electronics

MDPI

Article

Efficient Decoder for Turbo Product Codes Based on Quadratic Residue Codes

Jie Dong [1], Yong Li [1,*], Rui Liu [1], Taolin Guo [1] and Francis C. M. Lau [2]

[1] College of Computer Science, Chongqing University, Chongqing 400044, China; jiedong_cs@cqu.edu.cn (J.D.); liurui_cs@cqu.edu.cn (R.L.); tguo@cqu.edu.cn (T.G.)
[2] Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong, China; francis-cm.lau@polyu.edu.hk
* Correspondence: yongli@cqu.edu.cn

Abstract: In this letter, we study turbo product codes with quadratic residue codes (called QR-TPCs) as the component codes. We propose an efficient decoder based on Chase-II algorithm with two convergence conditions for the iterative decoding of QR-TPCs. For each row and column, the Chase-II decoder will stop immediately when one of the conditions is met. The simulation results show that the proposed algorithm has a lower computational complexity compared with existing decoding methods. Moreover, a comparison with 5G low-density parity-check codes shows that the proposed turbo product codes have better performance for short code lengths.

Keywords: Chase algorithm; efficient decoding; quadratic residue codes; turbo product codes

check for updates

Citation: Dong J.; Li Y.; Liu R.; Guo L.; Lau, F.C.M. Efficient Decoder for Turbo Product Codes Based on Quadratic Residue Codes. *Electronics* **2022**, *11*, 3598. https://doi.org/10.3390/electronics11213598

Academic Editor: Giovanni Crupi

Received: 26 August 2022
Accepted: 26 September 2022
Published: 3 November 2022

1. Introduction

Combining the ideas of product codes [1] and turbo codes [2], a new iterative decoding algorithm for product codes was proposed by Pyndiah [3,4]. Since this concept is very similar to the iterative decoding of turbo codes built on convolutional component codes, it is known as block turbo code (BTC) and turbo product code (TPC). With its good error correction capability, which is close to the Shannon limit, TPC has great research value and application potential.

(Extended) Bose-Chaudhuri-Hocquenghem (BCH) codes, (extended) Hamming codes and Reed-Solomon (RS) codes are the common types of linear block codes selected as the component codes of TPCs. In this letter, we choose quadratic residue (QR) codes [5] as the component code. As one class of BCH codes, QR codes do not have an error floor at high signal-to-noise ratio (SNR) and most of the known QR codes usually have large minimum distances with code rates greater than or equal to 1/2 so that the code rates of TPCs with QR component codes are around 1/4. Since QR codes usually have larger minimum distances than traditional BCH codes with approximately equal code lengths, they are typically highly prospected in the fields of error correction for reliable data transmission over communication channels with noise. Considering the complexity, the QR codes of lengths less than 100 may have more considerable application potential, especially for short packet transmission with low latency. Take the case of the (24, 12, 8) extended QR code: it has been used in high frequency radio systems [6].

In general, the decoding of TPCs is based on several soft-input/soft-output (SISO) decoders. The iterative operation of the algorithm results in increased computational complexity. In order to reduce the calculation time without noticeable performance loss many efficient decoding algorithms have been investigated to date. In [7], an efficient Chase decoder used for TPC decoding was proposed for one-error-correcting extended BCH codes with substantial complexity reduction and no performance degradation. In [8], Al-Dweik et al. proposed a novel hybrid decoder, which is composed of a standard SISO decoder with a small number of iterations followed by a hard-input/hard-output (HIHO)

decoder. In [9], Lu et al. further modified the hybrid decoder developed in [8] and derived a simple formula for estimating the extrinsic information. Simulations showed that the proposed hybrid decoder gained almost the same error correction performance compared to the algorithm in [8] with complexity reduction. However, the hybrid decoders in [8,9] always required a predetermined number of SISO and HIHO modes and did not take into account that the number of error bits would vary according to the SNR or the number of iterations throughout the decoding process. To overcome this deficiency, in [10], Ahn et al. presented a low complexity syndrome-based decoding algorithm based on the syndrome characteristics of extended Hamming codes. When a single error was detected, the decoder was possible to determine the codeword by using only a single hard decision decoding (HDD) operation. Nevertheless, the SISO decoders had to be used if a double-error syndrome was identified. In [11], Yoon et al. proposed an advanced syndrome-based decoder for TPCs with extended Hamming codes as the component codes and introduced an early termination technique. The advanced syndrome-based decoder applied a HDD operation in the proposed hard-input/soft-output (HISO) conditionally for a double-error syndrome, which reduced the overall complexity but remained slight decoding performance gaps in comparison with that of conventional methods. In addition, a number of test sequences that might result in the same codewords or decoding failures had been avoided in [12], and the computation of the extrinsic information was optimized when there was no competing codeword. In [13], Wang et al. introduced a low-complexity decoder based on the Fast Chase algorithm with a new method of calculating the extrinsic information using only two candidate codewords. And in [14], the authors further introduced a high-speed TPC decoder based on the algorithm in [13] with a fully parallel SISO module.

It is worth noting that the research on algorithms for TPC based on QR component codes is relatively small. Moreover, many of the previously studied algorithms cannot be directly used for the decoding of QR-TPCs. In [15], an iterative soft permutation decoding algorithm (ISPDA) used for QR-TPCs is proposed. Its key idea is to move the errors out of the information bits by permutations and then compute the extrinsic information according to the distance-based decoding (DBD) algorithm [16]. However, a large number of permutations need to be performed to obtain good error performance.

In order to tackle the high computational complexity of decoding QR-TPCs, in this letter, We present an efficient decoder for QR-TPCs with two terminating criteria [17,18] based on Chase-II algorithm [19]. Specifically, we apply this approach to decoding QR-TPCs and give a new way of evaluating the extrinsic information. The difference of syndromes (DS) algorithm [20] is employed for the HDD needed in the Chase-II algorithm, which can be utilized to decode any binary systematic QR code and is one of the most effective look-up table methods for decoding QR codes. Note that Duan et al. proposed an improved difference-of-syndromes (IDS) decoding algorithm [21] for decoding QR codes up to the minimum distance to speed up the computations. However, if the error-correction capability $\lfloor t/2 \rfloor \leq 2$, the IDS algorithm is no longer necessary, as its computational complexity is higher than that of the DS algorithm. The proposed decoder avoids a number of unnecessary HDD operations and calculates the extrinsic information in an easy way. With a more efficient decoding process and an excellent error performance, QR-TPCs may have potential applications in modern communication systems for short-blocklength and high-reliability requirements.

The rest of this paper is organized as follow. Section 2 describes the background of QR codes, the sufficient optimality convergence conditions and TPCs. Section 3 provides a description of QR-TPC and the method of calculating the extrinsic information under different conditions. Section 4 presents the simulation results and complexity analysis. Finally, a brief summary is given in Section 5.

2. Background

2.1. Generator Polynomial of QR Codes

Let l be a positive integer and n be a prime number of the form $n = 8l \pm 1$. Denote Q_n as the set of quadratic residues modulo n, i.e.,

$$Q_n = \{q | q \equiv j^2 \bmod n,\ 1 \le j \le n-1\}. \tag{1}$$

Let θ be the smallest positive integer such that n divides $2^\theta - 1$ and ε be a generator of the multiplicative group consisting of all nonzero elements in GF(2^θ). Then the element $\zeta = \varepsilon^u$ is a primitive n-th root of unity in GF(2^θ) where $u = (2^\theta - 1)/n$. The generator polynomial $g(x)$ of a binary QR code [5] which has the code length n is given by

$$g(x) = \prod_{q \in Q_n} (x - \zeta^q). \tag{2}$$

2.2. Decoding of QR Codes

We first consider the decoding of a single QR code. For a QR $(n, k, d_{\min})$ code, let $\mathbf{x} \in \{0,1\}^n$ be a transmitted codeword and $\hat{\mathbf{x}} \in \{-1,+1\}^n$ be the corresponding transmitted binary phase-shift keying (BPSK) signal where $\hat{x}_i = (-1)^{x_i}\ \forall i$. Assuming an additive white Gaussian noise (AWGN) channel with zero mean and variance σ^2, we denote the received signal by $\mathbf{r} \in \mathbb{R}^n$ and the hard decoding vector of $\mathbf{r}$ by $\mathbf{z} \in \{0,1\}^n$ where

$$z_i = \begin{cases} 0, & r_i \ge 0, \\ 1, & r_i < 0. \end{cases} \tag{3}$$

Supposing $\mathbf{a} \in \{0,1\}^n$ is a codeword of the QR $(n, k, d_{\min})$ code, we define the set of indices where $a_i = z_i$ by $\mathcal{D}_0(\mathbf{a}, \mathbf{z})$ and define the set of indices where $a_i \ne z_i$ by $\mathcal{D}_1(\mathbf{a}, \mathbf{z})$, i.e.,

$$\mathcal{D}_0(\mathbf{a}, \mathbf{z}) \triangleq \{i : a_i = z_i\} \tag{4}$$

$$\mathcal{D}_1(\mathbf{a}, \mathbf{z}) \triangleq \{i : a_i \ne z_i\} = \{1, 2, \cdots, n\} \setminus \mathcal{D}_0(\mathbf{a}, \mathbf{z}). \tag{5}$$

Moreover, the correlation discrepancy between $\mathbf{a}$ and $\mathbf{r}$ is defined as

$$\lambda(\mathbf{r}, \mathbf{a}) = \sum_{i \in \mathcal{D}_1(\mathbf{a}, \mathbf{z})} |r_i|. \tag{6}$$

We arrange the elements in $\mathcal{D}_0(\mathbf{a}, \mathbf{z})$ into a vector $\mathbf{l}$ such that the corresponding magnitudes r_i is increasing, i.e.,

$$\mathbf{l}(\mathcal{D}_0(\mathbf{a}, \mathbf{z})) \triangleq (l_1, l_2, \ldots, l_{|\mathcal{D}_0(\mathbf{a}, \mathbf{z})|}) \tag{7}$$

with $|r_{l_i}| < |r_{l_j}|$ for $1 \le i < j \le |\mathcal{D}_0(\mathbf{a}, \mathbf{z})|$ where $|\mathcal{D}|$ represents the cardinality of the set $\mathcal{D}$. We also denote the first κ elements of $\mathbf{l}$ by $\mathbf{l}(\mathcal{D})^{1:\kappa}$, where $\mathbf{l}(\mathcal{D})^{1:\kappa} \triangleq \varnothing$ for $\kappa < 1$ and $\mathbf{l}(\mathcal{D})^{1:\kappa} \triangleq \mathbf{l}(\mathcal{D})^{1:|\mathcal{D}|}$ for $\kappa > |\mathcal{D}|$. Assuming the received signal $\mathbf{r}$ is decoded by the Chase-II decoding algorithm, the following two theorems can be applied to find the maximum likelihood (ML) codeword.

Theorem 1. *Supposing* $\mathbf{a}$ *is the first valid codeword found by the Chase-II decoder, we define* $\rho \triangleq d_{\min} - |\mathcal{D}_1(\mathbf{a}, \mathbf{z})|$ *and* $G_T(\mathbf{a}, d_{\min}) \triangleq \sum_{i \in \{l_1, l_2, \ldots, l_\rho\}} |r_i|$. *If*

$$\lambda(\mathbf{r}, \mathbf{a}) \le G_T(\mathbf{a}, d_{\min}), \tag{8}$$

then $\mathbf{a}$ *is the ML codeword corresponding to the received signal* $\mathbf{r}$.

Proof of Theorem 1. For detailed proof, see [17]. □

Theorem 2. *Let* $\mathbf{a}$ *and* $\mathbf{b}$ *be two valid codewords and the correlation discrepancy between* $\mathbf{a}$ *and* $\mathbf{r}$ *is smaller than that between* $\mathbf{b}$ *and* $\mathbf{r}$. *Define* $\mathcal{D}_{00} \triangleq \mathcal{D}_0(\mathbf{a},\mathbf{z}) \cap \mathcal{D}_0(\mathbf{b},\mathbf{z})$, $\mathcal{D}_{01} \triangleq \mathcal{D}_0(\mathbf{a},\mathbf{z}) \cap \mathcal{D}_1(\mathbf{b},\mathbf{z})$, $\rho_1 \triangleq d_{\min} - |\mathcal{D}_1(\mathbf{a},\mathbf{z})|$, $\rho_2 \triangleq d_{\min} - |\mathcal{D}_1(\mathbf{b},\mathbf{z})|$. *Without loss of generality, assume* $\rho_1 \geq \rho_2$, *define* $I(\mathbf{a},\mathbf{b}) \triangleq \mathbf{l}(\mathcal{D}_{00} \cup \mathbf{l}(\mathcal{D}_{01})^{1:\lfloor(\rho_1-\rho_2)/2\rfloor})^{1:\rho_1}$ *(for* $\rho_1 < \rho_2$, $I(\mathbf{a},\mathbf{b}) \triangleq \mathbf{l}(\mathcal{D}_{00} \cup \mathbf{l}(\mathcal{D}_{01})^{1:\lfloor(\rho_2-\rho_1)/2\rfloor})^{1:\rho_2}$*) and* $G(\mathbf{a}, d_{\min}; \mathbf{b}, d_{\min}) \triangleq \sum_{i\in I(\mathbf{a},\mathbf{b})} |r_i|$. *If*

$$\lambda(\mathbf{r},\mathbf{a}) \leq G(\mathbf{a}, d_{\min}; \mathbf{b}, d_{\min}), \tag{9}$$

then $\mathbf{a}$ *is the ML codeword of* $\mathbf{r}$.

Proof of Theorem 2. For detailed proof, see [18]. □

The Chase-II algorithm with two terminating criteria can be summarized as Algorithm 1. Note that Theorem 1 applies only to the first valid codeword found by the Chase-II decoder while Theorem 2 is applicable whenever a valid codeword is found subsequently. If the first valid codeword satisfies Theorem 1, the Chase-II decoder will stop immediately and output the codeword. If not, the decoder will continue decoding and Theorem 2 will be considered whenever a valid codeword is found. The two theorems are applied to help determining the ML codeword, terminating the Chase-II decoder earlier and hence improving its convergence rate. The same ML codeword, however, will be decoded with or without the use of the two theorems.

Algorithm 1: Chase-II algorithm with two terminating criteria

```
z = 0.5(−sgn(r) + 1), d = z;
Find the least reliable p bits in z, generate 2^p error patterns e^{2^p};
for j = 1 : 2^p do
    Generate the j-th test pattern g^j = z + e^j;
    Decode g^j using HDD and get the output d^j;
    if d^j is the first valid codeword then
        d = d^j;
        if d^j satisfies Theorem 1 then
            Proceed to Step 16;
        j = j + 1. Proceed to Step 4;
    if d^j is valid & d^j ≠ d then
        if the correlation discrepancy between d^j and r is smaller than that between d and r then
            Swap d and d^j;
        if d and d^j satisfy Theorem 2 then
            Proceed to Step 16;
return d;
```

In the following section, we apply the aforementioned decoding algorithm in the iterative decoding of TPC codes formed by QR component codes.

2.3. 2-D TPC

We denote two systematic linear block codes by $\mathcal{C}^1(n_1,k_1,d_{\min,1})$ and $\mathcal{C}^2(n_2,k_2,d_{\min,2})$, where n_i, k_i and $d_{\min,i}$ denote, respectively, the code length, the information length and the minimum Hamming distance of the i-th code ($i = 1,2$). Using these two block codes as component codes, a 2-D TPC can be constructed as follows.

1. Place $(k_1 \times k_2)$ information bits in an array having k_1 rows and k_2 columns.
2. Encode the k_1 rows using Code $\mathcal{C}^2$.
3. Encode the n_2 columns using Code $\mathcal{C}^1$.

The structure of the 2-D TPC is shown in Figure 1 and the TPC can be represented by $\mathbf{X} \in \{0,1\}^{n_1 \times n_2}$. Moreover, the 2-D TPC is characterized by $(n_1 \times n_2, k_1 \times k_2, d_{\min,1} \times d_{\min,2})$ and has a code rate of $(k_1 \times k_2)/(n_1 \times n_2)$. Usually, the same block code is chosen as the component code, i.e., $\mathcal{C}^1 = \mathcal{C}^2$, so as to reduce the complexity of encoder and decoder. In the following, we assume $\mathcal{C}^1 = \mathcal{C}^2 \triangleq \mathcal{C}$, $n_1 = n_2 \triangleq n$, $k_1 = k_2 \triangleq k$, and $d_{\min,1} = d_{\min,2} \triangleq d_{\min}$.

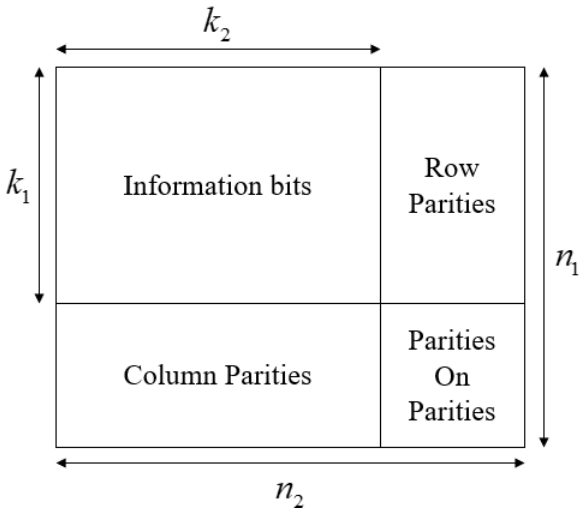

Figure 1. A 2-D TPC consisting of n_1 rows and n_2 columns. The overall codeword is represented by $\mathbf{X} \in \{0,1\}^{n_1 \times n_2}$.

From this point onwards, unless specified otherwise, we use the following notations. We represent a vector by a bold font small letter or symbol, e.g., $\mathbf{w}$ and $\boldsymbol{\alpha}$; and its i-th element by the same small letter or symbol with subscript "i", e.g., w_i and α_i. We represent a two dimensional array by a bold font capital letter, e.g., $\mathbf{W}$; its i-th row (or j-column) by the same bold font capital letter with subscript "$i,:$" (and "$:,j$"), e.g., $\mathbf{W}_{i,:}$ (and $\mathbf{W}_{:,j}$); and its (i,j)-th element by the same capital letter with subscript "i,j", e.g., $W_{i,j}$.

2.4. Pyndiah-Chase-II Algorithm

One of the most common iterative decoding algorithms of TPC is the Pyndiah-Chase-II algorithm [4]. It operates in a SISO mode consisting of several half-iterations. A half-iteration in a row-wise manner and another one in a column-wise manner form one complete iteration. Figure 2 illustrates the operation of a half-iteration. Here, $\mathbf{r} \in \mathbb{R}^n$ denotes the received signal vector corresponding to one particular (row-wise or column-wise) component code; m denotes the half-iteration number; $\mathbf{w}^{(m+1)} \in \mathbb{R}^n$ denotes the extrinsic information vector generated at the m-th half-iteration with $\mathbf{w}^{(1)}$ being a zero vector; α_m and β_m are predefined scaling factors; $\bar{\mathbf{r}}^{(m)} = \mathbf{r} + \alpha_m \mathbf{w}^{(m)}$ is the software information input to the Chase-II decoder; $\mathbf{d} \in \mathcal{C}$ is an output codeword from the decoder having the minimum squared Euclidean distance to $\bar{\mathbf{r}}^{(m)}$; and $\Omega = \{\mathbf{c}^1, \mathbf{c}^2, \ldots\} \subset \mathcal{C}$ is a set containing all other valid codewords $\mathbf{c}^1, \mathbf{c}^2, \ldots$ found by the decoder. For example, the values of α_m and β_m for the 8 half-iterations in [4] are given by

$$\boldsymbol{\alpha} = (0.0, 0.2, 0.3, 0.5, 0.7, 0.9, 1.0, 1.0); \quad (10)$$

$$\boldsymbol{\beta} = (0.2, 0.4, 0.6, 0.8, 1.0, 1.0, 1.0, 1.0). \quad (11)$$

In fact, $\boldsymbol{\beta}$ can be computed dynamically (see (20) in [4]).

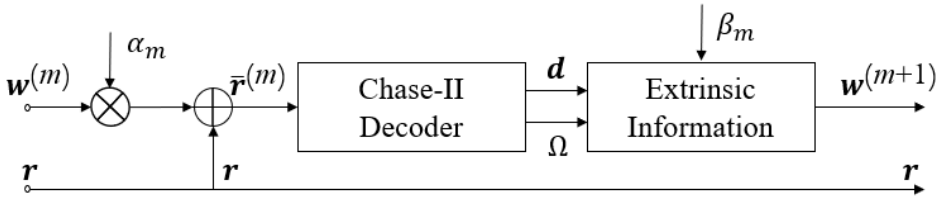

Figure 2. Block diagram of a half-iteration decoder.

In the m-th half-iteration, the i-th element of extrinsic information vector $\mathbf{w}^{(m+1)}$, i.e., $w_i^{(m+1)}$, is calculated as follows.

1. If (i) Ω is an empty set or (ii) all codewords in Ω have their i-th elements the same as d_i, i.e., $c_i^g = d_i \; \forall g$, then
$$w_i^{(m+1)} = \beta_m \times \hat{d}_i \tag{12}$$
where $\hat{d}_i = (-1)^{d_i}$.
2. If one or more codewords in Ω have their i-th elements different from d_i, i.e., $c_i^g \neq d_i \; \exists g$, we compute the squared Euclidean distance between each of these codewords and $\bar{\mathbf{r}}^{(m)}$. In other words, we compute $|\bar{\mathbf{r}}^{(m)} - \hat{\mathbf{c}}^g|^2$ where $\hat{c}_i^g = (-1)^{c_i^g}$. Then we select the codeword $\mathbf{c}$ that is closest to $\bar{\mathbf{r}}^{(m)}$ in terms of squared Euclidean distance. Subsequently, $w_i^{(m+1)}$ is calculated using
$$w_i^{(m+1)} = \left(\frac{|\bar{\mathbf{r}}^{(m)} - \hat{\mathbf{c}}|^2 - |\bar{\mathbf{r}}^{(m)} - \hat{\mathbf{d}}|^2}{4} \right) \times \hat{d}_i - \bar{r}_i^{(m)} \tag{13}$$
where $\hat{c}_i = (-1)^{c_i}$.

3. Proposed QR-TPC and Decoders

3.1. Proposed Scaling Vector

We use the same QR $(n, k, d_{\min})$ code as the component codes in the 2-D TPC shown in Figure 1. We call the resultant TPC as QR-TPC$(n, k, d_{\min})^2$ and denote its codeword by $\mathbf{X} \in \{0, 1\}^{n \times n}$. Assuming BPSK modulation is used and the transmitted signal is passed through an AWGN channel, we denote the received signal by an array $\mathbf{R} \in \mathbb{R}^{n \times n}$ in which each row/column corresponds to one QR $(n, k, d_{\min})$ code.

We first apply the Pyndiah-Chase-II algorithm described in Section 2.4 [4] to decode different QR-TPCs. We assume 8 half-iterations are performed. Moreover, the extrinsic information magnitudes $|w_i^{(m+1)}|$ that are derived using (13) have been recorded during the simulations. For each of QR-TPC$(17, 9, 5)^2$, QR-TPC$(31, 16, 7)^2$ and QR-TPC$(47, 24, 11)^2$, we plot the average value of $|w_i^{(m+1)}|$ that are derived using (13) in Figure 3. The average $|w_i^{(m+1)}|$ is plotted against the half-iteration number m for different E_b/N_0 values. Furthermore, for each half-iteration number m, we evaluate the mean of all the curves and denote this value by γ_m (see the thick blue curve in Figure 3). Finally, we obtain the vector

$$\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \cdots, \gamma_8) = (1.8, 2.0, 2.3, 3.1, 4.4, 6.2, 7.3, 7.7). \tag{14}$$

3.2. Proposed QR-TPC Decoder

We propose modifying the Chase-II decoder in the Pyndiah-Chase-II algorithm to make the algorithm more efficient. Details of the proposed QR-TPC decoding algorithm with a modified Chase-II decoder are shown in Algorithm 2.

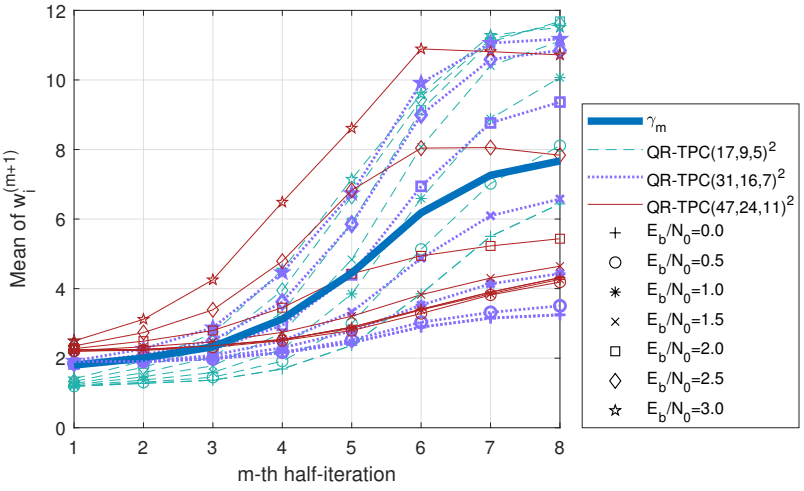

Figure 3. Average extrinsic information magnitudes $|w_i^{(m+1)}|$ derived using (13) versus half-iteration number m. The thick blue curve shows the mean of the averaged values at each m.

Algorithm 2: Proposed QR-TPC Decoding Algorithm

1 **for** $m = 1 : m_{\max}$ **do**
2 **for** $l = 1 : n$ **do**
3 $\mathbf{r} = \mathbf{R}_{l,:}$, $\mathbf{w}^{(m)} = \mathbf{W}_{l,:}^{(m)}$;
4 $\bar{\mathbf{r}}^{(m)} = \mathbf{r} + \alpha_m \mathbf{w}^{(m)}$;
5 $\mathbf{z} = 0.5(-\mathrm{sgn}(\bar{\mathbf{r}}^{(m)}) + 1)$;
6 Find the least reliable p bits in $\mathbf{z}$, generate 2^p error patterns $\mathbf{e}^{2^p}$;
7 **for** $j = 1 : 2^p$ **do**
8 Generate the j-th test pattern $\mathbf{g}^j = \mathbf{z} + \mathbf{e}^j$;
9 Decode $\mathbf{g}^j$ using HDD and get the output $\mathbf{d}^j$;
10 **if** $\mathbf{d}^j$ *is the first valid candidate codeword* **then**
11 $\mathbf{d} = \mathbf{d}^j$;
12 **if** $m \geq m_\delta$ *&* $\mathbf{d}^j$ *satisfies Theorem 1* **then**
13 Compute $w_i^{(m+1)}$ by (15);
14 Proceed to Step 27;
15 $j = j + 1$. Proceed to Step 8;
16 **if** $\mathbf{d}^j$ *is valid &* $\mathbf{d}^j \neq \mathbf{d}$ *&* $\mathbf{d}^j \notin \Omega$ **then**
17 Store $\mathbf{d}^j$ in Ω as a competing codeword $\mathbf{c}^j$;
18 **if** *the correlation discrepancy between* $\mathbf{d}^j$ *and* $\bar{\mathbf{r}}^{(m)}$ *is smaller than that between* $\mathbf{d}$ *and* $\bar{\mathbf{r}}^{(m)}$ **then**
19 Swap $\mathbf{d}$ and $\mathbf{d}^j$;
20 **if** $\mathbf{d}$ *and* $\mathbf{d}^j$ *satisfy Theorem 2* **then**
21 Proceed to Step 22;
22 **for** $i = 1 : n$ **do**
23 **if** *competing codewords in* Ω *with* $c_i \neq d_i$ *can be found, select the one* $\mathbf{c}$ *that is closest to* $\bar{\mathbf{r}}^{(m)}$ *in terms of squared Euclidean distance* **then**
24 Compute $w_i^{(m+1)}$ by (13);
25 **else**
26 Compute $w_i^{(m+1)}$ by (12);
27 $\mathbf{D}_{l,:} = \mathbf{d}$, $\mathbf{W}_{l,:}^{(m+1)} = \mathbf{w}^{(m+1)}$;
28 $\mathbf{R} = \mathbf{R}^T$, $\mathbf{D} = \mathbf{D}^T$, $\mathbf{W}^{(m)} = (\mathbf{W}^{(m)})^T$;
29 return $\mathbf{D}$;

In the modified Chase-II decoder, we apply both Theorems 1 and 2 (described in Section 2.2) as the terminating criteria to complete a half-iteration earlier. Depending on the theorem that has been satisfied, the extrinsic information is calculated as follows.

1. When Theorem 1 is satisfied, $\mathbf{d}$ is the only output codeword from the decoder and Ω is an empty set in Figure 2. One simple way to compute the extrinsic information $w_i^{(m+1)}$ is to apply (12) because Ω is an empty set. However, as will be shown in Section 4, the error performance will be degraded quite significantly (particularly for short codes) compared with that of the original algorithm. We conjecture that the weights used in (12), i.e., β_m's, are not large enough to correct the sign of an incorrect bit. Thus, we propose to use γ in (14) to replace $\boldsymbol{\beta}$ in computing $w_i^{(m+1)}$. In other words, when Theorem 1 is applied and is satisfied, we compute $w_i^{(m+1)}$ using

$$w_i^{(m+1)} = \gamma_m \times \hat{d}_i. \tag{15}$$

   Moreover, we set a threshold half-iteration number m_δ, which defines the occasion that Theorem 1 starts being applied to the Chase-II decoder. The rationale of setting such a threshold is as follows. During the first few half-iterations, the inputs to the Chase-II decoder, i.e., $\bar{\mathbf{r}}^{(m)}$, may not have very high reliabilities. Thus, the ML codeword output $\mathbf{d}$ is not likely to be the transmitted component codeword. If we apply Theorem 1 and (15) to compute the extrinsic information now, the extrinsic information may not be accurate. But after a number of half-iterations, $\bar{\mathbf{r}}^{(m)}$ should become much more reliable and the ML codeword output $\mathbf{d}$ is more likely to be the transmitted component codeword. Using Theorem 1 and (15) to compute the extrinsic information therefore becomes more accurate. Based on the above rationale, Theorem 1 will be applied when $m \geq m_\delta$ (refer to Line 12 to Line 14 of in Algorithm 2). Moreover, the first $m_\delta - 1$ entries in γ will not be used in the Chase-II decoder when $m_\delta > 1$.
2. When Theorem 2 is satisified, Chase-II decoder will stop. It will output $\mathbf{d}$ and Ω which contains at least one valid codeword. Then the extrinsic information is calculated by (12) and (13).

4. Simulation Results

In this section, we consider the QR-TPC$(n, k, d_{\min})^2$ shown in Table 1. In our algorithm, all coded bits are modulated by BPSK and transmitted over the AWGN channel. The algorithm will stop when the decoder gathers 100 incorrect codewords. The number of least-reliable-bit (LRB) positions p is set to 4 for the QR-TPC$(17, 9, 5)^2$, QR-TPC$(23, 12, 7)^2$, QR-TPC$(31, 16, 7)^2$ and 5 for the QR-TPC$(47, 24, 11)^2$. The DS algorithm [20] is employed for the HDD.

Table 1. Parameters of QR-TPCs.

QR-TPC	Codeword Length	Information Length	Code Rate
$(17, 9, 5)^2$	289	81	0.280
$(23, 12, 7)^2$	529	144	0.272
$(31, 16, 7)^2$	961	256	0.266
$(47, 24, 11)^2$	2209	576	0.261

4.1. BER Performance Comparison

Figure 4 shows the bit error rate (BER) performance of the proposed algorithm. The number of half-iterations $m_{\max}$ is set to 8, and the threshold m_δ is set to 1 and 4 separately. We plot three BER simulation results for each QR-TPC when the proposed algorithm is used. In the same figure, we also plot the BER simulation results when the algorithm in [4] (i.e., the one described in Section 2.4) is used. From the result of Figure 4, we can conclude the followings:

- Proposed scaling vector: When $m_\delta = 1$, we can observe that our proposed algorithm can achieve a better BER when using (15) instead of (12) to compute the extrinsic information. For example at a BER of 10^{-6}, a 0.5 dB gain is achieved for the QR-TPC$(23, 12, 7)^2$ and about 0.4 dB gain is achieved for the QR-TPC$(31, 16, 7)^2$.
- Threshold m_δ: For certain QR-TPCs, the threshold m_δ provides a trade-off between BER performance and computational complexity. It can be seen that increasing m_δ from 1 to 4 can further improve the BER performance of our proposed algorithm for the QR-TPC$(17, 9, 5)^2$.
- Overall BER performance: We can see that no significant BER deviation between the proposed algorithm and the algorithm in [4]. To be more specific, Figure 4 shows that there are slight BER performance gaps compared with that of the algorithm in [4] for the QR-TPC$(17, 9, 5)^2$ and QR-TPC$(23, 12, 7)^2$. Nonetheless, the BER performance loss is tolerable. Furthermore, the proposed algorithm provides better performance with significant computational complexity reduction for the QR-TPC$(31, 16, 7)^2$ and QR-TPC$(47, 24, 11)^2$. As a result, the proposed algorithm has a similar BER performance as the algorithm in [4].
- Comparison with other non-chase algorithm: When E_b/N_0 is larger than 2.0 dB, the decoding performance of the proposed algorithm for the QR-TPC$(47, 24, 11)^2$ is similar to that reported in [15] (see the curve for 4 iterations in Figure 6 given in [15]).

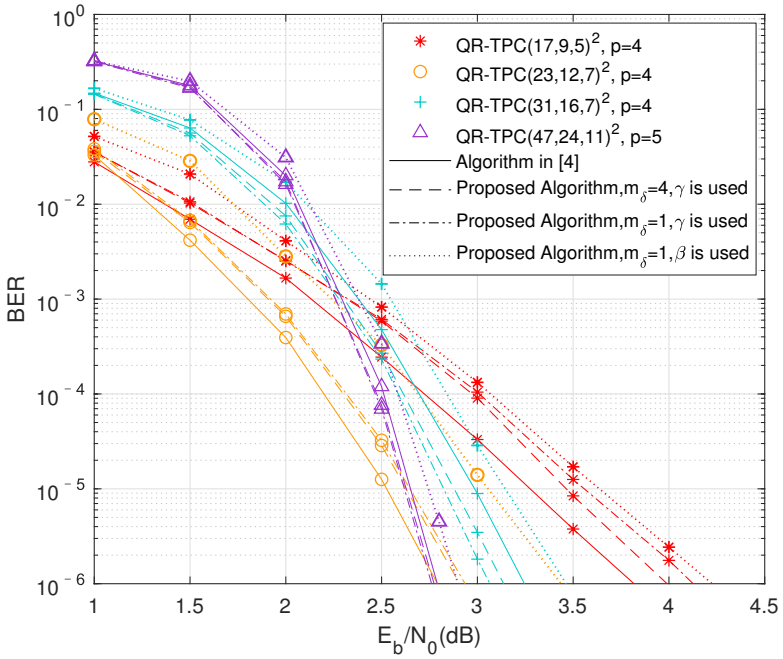

Figure 4. BER performance of QR-TPCs under different decoder settings.

4.2. Complexity Comparison

In Figure 5, we plot the average number of HDD operations required for decoding each QR-TPC codeword at different E_b/N_0 for our proposed algorithm and the algorithm in [4]. Since the algorithm in [4] will exhaust all the 2^p test patterns, the average number of HDD operations required is fixed for each QR-TPC code and equals $m_{max} \cdot n \cdot 2^p$. For our proposed algorithm, the modified Chase-II decoder will stop testing any remaining patterns when Theorem 1 or Theorem 2 is satisfied. As a result, the average number of HDD operations required is reduced when our proposed algorithm is used. Figure 5 also shows that:

- Proposed scaling vector: When $m_\delta = 1$, we can observe that our proposed algorithm can always avoid more HDD operations when using (15) instead of (12) to compute the extrinsic information. For instance, when $E_b/N_0 = 3.0$ dB, the average number

of HDD operations for the QR-TPC$(17,9,5)^2$, QR-TPC$(23,12,7)^2$, QR-TPC$(31,16,7)^2$ and QR-TPC$(47,24,11)^2$ are reduced more by approximately 4.27%, 3.70%, 6.90% and 4.67%, respectively, using (15) instead of (12).

- Threshold m_δ: Using $m_\delta = 1$ compared with using $m_\delta = 4$ further reduces the average number of HDD operations because Theorems 1 and 2 are applied earlier (i.e., starting the first half-iteration) in the modified Chase-II decoder.
- The reduction in the average number of HDD operations increases as E_b/N_0: In the low E_b/N_0 region, the number of error bits that occur in each codeword is more likely to exceed the error-correction capability t of the QR code. Thus a valid codeword that satisfies Theorem 1 or Theorem 2 cannot be found easily. As E_b/N_0 increases, the ML codeword can be obtained more readily found using Theorem 1 or Theorem 2 and hence the modified Chase-II decoder can terminate earlier. We take $m_\delta = 1$ and $E_b/N_0 = 1.0$ dB as an example. For the QR-TPC$(17,9,5)^2$, QR-TPC$(23,12,7)^2$, QR-TPC$(31,16,7)^2$ and QR-TPC$(47,24,11)^2$, the average numbers of HDD operations used in our proposed algorithm are reduced by 65.90%, 62.26%, 19.92% and 3.82%, respectively, compared with those used in the algorithm in [4]; when E_b/N_0 is increased to 3.0 dB, the numbers of HDD operations are reduced by 85.52%, 86.17%, 71.51% and 72.14%, respectively.

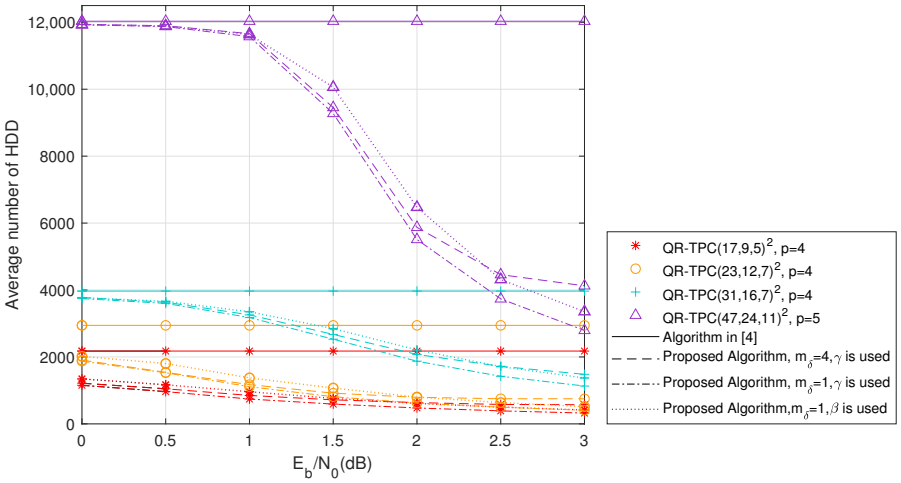

Figure 5. Average number of HDD operations performed to decode one codeword.

The complexity of SISO QR-TPC decoders depends mainly on the number of operations of HDD. Generally, it takes $O(k + \sum_{i=1}^{\lfloor t/2 \rfloor} iC_k^i)$ GF(+), $O(k)$ GF($\ll$) and $O((k-1) + (k-1)\sum_{i=1}^{\lfloor t/2 \rfloor} C_k^i)$ R(+) for a QR code to run the DS algorithm once, where GF(+) and GF($\ll$) means the addition and shift operation in the Galois field, and R(+) means the addition operation in the real field. To calculate (8) and (9), the proposed algorithm also requires $O(d_{\min})$ R(+). But it increases only a minimal computational complexity due to the small number of operations. It is worth pointing out that when Theorem 1 is satisfied, the proposed algorithm calculates the $w_i^{(m+1)}$ by (15) directly and therefore no longer needs to compute the squared Euclidean distance to $\bar{\mathbf{r}}^{(m)}$. Thus it requires much fewer R(+) and the multiplication operations over the real field (i.e., R(*)).

The complexity comparison of our proposed algorithm and the algorithm in [15] is shown in Table 2, where L represents the average number of HDD operations shown in Figure 5 and P represents the number of the permutations used in [15]. The error performance of the algorithm in [15] depends on P and $m_{\max}$. In Table 2, when $m_{\max} = 8$, $E_b/N_0 = 3.0$ dB, $m_\delta = 1$ and $P = 1000$, for the QR-TPC$(47,24,11)^2$, we observe that the proposed algorithm has a much smaller number of GF(+), GF($\ll$) and R(*) performed to decode one codeword than the algorithm in [15]. In fact, our proposed algorithm may not require such a large number of GF(+), GF($\ll$), and R(+). In the best case, if the weight of

the syndrome **s** of the test pattern is smaller or equal to t, the DS algorithm can directly obtain the candidate codeword. For this reason, the DS algorithm requires only $O(k)$ GF(+) and $O(k-1)$ R(+) when decoding a QR code. This situation occurs especially in the high E_b/N_0 region.

Table 2. Complexity comparison of the algorithms and relative number of the operations performed to decode one codeword.

Operation	The Proposed Algorithm		Algorithm in [15]	
	Complexity Comparison	Relative Number	Complexity Comparison	Relative Number
GF(+)	$O(L \cdot (k + \sum_{i=1}^{\lfloor t/2 \rfloor} iC_k^i))$	2,011,200	$O(P \cdot m_{\max} \cdot n^2)$	17,672,000
GF($\ll$)	$O(L \cdot k)$	80,448	$O(P \cdot m_{\max} \cdot n \cdot (k-1))$	8,648,000
R(+)	$O(L \cdot (k-1) \cdot \sum_{i=0}^{\lfloor t/2 \rfloor} C_k^i)$	23,205,896	$O(P \cdot m_{\max} \cdot n^2)$	17,672,000
R(*)	$O(m_{\max} \cdot n^2)$	17,672	$O(P \cdot m_{\max} \cdot n^2)$	17,672,000

In [22], the authors introduce an efficient decoding algorithm for TPCs based on Kaneko's algorithm [23], which provides similar BER performance to the algorithm in [4]. It is interesting to note that Lemma 1 in Kaneko's algorithm is identical to Theorem 1 when $|\mathcal{D}_1(\mathbf{x}, \mathbf{z})| = |\mathcal{D}_1(\mathbf{a}, \mathbf{x}')|$, where $\mathbf{x}'$ is an arbitrary codeword such that $\mathbf{x}' \neq \mathbf{x}$. Figure 2a given in [23] shows that Kaneko's algorithm reduces the average number of HDD operations by about 65.6% compared to the conventional Chase-II algorithm when $E_b/N_0 = 3.0$ dB for QR $(23, 12, 7)$ code. However, in the range under 1.5 dB, Kaneko's algorithm requires more complexity than that of the Chase-II decoder. To avoid this drawback, the maximum number of test patterns for each row and column in the algorithm in [22] is made equal to that of the Chase-II decoder. Figure 5 shows that our proposed algorithm can reduced by 86.17% HDD operations for QR-TPC$(23, 12, 7)^2$ in the same E_b/N_0. Given that the procedure used for the extrinsic information calculations in [22] is the same as the algorithm in [4], our proposed algorithm uses fewer HDD operations and therefore provides lower computational complexity.

4.3. Comparison with 5G LPDC Codes

QR-TPCs have potential applications in modern communication systems for short-blocklength and high-reliability requirements. We compare the BER performance of QR-TPCs and 5G low-density parity-check (LDPC) codes [24] having similar code parameters. For these LDPC codes, we take the first 26/27 rows and 36/37 columns of base graph (BG) 2 as the base matrices (parameters are given in Table 3), and use the sum-product decoding algorithm with a maximum of 50 iterations. It is well known that LDPC codes perform better as codelength increases. As shown in Figure 6, when the codes are short, QR-TPCs have better performance than 5G LDPC codes. Specifically, QR-TPC $(17, 9, 5)^2$ achieves a 1.1 dB gain over 5G LDPC (288, 80) at a BER of 10^{-6}, and QR-TPC $(23, 12, 7)^2$ outperforms 5G LDPC (518, 140) by about 0.7 dB. However, when the codelength becomes long, the 5G LDPC (962,260) outperforms QR-TPC $(31, 16, 7)^2$ by about 0.8 dB at a BER of 10^{-6}.

Table 3. Parameters of 5G LDPC Codes.

Size of BG2	Lifting Size	Codeword Length	Information Length	Code Rate
26×36	8	288	80	0.278
27×37	14	518	140	0.270
27×37	26	962	260	0.270

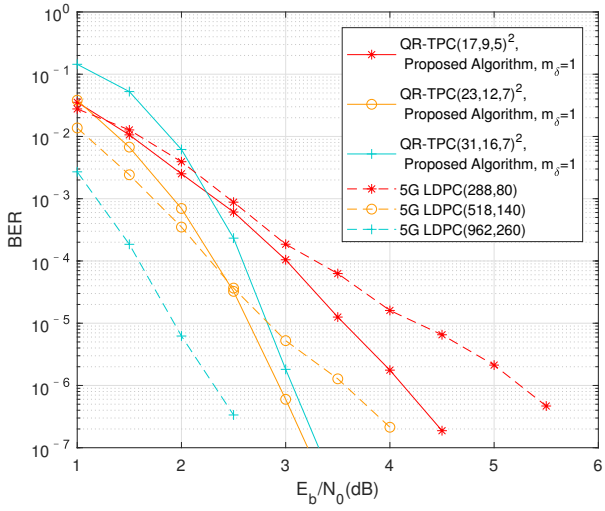

Figure 6. Comparison of BER performance between QR-TPCs and 5G LDPC codes.

5. Conclusions

This letter presents an efficient decoding algorithm for TPCs with QR component codes. By introducing two convergence conditions and a new scaling vector γ, the proposed algorithm significantly improves the decoding efficiency with almost the same BER performance compared to the Pyndiah-Chase-II decoding algorithm and the ISPDA. Simulation results also reveal that QR-TPCs outperform 5G LDPC codes with similar parameters when the block lengths are below 500.

Author Contributions: All the authors contributed to various degrees to ensure the quality of this work (Conceptualization, Y.L.; methodology, J.D.; software, J.D.; validation, J.D. and Y.L.; formal analysis, J.D.; investigation, J.D. and R.L.; resources, Y.L., R.L. and T.G.; data curation, J.D.; writing—original draft preparation, J.D.; writing—review and editing, F.C.M.L.; visualization, J.D.; supervision, Y.L.; project administration, T.G.; funding acquisition, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by China Natural Science Foundation (NSF) under Grant 61771081, the Fundamental Research Funds for the Central Universities (China) under Grant cstc2019jcyjmsxmX0110.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Elias, P. Error-free coding. *Trans. IRE Prof. Group Inf. Theory* **1954**, *4*, 29–37. [CrossRef]
2. Berrou, C.; Glavieux, A. Near optimum error correcting coding and decoding: Turbo-codes. *IEEE Trans. Commun.* **1996**, *44*, 1261–1271. [CrossRef]
3. Pyndiah, R.; Glavieux, A.; Picart, A.; Jacq, S. Near optimum decoding of product codes. In Proceedings of the 1994 IEEE GLOBECOM. Communications: The Global Bridge, San Francisco, CA, USA, 28 Novermber–2 December 1994; Volume 1, pp. 339–343. [CrossRef]
4. Pyndiah, R. Near-optimum decoding of product codes: Block turbo codes. *IEEE Trans. Commun.* **1998**, *46*, 1003–1010. [CrossRef]
5. Prange, E. *Some Cyclic Error-Correcting Codes with Simple Decoding Algorithms*; Technical Report TN-58-156; Air Force Cambridge Res. Center: Cambridge, MA, USA, 1958.
6. Honary, B.; Hunt, B.; Maundrell, M. Improving automatic link establishment through a new soft decision trellis decoder for the (24,12) Golay code. In Proceedings of the Sixth International Conference on HF Radio Systems and Techniques, York, UK, 4–7 July 1994; pp. 182–185. [CrossRef]
7. Argon, C.; McLaughlin, S.W. An efficient chase decoder for turbo product codes. *IEEE Trans. Commun.* **2004**, *52*, 896–898. [CrossRef]
8. Al-Dweik, A.; Goff, S.; Sharif, B. A Hybrid Decoder for Block Turbo Codes. *IEEE Trans. Commun.* **2009**, *57*, 1229–1232. [CrossRef]
9. Lu, P.; Lu, E.; Chen, T. An efficient hybrid decoder for block turbo codes. *IEEE Commun. Lett.* **2014**, *18*, 2077–2080. [CrossRef]
10. Ahn, B.; Yoon, S.; Heo, J. Low Complexity Syndrome-Based Decoding Algorithm Applied to Block Turbo Codes. *IEEE Access* **2018**, *6*, 26693–26706. [CrossRef]

11. Yoon, S.; Ahn, B.; Heo, J. An advanced low-complexity decoding algorithm for turbo product codes based on the syndrome. *Eurasip J. Wirel. Commun. Netw.* **2020**, *2020*, 1–31. [CrossRef]
12. Son, J.; Kong, J.J.; Yang, K. Efficient decoding of block turbo codes. *J. Commun. Netw.* **2018**, *20*, 345–353. [CrossRef]
13. Wang, Y.; Lin, J.; Wang, Z. A low-complexity decoder for turbo product codes based on extended hamming codes. In Proceedings of the 2018 IEEE 18th International Conference on Communication Technology (ICCT), Chongqing, China, 8–11 October 2018; pp. 99–103. [CrossRef]
14. Kishore, J.H.; Yamuna, B.; Balasubramanian, K. Design of a Fast Chase Algorithm based High Speed Turbo Product Code Decoder. In Proceedings of the 2021 International Conference on Advances in Computing and Communications (ICACC), Kochi, India, 21–23 October 2021; pp. 1–5. [CrossRef]
15. Askali, M.; Ayoub, F.; Chana, I.; Belkasmi, M. Iterative soft permutation decoding of product codes. *Comput. Inf. Sci.* **2016**, *9*, 128–135. [CrossRef]
16. Le, N.; Soleymani, A.R.; Shayan, Y.R. Distance-based-decoding of block turbo codes. *IEEE Commun. Lett.* **2005**, *9*, 1006–1008. [CrossRef]
17. Taipale, D.J.; Pursley, M.B. An improvement to generalized-minimum-distance decoding. *IEEE Trans. Inf. Theory* **1991**, *37*, 167–172. [CrossRef]
18. Wang, L.; Li, Y.; Truong, T.K.; Lin, T. On decoding of the (89, 45, 17) quadratic residue code. *IEEE Trans. Commun.* **2013**, *61*, 832–841. [CrossRef]
19. Chase, D. A class of algorithms for decoding block codes with channel measurement information. *IEEE Trans. Inf. Theory* **1972**, *18*, 170–182. [CrossRef]
20. Li, Y.; Duan, Y.; Chang, H.C.; Liu, H.; Truong, T.K. Using the difference of syndromes to decode quadratic residue codes. *IEEE Trans. Inf. Theory* **2018**, *64*, 5179–5190. [CrossRef]
21. Duan, Y.; Li, Y. An Improved Decoding Algorithm to Decode Quadratic Residue Codes Based on the Difference of Syndromes. *IEEE Trans. Inf. Theory* **2020**, *66*, 5995–6000. [CrossRef]
22. Dave, S.; Kim, J.; Kwatra, S.C. An efficient decoding algorithm for block turbo codes. *IEEE Trans. Commun.* **2001**, *49*, 41–46. [CrossRef]
23. Kaneko, T.; Nishijima, T.; Inazumi, H.; Hirasawa, S. An efficient maximum-likelihood-decoding algorithm for linear block codes with algebraic decoder. *IEEE Trans. Inf. Theory* **1994**, *40*, 320–327. [CrossRef]
24. Chairman, S. Chairman's Notes of Agenda Item 7.1.5 Channel Coding and Modulation. In Proceedings of the TSG RAN WGI Meeting 87, R1-1613710, 3GPP Session Chairman (Nokia), Reno, Nevada, USA, 14–18 November 2016.