



HAL
open science

Construction of Symbol Transformations for Non-Binary Turbo Codes with Lowered Error Floor

Jonas Wilking, Stefan Weithoffer, Charbel Abdel Nour

► **To cite this version:**

Jonas Wilking, Stefan Weithoffer, Charbel Abdel Nour. Construction of Symbol Transformations for Non-Binary Turbo Codes with Lowered Error Floor. ISTC 2023: 12th International Symposium on Topics in Coding, Sep 2023, Brest, France. 10.1109/ISTC57237.2023.10273516 . hal-04168146

HAL Id: hal-04168146

<https://hal.science/hal-04168146>

Submitted on 21 Jul 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Construction of Symbol Transformations for Non-Binary Turbo Codes with Lowered Error Floor

Jonas Wilking, Stefan Weithoffer, Charbel Abdel Nour
 IMT Atlantique, Dpt. MEE, Lab-STICC - UMR 6285
 Email: {jonas.wilking, stefan.weithoffer, charbel.abdelnour}@imt-atlantique.fr

Abstract—Non-binary variants of Turbo and LDPC codes are known to provide significantly improved error correcting performance over their binary counterparts in particular for short frame sizes. One of the reasons is the possibility to directly map the code symbols to constellation symbols of higher order modulations. Moreover, being defined over higher-order Galois fields $GF(q)$, Non-binary Turbo Codes offer high degrees of freedom in the code and interleaver design. In this work, first we analyse the interplay between component codes through the interleaver. Then, we propose a suitable design methodology for a $GF(q)$ symbol transformation applied to the encoded frames by one of the component codes. By modifying the values of encoded symbols by one component code with respect to the other, the aim of such a transformation is to avoid reproducing error-prone sequences while taking into account the effect of the interleaver. Without added complexity to the encoding or to the decoding process, the transformations constructed through the proposed methodology significantly lower the error floor of non-binary turbo codes. Indeed, in a case study for two $GF(16)$ codes, we show an improvement of up to 3 decades in the error floor.

Keywords—Forward Error Correction, Non-binary, turbo codes, coded-modulation, high order constellation

I. INTRODUCTION

Thanks to their iterative decoding, binary turbo and LDPC codes have been adopted into a wide range of wireless communication standards with increasing requirements on decoding throughput and latency. For IoT applications, however, performance metrics like energy- and spectral efficiency are more important leading to short transmit packets and the use of higher order constellations [1].

The non-binary variants of turbo codes (NB-TC) and LDPC codes (NB-LDPC), which are defined over Galois fields $GF(q)$, are often considered important candidates for IoT and short packet applications [1]. This is thanks to the possibility of a direct mapping to high order constellation symbols [2], [3], a significantly improved error correcting performance and lower error floor, outperforming their binary counterparts. Their drawback of having a significant decoding complexity is being addressed both for NB-LDPC [4]–[6] as well as for NB-TC [7], [8] in recent works. To this end, proposed code design methods for NB-TC explicitly aim at a lowering the decoding complexity [9]. The resulting encoder structure with a single memory element improves the accuracy of code design metrics [10] and analysis tools. Hence, it facilitates the emergence of new design methodologies [11].

In [11], it was proposed to employ a *Symbol Transformation* between the inputs of the two component encoders of a NB-

TC in order to improve its error floor at no additional cost in terms of decoding complexity. While [11] gave first results for individual codes, a rigorous design methodology was not given. In this work, we propose a dedicated methodology to construct a symbol transformation for a given NB-TC that explicitly takes into account the interplay between interleaving and error-prone sequences. The symbol transformations found with our methodology are shown to lower the error floor of a given NB-TC by up to three decades.

The remainder of this paper is structured as follows: Section II introduces the necessary mathematical background. Then, Section III describes the proposed design methodology to construct a symbol transformation for a given NB-TC explicitly targeting sequences which predominantly determine the error floor of the code. The design problem is formulated as a decision problem and a SAT-solver is employed to solve it. Section IV proves the NP-completeness of the design problem justifying the use of the SAT-solver before Section V validates the superior error correcting performance when employing the constructed Transformations. Section VI concludes the paper.

II. BACKGROUND

In this section we give the necessary mathematical and contextual background required for the remainder of the paper and introduce the terminology that will be used in Section III.

A. Non-binary turbo codes: structure and constituent codes

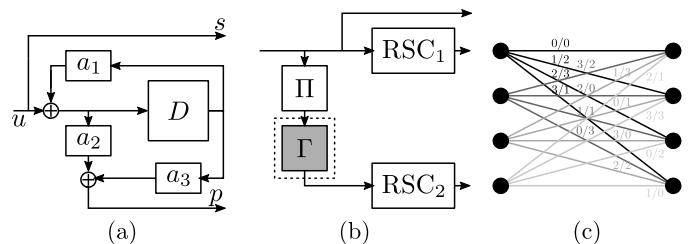


Figure 1: (a) NB-CC Encoder (b) NB-TC Encoder (with Transformation Γ) (c) Trellis of a $GF(4)$ recursive NB-CC.

The encoded non-binary (NB) data symbols are elements of the Galois Field of order q , namely $GF(q)$, where $q = 2^m$ for some integer m . The NB-TCs considered in this work follow the structure proposed in [9] which is illustrated in Fig. 1(b): two NB-Convolutional (NB-CC) encoders concatenated in parallel through an interleaver Π . Additionally,

a *Symbol Transformation* Γ may be employed to improve performance [11]. Applying a single memory element [9], the NB-CC structure of Fig. 1(a) achieves a trade-off between error correcting performance and low decoding complexity for the NB family of codes. Note that for such NB-CCs the number of states is equal to the order of $\text{GF}(q)$ and that the trellis is fully connected for any choice of coefficients $a_i \in \text{GF}(q)$. Furthermore, $a_1 \neq 0$ to guarantee that the NB-CC is a recursive systematic convolutional (RSC) code. The two constituent NB-CCs are from now on called RSC_1 and RSC_2 (see Fig. 1(b)).

The symbols over $\text{GF}(q)$ can be directly mapped to a q -order Quadrature Amplitude Modulation (QAM) or onto several constellation symbols of lower order q' -QAM ($q' < q$) where the q -ary code symbols are decomposed into q' -ary symbols [10]. In the case of a 1-to-1 relation between code symbols and constellation signals, we define the squared *Euclidean Distance* (ED) between symbols, denoted as $\delta(s, s')$, as the ED between the associated signals in the constellation space. Similar for the case ($q' < q$), where the ED between symbols is now defined as the sum of the ED of their composed signals. Further, the cumulated ED between two codewords $x = (x_0, x_1, \dots)$, $y = (y_0, y_1, \dots)$ is defined as $\delta(x, y) = \sum_i \delta(x_i, y_i)$. The ED between input-sequences is denoted as the ED of their encodings. For higher q -QAM constellations, the assumption of a uniform distribution of errors across the constellation signals does not hold anymore [12].

The *Euclidean Distance Spectrum* of a code is the set of all EDs between pairs of codewords and their corresponding multiplicities. The Union Bound (UB) gives an upper bound on the symbol-error rate via the distance-spectrum [9], the first few terms of distance-spectrum (with lowest ED) and their multiplicities determine the error-floor of the code. Therefore codeword pairs with low ED (in the following *low ED codewords* or *low ED sequences*) determine the error-floor of the code. However, due to the non-uniform distribution of errors of the constellation for higher order q -QAM, the distance spectrum computation for the code can no longer rely on the assumption of the transmission of the all-zero NB-symbol codeword.

For NB-CC with fully connected trellis as in Fig 1(c), we can compute the distance-spectrum by enumerating all possible sequence pairs diverging from the same state and later converging (*diverging and converging*, DC) to a common state. DC-sequences have length- L if the two sequences in the trellis are unequal during L stages. It was shown in [11], that the effect of DC sequences of length-2 and length-3 dominates the distance-spectrum of the first terms of NB-CCs of Fig. 1(a). An example of such DC-sequences is shown in Fig. 2 for RSC_1 and RSC_2 . These are composed of two or three segments, their divergence, convergence and optional middle segment.

B. Low-Distance DC Sequences in NB-TCs

Low ED sequences for NB-TCs are obtained when short DC sequences for RSC_1 coincide with short DC sequences for RSC_2 [11] despite the presence of the interleaver. DC-

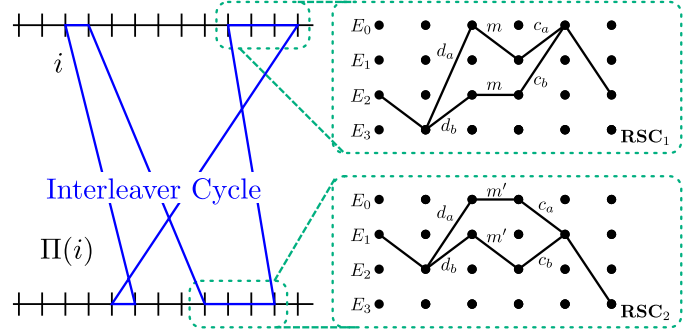


Figure 2: pDC-sequences connected by an interleaver cycle.

Sequences need to differ in at least 2 symbols, namely during divergence and convergence. In the following we denote with d_a, d_b the two symbols during divergence and with c_a, c_b the symbols during convergence of DC-Sequences. Moreover, we call DC-Sequences that differ only during their divergence and convergence *parallel DC-Sequences* or *pDC-Sequences*. NB-TC Sequences with the lowest ED that dominate the UB differ in multiples of pDC-Sequences for a carefully designed interleaver [10]. For these cases, the pDC-Sequences in each constituent encoder are connected by an interleaver cycle as indicated in Fig. 2. There, the symbols of the 2 pDC-Sequences in RSC_1 are rearranged by the interleaver, however, their interleaved version still leads to 2 pDC-Sequences in RSC_2 . Note that the length of a pDC-Sequence can be larger than two (or three as in Fig. 2) when the symbols of the middle-segments of the competing sequences are considered equal for each RSC but can be different between RSCs (we can have $m = m'$ or $m \neq m'$ in Fig. 2). Hence, the length of the interleaver cycle sets a lower bound on the achievable minimum ED for the NB-TC.

C. Symbol Transformation

A *symbol Transformation* Γ is a bijective function defined over the elements of $\text{GF}(q)$ and applied to the values of data symbols before being encoded by RSC_2 . Hence, the input sequence of symbols encoded by RSC_1 undergoes, not only a change in the positions of the symbols due to the interleaver Π , but also a change in the actual values of the symbols through the proposed transformation. This change in the values of the symbols should be designed to improve the lower bound on the minimum ED value set by the interleaver cycle. The resulting structure of Fig. 1(b) can be seen as a generalization of the NB-TC architecture [13]. Indeed, recovering the original structure is achieved when Γ is chosen to be the identity function. There exist $q!$ possible Γ and therefore an exhaustive search is not feasible even for small values of q . In [11] one approach to construct Γ was proposed which is based upon the observation that the competing systematic symbols were always mapped to constellation points whose ED was low. Γ was designed to artificially increase the ED between those symbols for the second component code. Note that similarly an intra-symbol permutation for double-binary TCs was introduced in [2].

While first results in [11] were promising, they were limited to the special case of a Γ for 64-QAM and did not involve dedicated optimization of the symbol mapping with respect to the underlying NB-TC.

III. PROPOSED SYMBOL TRANSFORMATION DESIGN

This section details a dedicated methodology to construct Γ for a given NB-TC that explicitly takes into account the interplay between interleaving and low-ED sequences. To this end we propose to

- 1) Identify the interleaver cycles most likely to constitute the minimal ED by a worst-case analysis then compute the distance-spectrum of pDC-Sequences with lengths dictated by the considered interleaver cycles.
- 2) Apply a design methodology for Γ which prevents worst-case EDs for a given interleaver cycle. It exploits that the corresponding DC-sequences share the same symbol values (albeit interleaved) during convergence and divergence in both constituent encoders. This is associated to the fact that the ED of pDC-Sequences depends on the specific values of divergence and convergence symbol pairs. Thereby, the proposed design methodology for Γ avoids duplicating in both RSCs the specific values of divergence and convergence symbol pairs that lead to the minimum ED, improving the minimum ED of the code.

A. Distance-Spectrum of parallel DC-Sequences

Following a worst-case analysis for short interleaver cycles, we can identify the subset of cycles which lead to the minimal ED of the code. In fact, it was observed that for an algebraic interleaver such as ARP, composite pDC-sequences connected through the interleaver cycles largely contribute to low-ED codewords [11]. In order to evaluate their effect, we need to compute the distance-spectrum of pDC-sequences in RSC₁ and RSC₂. To this end, let us define $A_{s,i}, B_{s,i}$ as the systematic symbols and $A_{p,i}, B_{p,i}$ as the parity symbols of codewords $A = (A_{s,0}, A_{p,0}, \dots)$ and $B = (B_{s,0}, B_{p,0}, \dots)$ respectively that make up a pDC-sequence. A NB-TC DC-sequence consists of the contributions of composite pDC sequences in each component code. Therefore, for the computation of the ED-spectrum of pDC-Sequences in RSC₁, we consider both the systematic and the parity symbols following

$$\delta_1(A, B) := \sum_{k=0 \dots K-1} (\delta(A_{s,k}, B_{s,k}) + \delta(A_{p,k}, B_{p,k})). \quad (1)$$

where $\delta_1(A, B)$ denotes the ED of codewords A and B over their length K . To avoid double-counting of the systematic symbols, for pDC-Sequences in RSC₂ we only consider the parity symbols

$$\delta_2(A, B) := \sum_{k=0 \dots K-1} \delta(A_{p,k}, B_{p,k}). \quad (2)$$

Recall, that short sequence lengths L dominate the error probability. Therefore, we denote by $\delta_{j,L}^{trunc}$ the *truncated ED* for a pDC-Sequence of a specific length L in RSC _{j} .

We denote by $\Delta_{j,L}$ the set set of all $\delta_{j,L}^{trunc}$, providing the distance-spectrum of pDC-Sequences of length L in RSC _{j} . The required complexity for their computation is in $\mathcal{O}(q^{2+L})$. Aside from $\Delta_{j,L}$, we are interested in the symbol pairs (d_a, d_b) at the divergence/convergence trellis steps of pDC-Sequences of length L with a particular $\delta_{j,L}^{trunc}$. We define for each RSC _{j}

$$D_{j,L}(\delta) := \{(d_a, d_b) \in GF(q)^2 \mid \exists \text{ pDC-Sequences in RSC}_j \\ \text{with } \delta = \delta_{j,L}^{trunc}, \text{ where } (d_a, d_b) \text{ are the symbols} \\ \text{during divergence}\} \text{ and}$$

$$C_{j,L}(\delta) := \{(c_a, c_b) \in GF(q)^2 \mid \exists \text{ pDC-Sequences in RSC}_j \\ \text{with } \delta = \delta_{j,L}^{trunc}, \text{ where } (c_a, c_b) \text{ are the symbols} \\ \text{during convergence}\}$$

B. Symbol Transformation Γ and pDC-Sequences

In this section we reintroduce a symbol transformation Γ which precedes RSC₂ as shown in Fig.1(b) and study its effects upon the definitions from the previous subsection. First it can be noted, that Γ has no effect upon $\Delta_{1,L}$ and $\Delta_{2,L}$ as these can be computed independently on the component codes by brute-force. Furthermore, $D_{1,L}$ and $C_{1,L}$ stay unchanged as well, since Γ has no effect on the input symbols of RSC₁. On the other hand, $D_{2,L}$ and $C_{2,L}$ become

$$D_{2,L,\Gamma}(\delta) := \{(d_a, d_b) \in GF(q)^2 \mid \exists \text{ pDC-Sequences in RSC}_2 \\ \text{with } \delta = \delta_{2,L}^{trunc}, \text{ where } (\Gamma(d_a), \Gamma(d_b)) \text{ are the} \\ \text{symbols during divergence}\} \text{ and}$$

$$C_{2,L,\Gamma}(\delta) := \{(c_a, c_b) \in GF(q)^2 \mid \exists \text{ pDC-Sequences in RSC}_2 \\ \text{with } \delta = \delta_{2,L}^{trunc}, \text{ where } (\Gamma(c_a), \Gamma(c_b)) \text{ are the} \\ \text{symbols during convergence}\}$$

Due to the preceding definitions the following holds:

$$D_{2,L,\Gamma}(\delta) = \Gamma^{-1}(D_{2,L}(\delta)) \quad (3)$$

where applying Γ^{-1} to a set of symbol pairs during the divergence step is defined as applying Γ^{-1} to each element of each pair of the set. Note that the same holds for Γ as well. Similarly, for the convergence step :

$$C_{2,L,\Gamma}(\delta) = \Gamma^{-1}(C_{2,L}(\delta)) \quad (4)$$

Next, these functions will serve as basis to prevent that low ED pDC-Sequences coincide in both component codes.

C. Design Methodology

By making use of the definitions of the previous sections, in this section we state the design methodology to construct a symbol Transformation Γ for a given NB-TC, which aims to increase the minimal ED of the code. The proposed methodology can be generalized for any shape of interleaver cycles to further improve the minimal ED. However, for the sake of clarity, we limit the discussion here to shapes as illustrated in Fig. 2 on the example of pDC-Sequences of length l_1 and l_2 in RSC₁ and RSC₂ respectively.

First, we define $\Delta = \min(\Delta_{1,l_1})$, $\Delta' = \min(\Delta_{2,l_2})$ as the minimum ED of the spectrum of each component code and

assume an arbitrary Γ . Then, the worst-case NB-TC ED is achieved if any pDC-Sequence with ED Δ in RSC_1 coincides with any pDC-sequence in RSC_2 with ED Δ' . Due to the interleaver cycle the pDC-Sequences in RSC_1 and RSC_2 share the same symbols during divergence and convergence, and therefore the worst-case can only occur, if there exists some (d_a, d_b) and (c_a, c_b) such that:

$$(d_a, d_b) \in D_{1,l_1}(\Delta) \wedge (c_a, c_b) \in C_{1,l_1}(\Delta) \\ \bigwedge \\ (d_a, d_b) \in D_{2,l_2,\Gamma}(\Delta') \wedge (c_a, c_b) \in C_{2,l_2,\Gamma}(\Delta')$$

The condition on Γ to avoid the occurrence of these worst cases follows consequently as:

$$D_{1,l_1}(\Delta) \cap D_{2,l_2,\Gamma}(\Delta') = \emptyset \bigvee C_{1,l_1}(\Delta) \cap C_{2,l_2,\Gamma}(\Delta') = \emptyset$$

By (3) and (4), this is equivalent to:

$$D_{1,l_1}(\Delta) \cap \Gamma^{-1}(D_{2,l_2}(\Delta')) = \emptyset \\ \bigvee \\ C_{1,l_1}(\Delta) \cap \Gamma^{-1}(C_{2,l_2}(\Delta')) = \emptyset \quad (5)$$

(5) therefore serves as a requirement for the construction of Γ . Assuming that the considered pDC-sequences constitute the minimal ED of the code, we are increasing the minimal ED by employing a Γ that fulfills this requirement.

Eq. (5) can be expressed as boolean formulas with $\ln(q)$ boolean variables for each of the q values of the bijection Γ^{-1} . These boolean variables define the values of Γ^{-1} in binary representation. To this end, we can utilize a SAT-solver¹ to find a Γ that fulfills our requirements, by constructing Γ from a satisfying assignment to the boolean variables.

This idea is employed in the proposed symbol transformation design in Algorithm 1, where we start with the worst-case and add further requirements until the SAT-solver can no longer find a variable assignment satisfying the requirements. There, Ψ_1 denotes the basic constraint that the symbol transformation has to be a bijection and Ψ_2 defines a set of variables that encode integer constants from 0 to $q-1$. $\kappa(U \cap \Gamma^{-1}(V) = \emptyset)$ is defined as the boolean formula which holds if and only if two sets U and $\Gamma^{-1}(V)$ are disjoint by using the previously mentioned boolean variables, where $U, V \subseteq GF(q)^2$.

It is worth noting, that after adding a certain amount of requirements, other shapes of interleaver-cycles will start to become the most critical in their contribution to codewords with the new increased minimal ED. However, as stated above for those cycles the same approach can be also applied to conjunct new requirements to the already existing ones. Further, low ED codewords can consist of more than one pDC segment in each constituent trellis which is equivalent to more than two symbol errors. Regardless, the presented method can be generalized for this case as well.

¹In section IV we show that the problem of finding Γ that fulfills a set of requirements is NP-complete which justifies the use of SAT-solver.

Algorithm 1 Design symbol transformation Γ

```

 $\varphi_0 := \Psi_1 \wedge \Psi_2$ 
 $\Lambda := \{\lambda \in \mathbb{R} \mid \lambda = \Delta + \Delta', \Delta \in \Delta_{1,l_1} \wedge \Delta' \in \Delta_{2,l_1}\}$ 
 $i := 0$ 
while  $\varphi_i$  is satisfied and  $\Lambda \neq \emptyset$  do
   $i = i + 1$ 
   $\theta = \min(\Lambda)$ 
   $\psi := true$ 
  for all  $\Delta \in \Delta_{1,l_1}, \Delta' \in \Delta_{2,l_2}$ , such that  $\theta = \Delta + \Delta'$  do
     $\alpha = \kappa(D_{1,l_1}(\Delta) \cap \Gamma^{-1}(D_{2,l_2}(\Delta')) = \emptyset)$ 
     $\beta = \kappa(C_{1,l_1}(\Delta) \cap \Gamma^{-1}(C_{2,l_2}(\Delta')) = \emptyset)$ 
     $\gamma = \alpha \vee \beta$ 
     $\psi = \psi \wedge \gamma$ 
  end for
   $\varphi_i = \varphi_{i-1} \wedge \psi$ 
   $\Lambda = \Lambda \setminus \{\theta\}$ 
end while
Construct  $\Gamma$  from satisfying assignment of  $\varphi_{i-1}$ 

```

IV. NP-COMPLETENESS OF DESIGN-PROBLEM

First, we state the encountered design problem as a decision problem, then we give a poly-time reduction from the k-Clique decision problem.

Definition IV.1 (design problem). *Assume two families of sets U_0, \dots, U_{k-1} and W_0, \dots, W_{k-1} over a finite set X and $U_i, W_i \subseteq X^2$.*

Problem: *Does there exist a bijection $\varphi : X \rightarrow X$ such that:*

$$\bigwedge_{i=0}^{k-1} U_i \cap \varphi(W_i) = \emptyset$$

where

$$\varphi(\{x_0, \dots, x_n\}) := \{\varphi(x_0), \dots, \varphi(x_n)\}$$

1) *Design Problem is in NP:* Given a bijection φ , it can be checked in polynomial time if the condition holds. The problem is therefore in NP according to the certificate definition.

2) *Design Problem is NP-hard:* Recall that an undirected graph G has a k -clique if there exists a subset of k vertices such that for every pair of vertices u, v of this subset, there exists the edge $\{u, v\}$ in G . We denote an undirected graph as $G = (V, E)$, with $V := \{1, \dots, n\}$ and $E \subseteq V^2$. Further, for $k \leq n$ we denote $C_k := \{\{u, v\} \in V^2 \mid u, v \leq k, u \neq v\}$, therefore $G = (V, C_k)$ has a k -Clique.

Lemma IV.1. *An undirected graph $G = (V, E)$ has a Clique of size larger or equal to k if and only if there exists a bijection $\varphi : V \rightarrow V$ such that $\bar{E} \cap \varphi(C_k) = \emptyset$, where $\bar{E} := V^2 \setminus E$*

Proof.

\Rightarrow Since G has a Clique of size larger or equal to k it has also a clique of exactly size k , thus there exists a subset $E_k \subseteq E$ that constitutes a clique of size k . Also,

$E_k \cap \bar{E} = \emptyset$, therefore there exists a bijection $\varphi : V \rightarrow V$ with $\varphi(C_k) = E_k$, which implies $\bar{E} \cap \varphi(C_k) = \emptyset$.
 \Rightarrow If \exists a bijection φ such that $\bar{E} \cap \varphi(C_k) = \emptyset$ then $\varphi(C_k) \subseteq E$. Thus, G has a Clique of at least size k . \square

Definition IV.2 (Reduction). Given $k \in \mathbb{N}$ and an undirected graph $G = (V, E)$. Construct $\bar{E} := V^2 \setminus E$ and $C_k \subseteq V^2$. If $k \leq |V|$, formulate a design problem instance with:

$$U_0 = \bar{E} \text{ and } W_0 = C_k$$

else:

$$U_0 = V^2 \text{ and } W_0 = V^2$$

The correctness follows from the previous Lemma and the reduction can be done in polynomial time. This reduces the k -Clique Problem to our design problem.

V. SIMULATION RESULTS

We evaluate the proposed transformation design method in a case study for two NB-TsC over GF(16) with primitive-polynomial $x^4 + x^3 + 1$, which are mapped to a 16-QAM constellation: C_1 with component code parameters $a_1 = 2$, $a_2 = 5$ and $a_3 = 11$, which has a good component code min ED of 4.0, and C_2 with $a_1 = 4$, $a_2 = 2$ and $a_3 = 1$ with a comparatively low component code min ED of 1.6. Note, that the employed interleaver (ARP interleaver with $P = 143$, $Q = 8$, $S = (0, 150, 62, 10, 110, 22, 20, 98)$) was not optimized with respect to C_1 and C_2 to better illustrate the effect of the transformation. Their respective transformations constructed through Algorithm 1 are listed in table I.

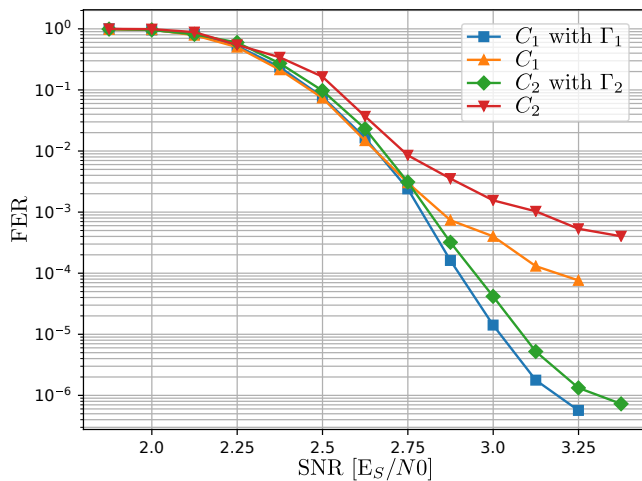


Figure 3: FER comparison ($K = 160$, 16-QAM, 8 iterations max-Log-MAP decoding, AWGN Channel).

When used without Γ , both C_1 and C_2 suffer from high error floors at Frame Error Rate (FER) of 10^{-3} and 10^{-4} respectively, as seen in Fig.3. The observed floors are significantly reduced to a FER of 10^{-6} when using C_1 and C_2 as component codes together with Γ_1 , Γ_2 respectively.

y	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\Gamma_1(y)$	6	5	12	15	10	9	0	3	2	1	8	11	14	13	4	7
$\Gamma_2(y)$	12	11	5	2	4	3	13	10	14	9	7	0	6	1	15	8

Table I: Symbol transformations for use with C_1 and C_2 .

VI. CONCLUSION

In this work we presented a new methodology to construct a symbol transformation dedicated to a given NB-TC over GF(q). This methodology aims to increase the minimal ED of the code by explicitly targeting low-ED sequences which are caused by interleaving. Compared to a NB-TC without symbol transformation, we were able to significantly increase the error-correcting performance, hence lowering the error floor by as much as three decades in a case study. The encountered design problem was proven to be NP-complete justifying the utilization of SAT-Solver in the proposed methodology. In future work, it is expected that investigating joint optimization of interleaver and symbol transformation will lead to further significant improvements.

ACKNOWLEDGMENTS

We acknowledge the kind support by Prof. Dr.-Ing. Hans D. Schotten of RPTU Kaiserslautern-Landau. This work was partially funded by the French National Research Agency TurboLEAP project (ANR-20-CE25-0007).

REFERENCES

- [1] "QCSP project: Quasi cyclic small packet," <https://qcsproj.univ-ubs.fr/>.
- [2] C. Douillard and C. Berrou, "Turbo codes with rate- $m/(m+1)$ constituent convolutional codes," *IEEE Trans. on Commun.*, vol. 53, no. 10, pp. 1630–1638, 2005.
- [3] R. Klaimi, S. Weithoffer, and C. Abdel Nour, "Improved non-uniform constellations for non-binary codes through deep reinforcement learning," in *IEEE 23rd Int. Workshop on Signal Process. Advances in Wireless Commun. (SPAWC)*, 2022, pp. 1–5.
- [4] A. Voicila, D. Declercq, F. Verdier, M. Fossorier, and P. Urard, "Low-complexity, low-memory EMS algorithm for non-binary LDPC codes," in *IEEE Int. Conf. on Commun.*, Glasgow, UK, 2007, pp. 671–676.
- [5] E. Boutillon and L. Conde-Canencia, "Simplified check node processing in nonbinary LDPC decoders," in *6th Int. Symp. on Turbo Codes and Iter. Proc. (ISTC)*, Brest, France, Sept 2010, pp. 201–205.
- [6] —, "Bubble check: a simplified algorithm for elementary check node processing in extended min-sum non-binary LDPC decoders," *Electron. Lett.*, vol. 46, no. 9, pp. 633–634, 2010.
- [7] R. Klaimi, C. Abdel Nour, C. Douillard, and J. Farah, "Low-complexity decoders for non-binary turbo codes," in *10th IEEE Intern. Symp. on Turbo Codes Iter. Inf. Proc. (ISTC)*, Hong Kong, China, Dec. 2018.
- [8] H. L. Blevic, R. Klaimi, S. W. C. A. Nour, and A. Baghdadi, "Low complexity non-binary turbo decoding based on the local-sova algorithm," in *11th IEEE Int. Symp. on Topics in Coding (ISTC)*, 2021, pp. 1–5.
- [9] R. Klaimi, C. Abdel Nour, C. Douillard, and J. Farah, "Design of Low-Complexity Convolutional Codes over GF(q)," in *IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, UAE, Dec 2018.
- [10] R. Klaimi, C. Abdel Nour, C. Douillard, and J. Farah, "Union bound evaluation for non-binary turbo coded modulations," *IEEE Commun. Letters*, vol. 24, no. 6, pp. 1178 – 1182, 2020. [Online]. Available: <https://imt-atlantique.hal.science/hal-02502303>
- [11] R. Klaimi, "Study of non-binary turbo codes for future communication and broadcasting systems," Theses, IMT Atlantique, Jul. 2019. [Online]. Available: <https://theses.hal.science/tel-02543195>
- [12] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE trans. on info. theory*, vol. 44, no. 3, pp. 927–946, 1998.
- [13] C. Berrou and M. Jezequel, "Non-binary convolutional codes for turbo coding," *Electron. Lett.*, vol. 35, no. 1, pp. 39–40, 1999.