Generalized Automorphisms of Channel Codes: Properties, Code Design, and a Decoder

Jonathan Mandelbaum, Holger Jäkel, and Laurent Schmalen

Communications Engineering Lab, Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany

jonathan.mandelbaum@kit.edu

Abstract—Low-density parity-check codes together with belief propagation (BP) decoding are known to be well-performing for large block lengths. However, for short block lengths there is still a considerable gap between the performance of BP decoding and maximum likelihood decoding. Different ensemble decoding schemes such as, e.g., automorphism ensemble decoding (AED), can reduce this gap in short block length regime. We propose generalized AED (GAED) that uses automorphisms according to the definition in linear algebra. Here, an automorphism of a vector space is defined as a linear, bijective self-mapping, whereas in coding theory self-mappings that are scaled permutations are commonly used. We show that the more general definition leads to an explicit joint construction of codes and automorphisms, and significantly enlarges the search space for automorphisms of existing linear codes. Furthermore, we prove the concept that generalized automorphisms can indeed be used to improve decoding. Additionally, we propose a code construction of linear codes enabling the construction of codes with suitably designed automorphisms. Finally, we analyze the decoding performances of GAED for some of our constructed codes.

Index Terms—generalized automorphism groups; generalized automorphism ensemble decoding; short block lengths codes

I. INTRODUCTION

Low-density parity-check (LDPC) codes are a prominent example of error correcting codes that are used in a large variety of applications. They were first proposed by Gallager together with a low-complexity message passing decoding algorithm [1], often called belief propagation (BP) decoding. LDPC codes with large block lengths can achieve low error rates and close-to-capacity performance [2]. Yet, many low-latency communication systems, such as the internet of things, autonomous driving, or communicating control commands, require codes of short block lengths. For such codes, there is still a considerable gap between the performance of BP decoding and maximum likelihood (ML) decoding. This can be attributed to the poor structural properties of the parity-check matrix (PCM) of short codes for BP decoding, e.g., a large number of short cycles and of non-zero entries, together with the sub-optimality of the message passing decoding algorithm.

For any binary-input memoryless symmetric output channel (BMSC), such as the additive white Gaussian noise (AWGN) channel, correctly decoding a received word with a symmetric

message passing decoding algorithm depends only on the noise superimposed by the channel, not on the transmitted codeword itself [2, Lemma 4.90]. Therefore, different variations of the algorithm such as multiple bases belief propagation (MBBP) [3] and automorphism ensemble decoding (AED) [4] were proposed. Herein, an ensemble of noise representations or decoding algorithms aims to improve decoding. Following the latter statement, MBBP decoding performs ensemble decoding, in which the received sequence is decoded by multiple different BP decoders in parallel. Similarly, the idea of the AED, as introduced in [5], is to use the automorphism group defined in [6] to exploit different noise representations in parallel paths which is discussed in more detail in Sec. II-B.

From a structural perspective, it is reasonable to restrict the definition of the automorphism groups of linear codes to consist solely of suitable *scaled permutations* (Sec. II-A) such that, i.a., codewords of the same Hamming weight are mapped onto each other. The (permutation) automorphism groups of classical and modern codes are extensively discussed in the literature [4], [6], [7], [8]. To improve decoding, automorphisms must be chosen carefully because the generated diversity might be absorbed by the symmetry of the decoding algorithm [4], [5], [9]. In linear algebra, the definition of automorphisms of a vector space is broader and includes all linear, bijective self-mappings, a fact that is going to be used and analyzed in this paper.

We show that automorphisms according to this more general definition can be beneficial for decoding. Thus, we significantly enlarge the search space for suitable automorphisms. To do so, we describe the (generalized) automorphism group of linear codes defined by a PCM. Additionally, we propose a generalized AED (GAED) algorithm such that generalized automorphisms can be used for decoding. Furthermore, we propose a code construction algorithm for linear codes together with specific automorphisms which enables designing suitable codes for GAED. Finally, we present and compare the performance of GAED for some of our constructed codes.

II. PRELIMINARIES

A linear block code $\mathcal{C}(n, k)$ over a field $\mathbb{F}$ forms a subspace of the vector space $\mathbb{F}^n$. It consists of $|\mathbb{F}|^k$ distinct elements from $\mathbb{F}^n$, where the parameters $n \in \mathbb{N}$ and $k \in \mathbb{N}$ are called block length and information length, respectively. A linear code $\mathcal{C}(n, k)$ can be described as the row span of a

generator matrix $\boldsymbol{G} \in \mathbb{F}^{k \times n}$ or as the null space of its PCM $\boldsymbol{H} \in \mathbb{F}^{(n-k) \times n}$, which we assume to be of full rank [6]:

$$\mathcal{C}(n,k) = \{\boldsymbol{x} \in \mathbb{F}^n : \boldsymbol{H}\boldsymbol{x} = \boldsymbol{0}\} = \mathrm{Null}(\boldsymbol{H}).$$

Note that in contrast to most coding literature, we denote vectors as column vectors in order to directly account for matrix-vector operations common in linear algebra.

For the sake of simplicity, the parameters of the code will be (n,k) and omitted if they are clear from the context.

BP decoding is an iterative message passing algorithm over the Tanner graph of the code. Messages are log-likelihood ratios (LLRs) that are iteratively propagated along the edges and updated in the nodes of the graph [2]. Every linear code can be represented by possibly different PCMs or, equivalently, different Tanner graphs. Although the code is the same, BP decoding behaves differently since the degrees of the nodes and the short cycles within the Tanner graph mainly dominate their performance. For more details on BP decoding, the interested reader is referred to [2].

A. Automorphism Group

In this section, we discuss two different definitions of the automorphism group of a code. To this end, let $\mathcal{C} \subset \mathbb{F}^n$ be a linear code defined over an arbitrary finite field $\mathbb{F}$ and S_n be the symmetric group.

First, according to [6], the automorphism group is defined as the set of mappings $\pi^{(a)}$ with

$$\mathrm{Aut}(\mathcal{C}) := \left\{ \pi^{(a)} : \mathcal{C} \to \mathcal{C}, \boldsymbol{x} \mapsto a\pi(\boldsymbol{x}) : \pi \in \mathrm{S}_n, a \in \mathbb{F} \backslash \{0\} \right\},$$

where $a\pi(\boldsymbol{x}) = \left(ax_{\pi(1)}, \quad \cdots \quad , ax_{\pi(n)} \right)^{\mathsf{T}}$ can be interpreted as a *scaled permutation*. Note that for binary codes, the scaling factor a must be 1 and, hence, is omitted, i.e., $\pi := \pi^{(1)}$.

Second, in linear algebra another definition is standard. Here, the automorphism group $\mathrm{GAut}(\mathcal{C})$ of a vector space $\mathcal{C}$ is defined as all linear, bijective self-mappings [10], i.e.,

$$\mathrm{GAut}(\mathcal{C}) := \{\tau : \mathcal{C} \to \mathcal{C} : \tau \text{ linear}, \tau \text{ bijective}\}.$$

Since scaled permutations are linear, bijective mappings and, thus, $\mathrm{Aut}(\mathcal{C}) \subseteq \mathrm{GAut}(\mathcal{C})$, the latter definition is more general.

B. Automorphism Ensemble Decoding

Fig. 1 shows the block diagram of AED as proposed in [5] for binary codes. A codeword $\boldsymbol{x} \in \mathcal{C} \subset \mathbb{F}_2^n$ is transmitted over a BMSC yielding a received word $\boldsymbol{y} \in \mathcal{Y}^n$, where $\mathcal{Y}$ denotes the channel output alphabet, and resulting in the bit-wise LLR vector $\boldsymbol{L} := (L(y_j|x_j))_{j=1}^n \in \mathbb{R}^n$. Instead of decoding the LLR vector $\boldsymbol{L}$ with only one decoder, it is propagated along K different paths. In each path $i \in \{1, \ldots, K\}$, the LLR vector $\boldsymbol{L}$ is preprocessed according to a permutation automorphism $\pi_i \in \mathrm{Aut}(\mathcal{C})$ as $\pi_i(\boldsymbol{L})$. Afterward, the permuted LLRs $\pi_i(\boldsymbol{L})$ are decoded using an arbitrary decoding algorithm of the code $\mathcal{C}$. This yields several estimates of the permuted versions of the codeword $\pi_i(\hat{\boldsymbol{x}}_i)$. Hereby, every path might possess a distinct decoder. In the next step, applying the inverse automorphisms results in K estimates $\hat{\boldsymbol{x}}_i \in \mathbb{F}_2^n$ of the transmitted codeword $\boldsymbol{x} \in \mathbb{F}_2^n$. Finally, the best candidate is chosen according to an *ML-in-the-list* rule [5].
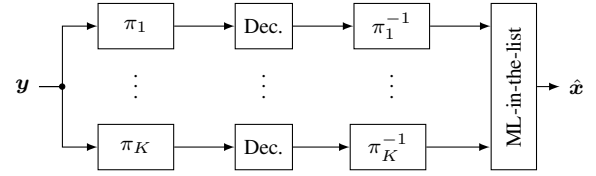

Fig. 1. Block diagram of an AED. K different automorphism $\pi_i \in \mathrm{Aut}(\mathcal{C})$ are chosen from the permutation automorphism group [5].

AED relies on the assumption that if decoding fails in one path, it may succeed in another path. This is not always the case. First, it depends on the interaction of the respective automorphism and the decoder in a path. Second, the ensemble of automorphisms for the different paths must be chosen carefully to improve decoding performance [8].

We consider linear codes that are decoded with a BP decoder using a flooding schedule. As discussed in [4] and [9], the known automorphisms from the permutation automorphism group cannot be used to improve BP decoding for a large variety of LDPC codes since their diversity is absorbed by the symmetry of the PCM resulting from code construction. Thus, either the PCM must be altered after construction as in [4] or other construction methods must be used as in [9].

C. Frobenius Normal Form

The code construction presented below in Sec. IV is based on the Frobenius normal form. To this end, let $\boldsymbol{T} \in \mathrm{GL}_n(\mathbb{F})$, with $\mathrm{GL}_n(\mathbb{F})$ denoting the general linear group, be a non-singular matrix describing a linear, bijective self-mapping $\tau : \mathbb{F}^n \to \mathbb{F}^n$ via $\tau(\boldsymbol{x}) := \boldsymbol{T}\boldsymbol{x}$. Then, there exists a matrix $\boldsymbol{F} \in \mathrm{GL}_n(\mathbb{F})$ of the form

$$\boldsymbol{F} = \begin{pmatrix} \boldsymbol{B}_{f_1} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \boldsymbol{B}_{f_j} \end{pmatrix}$$

with $j \in \mathbb{N}$, consisting of companion matrices $\boldsymbol{B}_{f_i}$ [6, p. 106] of size $d_i \times d_i$ of a polynomial $f_i(x) = \sum_{\ell=0}^{d_i} \alpha_{i,\ell} x^\ell$ with $\alpha_{i,\ell} \in \mathbb{F}$, together with a matrix $\boldsymbol{S}_{\mathrm{F}} \in \mathrm{GL}_n(\mathbb{F})$ such that $\boldsymbol{T} = \boldsymbol{S}_{\mathrm{F}}^{-1} \boldsymbol{F} \boldsymbol{S}_{\mathrm{F}}$ [10]. The matrix $\boldsymbol{F}$ is called the *Frobenius normal form* or, equivalently, *rational canonical form* of $\boldsymbol{T}$ and is determined by $\boldsymbol{T}$ except for the order of the blocks $\boldsymbol{B}_{f_i}$.

III. GENERALIZED AUTOMORPHISMS OF CODES

This section investigates the general automorphism group $\mathrm{GAut}(\mathcal{C})$ of linear codes. Afterward, we propose code design and an adaption of AED that take advantage of using general automorphisms from $\mathrm{GAut}(\mathcal{C})$.

A. Automorphism Group of Parity-Check Codes

In the following, we will derive algebraic properties providing automorphisms of arbitrary codes and enabling a joint construction of linear codes together with their automorphisms. This will be approached by finding automorphisms of $\mathbb{F}^n$ and restricting them to the code subspace $\mathcal{C} \subset \mathbb{F}^n$. Directly finding

automorphisms of $\mathcal{C}$ and exploiting the associated flexibility is part of our ongoing research.

A linear, bijective self-mapping $\tau : \mathbb{F}^n \to \mathbb{F}^n$ based on a non-singular transformation matrix $\boldsymbol{T} \in \mathrm{GL}_n(\mathbb{F})$ is an automorphism of a code $\mathcal{C}$ if and only if

$$\forall \boldsymbol{x} \in \mathcal{C}: \quad \boldsymbol{H}\boldsymbol{x} = \boldsymbol{0} \Longleftrightarrow \boldsymbol{H}\boldsymbol{T}\boldsymbol{x} = \boldsymbol{0}.$$

Thus, in order to identify the automorphisms of a code, non-singular matrices can be used that retain the null space of $\boldsymbol{H}$ under right multiplication, i.e.,

$$\mathrm{Null}(\boldsymbol{H}) = \mathrm{Null}(\boldsymbol{H}\boldsymbol{T}), \tag{1}$$

which is investigated below in Theorem 1. In order to state the theorem, the following definitions are useful:

$$\mathcal{T} := \{\boldsymbol{T} \in \mathrm{GL}_n(\mathbb{F}) : \boldsymbol{T} \text{ fulfills (1)}\},$$

$$\mathcal{Z}(n,k) := \left\{ \boldsymbol{Z} \in \mathrm{GL}_n(\mathbb{F}) : \boldsymbol{Z} = \begin{bmatrix} \boldsymbol{C} & \boldsymbol{0}_{(n-k)\times k} \\ \boldsymbol{D} & \boldsymbol{E} \end{bmatrix} \right\},$$

with $\boldsymbol{C} \in \mathbb{F}^{(n-k)\times(n-k)}$, $\boldsymbol{D} \in \mathbb{F}^{k\times(n-k)}$, and $\boldsymbol{E} \in \mathbb{F}^{k\times k}$. As before, for the sake of simplicity, the parameters (n,k) will be omitted if they are clear from the context.

Theorem 1. *Let $\boldsymbol{H} \in \mathbb{F}^{(n-k)\times n}$ be of rank $n-k$. Then, $\mathcal{T}$ forms a subgroup of the general linear group $\mathrm{GL}_n(\mathbb{F})$ which is conjugated to the matrices in $\mathcal{Z}$, i.e.: $\boldsymbol{T} \in \mathcal{T}$ if and only if there exists $\boldsymbol{Z} \in \mathcal{Z}$ and $\boldsymbol{A} \in \mathrm{GL}_n(\mathbb{F})$ such that $\boldsymbol{T} = \boldsymbol{A}\boldsymbol{Z}\boldsymbol{A}^{-1}$, where $\boldsymbol{A} \in \mathrm{GL}_n(\mathbb{F})$ is a non-singular matrix, termed* code characterization matrix (CCM)*, such that*

$$\boldsymbol{H}\boldsymbol{A} = \tilde{\boldsymbol{H}} = \begin{bmatrix} \boldsymbol{I}_{(n-k)\times(n-k)} & \boldsymbol{0}_{(n-k)\times k} \end{bmatrix}. \tag{2}$$

Proof. First, we consider the case $\boldsymbol{H} = \tilde{\boldsymbol{H}}$. The null space of $\tilde{\boldsymbol{H}}$ is the linear span of the k canonical vectors $\boldsymbol{e}_{n-k+1}, \ldots, \boldsymbol{e}_n \in \mathbb{F}^n$ where the ith entry of $\boldsymbol{e}_i$ is 1 and all other entries are 0. Then, the subspace $\mathrm{Null}(\tilde{\boldsymbol{H}})$ is mapped on itself by all matrices $\boldsymbol{Z} \in \mathcal{Z}$, i.e.,

$$\mathrm{span}\{\boldsymbol{Z}\boldsymbol{e}_{n-k+1}, \ldots, \boldsymbol{Z}\boldsymbol{e}_n\} = \mathrm{span}\{\boldsymbol{e}_{n-k+1}, \ldots, \boldsymbol{e}_n\}, \tag{3}$$

proving the theorem in the case of $\boldsymbol{H} = \tilde{\boldsymbol{H}}$.

An arbitrary matrix $\boldsymbol{H} \in \mathbb{F}^{(n-k)\times n}$ of rank $n-k$ can be transformed into $\tilde{\boldsymbol{H}}$ by applying Gaussian elimination on the columns which can be represented by multiplication of $\boldsymbol{H}$ from the right with a non-singular matrix $\boldsymbol{A}$ as in (2).

Consider again the basis $\{\boldsymbol{e}_{n-k+1}, \ldots, \boldsymbol{e}_n\}$ of $\mathrm{Null}(\tilde{\boldsymbol{H}})$. Then, using (2) and arbitrary $i \in \{1, \ldots, k\}$, it follows that $\boldsymbol{0} = \tilde{\boldsymbol{H}}\boldsymbol{e}_{n-k+i} = \boldsymbol{H}\boldsymbol{A}\boldsymbol{e}_{n-k+i}$. Thus, $\boldsymbol{A}\boldsymbol{e}_{n-k+i}$ is an element of $\mathrm{Null}(\boldsymbol{H})$. In addition, since $\boldsymbol{A}$ is non-singular and the vectors in $\{\boldsymbol{e}_{n-k+1}, \ldots, \boldsymbol{e}_n\}$ are linearly independent, it follows that the vectors in $\{\boldsymbol{A}\boldsymbol{e}_{n-k+1}, \ldots, \boldsymbol{A}\boldsymbol{e}_n\}$ are also linearly independent. Therefore, $\{\boldsymbol{A}\boldsymbol{e}_{n-k+1}, \ldots, \boldsymbol{A}\boldsymbol{e}_n\}$ is a basis of $\mathrm{Null}(\boldsymbol{H})$.

The CCM $\boldsymbol{A}$ is a non-unique change-of-basis matrix mapping the basis $\{\boldsymbol{e}_{n-k+i}\}_{i=1}^{k}$ of $\mathrm{Null}(\tilde{\boldsymbol{H}})$ to the basis $\{\boldsymbol{A}\boldsymbol{e}_{n-k+i}\}_{i=1}^{k}$ of $\mathrm{Null}(\boldsymbol{H})$. Hence, $\boldsymbol{A}^{-1}$ is a change-of-basis matrix from the set $\{\boldsymbol{A}\boldsymbol{e}_{n-k+i}\}_{i=1}^{k}$ to the set $\{\boldsymbol{e}_{n-k+i}\}_{i=1}^{k}$. Thus, conjugation of arbitrary $\boldsymbol{Z} \in \mathcal{Z}$ with $\boldsymbol{A}$ leads to all matrices that fulfill (1). $\square$
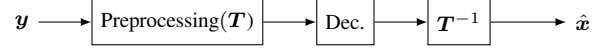

Fig. 2. Path of a GAED if an automorphism $\boldsymbol{T} \in \mathrm{Aut}(\mathcal{C})$ is used.

Theorem 1 provides some structural insights into the automorphism group of a code and states an explicit construction method of automorphisms. Another relevant property of its CCM for the code design is highlighted in Theorem 2 and proven in the appendix.

Theorem 2. *All characteristics of a linear code with PCM $\boldsymbol{H} \in \mathbb{F}^{(n-k)\times n}$, except for its code rate, are contained within the non-singular CCM $\boldsymbol{A} \in \mathrm{GL}_n(\mathbb{F})$. In addition, the inverse of $\boldsymbol{A}$ is of the form:*

$$\boldsymbol{A}^{-1} = \begin{pmatrix} \boldsymbol{H} \\ \boldsymbol{\Lambda} \end{pmatrix}, \tag{4}$$

where $\boldsymbol{\Lambda} \in \mathbb{F}^{k\times n}$ must be chosen such that $\boldsymbol{A}^{-1}$ is of full rank. Furthermore, the CCM $\boldsymbol{A}$ is not unique.

B. Generalized Automorphism Ensemble Decoding

The statements in Sec. III-A hold for arbitrary fields. In this section, we confine ourselves to binary codes $\mathcal{C} \subset \mathbb{F}_2^n$ to adapt AED for automorphisms of $\mathrm{GAut}(\mathcal{C}) \setminus \mathrm{Aut}(\mathcal{C})$. We require a preprocessing of the bit-wise LLRs, as depicted in Fig. 2, for using generalized automorphisms in GAED. In order to describe the effects of $\mathbb{F}_2$-sums on the LLRs, let $(X_i)_{i=1}^{s} \in \mathbb{F}_2^s$ be a sequence of binary random variables. Accordingly, let $(L_i)_{i=1}^{s} \in \mathbb{R}^s$ be their corresponding LLRs. Then, the LLR of the $\mathbb{F}_2$-sum is given by the box-plus operator [11]:

$$L\left(\sum_{i=1}^{s} X_i\right) = 2 \cdot \tanh^{-1}\left(\prod_{i=1}^{s} \tanh\left(\frac{L_i}{2}\right)\right) =: \underset{i=1}{\overset{s}{\boxplus}} L_i. \tag{5}$$

Let $\boldsymbol{x} \in \mathcal{C}$ be an arbitrary codeword that is transmitted over a binary memoryless channel. Consider an automorphism $\boldsymbol{T} \in \mathrm{GAut}(\mathcal{C})$, and define

$$\tilde{\boldsymbol{x}} := \boldsymbol{T}\boldsymbol{x} = \left(\sum_{i=1}^{n} T_{1,i}x_i, \quad \ldots \quad, \sum_{i=1}^{n} T_{n,i}x_i\right)^{\top}. \tag{6}$$

Then, to mimic the effect of this automorphism at the receiver the bit-wise LLRs $L(y_i|x_i)$ are processed according to

$$L(y_j|\tilde{x}_j) = \underset{\substack{i=1, \\ T_{j,i}=1}}{\overset{n}{\boxplus}} L(y_i|x_i), \quad \forall j \in \{1, \ldots, n\}, \tag{7}$$

which follows immediately from (5). Note that permuting the bit-wise LLR vector $\pi_i(\boldsymbol{L})$ with $\pi_i \in \mathrm{Aut}(\mathcal{C})$ in Sec. II-B is a special case of (7). Hence, the proposed GAED algorithm naturally generalizes AED proposed in [5].

According to (7), the numbers of non-zero entries per row of $\boldsymbol{T}$ indicate the number of LLR values that participate in the boxplus summations. To quantify those effects, we conveniently define the *weight* $\Omega(\boldsymbol{T})$ as the number of non-zero elements of $\boldsymbol{T}$ and $\Delta(\boldsymbol{T}) = \Omega(\boldsymbol{T}) - n$ as the *weight over permutation*. Note that, due to $\boldsymbol{T} \in \mathrm{GL}_n(\mathbb{F})$, $\Omega(\boldsymbol{T})$ is lower bounded by n which is only obtained if $\boldsymbol{T} \in \mathrm{Aut}(\mathcal{C})$.

Since the magnitude of an LLR indicates the reliability of the message, it is important to understand the influence of the weight $\Omega(\boldsymbol{T})$ of an automorphism on the magnitude of the LLR after the proposed preprocessing. It can be shown that the magnitude of the outgoing LLR is decreasing if more finite LLRs participate in the boxplus summation. Thus, the preprocessing with an automorphism $\boldsymbol{T} \in \mathrm{GAut}(\mathcal{C}) \backslash \mathrm{Aut}(\mathcal{C})$ leads to an information loss. Hence, we expect the paths of GAED with such automorphisms to individually possess a higher decoding error probability. Still, as long as the performance degradation per path is not too severe, the decoding performance of the full GAED algorithm can improve. Note that the complexity of GAED is comparable to AED because the preprocessing step can be interpreted as one check node update.

IV. CONSTRUCTION OF LINEAR CODES WITH SPARSE AUTOMORPHISMS

Next, we propose a method to construct linear codes $\mathcal{C}$ with a specific, potentially sparse, automorphism $\boldsymbol{T} \in \mathrm{GAut}(\mathcal{C})$. We observe that if $\boldsymbol{T}$ is sufficiently sparse, then often $\boldsymbol{T}^{-1}$ and $\boldsymbol{T}^2$ are also sparse automorphisms usable in GAED. A possible approach is to find a non-singular CCM $\boldsymbol{A}$ such that $\boldsymbol{A}^{-1}\boldsymbol{T}\boldsymbol{A} \in \mathcal{Z}$. The following construction method designs a code $\mathcal{C}$ along with an automorphism of weight Ω_{obj} and is based on the observation that the Frobenius normal form is, in some cases, an element of $\mathcal{Z}$:

1) Choose Ω_{obj} close to n, i.e., $\Delta(\boldsymbol{T})$ close to zero.
2) Sample a matrix $\boldsymbol{T} \in \mathrm{GL}_n(\mathbb{F})$ with $\Omega(\boldsymbol{T}) = \Omega_{\mathrm{obj}}$
   - Determine the sizes d_i of the Frobenius normal form $\boldsymbol{F}$ by solving a set of linear equations [10].
   - Evaluate if the matrices $\boldsymbol{B}_{f_i}$ can be ordered such that $\boldsymbol{F}$ is an element of $\mathcal{Z}$, e.g., by using Theorem 3.
   - Otherwise, repeat step 2).
3) Calculate $\boldsymbol{S}_{\mathrm{F}} = \boldsymbol{A}^{-1}$, such that $\boldsymbol{F} = \boldsymbol{A}^{-1}\boldsymbol{T}\boldsymbol{A} \in \mathcal{Z}$.
4) Extract the PCM, denoted $\boldsymbol{H}_{\mathrm{c}}$, according to Theorem 2.
5) Find an optimized PCM $\boldsymbol{H}_{\mathrm{opt}}$ based on $\boldsymbol{H}_{\mathrm{c}}$.

Theorem 3 states a sufficient condition for the existence of a CCM for a given $\boldsymbol{T}$. A proof is given in the appendix.

Theorem 3. *Let $d_1, \ldots d_j$ be the sizes of the block matrices of the Frobenius normal form $\boldsymbol{F}$ of $\boldsymbol{T}$. If there exists a subset $\mathcal{J} \subseteq \{1, \ldots, j\}$ with $\sum_{i \in \mathcal{J}} d_i = k$, then there exists an ordering of the $\boldsymbol{B}_{f_i}$ yielding an upper-right all-zero block of size $(n - k) \times k$ or $k \times (n - k)$ within $\boldsymbol{F}$. Hence, $\mathcal{C}(n, k)$ and $\mathcal{C}(n, n - k)$ can be constructed.*

Note that the method only yields a PCM. A generator matrix must still be determined, which may contain zero columns. Then, a suitable reduction of $\boldsymbol{G}$, $\boldsymbol{H}$ and $\boldsymbol{T}$ can be performed resulting in a code with smaller block length.

It is not guaranteed that a code constructed with the proposed method has good properties as, e.g., large minimum Hamming distance. In addition, structural properties of the resulting PCM are not yet considered in the first four steps, but is subject to ongoing research. Thus, we currently perform a heuristic optimization in which we randomly choose low-weight dual codewords to construct a full rank PCM. Note

TABLE I
CODE PARAMETERS

Code	$\mathcal{C}_1$	$\mathcal{C}_{1,\mathrm{ref}}$	$\mathcal{C}_2$	$\mathcal{C}_{2,\mathrm{ref}}$	$\mathcal{C}_3$	$\mathcal{C}_{\mathrm{BCH}}$
n	39	39	32	32	63	63
k	24	24	16	16	45	45
$d_{\min}$	6	6	5	8	5	7

that the full dual codebook can be determined for all of the upcoming codes.

V. RESULTS

We analyze the performance of three constructed binary codes $\mathcal{C}_1$-$\mathcal{C}_3$, two reference codes $\mathcal{C}_{1,\mathrm{ref}}, \mathcal{C}_{2,\mathrm{ref}}$ from [12] and a BCH code $\mathcal{C}_{\mathrm{BCH}}$, with parameters outlined in Table I. The minimum Hamming distances were obtained using the methods proposed in [13]. To evaluate the frame error rate (FER), we perform Monte-Carlo simulations using an AWGN channel, accumulating at least 300 frame errors for each SNR. The notation GAED-ℓ-BP–p denotes GAED consisting of ℓ BP path decoders performing p iterations of normalized min-sum decoding (normalization constant $\frac{3}{4}$). All GAEDs rely on three different automorphisms, namely the identity mapping $\boldsymbol{I}$, an automorphism $\boldsymbol{T}$ constructed according to Sec. IV, and its inverse $\boldsymbol{T}^{-1}$. Additionally, as reference, we show results of a redundant row BP decoder, named R-ℓ-BP–p decoder, which performs BP decoding with p iterations using an overcomplete PCM consisting of $\ell \cdot (n - k)$ low-weight dual codewords. This approach is known to potentially improve BP decoding of short block codes [3]. Ordered statistics decoders are used to approximate the ML performances of all codes [14].

Fig. 3-5 depict the FER over E_{b}/N_0 for codes $\mathcal{C}_1$-$\mathcal{C}_3$ based on automorphisms with different weights over permutation. Code $\mathcal{C}_1$ was constructed based on a permutation ($\Delta(\boldsymbol{T}) = 0$), hence GAED equals AED. Codes $\mathcal{C}_2$ and $\mathcal{C}_3$ were designed to have automorphisms with $\Delta(\boldsymbol{T}) = 10$ and $\Delta(\boldsymbol{T}) = 5$, respectively, to show validity of the general approach. For all constructed codes, the performance of GAED-3-BP-10 is compared against two decoders with comparable complexity, namely the BP-30 and the R-3-BP-10. Additionally, Fig. 3-5 depict the performance of the ML decoder for the constructed codes and some reference codes. Both are intended as an indication of the best achievable performance. Since the paper at hand is intended as a proof of concept, a natural gap to this performance is still observed.

We observe that GAED-3-BP-10 results in lower error rates compared to BP-30. For code $\mathcal{C}_1$, GAED-3-BP-10 shows a gain of $0.6\,\mathrm{dB}$ compared to BP-30 at an FER of 10^{-3} and also outperforms R-3-BP-10 by $1\,\mathrm{dB}$. When decoding $\mathcal{C}_2$ and $\mathcal{C}_3$, GAED-3-BP-10 also is able to yield a gain over BP-30. For code $\mathcal{C}_2$, R-3-BP-10 obtains a gain compared to GAED-3-BP-10. However, in higher SNR regime, GAED-3-BP-10 is able to close the gap to R-3-BP-10.

Comparing GAED-3-BP-30 and BP-90, i.e., when decoding with increased complexity, similar improvements can be observed. For code $\mathcal{C}_3$, the performance of GAED-3-BP-10 and BP-90 coincide. Hence, the latter was omitted for clarity. As
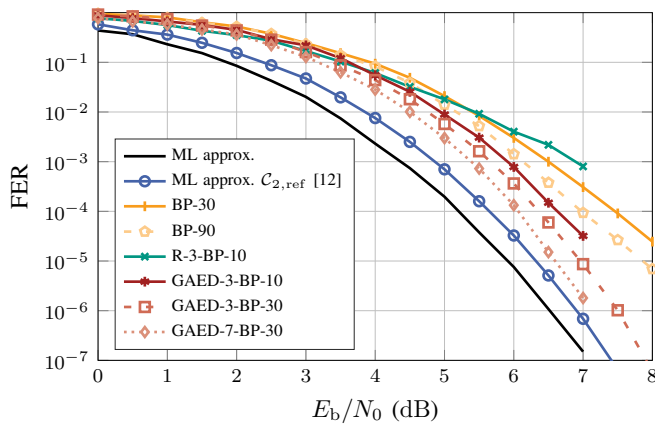
Fig. 3. Performance of different decoders for code $\mathcal{C}_1(39, 24)$. The GAED-7 relies on $\boldsymbol{T}^\alpha$ with $\Delta(\boldsymbol{T}^\alpha) = 0$ and $\alpha \in \{-3, \ldots, 3\}$ in its paths.

Fig. 4. Performance of different decoders for code $\mathcal{C}_2(32, 16)$ with $\Delta(\boldsymbol{T}) = 10$ and $\Delta(\boldsymbol{T}^{-1}) = 13$.

Fig. 5. Performance of different decoders for code $\mathcal{C}_3(63, 45)$ with $\Delta(\boldsymbol{T}) = 5$ and $\Delta(\boldsymbol{T}^{-1}) = 6$.

to construct linear codes together with potentially sparse automorphisms. Finally, we discussed the decoding performance of GAED for three exemplary constructed codes with varying code sizes and rates. In all cases, GAED improved decoding when compared to equal complexity BP decoding. Therefore, this approach is very promising to enable alternative code designs and to improve decoding performance.

REFERENCES

[1] R. G. Gallager, "Low-density parity-check codes," Ph.D. dissertation, Mass. Inst. Tech., Cambridge, 1960.

[2] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, 2008.

[3] T. Hehn, J. B. Huber, S. Laendner, and O. Milenkovic, "Multiple-bases belief-propagation for decoding of short block codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2007.

[4] M. Geiselhart, M. Ebada, A. Elkelesh, J. Clausius, and S. ten Brink, "Automorphism ensemble decoding of quasi-cyclic LDPC codes by breaking graph symmetries," *IEEE Commun. Lett.*, 2022.

[5] M. Geiselhart, A. Elkelesh, M. Ebada, S. Cammerer, and S. ten Brink, "Automorphism ensemble decoding of Reed–Muller codes," *IEEE Trans. Commun.*, 2021.

[6] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error Correcting Codes*. Elsevier, 1977.

[7] T. Berger, "The automorphism group of double-error-correcting BCH codes," *IEEE Trans. Inf. Theory*, 1994.

[8] M. Geiselhart, A. Elkelesh, M. Ebada, S. Cammerer, and S. ten Brink, "On the automorphism group of polar codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2021.

[9] C. Chen, B. Bai, X. Yang, L. Li, and Y. Yang, "Enhancing iterative decoding of cyclic LDPC codes using their automorphism groups," *IEEE Trans. Commun.*, 2013.

[10] P. B. Bhattacharya, S. K. Jain, and S. Nagpaul, *Basic Abstract Algebra*. Cambridge University Press, 1994.

[11] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inf. Theory*, 1996.

[12] M. Grassl, "Bounds on the minimum distance of linear codes and quantum codes," Online available at http://www.codetables.de, 2023, accessed 2023-01-04.

[13] M. Punekar, F. Kienle, N. Wehn, A. Tanatmis, S. Ruzika, and H. W. Hamacher, "Calculating the minimum distance of linear block codes via integer programming," in *Proc. Int. Symp. on Turbo Codes & Iterative Inf. Process. (ISTC)*, 2010.

[14] M. Fossorier and S. Lin, "Soft-decision decoding of linear block codes based on ordered statistics," *IEEE Trans. Inf. Theory*, 1995.

[15] M. Helmling, S. Scholl, F. Gensheimer, T. Dietz, K. Kraft, S. Ruzika, and N. Wehn, "Database of Channel Codes and ML Simulation Results," www.uni-kl.de/channel-codes, 2019, accessed 2023-01-04.

additional reference, we simulated the performance of BP-30 for $\mathcal{C}_{\mathrm{BCH}}$ using the optimized PCM *1min* from [15]. It can be observed that the performance of BP-30 for codes $\mathcal{C}_3$ and $\mathcal{C}_{\mathrm{BCH}}$ coincide over the whole SNR regime. Our simulations indicate that increasing the number of iterations does not yield further improvement for all decoders of $\mathcal{C}_3$. Note that GAED for $\mathcal{C}_2$ and $\mathcal{C}_3$ relies on elements from $\mathrm{GAut}(\mathcal{C}) \setminus \mathrm{Aut}(\mathcal{C})$, serving as proof that the generalized automorphism group can, in fact, improve decoding.

VI. CONCLUSION

In this paper, we have shown that the application of the more general definition of automorphisms prevailing in linear algebra in GAED can be used to improve decoding compared to BP decoding. One important advantage is that this more general definition is expected to simplify the search for suitable transformation significantly. To this end, we first analyzed generalized automorphisms of linear codes based on non-singular mappings of $\mathbb{F}^n$ and proved a specification of their structure introducing the CCM. Then, we described the resulting effects at the receiver and reasoned that generalized automorphisms should possess sparse matrices to prevent severe information loss. Additionally, we introduced a method
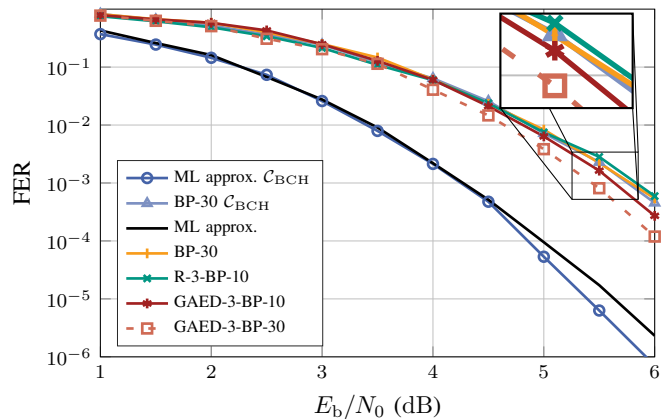
Proof of Theorem 2. Consider a linear code $\mathcal{C}$ with PCM $\boldsymbol{H}$. From Theorem 1, it follows that there exists at least one CCM $\boldsymbol{A} \in \mathrm{GL}_n(\mathbb{F})$ to transform $\boldsymbol{H}$ into $\hat{\boldsymbol{H}}$. If $\hat{\boldsymbol{H}}$ is multiplied from the right with some matrix $\boldsymbol{Z} \in \mathcal{Z}$, then $\hat{\boldsymbol{H}}\boldsymbol{Z} = \hat{\boldsymbol{H}}$ holds. Thus, $\boldsymbol{A}_1 = \boldsymbol{A}\boldsymbol{Z}$ also transforms $\boldsymbol{H}$ into the desired form. Therefore, the matrix $\boldsymbol{A}$ is not unique.

If $\boldsymbol{A}^{-1}$ has the structure described in (4) and the code rate r is known, then the PCM $\boldsymbol{H}$ of $\mathcal{C}$ can be extracted from the inverse CCM. Thus, $\boldsymbol{A}^{-1}$ characterizes $\mathcal{C}$ because a linear code is fully defined by its PCM. Consequently, $\boldsymbol{A}$ also must characterize the code. To prove (4), $\hat{\boldsymbol{H}}$ is multiplied with $\boldsymbol{A}^{-1}$ from the right. Assuming that

$$\boldsymbol{A}^{-1} = \begin{pmatrix} \boldsymbol{U}_{(n-k)\times n} \\ \boldsymbol{\Lambda}_{k\times n} \end{pmatrix},$$

it can be seen that $\hat{\boldsymbol{H}}\boldsymbol{A}^{-1} = \boldsymbol{H}\boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{H}$ and

$$\hat{\boldsymbol{H}}\boldsymbol{A}^{-1} = \begin{bmatrix} \boldsymbol{I}_{(n-k)\times(n-k)} & \boldsymbol{0}_{(n-k)\times k} \end{bmatrix} \begin{pmatrix} \boldsymbol{U} \\ \boldsymbol{\Lambda} \end{pmatrix} \overset{!}{=} \boldsymbol{H}$$

$$\Longleftrightarrow \boldsymbol{U} + \boldsymbol{0}_{(n-k)\times k} = \boldsymbol{U} \overset{!}{=} \boldsymbol{H}.$$

Therefore, the PCM is contained within $\boldsymbol{A}^{-1}$. $\qquad\square$

Proof of Theorem 3. Let $d_1,\ldots d_j$ be the sizes of the block matrices of the Frobenius normal form $\boldsymbol{F}$ of $\boldsymbol{T}$ and let there exist a subset

$$\mathcal{J} \subseteq \{1,\ldots,j\} =: \mathcal{I}$$

with $\sum_{i\in\mathcal{J}} d_i = k$. Then, because the sizes of the block matrices of the Frobenius normal form necessarily sum up to n, i.e., $\sum_{i\in\mathcal{I}} d_i = n$, it follows that $\sum_{i\in\mathcal{I}\setminus\mathcal{J}} d_i = n - k$. Without loss of generality, assume that $\mathcal{J} = \{1,\ldots,m\}$. Define the matrices

$$\boldsymbol{F}_{\mathcal{J}} := \begin{pmatrix} \boldsymbol{B}_{f_1} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \boldsymbol{B}_{f_m} \end{pmatrix} \in \mathbb{F}^{k\times k}, \text{ and}$$

$$\boldsymbol{F}_{\mathcal{I}\setminus\mathcal{J}} := \begin{pmatrix} \boldsymbol{B}_{f_{m+1}} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \boldsymbol{B}_{f_j} \end{pmatrix} \in \mathbb{F}^{(n-k)\times(n-k)}.$$

Then, $\boldsymbol{F}_{\mathcal{J}}$ and $\boldsymbol{F}_{\mathcal{I}\setminus\mathcal{J}}$ are of the form

$$\begin{pmatrix} * & * & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ * & \cdots & \cdots & * & * \\ * & * & \cdots & \cdots & * \end{pmatrix},$$

and there exists

$$\boldsymbol{F} = \begin{pmatrix} \boldsymbol{F}_{\mathcal{I}\setminus\mathcal{J}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{F}_{\mathcal{J}} \end{pmatrix}$$

$$= \begin{pmatrix} \begin{matrix} * & * & & 0 & \cdots & 0 \\ \vdots & \ddots & & \ddots & \ddots & \vdots \\ \vdots & \ddots & & \ddots & \ddots & \vdots \\ * & \cdots & & \cdots & * & * \\ * & * & & \cdots & \cdots & * \end{matrix} & \boldsymbol{0}_{(n-k)\times k} \\[4ex] \boldsymbol{0}_{k\times(n-k)} & \begin{matrix} * & * & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ * & \cdots & \cdots & * & * \\ * & * & \cdots & \cdots & * \end{matrix} \end{pmatrix}$$

$$\in \mathcal{Z}(n,k)$$

Similarly, there exists $\boldsymbol{F} \in \mathcal{Z}(n, n-k)$. Therefore, according to Theorem 2, $\mathcal{C}(n,k)$ as well as $\mathcal{C}(n, n-k)$ can be constructed. $\qquad\square$