_{\boldsymbol{h}(\iota_1),\ldots,\boldsymbol{h}(\iota_d),\boldsymbol{w}(\iota_1),\ldots,\boldsymbol{w}(\iota_d)} \left[e^{-s\underline{\underline{\Delta}}} \right] = \prod_{b=\iota_1,\ldots,\iota_d} \Phi_{\Delta(\boldsymbol{x}(b),\boldsymbol{z}(b))}(s) \tag{44}$$

is the bilateral Laplace transform of the cumulative metric difference and ϵ belongs to the intersection of the regions of convergence of all $\Phi_{\Delta(\boldsymbol{x}(b),\boldsymbol{z}(b))}$ terms with the real positive line. The last transformation in (44) is valid as all $\boldsymbol{h}(\iota_1), \ldots, \boldsymbol{h}(\iota_d), \boldsymbol{w}(\iota_1), \ldots, \boldsymbol{w}(\iota_d)$ terms are mutually independent RVs. Consequently,

$$\begin{aligned}
f_{\mathrm{EF}}(d, \boldsymbol{\omega}, \boldsymbol{\chi}) &= \frac{1}{2\pi j} E_{\boldsymbol{x}(\iota_1),\ldots,\boldsymbol{x}(\iota_d),m(\iota_1),\ldots,m(\iota_d)} \left[\int_{\epsilon - j\infty}^{\epsilon + j\infty} \prod_{b=\iota_1,\ldots,\iota_d} \Phi_{\Delta(\boldsymbol{x}(b),\boldsymbol{z}(b))}(s) \frac{\mathrm{d}s}{s} \right] \\
&= \frac{1}{2\pi j} \int_{\epsilon - j\infty}^{\epsilon + j\infty} E_{\boldsymbol{x}(\iota_1),\ldots,\boldsymbol{x}(\iota_d),m(\iota_1),\ldots,m(\iota_d)} \left[\prod_{b=\iota_1,\ldots,\iota_d} \Phi_{\Delta(\boldsymbol{x}(b),\boldsymbol{z}(b))}(s) \right] \frac{\mathrm{d}s}{s} \\
&\overset{\text{a)}}{=} \frac{1}{2\pi j} \int_{\epsilon - j\infty}^{\epsilon + j\infty} \prod_{b=\iota_1,\ldots,\iota_d} E_{\boldsymbol{x}(b),m(b)} \left[\Phi_{\Delta(\boldsymbol{x}(b),\boldsymbol{z}(b))}(s) \right] \frac{\mathrm{d}s}{s} \\
&\overset{\text{b)}}{=} \frac{1}{2\pi j} \int_{\epsilon - j\infty}^{\epsilon + j\infty} \left\{ E_{\boldsymbol{x},m} \left[\Phi_{\Delta(\boldsymbol{x},\boldsymbol{z})}(\mathrm{s}) \right] \right\}^d \frac{\mathrm{d}s}{s} \\
&= \frac{1}{2\pi j} \int_{\epsilon - j\infty}^{\epsilon + j\infty} \left(\frac{1}{M 2^M} \sum_{\boldsymbol{x} \in \boldsymbol{\chi}} \sum_{m=1}^{M} \prod_{n=1}^{N} \frac{1}{1 + |x_n - x_n^{\neg m}|^2 / 4N_0} \right) \frac{\mathrm{d}s}{s},
\end{aligned} \tag{45}$$

where (a) is valid because all $\boldsymbol{x}(b)$ and $m(b)$ RVs are mutually independent, and (b) is obtained because of the identical (uniform) probability distribution of all $\boldsymbol{x}(b)$'s and the same for all $m(b)$'s. Computing the integral from (45) seems to be a challenging task; the author of the paper applies a method based on numerical integration, detailed in (Section 13.3.2 in [21]).

The developed tight EF bounds will be used in the further part of the paper to verify the accuracy of simulation experiments.

3.2. Properties of Optimal Hypercube Labeling

The tight EF bounds are quite impractical when comparing different labeling maps in terms of their impact on PEP. Therefore, let us resort to the loose Chernoff PEP bound (Sections 12.1.1, 13.3 and 13.5 in [21]):

$$\begin{aligned}
f_{\mathrm{EF,Chernoff}}(d, \boldsymbol{\omega}, \boldsymbol{\chi}) &\leq \min_{\epsilon \in \mathbb{R}, \epsilon > 0} E_{\boldsymbol{x}(\iota_1),\ldots,\boldsymbol{x}(\iota_d),m(\iota_1),\ldots,m(\iota_d)} \left[\Phi_{\underline{\underline{\Delta}}}(\epsilon) \right] \\
&= \left(\frac{1}{M 2^M} \sum_{\boldsymbol{x} \in \boldsymbol{\chi}} \sum_{m=1}^{M} \prod_{n=1}^{N} \frac{1}{1 + |x_n - x_n^{\neg m}|^2 / 4N_0} \right)^d.
\end{aligned} \tag{46}$$

Now, it is easy to determine the impact of the constellation type and labeling on the system performance by introducing a distance parameter

$$\xi(\boldsymbol{\omega}, \boldsymbol{\chi}) = \sum_{\boldsymbol{x} \in \boldsymbol{\chi}} \sum_{m=1}^{M} \prod_{n=1}^{N} \left(1 + \frac{|x_n - x_n^{\neg m}|^2}{4N_0} \right)^{-1}. \tag{47}$$

It is clear that $f_{\mathrm{EF,Chernoff}}(d, \boldsymbol{\omega}, \boldsymbol{\chi})$ is proportional to $[\xi(\boldsymbol{\omega}, \boldsymbol{\chi})]^d$, so the optimal hypercube labeling map for the given hypercube is

$$\boldsymbol{\omega}_{\mathrm{opt}}(\boldsymbol{\chi}) = \arg\min_{\boldsymbol{\omega}} \xi(\boldsymbol{\omega}, \boldsymbol{\chi}). \tag{48}$$

In [12], the authors considered the optimal labeling map for the case of an $(M = 4)$-cube consisting of $n = 2$ QPSK signals. However, their approach was more general, as they focused on hypercube dimensions rather than component signals. In particular, the 4-cube is made up of four orthogonal coordinates, which may represent either four subsequent BPSK symbols or two pairs of real and imaginary QPSK symbol components. Consequently, let us represent the vertices by a series of M coordinates: $\boldsymbol{x}$ by $\boldsymbol{c} \equiv (c_1, \ldots c_M)$, and $\boldsymbol{x}^{\neg m}$ by $\boldsymbol{c}^{\neg m} \equiv (c_1^{\neg m}, \ldots, c_M^{\neg m})$. The individual elements of $\boldsymbol{c}$ and $\boldsymbol{c}^{\neg m}$ will be referred to using index n'.

The distance parameter (47), expressed explicitly with the coordinates, takes a different form for the two hypercube variants, i.e.,

$$\xi_{\mathrm{BPSK}}(\boldsymbol{\omega}, \boldsymbol{\chi}) = \sum_{\boldsymbol{x} \in \boldsymbol{\chi}} \sum_{m=1}^{4} \prod_{n'=1}^{4} \left(1 + \frac{\left| c_{n'} - c_{n'}^{\neg m} \right|^2}{4N_0} \right)^{-1}, \tag{49}$$

and

$$\begin{aligned} \xi_{\mathrm{QPSK}}(\boldsymbol{\omega}, \boldsymbol{\chi}) &= \sum_{\boldsymbol{x} \in \boldsymbol{\chi}} \sum_{m=1}^{4} \prod_{n=1}^{2} \left(1 + \frac{\left| x_n - x_n^{\neg m} \right|^2}{4N_0} \right)^{-1} \\ &= \sum_{\boldsymbol{x} \in \boldsymbol{\chi}} \sum_{m=1}^{4} \left(1 + \frac{1}{4N_0} \underbrace{\left(\left| c_1 - c_1^{\neg m} \right|^2 + \left| c_2 - c_2^{\neg m} \right|^2 \right)}_{\left| x_1 - x_1^{\neg m} \right|^2} \right)^{-1} \cdot \left(1 + \frac{1}{4N_0} \underbrace{\left(\left| c_3 - c_3^{\neg m} \right|^2 + \left| c_4 - c_4^{\neg m} \right|^2 \right)}_{\left| x_2 - x_2^{\neg m} \right|^2} \right)^{-1} \end{aligned} \tag{50}$$

for BSPK and QPSK, respectively.

Let us consider the distances between vertices $\boldsymbol{x}$ and $\boldsymbol{x}^{\neg m}$, whose labels ($\mathbf{v}$ and $\mathbf{v}^{\neg m}$) differ at position m. Having M hypercube dimensions, the same or opposite coordinates for $\boldsymbol{c}$ and $\boldsymbol{c}^{\neg m}$ can be assigned in any dimension, as

$$c_{n'},\, c_{n'}^{\neg m} \in \{-1, +1\}, n' = 1, \ldots, M,\, m = 1, \ldots, M.$$

Consequently, the distance between $c_{n'}$ and $c_{n'}^{\neg m}$ can be either 0 or $|+1 - (-1)|^2 = 4$.

Let us define a set containing all one-dimensional distances for a given $(\boldsymbol{x}, \boldsymbol{x}^{\neg m})$ pair

$$\mathcal{D}(\boldsymbol{x}, m) = \left\{ \left| c_{n'} - c_{n'}^{\neg m} \right|^2 : n' = 1, \ldots, M \right\}, \tag{51}$$

and respective Euclidean distance

$$\zeta(\boldsymbol{x}, m) = \sum_{n'=1}^{M} \left| c_{n'} - c_{n'}^{\neg m} \right|^2. \tag{52}$$

As one is interested in maximizing the actual distance parameter related to labeling (47), they should maximize ζ for all vertices and all m, regardless of the hypercube variant considered. It would be achieved if the vertices $\boldsymbol{x}$ and $\boldsymbol{x}^{\neg m}$ differed from each other in all dimensions, for every m, so that $\forall_{\boldsymbol{x}} \forall_m \mathcal{D}(\boldsymbol{x}, m) = \{4, 4, 4, 4\}$ and $\zeta(\boldsymbol{x}, m) = 16$. But here the constraints caused by the hypercube properties must be taken into account [13]. That is, for a given $\boldsymbol{x}$, the desired content of $\mathcal{D}$ can appear only for one value of m. To better explain it, without loss of generality, let $\boldsymbol{c} = (-1, -1, -1, -1)$ be assigned to a binary block $\mathbf{v} = (1000)$. It is obvious that there is only one competing vertex, i.e., $(+1, +1, +1, +1)$, with the highest

distance ζ from $\boldsymbol{c}$. It can be labeled, e.g., by $\mathbf{v}^{\neg 1} = (\mathbf{0}000)$, so let us call it $\boldsymbol{c}^{\neg 1}$. Now let us consider the remaining second furthest vertices that can be assigned to the other binary blocks at Hamming distance 1 to $\mathbf{v}$ ($\mathbf{v}^{\neg 2} = (0\mathbf{1}00)$, $\mathbf{v}^{\neg 3} = (00\mathbf{1}0)$, $\mathbf{v}^{\neg 4} = (000\mathbf{1})$). There are four equally good candidates: the vertices differing in three out of four coordinates from $\boldsymbol{c}$, namely $(-1,+1,+1,+1)$, $(+1,-1,+1,+1)$, $(+1,+1,-1,+1)$, $(+1,+1,+1,-1)$. Each of them yields $\mathcal{D} = \{4,4,4,0\}$ and $\zeta = 12$. To conclude this part, let us formulate Corollary 1.

Corollary 1. *For the best 4-cube labeling in the BICM-ID system, a distance spectrum*

$$\mathcal{Z}(\boldsymbol{x}) = \{\zeta(\boldsymbol{x}, m) : m \in 1, \ldots, M\} \tag{53}$$

contains the entries $\{16, 12, 12, 12\}$ *for every vertex* $\boldsymbol{x}$.

The corollary reflects the limits caused by the construction of the 4-cube. Any labeling that conforms to these limits will be considered the ideal labeling. However, in advance, it is not guaranteed that one exists. Note that Corollary 1 is valid for both BPSK and QPSK constellations used to create the 4-cube. In (53), there is only one entry bigger than the rest—in the case of the QPSK constellation it might be a real or imaginary component of either the first ($n = 1$) or the second ($n = 2$) symbol of the vertex. The final assignment has no impact on the content of $\mathcal{Z}(\boldsymbol{x})$.

Now, let us recall (50)—the distance parameter for QPSK—in the context of the considered example. For the most distant competing vertex, both component QPSK signals differ from their $\boldsymbol{x}$ counterparts in both real and imaginary coordinates. Meanwhile, for the remaining three competing vertices (with the second furthest distance), one of the coordinates (real or imaginary, of either the first or second component QPSK signal) is the same as for $\boldsymbol{x}$. In the case of the ideal labeling (constrained only by the properties of the hypercube), the same pattern appears for every vertex $\boldsymbol{x}$. Consequently, the limit for any QPSK hypercube labeling is constituted by

$$\begin{aligned}
\xi_{\mathrm{QPSK}}(\boldsymbol{\omega}_{\mathrm{opt}}, \boldsymbol{\chi}) &\geq \frac{1}{M \cdot 2^M} \sum_{\boldsymbol{x} \in \boldsymbol{\chi}} \left\{ \left[\left(1 + \frac{8}{4N_0}\right)^{-1} \right]^2 + (M-1) \left(1 + \frac{8}{4N_0}\right)^{-1} \left(1 + \frac{4}{4N_0}\right)^{-1} \right\} \\
&= \frac{1}{M} \left\{ \left[\left(1 + \frac{2}{N_0}\right)^{-1} \right]^2 + (M-1) \left(1 + \frac{2}{N_0}\right)^{-1} \left(1 + \frac{1}{N_0}\right)^{-1} \right\},
\end{aligned} \tag{54}$$

where the distance of 8 represents the case where both QPSK coordinates of $x_n^{\neg m}$ are different from their counterparts x_n, and 4 appears for the cases where only one coordinate (real or imaginary) is different. The last transition results from the assumption that the optimal distance pattern features in every $\boldsymbol{x} \in \boldsymbol{\chi}$, so the expectation over $\boldsymbol{x}$ disappears. For $E_b/N_0 \to \infty$, one can assume $\left(1 + \frac{E_b}{N_0}\right)^{-1} \approx \left(\frac{E_b}{N_0}\right)^{-1}$, which yields a simpler loose bound:

$$\begin{aligned}
\xi_{\mathrm{QPSK}}(\boldsymbol{\omega}_{\mathrm{opt}}, \boldsymbol{\chi}) &\gtrapprox \frac{1}{M} \left[\left(\frac{2}{N_0}\right)^{-2} + (M-1) \left(\frac{2}{N_0}\right)^{-1} \left(\frac{1}{N_0}\right)^{-1} \right] \\
&= \frac{1}{M} \left[\frac{N_0^2}{4} + (M-1) \cdot \frac{N_0^2}{2} \right].
\end{aligned} \tag{55}$$

In the case of a hypercube based on BPSK symbols, the individual hypercube coordinates are de facto single BPSK symbols, transmitted during separate modulation intervals. Consequently,

$$\xi_{\mathrm{BPSK}}(\boldsymbol{\omega}_{\mathrm{opt}},\boldsymbol{\chi}) \geq \frac{1}{M\cdot 2^M}\sum_{\boldsymbol{x}\in\boldsymbol{\chi}}\left\{\left[\left(1+\frac{4}{4N_0}\right)^{-1}\right]^4+(M-1)\left[\left(1+\frac{4}{4N_0}\right)^{-1}\right]^3\right\}$$
$$=\frac{1}{M}\left\{\left(1+\frac{1}{N_0}\right)^{-4}+(M-1)\left(1+\frac{1}{N_0}\right)^{-3}\right\}. \tag{56}$$

Again, for $E_b/N_0 \to \infty$

$$\xi_{\mathrm{BPSK}}(\boldsymbol{\omega}_{\mathrm{opt}},\boldsymbol{\chi}) \gtrapprox \frac{1}{M}\left[\left(\frac{1}{N_0}\right)^{-4}+(M-1)\left(\frac{1}{N_0}\right)^{-3}\right]=\frac{N_0^3}{M}(N_0+M-1). \tag{57}$$

If one assumes reasonably small $N_0(<0.1)$ and $M=4$ when comparing the approximate minimal values of ξ_{BPSK} and ξ_{QPSK}, they obtain

$$\xi_{\mathrm{BPSK}}(\boldsymbol{\omega}_{\mathrm{opt}},\boldsymbol{\chi})/\xi_{\mathrm{QPSK}}(\boldsymbol{\omega}_{\mathrm{opt}},\boldsymbol{\chi})=\frac{4N_0\cdot(N_0+M-1)}{2M-1}\ll 1. \tag{58}$$

The ratio becomes smaller and smaller with decreasing N_0. This is a clue that the considered system exhibits better asymptotic performance when employing BPSK hypercube labeling; the cost is a halved data rate.

3.3. Search for the Optimal Hypercube Labeling

In [13], an algorithm was developed to obtain the best labeling for a hypercube of any dimension M. For the case of $M=4$, it complies with the ideal labeling conditions, specified by Corollary 1. However, for LABRID applications, one needs hundreds of such optimal labelings. One way to generate such a labeling set is to start with the optimal labeling from [13] and consider some assignment manipulations. Another solution is to start from scratch and utilize adequate optimization algorithms, such as the Binary Search Algorithm [22], or Reactive Tabu Search (RTS) [23], commonly used to solve combinatorial problems, such as label assignment in BICM-ID. In a general case, the advantage of the second approach is that it is not constrained by the outcomes of the previous contributions and allows the exploration of new areas of the search space. However, in the considered case, no optimal labeling can break the rules specified for the ideal labeling in Corollary 1, so the optimal solution can be easily identified.

Having taken into account the above remarks, the author decided to run the RTS algorithm, which has been successfully used in the past to find equivalent optimal labelings for 16-QAM [4] and 64-QAM [10]. The optimization goal is to minimize the distance parameter (47), which represents the impact of labeling on system performance, and acts as the cost function for the optimization algorithm. It is instantiated for QPSK and BPSK hypercubes in (50) and (49), respectively. When searching for the optimal labeling map at $E_b/N_0 \to \infty$, a reasonably small N_0 (i.e., 0.1) is assumed in the cost function. Every labeling visited by the RTS algorithm is checked for compliance with Corollary 1. If the labeling passes the check and has not been found before, it is archived, and the algorithm starts another optimization round from the initial random labeling assignment. After an all-day algorithm run, a collection of 1536 unique optimal labelings is found. All equivalent labelings are equally good for QPSK- and BPSK hypercubes.

A small part of the equivalent optimal labelings are presented in Table 1. The labelings, denoted by $\boldsymbol{\omega}_1,\ldots,\boldsymbol{\omega}_9$, are expressed by integers such that each integer is a unique identifier of given label $\mathbf{v}$, e.g., 11 stands for $\mathbf{v}=(1011)$. For better illustration, let us consider the distance spectrum $\mathcal{Z}(\boldsymbol{x})$ for all vertices labeled according to $\boldsymbol{\omega}_1$ from Table 1. The spectrum is shown in Table 2. Individual rows represent $\boldsymbol{c}$ vectors (unequivocally associated with

respective vertices $\boldsymbol{x}$). The order of $\boldsymbol{c}$ vectors in rows results from the considered labeling $\boldsymbol{\omega}_1$—note the incremental order of the label identifiers. The columns of the table represent the same $\boldsymbol{c}$ vectors assigned to the respective column labels according to the same labeling ω_1. The spectrum entries in a given row appear only in four cells for which the column labels differ in exactly one bit position from their row labels, thereby generating valid $(\mathbf{v}, \mathbf{v}^{\neg m})$ pairs. It can be easily noticed that each row of the table contains the set of entries: {16,12,12,12}, which is specific for the optimal 4-cube labeling, as claimed in Corollary 1. Thus, $\boldsymbol{\omega}_1$ is optimal.

Table 1. Collection of representative equivalent optimal hypercube labelings.

$\boldsymbol{c}$	$\boldsymbol{\omega_1}$	$\boldsymbol{\omega_2}$	$\boldsymbol{\omega_3}$	$\boldsymbol{\omega_4}$	$\boldsymbol{\omega_5}$	$\boldsymbol{\omega_6}$	$\boldsymbol{\omega_7}$	$\boldsymbol{\omega_8}$	$\boldsymbol{\omega_9}$
$(-1,-1,-1,-1)$	1	11	13	13	8	8	8	8	8
$(-1,-1,-1,+1)$	6	6	6	3	3	15	5	3	15
$(-1,-1,+1,-1)$	8	8	8	8	13	1	11	13	1
$(-1,-1,+1,+1)$	15	5	3	6	6	6	6	6	6
$(-1,+1,-1,-1)$	13	13	11	14	14	2	2	4	4
$(-1,+1,-1,+1)$	10	0	0	0	5	5	15	15	3
$(-1,+1,+1,-1)$	4	14	14	11	11	11	1	1	13
$(-1,+1,+1,+1)$	3	3	5	5	0	12	12	10	10
$(+1,-1,-1,-1)$	11	1	1	4	4	4	14	14	2
$(+1,-1,-1,+1)$	12	12	10	10	15	3	3	5	5
$(+1,-1,+1,-1)$	2	2	4	1	1	13	13	11	11
$(+1,-1,+1,+1)$	5	15	15	15	10	10	0	0	12
$(+1,+1,-1,-1)$	7	7	7	7	2	14	4	2	14
$(+1,+1,-1,+1)$	0	10	12	9	9	9	9	9	9
$(+1,+1,+1,-1)$	14	4	2	2	7	7	7	7	7
$(+1,+1,+1,+1)$	9	9	9	12	12	0	10	12	0

Note that the spectrum table is symmetrical along its main diagonal, and the rule from Corollary 1 is valid also for every column label.

What is interesting is that the cells occupied in the spectrum table form a pattern that is pleasing to the eye: two ♋ symbols between two lines. The pattern is not specific to $\boldsymbol{\omega}_1$ or any labeling used. Instead, it is connected only with the order of row- and column labels (the natural binary order is used here). However, for the presented $\boldsymbol{\omega}_1$ labeling, another feature is attractive. That is, the longest entries of the distance spectrum (16) appear only along the lines of the pattern, creating the 8-diagonal and (-8)-diagonal of the table, never as a part of the Zodiac symbols. The same feature is present for the optimal 4-cube labeling constructed in [13], giving the impression that the feature is a prerequisite for any optimal 4-cube labeling. Actually, it is specific only for these optimal labelings, for which the longest spectral distance is always assigned to such pairs of vertices, whose labels differ from each other on the most significant bit, i.e., $\zeta(\boldsymbol{x}, 1) = 16,\ \zeta(\boldsymbol{x}, m \neq 1) = 12,\ \forall \boldsymbol{x}$. Such a design rule is expressed literally in the algorithm to obtain the optimal hypercube labeling from [13]. Among the 1536 optimal labelings found by RTS in the current work, there are exactly 384 (25 percent of all) that have this feature. For the rest (in equal proportion), the longest spectral distance is assigned to the pairs of vertices with the labels always differing on bit $m = 2$, 3, or 4, respectively. In such cases, the highest entries in the spectrum table lie on respective $\pm(2^{4-m})$-diagonals of the spectrum table, so that the diagonal offset corresponds with the bit position on which the labels assigned to the furthest vertices differ from each other. An important observation for any optimal 4-cube labeling is that the furthest spectral distance is assigned to the same bit position in the label for all vertices.

Table 2. Distance spectra for individual vertices for labeling ω_1.

		$c^{\neg m}$	$(+1,+1,-1,+1)$	$(-1,-1,-1,-1)$	$(+1,-1,+1,-1)$	$(-1,+1,+1,+1)$	$(-1,+1,+1,-1)$	$(+1,-1,+1,+1)$	$(-1,-1,-1,+1)$	$(+1,+1,-1,-1)$	$(-1,-1,+1,-1)$	$(+1,+1,+1,+1)$	$(-1,+1,-1,+1)$	$(+1,-1,-1,-1)$	$(+1,-1,-1,+1)$	$(-1,+1,-1,-1)$	$(+1,+1,+1,-1)$	$(-1,-1,+1,+1)$
		$\mathbf{v}^{\neg m}$	(0000)	(0001)	(0010)	(0011)	(0100)	(0101)	(0110)	(0111)	(1000)	(1001)	(1010)	(1011)	(1100)	(1101)	(1110)	(1111)
c	identifier	$\mathbf{v}$																
$(+1,+1,-1,+1)$	0	(0000)		12	12		12				16							
$(-1,-1,-1,-1)$	1	(0001)	12			12		12				16						
$(+1,-1,+1,-1)$	2	(0010)	12			12			12				16					
$(-1,+1,+1,+1)$	3	(0011)		12	12					12				16				
$(-1,+1,+1,-1)$	4	(0100)	12					12	12						16			
$(+1,-1,+1,+1)$	5	(0101)		12			12			12						16		
$(-1,-1,-1,+1)$	6	(0110)			12		12			12							16	
$(+1,+1,-1,-1)$	7	(0111)				12		12	12									16
$(-1,-1,+1,-1)$	8	(1000)	16									12	12		12			
$(+1,+1,+1,+1)$	9	(1001)		16							12			12		12		
$(-1,+1,-1,+1)$	10	(1010)			16						12			12			12	
$(+1,-1,-1,-1)$	11	(1011)				16						12	12					12
$(+1,-1,-1,+1)$	12	(1100)					16				12					12	12	
$(-1,+1,-1,-1)$	13	(1101)						16				12			12			12
$(+1,+1,+1,-1)$	14	(1110)							16				12		12			12
$(-1,-1,+1,+1)$	15	(1111)								16				12		12	12	

4. Illustrative Results

4.1. Overall System Performance

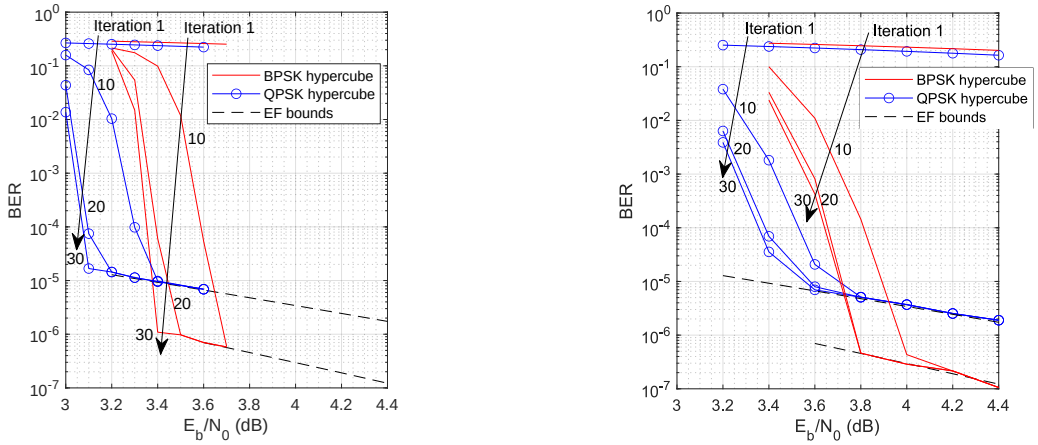
The first step in verifying the usefulness of the proposed LABRID with hypercube labeling is to evaluate the BER(E_b/N_0) performance of the proposed system using MATLAB R2020a software. The system model is implemented according to the description in Section 2.1, with the exception that only one receiver (the actual recipient of the given frame) is considered at a given time. The number of errors among other receivers is not important, as the frame should be classified by them as foreign and rejected. The transmitter labeling map ω_i is randomly selected per frame to not limit the conclusions to only one Tx-Rx link. However, due to the identical features of all equivalent labelings, one could expect the same results having fixed both the labeling map and the relevant receiver. The optimal log-MAP SISO decoding algorithm is considered.

It is known that any BICM-ID system needs a high-depth interleaver, which could brake the correlation between channel gains affecting subsequent codeword bits; assuming high data frame size could help. The smaller the data frame, the higher the risk of "lazy convergence" caused by the fact that the a priori demapper knowledge is not diversified enough. As a result, a wider *turbo cliff* region can be expected in the BER(E_b/N_0) plot.

In the current setup, we consider the data frame size to be 100,000 or 10,000 bits. The first case is quite unrealistic, but it can perfectly demonstrate BICM-ID capabilities in terms of decoding convergence. For statistical reliability reasons, as many as 10^6 data frames are passed for each E_b/N_0 setup. The channel gain and the noise power spectral density N_0 are perfectly estimated at the receiver.

The simulation results are shown in Figure 2 together with the relevant EF bounds defined by (41), calculated for one arbitrarily chosen optimal hypercube labeling by nu-

merical integration of PEP, as shown in [24]. The position of the EF bounds is irrespective of the assumed size of the data frame. The conclusions from the obtained results can be formulated as follows:



(**a**) 100,000 bits per frame (**b**) 10,000 bits per frame

Figure 2. BER vs. E_b/N_0 plot for different data frame sizes.

- The simulation BER curves meet the EF bounds, which proves the accuracy of simulation setup and scripts. Moreover, it shows that LABRID is transparent for the overall system performance.
- The position of the *turbo cliff* for the case of the 10,000 data frame size and QPSK modulation corresponds to its non-LABRID QPSK hypercube BICM-ID reference (Figure 2 in [12]; note the difference in the number of iterations performed and the frame size). It is another proof that LABRID does not deteriorate the overall system performance.
- As expected, the *turbo cliff* for the longer data frame appears at a lower E_b/N_0 and is steeper than for the setup with the shorter data frame.
- Particularly for the shorter data frame, there is no real benefit in running as many as 30 decoding iterations (20 are fully enough).
- The EF bound for BPSK hypercube labeling runs significantly lower than for QPSK hypercube labeling and diverges with increasing E_b/N_0. It is clear evidence of better asymptotic performance of the former. It agrees with the conclusion ending Section 3.2.

4.2. EXIT Chart Analysis

The EXIT chart [15,16] at $E_b/N_0 = 3.4\,\mathrm{dB}$ for the 4-cube modulations considered in the paper (BPSK, QPSK) is shown in Figure 3. The tunnel between the demapper and decoder transfer curves is opened, so $E_b/N_0 = 3.4\,\mathrm{dB}$ is definitely above the *pinch-off* limit, regardless of the modulation type considered. It suggests that the simulation BER points should lie on respective modulation-specific EF bounds at the considered E_b/N_0, which can be observed exactly in Figure 2a. The transfer curve for the QPSK demapper starts its growth with a higher value of extrinsic mutual information than that for the BPSK demapper, and the tunnel width in its narrowest place is wider for the former than for the latter. It proves faster decoding convergence in the case of the system using QPSK hypercube modulation, which is also reflected in the simulation results at $E_b/N_0 = 3.4\,\mathrm{dB}$, i.e., the EF bound is met in at most 10 iterations in the case of QPSK, while the system employing BPSK modulation needs more than 20 iterations. On the other hand, the maximal value of the demapper transfer function (observed in the EF case, i.e., at $\mathcal{I}(\underline{\mathbf{\Lambda}}^{[\mathrm{dem,A}]}; \underline{\mathbf{v}}) = 1$) is higher in the case of the BPSK demodulator. It gives more reliable decoder a priori LLRs in the final decoding pass, which, in turn, lowers the respective EF bound.

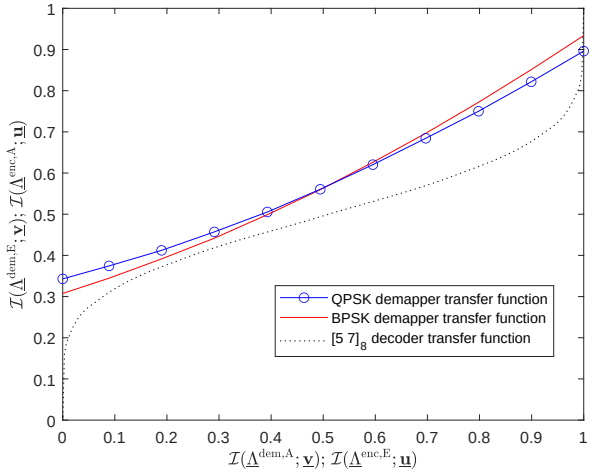


Figure 3. EXIT chart at $E_b/N_0 = 3.4$ dB; 100,000 bits per frame.

These conclusions regarding the EXIT tunnel width do not correspond to the BER(E_b/N_0) plot for the case of the 10,000-bit long data frame, shown in Figure 2b. In detail, at the considered $E_b/N_0 = 3.4$ dB the simulation BER points do not meet the respective EF bounds. So, let us study the reason. The subplots of Figure 4 refer to the following cases: QPSK modulation, 100,000 bits data frame (Figure 4a), QPSK modulation, 10,000 bits (Figure 4b), BPSK modulation, 100,000 bits (Figure 4c), and BPSK modulation, 10,000 bits (Figure 4d). Along with the demapper and decoder transfer curves, they display 10 decoding trajectories exhibiting the lowest demapper a priori mutual information in the last (the 30th) iteration among all transmitted data frames. For the cases where the longer data frame is transmitted (refer to Figure 4a,c), all the considered decoding trajectories tightly match the tunnel created by the transfer curves, so there is neither pdf shape mismatch nor convergence deficiency, and the conclusions based on the EXIT chart observation are fully accurate. On the contrary, if the data frame consists of 10,000 bits (Figure 4b,d), the decoding trajectories visually *die out* and eventually start to leave the EXIT tunnel, making it impossible to reach the EF range. Since the iterative decoding process cannot converge for at least a few "hopeless" frames, BER increases, substantially. It justifies the gap between the simulation BER curves and their EF bounds at the considered $E_b/N_0 = 3.4$ dB, observed in Figure 2b.

4.3. Efficacy of Frame Filtering

LABRID could be considered a robust solution only if the receivers can efficiently filter out foreign frames. So, the current task is to verify the empirical pdfs of the random variable $p_{\Lambda|\omega}(\lambda|\omega)$, representing the absolute extrinsic demapper LLR mean per frame, in two cases: (1) when the receiver obtains the foreign frame, and (2) when it obtains the desired frame. The pdfs shown in Figure 5 are based on the observation of 1000 desired and 1000 foreign frames, both of length 10,000 (such frame length is more realistic than 100,000). In the case of foreign frames, the labeling maps are selected randomly at both the transmitter and receiver sides from the collection of equivalent optimal labelings, and it is guaranteed that they are different from each other. The pdfs are considered in iterations $1 \ldots 30$, separately for QPSK and BPSK hypercube modulation. For better readability, null pdf entries are not shown. Measures are taken at such values of E_b/N_0 that the simulation BER curves from Figure 5a have just met their respective EF bounds.

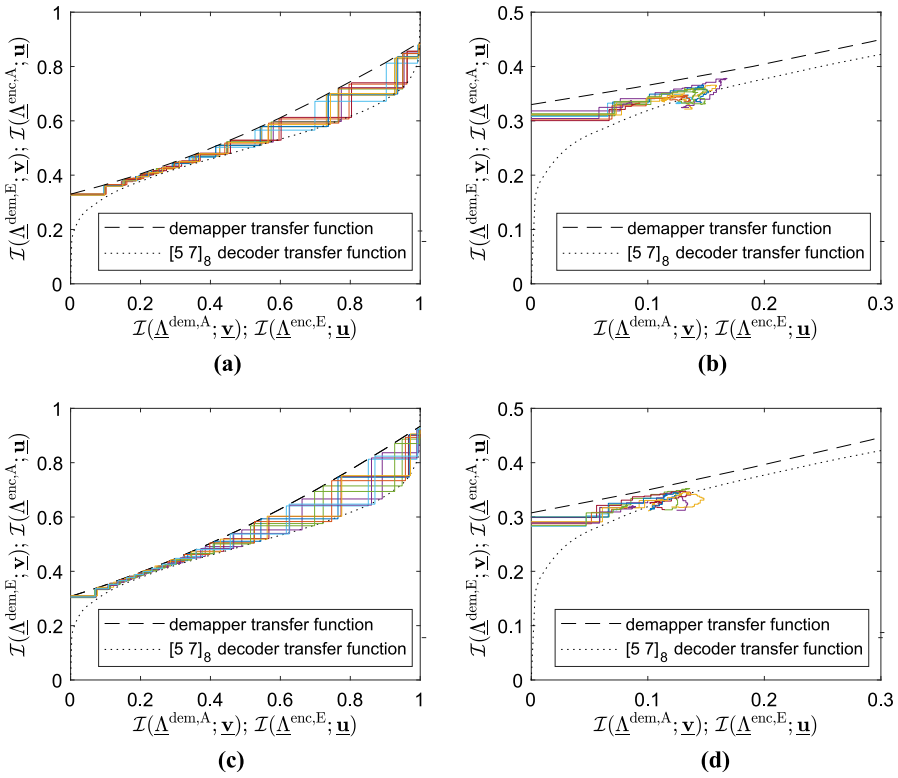
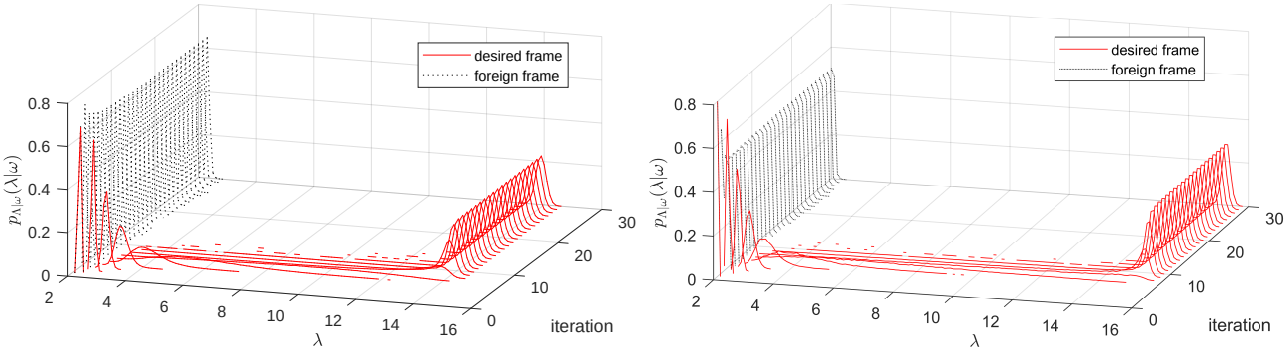


Figure 4. EXIT charts at $E_b/N_0 = 3.4$ dB; 100,000 bits per frame (**a**,**c**) or 10,000 bits per frame (**b**,**d**); QPSK modulation (**a**,**b**) or BPSK modulation (**c**,**d**); the zigzag lines represent the worst 10 decoding trajectories w.r.t. demapper a priori mutual information in the last iteration.



(**a**) QPSK modulation; $E_b/N_0 = 3.6$ dB (**b**) BPSK modulation; $E_b/N_0 = 3.8$ dB

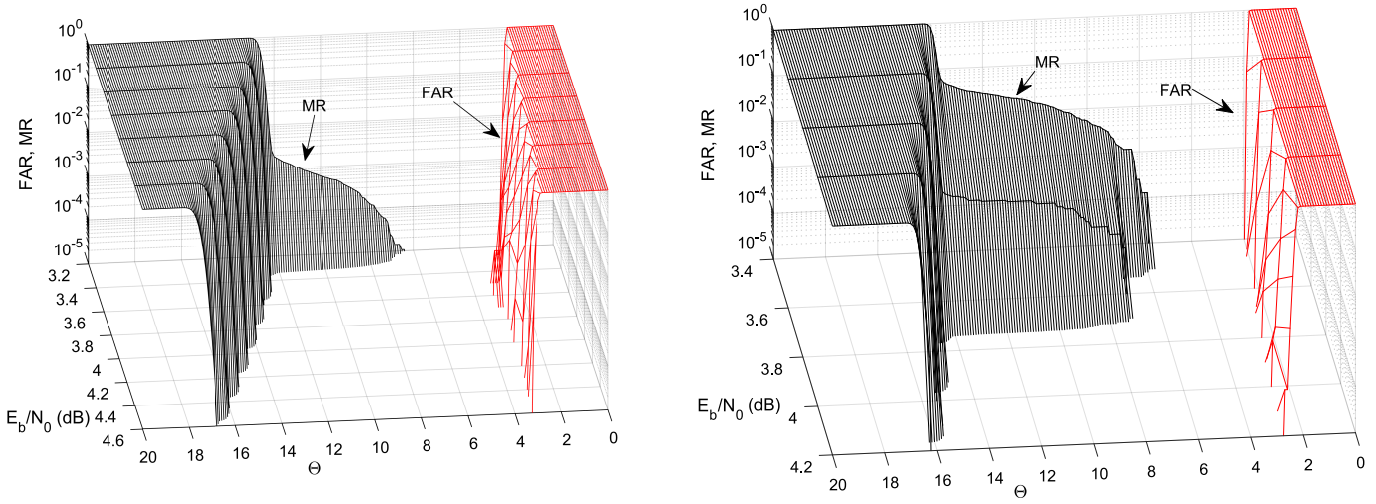
Figure 5. Growth of the absolute extrinsic demapper LLR mean through subsequent iterations.

In both plots it is evident that the absolute extrinsic demapper LLR mean increases in subsequent iterations only if the received frame is the desired frame. Consequently, the pdfs of the two kinds of frames start to disjoint in a few initial iterations, so one can easily classify a given frame as desired or foreign, judging by its Λ value. (As an example, for the QPSK hypercube labeling case considered in Figure 5a, in iteration No. 13 and subsequent, there is no desired frame that exhibits $\Lambda < 12$. At the same time, Λ for any foreign frame does not exceed 3, regardless of the iteration number.) It clearly proves that simple threshold-based decisions regarding frame destination can be error-free.

4.4. Desired Threshold Value

The promising results observed in the previous section are specific only to the given E_b/N_0 considered there. To achieve full success, a more systematic approach is necessary.

Therefore, let us measure FAR and MR (defined in Section 2.2) through a whole range of relevant E_b/N_0 values for arbitrarily chosen threshold values (Θ) and for the cases of QPSK and BPSK hypercubes. Our goal is to find a threshold value (constant, if possible) for which both FAR and MR are negligibly small. FAR and MR illustrated by Figure 6 are observed in iteration 20, assuming data frame length of 10,000 (according to Figure 2b, there is no reason for running more decoding passes). In each simulation setup, 10^6 desired frames and the same number of foreign frames are simulated for statistical reliability reasons.



(**a**) QPSK hypercube (**b**) BPSK hypercube

Figure 6. FAR and MR vs. E_b/N_0 vs. threshold applied to the frame filter.

For both considered hypercubes, a massive MR lobe can be observed for low E_b/N_0, the occurrence of which results in limited clearance between the MR and FAR surfaces, thereby narrowing the range of optimal Θ values. However, the lobe falls in the same area where the *turbo cliff* can be found on the respective BER(E_b/N_0) plots and where BER is usually considered insufficient. Thus, in practice, searching for the optimal threshold value in the *turbo cliff* region may be pointless.

However, the presence of the lobe does not actually prevent one from setting the optimal threshold for any E_b/N_0, as there is still enough room between MR and FAR. In fact, for Θ between, say, 4.0 and 5.0, both MR and FAR are negligibly low in the whole considered E_b/N_0 range, regardless of the hypercube taken into account.

There is still one more issue to be tackled. In contrast to MR, FAR has the form of a wall that crosses the level of 10^{-5} at a value of Θ that gradually grows with increasing E_b/N_0. Therefore, eventually (far from the considered E_b/N_0), FAR could appear to be unacceptably high if Θ is fixed. For this reason, the use of low-order modulations with hypercube labeling when E_b/N_0 is much higher than considered here would require variable Θ. Although such a case is somewhat unrealistic, because a common practice when the channel state improves is to change the modulation order to a higher one, thereby increasing the data rate—in the light of previous contributions [4,5,10] it would also work with LABRID.

4.5. Cost Comparison

The number of different SISO decoder's operations are specified in Table 3, where N_S is the number of convolutional encoder states, n_C is the number of encoded bits per one trellis segment. The decoder, first, calculates all γ components for the whole trellis (Step 1). Having obtained them, it can calculate α (Step 2) and β (Step 3) metrics for all states. Then, it can obtain the extrinsic LLRs related to codeword bits (in all but the last iteration, Step 4a) or the extrinsic LLRs for the data bits (in the last iteration, Step 4b).

Table 3. Complexity analysis of additive SISO decoder per trellis segment per iteration.

Element	Ref. Equation	Additions/Sub.	Multiplications	Max* Operations
Step 1 (γ costs)	(9)	$2N_S \cdot (n_C - 1)$	$2N_S n_C$	0
Step 2 (α metrics)	(10)	N_S	0	N_S
Step 3 (β metrics)	(11)	N_S	0	N_S
Step 4a (LLRs for codeword bits)	(12)	$2N_S \cdot 2 + 2$	0	$2(N_S - 1)$
Step 4b (LLRs for data bits)	(15)	$2N_S \cdot 2 + 1$	0	$2(N_S - 1)$

In the last decoding iteration, the decoder (if launched) executes Steps 1–3 and 4b. Then, including the number of trellis segments (K = 10,000), the coding rate ($1/n_C = 1/2$), and the number of encoder states ($N_S = 4$), one obtains 3.3×10^5 additions (or subtractions), 1.6×10^5 multiplications, and 1.4×10^5 max* calculations.

In the case of LABRID, the frame filter calculates the absolute extrinsic demapper LLR mean (it costs $2K - 1 \approx 1.99 \times 10^4$ additions) and compares it against the threshold (1 regular $\max(\cdot, \cdot)$ operation). Division by the number of samples might be omitted if the threshold is set as a higher respective value. The regular decoding steps are executed only for desired frames, and in these cases frame filtering is an additional cost. But for each foreign frame, the receiver avoids running the decoding pass in the last iteration, which is the LABRID advantage. Actually, its efficiency depends on the number of network users, N_U. The maximal number of users in a LABRID system is limited by the number of equivalent labelings (1536 for BPSK/QPSK hypercube labeling, as shown in Section 3.3).

Assuming a genie frame filter mechanism, each frame is classified correctly as a foreign or desired frame (in the light of the results from Section 4.4, the real LABRID receiver introduces just a marginal FAR and MR error in the E_b/N_0 region of interest). As a consequence, the probability of running the decoding pass in the last iteration is $1/N_U$. Figure 7 shows the number of operations executed in the last iteration by the conventional BICM-ID and LABRID receivers. The cost does not include the demapper routine, as it is common for the competing receivers. It can be seen that the computational complexity of the proposed LABRID mechanism declines rapidly with the number of users within a network. The number of the sum of additions and subtractions, asymptotically, meets a floor, which appears due to the need for Λ computation. The number of remaining operations continues to decline linearly on the log-scale with increasing N_U, and even becomes 1000 times lower than the reference BICM-ID in the case of 1536 users of the LABRID network.

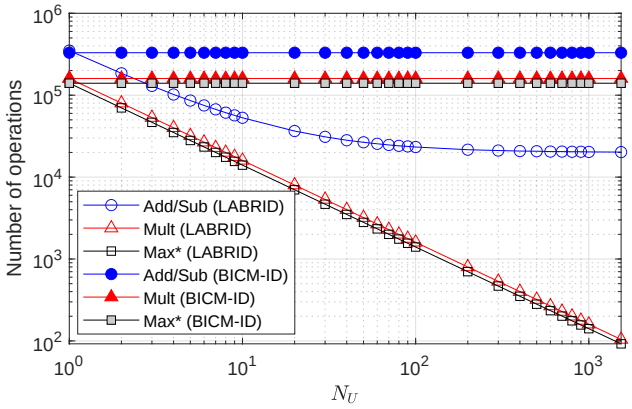


Figure 7. The number of operations of specific type executed in the last decoding pass after generation of the extrinsic demapper LLRs.

Note that the compared costs relate only to the operations in the physical layer. In the case of conservative MAC addressing, every data frame is transferred to the MAC layer, and undergoes integrity checking, which is, literally, another decoding task; it further increases the computational payload for the regular BICM-ID.

5. Conclusions

The concept of hypercube (or multidimensional) labeling has been known for a few years. It helps to overcome the problem of a weak iterative receiver convergence, observed in BICM-ID systems utilizing low-order modulations, such as BPSK or QPSK. In the current paper, another advantage of hypercube labeling has been displayed—the existence of a variety of equivalent optimal labelings which can be used within the framework of the LABRID mechanism to distinguish message recipients within the system. The number of equivalent labelings (more than 1500) is enough to address all stations in most wireless networks.

Since all equivalent labelings exhibit the same distance spectrum, the EF bounds of the LABRID system are identical to those of the regular hypercube BICM-ID system, given modulation order (BPSK or QPSK). In addition, LABRID does not favor any Tx-Rx link over the others in terms of BER for the same reason.

From the analysis of theoretical EF bounds, it is clear that the use of 4-cubes consisting of either four BPSK signals or two QPSK signals leads to a different performance in the considered scenario. Interestingly, the EF bound for the case of a BPSK hypercube lies below that for a QPSK hypercube. This is justified by the fact that the former exploits channel diversity to a higher extent—the signals of the vertex are transmitted within four (not only two) modulation periods, and in each modulation period the channel exhibits another fading gain.

Making decisions on the destination of the frame may rely on a simple threshold crossing rule. The experimental results showed that when the absolute extrinsic demapper LLR mean is checked against a constant threshold, both the false alarm ratio and the misdetection ratio place below 10^{-5}, regardless of the considered E_b/N_0 in the region of interest. The only cost of frame filtering results from the calculation of the mean of (available) absolute extrinsic demapper LLRs. It is incomparably smaller than the price of generating LLRs relative to the data bits in the case of conservative MAC addressing for every foreign frame. LABRID wins the competition in the scenario with just two receivers in a network, but its advantage grows with the number of Rx stations.

Funding: The APC was financed with own funds of the Institute of Radiocommunications, Poznan University of Technology.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Chindapol, A.; Ritcey, J.A. Design, analysis, and performance evaluation for BICM-ID with square QAM constellations in Rayleigh fading channels. *IEEE J. Sel. Areas Commun.* **2001**, *19*, 944–957. [CrossRef]
2. Caire, G.; Taricco, G.; Biglieri, E. Bit-interleaved coded modulation. *IEEE Trans. Inf. Theory* **1998**, *44*, 927–946. [CrossRef]
3. Zehavi, E. 8-PSK trellis codes for a Rayleigh channel. *IEEE Trans. Commun.* **1992**, *40*, 873–884. [CrossRef]
4. Krasicki, M. Labeling-Based Recipient Identification for 16-QAM BICM-ID. *EURASIP J. Wirel. Commun. Netw.* **2019**, 179. [CrossRef]
5. Krasicki, M. BICM-ID Labeling-Based Recipient Identification in a Heterogeneous Network. *Sensors* **2023**, *23*, 3605. [CrossRef] [PubMed]
6. Zhang, J.; Chen, D.; Wang, Y. A New Constellation Shaping Method and Its Performance Evaluation in BICM-ID. In Proceedings of the 2009 IEEE 70th Vehicular Technology Conference Fall (VTC 2009-Fall), Anchorage, AK, USA, 20–23 September 2009; pp. 1–5. [CrossRef]

7. Khoo, B.K.; Le Goff, S.; Sharif, B.; Tsimenidis, C. Bit-interleaved coded modulation with iterative decoding using constellation shaping. *IEEE Trans. Commun.* **2006**, *54*, 1517–1520. [CrossRef]
8. Basar, E.; Wen, M.; Mesleh, R.; Renzo, M.D.; Xiao, Y.; Haas, H. Index Modulation Techniques for Next-Generation Wireless Networks. *IEEE Access* **2017**, *5*, 16693–16746. [CrossRef]
9. Xu, C.; Xiong, Y.; Ishikawa, N.; Rajashekar, R.; Sugiura, S.; Wang, Z.; Ng, S.X.; Yang, L.L.; Hanzo, L. Space-, Time- and Frequency-Domain Index Modulation for Next-Generation Wireless: A Unified Single-/Multi-Carrier and Single-/Multi-RF MIMO Framework. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 3847–3864. [CrossRef]
10. Krasicki, M. Labeling-Based Recipient Identification for BICM-ID in 64-QAM case. In Proceedings of the 2020 International Wireless Communications and Mobile Computing (IWCMC), Limassol, Cyprus, 15–19 June 2020; pp. 1135–1139. [CrossRef]
11. Li, X.; Ritcey, J. Trellis-coded modulation with bit interleaving and iterative decoding. *IEEE J. Sel. Areas Commun.* **1999**, *17*, 715–724. [CrossRef]
12. Tran, N.; Nguyen, H. Improving the performance of QPSK BICM-ID by mapping on the hypercube. In Proceedings of the IEEE 60th Vehicular Technology Conference, VTC2004-Fall, Los Angeles, CA, USA, 26–29 September 2004; Volume 2, pp. 1299–1303. [CrossRef]
13. Tran, N.; Nguyen, H. Design and performance of BICM-ID systems with hypercube constellations. *IEEE Trans. Wirel. Commun.* **2006**, *5*, 1169–1179. [CrossRef]
14. Simoens, F.; Wymeersch, H.; Bruneel, H.; Moeneclaey, M. Multidimensional mapping for bit-interleaved coded modulation with BPSK/QPSK signaling. *IEEE Commun. Lett.* **2005**, *9*, 453–455. [CrossRef]
15. ten Brink, S. Convergence of iterative decoding. *Electron. Lett.* **1999**, *35*, 1117–1119. [CrossRef]
16. Hagenauer, J. The EXIT Chart-Introduction to Extrinsic Information Transfer. In Proceedings of the 12th European Signal Processing Conference (EUSIPCO), Vienna, Austria, 6–10 September 2004; pp. 1541–1548.
17. Benedetto, S.; Divsalar, D.; Montorsi, G.; Pollara, F. A Soft-Input Soft-Output maximum a posteriori (MAP) module to decode parallel and serial concatenated codes. *TDA Prog. Rep.* **1996**, *42*, 1–20.
18. Robertson, P.; Villebrun, E.; Hoeher, P. A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain. In Proceedings of the IEEE International Conference on Communications ICC'95, Seattle, WA, USA, 18–22 June 1995; Volume 2, pp. 1009–1013. [CrossRef]
19. Land, I.; Hoeher, P.; Gligorevic, S. Computation of symbol-wise mutual information in transmission systems with LogAPP decoders and application to EXIT charts. In Proceedings of the International ITG Conference on Source and Channel Coding, Erlangen, Germany, 14–16 January 2004; pp. 195–202.
20. Biglieri, E.; Caire, G.; Taricco, G.; Ventura-Traveset, J. Computing error probabilities over fading channels: A unified approach. *Eur. Trans. Telecommun.* **1998**, *9*, 15–25. [CrossRef]
21. Benedetto, S.; Biglieri, E. *Principles of Digital Transmission with Wireless Applications*; Kluwer Academic Publishers: New York, NY, USA, 1999.
22. Schreckenbach, F.; Gortz, N.; Hagenauer, J.; Bauch, G. Optimized symbol mappings for Bit-Interleaved Coded Modulation with Iterative Decoding. In Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'03), San Francisco, CA, USA, 1–5 December 2003; Volume 6, pp. 3316–3320. [CrossRef]
23. Battiti, R.; Tecchiolli, G. The reactive tabu search. *ORSA J. Comput.* **1994**, *6*, 126–140. [CrossRef]
24. Biglieri, E.; Caire, G.; Taricco, G.; Ventura-Traveset, J. Simple method for evaluating error probabilities. *Electron. Lett.* **1996**, *32*, 191–192. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.