# Short Block Length Code Design for Interference Channels

Shahrouz Sharifi*, Mehdi Dabirnia†, A. Korhan Tanc‡ and Tolga M. Duman†
*School of ECEE, Arizona State University, Tempe, AZ 85287-5706, USA
†Dept. of Electrical and Electronics Engineering, Bilkent University, Bilkent 06800, Ankara, Turkey
‡Dept. of Electrical and Electronics Engineering, Kirklareli University, Kayali 39100, Kirklareli, Turkey
Email: sh.sharifi@asu.edu, {mehdi, duman}@ee.bilkent.edu.tr, korhan.tanc@kirklareli.edu.tr

***Abstract*—We focus on short block length code design for Gaussian interference channels (GICs) using trellis-based codes. We employ two different decoding techniques at the receiver side, namely, joint maximum likelihood (JML) decoding and single user (SU) minimum distance decoding. For different interference levels (strong and weak) and decoding strategies, we derive error-rate bounds to evaluate the code performance. We utilize the derived bounds in code design and provide several numerical examples for both strong and weak interference cases. We show that under the JML decoding, the newly designed codes offer significant improvements over the alternatives of optimal point-to-point (P2P) trellis-based codes and off-the-shelf low density parity check (LDPC) codes with the same block lengths.**

***Index Terms*—Interference channel, short block length codes, convolutional codes, union bound.**

## I. Introduction

Communication of several sender-receiver pairs using a shared medium can be modelled as an interference channel. There has been extensive research on communication over such channels throughout the last several decades. Specifically, many information theoretic problems have been addressed, see, e.g., [1], [2]. On the other hand, there are only a limited number of studies on practical code designs, see [3]–[5] for some examples for Gaussian interference channels (GICs). These papers focus on design of low density parity check (LDPC) codes for two-user GICs at asymptotically long block lengths and obtain optimized codes operating close to capacity or achievable rate region boundaries. As a complementary study, in this paper, we consider design of short block length codes for GICs, which is motivated by practical applications with stringent decoding delay and complexity constraints.

The results of [5] show that in the large block length regime, optimized irregular LDPC codes have a performance close to the capacity or rate-region boundaries, however, for short block lengths the asymptotic design assumptions do not hold and the sample codes from the optimized ensembles perform considerably worse due to the facts that 1) the degree distribution of the short block length codes do not exactly match the optimized degree distribution, and 2) there are inevitable cycles in the Tanner graph of these codes deteriorating the iterative decoder performance. Different approaches for constructing short block length LDPC codes based on girth-conditioning and avoiding small stopping sets have been proposed in the literature [6]–[9], and performance improvements over random construction techniques have been observed for P2P transmissions. On the other hand, to the best of our knowledge, there is no joint design technique for LDPC codes over GICs in the short block length regime.

Recently, short block length codes have been designed for two-user Gaussian multiple access channels (GMACs) employing trellis-based codes [10]. Such codes are successfully employed for P2P channels, and it is shown that they can achieve superior performance in space-time coding scenarios particularly for quasi-static fading channels [11]. It is also possible to implement optimal decoders for such codes even in certain multi-user setups, and compute performance bounds in an efficient manner. With this motivation, we consider the case of two-user GICs employing trellis-based codes and derive error-rate bounds in order to design optimal codes with short block lengths and study their performance via several code design examples.

The paper is organized as follows. In Section II, we introduce the system model for a two-user GIC. We utilize the bounds derived for GMACs and Gaussian broadcast channels (GBCs) from the exisiting literature towards developing a framework for designing trellis-based codes for two-user GICs under strong and weak interference, and present specific design procedures in Section III. We provide several code design examples for both cases in Section IV, and finally, we conclude the paper in Section V.

## II. System Description

Fig. 1 illustrates the block diagram of a two-user GIC. Considering receiver $i$, the $n$-length received signal vectors can be written as

$$\mathbf{y}_i = \boldsymbol{\alpha}_i \mathbf{c} + \mathbf{z}_i, \quad i = 1, 2, \tag{1}$$

where $\mathbf{c}$ denotes the BPSK modulated transmitted codeword matrix as follows

$$\mathbf{c} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} = \begin{bmatrix} c_{1,1} \; c_{1,2} \; \cdots \; c_{1,n} \\ c_{2,1} \; c_{2,2} \; \cdots \; c_{2,n} \end{bmatrix}, \tag{2}$$

with $\mathbf{c}_1$ and $\mathbf{c}_2$ representing the codewords employed at transmitter 1 and transmitter 2, respectively. Note that here we restrict ourselves to the case of one codeword per user and do not implement Han-Kobayashi encoding with both public

This work was supported in part by the National Science Foundation under the grant NSF-CCF 1117174, and by the Turkish Scientific and Technological Research Council of Turkey (TUBITAK) under the grant 114E601.

and private messages. Future research can consider such a generalization. The channel gains from the transmitters to the receiver $i$ are denoted as $\boldsymbol{\alpha}_i = [\alpha_{1i}\ \alpha_{2i}]$, where $\alpha_{ji}$ is a real number denoting the gain of the channel from the transmitter $j$ to the receiver $i$. Note that for a more realistic channel model $\alpha_{ji}$ can be taken as complex but we consider real valued channel coefficients for the simplicity of the analysis. The independent and identically distributed (i.i.d.) zero mean Gaussian noise samples with variance $\frac{N_0}{2}$ at receiver $i$ are represented by the vector $\mathbf{z}_i$ of length $n$. The SNR and INR at receiver $i$ are defined as

$$SNR_i = \frac{\alpha_{ii}^2 P_i}{N_0}, \quad INR_i = \frac{\alpha_{ji}^2 P_j}{N_0}, \tag{3}$$

where $i, j = 1, 2$, and $P_i$ is the average transmit power per coded bit at the transmitter $i$. Based on the interference and signal levels, the interference can be categorized as strong (if $INR_i > SNR_j$), weak (if $SNR_i > INR_j$), or mixed (if $INR_i > SNR_j,\ INR_j < SNR_i$) with $i \neq j$ [12].

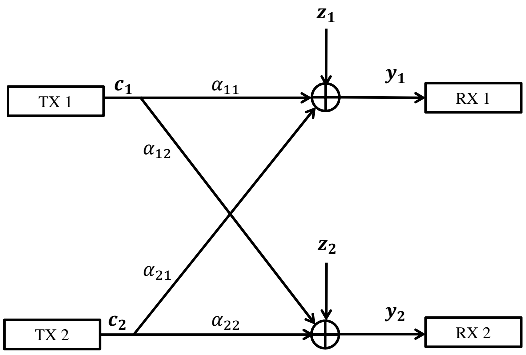


Fig. 1: Block diagram of a two-user GIC.

We consider two different decoding methods at the receiver side, namely, joint maximum likelihood decoding and single user decoding.

**Joint Maximum likelihood (JML) decoding**. Optimal decoding of both the messages is performed based on the ML criterion which can be written as

$$(\mathbf{c}_1^{(JML)}, \mathbf{c}_2^{(JML)}) = \underset{(\hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2)}{\arg\min}\ \|\mathbf{y}_i - \alpha_{1i}\hat{\mathbf{c}}_1 - \alpha_{2i}\hat{\mathbf{c}}_2\|^2, \quad i = 1, 2, \tag{4}$$

where $\|.\|$ denotes the Euclidean norm of the vector, and the minimization is performed over both codebooks.

**Single user (SU) decoding**. In this method the interfering signal is treated as noise and only the desired signal is decoded according to the minimum distance criterion, that is,

$$\mathbf{c}_i^{(SU)} = \underset{\hat{\mathbf{c}}_i}{\arg\min}\ \|\mathbf{y}_i - \alpha_{ii}\hat{\mathbf{c}}_i\|^2, \quad i = 1, 2, \tag{5}$$

where the minimization is done over the entire codebook for the $i$th message.

## III. Performance Analysis

### A. *Strong Interference Case*

For GIC with strong interference, we consider JML decoding which is the optimal decoding rule of both users' messages at each receiver. As a consequence, the performance analysis technique utilized in [10] for GMACs under joint ML decoding can be exploited to derive performance bounds for our set-up under strong interference aiming at decoding of both messages.

Using the union bound, the overall frame error probability (i.e., probability of the union of error events at both receivers) can be upper-bounded as

$$P_\varepsilon \leq \frac{1}{|\mathcal{C}|} \sum_{\mathbf{c}} \sum_{\hat{\mathbf{c}} \neq \mathbf{c}} \Big( P_{\varepsilon,1}(\mathbf{c}, \hat{\mathbf{c}}) + P_{\varepsilon,2}(\mathbf{c}, \hat{\mathbf{c}}) \Big) \tag{6}$$

where $\mathcal{C}$ denotes the set of codeword pairs $\mathbf{c}$, and $|.|$ denotes the cardinality of the set. $P_{\varepsilon,i}(\mathbf{c}, \hat{\mathbf{c}})$ is the pairwise error probability at $i$th receiver which is the probability that the received signal is closer to another codeword pair $\hat{\mathbf{c}}$ instead of $\mathbf{c}$ when $\mathbf{c}$ is transmitted. $P_{\varepsilon,i}(\mathbf{c}, \hat{\mathbf{c}})$ can be expressed as follows

$$P_{\varepsilon,i}(\mathbf{c}, \hat{\mathbf{c}}) = Q\left( \sqrt{\frac{Ed_i^2(\mathbf{c}, \hat{\mathbf{c}})}{2N_0}} \right), \tag{7}$$

where $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp(-\frac{t^2}{2}) dt$, and $Ed_i^2(.,.)$ is the squared Euclidean distance function computed at receiver $i$ as

$$Ed_i^2(\mathbf{c}, \hat{\mathbf{c}}) = \alpha_i \mathbf{D}_{\mathbf{c},\hat{\mathbf{c}}} \alpha_i^\dagger, \tag{8}$$

where "$\dagger$" denotes the transpose and $\mathbf{D}_{\mathbf{c},\hat{\mathbf{c}}}$ represents the codeword difference matrix given by

$$\mathbf{D}_{\mathbf{c},\hat{\mathbf{c}}} = (\mathbf{c} - \hat{\mathbf{c}})(\mathbf{c} - \hat{\mathbf{c}})^\dagger. \tag{9}$$

One main difficulty in computing (6) is the complexity of enumeration of the multiplicities of the codeword difference matrix $\mathbf{D}_{\mathbf{c},\hat{\mathbf{c}}}$ for all possible correct-erroneous codeword pairs. On the other hand, for certain cases such as convolutional codes, this matrix can be computed efficiently and in a systematic manner. In the following, we follow the approach developed in [10] to count the multiplicities of different $\mathbf{D}_{\mathbf{c},\hat{\mathbf{c}}}$ terms for use in the bound computations. The code optimization can then be simply performed by searching for pairs of codes minimizing the bound (6) computed at a specific SNR.

Consider a two-user joint trellis diagram with states labeled as $(s_1, s_2)$ with $s_i$ representing the state of the trellis for the code of $i$th user. The joint trellis has $n_{s_1} \times n_{s_2}$ states with $n_{s_i}$ denoting the number of states for the $i$th user's code. To track all possible codeword pairs $(\mathbf{c}, \hat{\mathbf{c}})$, a product state trellis diagram with states $(s_1, s_2, \hat{s}_1, \hat{s}_2)$ is formed wherein $s_i$ and $\hat{s}_i$ represent the states of the trellises corresponding to the codes $\mathbf{c}_i$ and $\hat{\mathbf{c}}_i$, respectively. To count the multiplicities of all possible $\mathbf{D}_{\mathbf{c},\hat{\mathbf{c}}}$'s over the joint code of the two users, a state transition matrix $S_{1,2}$ is assigned to the product state trellis whose element in the $k$th row and the $l$th column is either zero corresponding to the case where the transition from state $k$ to state $l$ is not allowed, or in the form of

$$[S_{1,2}]_{k,l} = D_{11}^{q_{11}^{k,l}} \times D_{12}^{q_{12}^{k,l}} \times D_{22}^{q_{22}^{k,l}}, \quad k, l = 1, ..., (n_{s_1} \times n_{s_2})^2 \tag{10}$$

where $D_{11}$, $D_{12}$, $D_{22}$ are dummy variables used to list the multiplicities of the different types of errors between two pairs of codewords [14]. The exponent $q_{i,i}^{k,l}$ is the number of indices in which the label corresponding to the transition from state $k$ to state $l$ in the $i$th user trellis differs from the all-zero label. The exponent $q_{i,j}^{k,l}$, $i \neq j$, is the number of coincidences between the previous indices for $i$th and $j$th user. Each of these components is used to compute the contribution of the transition from state $k$ to state $l$ to the corresponding entry of the $\mathbf{D}_{\mathbf{c},\hat{\mathbf{c}}}$.

It has been shown that considering all the pairwise error events in calculating the union bound results in a very loose bound [14]. In order to tighten the union bound we can use an expurgation technique which is nothing but considering only simple error events defined as errors associated with the paths that diverge from the correct path through the trellis in only one segment of the trellis diagram [14]. To efficiently count the simple error events, the technique given in [14] is adopted. That is, an error state in the product trellis is introduced. The transition to the error state occurs only when the two paths that diverged previously merge for the first time. Also, the only possible transition from this state is to itself. Considering $L$ stages of the joint trellis state transition, the complete list of possible $\mathbf{D}_{\mathbf{c},\hat{\mathbf{c}}}$ types for the transmitted codewords is obtained via calculating the $L$th power of $S_{1,2}$. Taking the trellis termination into account, the final stages of state transition matrix considered in the computation are modified accordingly [14].

Despite simplicity of the approach, the exact calculation of the bound through this method has a high computational cost, therefore it is not directly suitable for code design. Authors in [10] simplify the code design process by considering a shorter frame length than the intended design length. This is motivated by the fact that the decoding performance of the convolutional codes does not change significantly by considering a traceback length of four to five times of the constraint length of the code [15, Ch. 4]. In other words, even though the computed bounds would differ for different codeword lengths, the performance of the codes can be ordered based on their performance estimated for a sufficiently large (but relatively small) length code which is manageable.

Another simplification is performed to increase the computational efficiency where the number of terms for each entry of $S_{1,2}$ is restricted to those components ($q_{ij}$) with magnitudes less than a specific threshold knowing that the omitted terms do not affect the error bound considerably [14]. Although this greatly helps with the computation, the final computations based on this truncation approach should be considered as approximations rather than being true upper-bounds.

### B. Weak Interference Case

We consider the use of SU decoding for calculating the performance bounds under weak interference which resembles the first stage of the interference cancellation (IC) decoding for GBCs [13]. The SU decoding can be considered as treating both users' messages as private. Note that the employed decoding is not an instance of the ML decoding which is used for the strong interference case. In fact, we can obtain a similar bound to the case of strong interference if the JML decoding is utilized, however, we utilize the SU decoding to simplify performance bound computations, and accordingly the code design approach. In essence, the performance of the employed decoding scheme becomes close to that of the ML decoding when the interference levels at the receivers are negligible compared to the desired signals.

Using the union bound, the overall frame error probability under SU decoding can be upper-bounded as,

$$P_\varepsilon \leq P_{\varepsilon,1} + P_{\varepsilon,2}, \tag{11}$$

where

$$P_{\varepsilon,i} \leq \frac{1}{|\mathcal{C}|} \sum_{\mathbf{c}_i} \sum_{\hat{\mathbf{c}}_i \neq \mathbf{c}_i} P_{\varepsilon,i}(\mathbf{c}_i, \hat{\mathbf{c}}_i), \tag{12}$$

where $|\mathcal{C}|$ denotes the cardinality of $i$th user's codebook.

It is shown in [13] that the pairwise error probability for this case $P_{\varepsilon,i}(\mathbf{c}_i, \hat{\mathbf{c}}_i)$ can be calculated as

$$P_{\varepsilon,i}(\mathbf{c}_i, \hat{\mathbf{c}}_i) = Q\left(\sqrt{\frac{2f(d_{ii}, d_{ji}, \alpha_{ii}, \alpha_{ji})}{N_0}}\right), \quad i,j = 1,2, \quad i \neq j, \tag{13}$$

where $f$ is defined by

$$f\Big(d_{ii}, d_{ji}, \alpha_{ii}, \alpha_{ji}\Big) = \frac{\Big(\alpha_{ii} d_{ii} + \alpha_{ji}(d_{ii} - 2d_{ji})\Big)^2}{d_{ii}}, \tag{14}$$

Here $d_{ii}$ is the number of bit errors in $\hat{\mathbf{c}}_i$ compared to $\mathbf{c}_i$ and $d_{ji}$ is the number of positions where $\mathbf{c}_j$ and $\mathbf{c}_i$ differ among the positions where $\hat{\mathbf{c}}_i \neq \mathbf{c}_i$. In order to list the multiplicities of different values of $d_{ii}$ and $d_{ji}$, a product state trellis is constructed to which two state transition matrices $S'_{1,2}$ and $S'_{2,1}$ are associated. The entry in the $k$th and $l$th row of $S'_{i,j}$ is computed as

$$[S'_{i,j}]_{k,l} = D_{11}^{d_{ii}^{k,l}} \times D_{12}^{d_{ji}^{k,l}}, \quad k,l = 1,...,(n_{s_1} \times n_{s_2})^2, \tag{15}$$

where $i,j \in \{1,2\}, i \neq j$. Similar to the approach taken in [10], the computed state transition matrices are utilized towards listing the possible values of $d_{ij}$ with their multiplicities characterizing the upper bound (11). Similar simplifications are performed to cope with the memory limitations, therefore the computed values are treated as approximations rather than being actual performance upper bounds on the error probability.

The code design is carried out by searching for the code pair minimizing the upper bound in (11).

## IV. Code Design Examples

We consider codes with rates of $0.5$ and block lengths $N = 96$. The performances of the optimized trellis-based codes are compared against those of LDPC codes (96.33.964) and (96.33.966) taken from [16]. The memory of trellis-based codes is 2, therefore termination for each user's code is achieved via the last two information bits. The trellis-based codes are represented in octal form; i.e., $(m_1, n_1)/(m_2, n_2)$ represents the codes adopted for the GIC where the code $(m_i, n_i)$ represents the convolutional encoder in octal notation for the transmitter $i$. The code optimization is carried out through ordering the codes' performance by computing the approximate bounds in (6) and (11). To efficiently handle the matrix multiplications and cope with the memory limitations, the number of terms for each entry of the state transition matrix is truncated to 25.

### A. Strong Interference Case

For the first example, we consider a GIC with $SNR_1 - SNR_2 = 2$ dB, $INR_1 - SNR_2 = 1$ dB, and $INR_2 - SNR_1 = 2$ dB. Code design is performed by minimizing the performance bound (6) at $SNR_1 = 8$ dB over the codes with 4 states. The minimum value of the upper-bound is achieved for the code $(2,5)/(5,7)$. For comparison purposes, we also consider codes designed for P2P channels. In order to employ the P2P codes for the two-user setup, an *interleaved* scheme is adopted where the same code with different assignment of generator matrices to the output bits are used for different users. That is, for the first user we employ the code $(5,7)$, which has the largest minimum distance among the codes with memory 2. For the second user, we adopt the code $(7,5)$, which obviously is the same code with $(5,7)$ with a different assignment of coded bits.

Fig. 2 illustrates the decoding performance of the trellis-based codes and the LDPC codes employed for the considered GIC. The performance of LDPC codes is simulated for both soft interference cancellation (SIC) [17] and single user decoding (SUD) techniques. For the former technique, each receiver adopts a joint decoder and aims at *partially* decoding the interfering signal in an iterative manner helping the overall decoding process, while for the latter, each receiver treats the interfering signal as noise. It is evident that the SIC scheme provides a better performance than the SUD. It is also observed that the optimized trellis-based codes outperform the P2P optimal codes, both offering a better performance than LDPC codes even for the case of SIC.

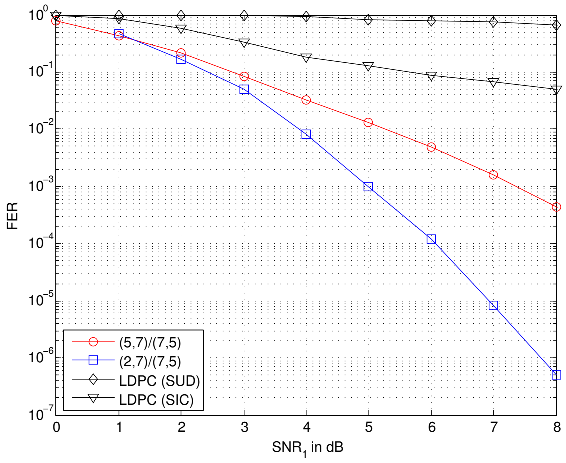


Fig. 2: Simulated overall frame error rates of LDPC codes and trellis-based codes employed for a GIC with strong interference $SNR_1 - SNR_2 = 2$ dB, $INR_1 - SNR_2 = 1$ dB, and $INR_2 - SNR_1 = 2$ dB.

As another example, code optimization is carried out for a GIC with $SNR_1 - SNR_2 = 1$ dB, $INR_1 - SNR_2 = 2$ dB, and $INR_2 - SNR_1 = 1.5$ dB. Unlike the previous example, code design is performed targeting different SNR values, that is, the upper-bound is minimized for codes with 4 states at low and high SNRs separately. For this example, we choose $SNR_1 = 3$ dB and $SNR_1 = 8$ dB for which $(2,7)/(7,5)$ and $(6,7)/(3,5)$ minimize the upper bound (6), respectively. Fig. 3 demonstrates the decoding results for the codes adopted for the considered GIC. The codes optimized at $SNR_1 = 3$ dB have the best performance at low SNRs while $(6,7)/(3,5)$ code pair has the best performance at high SNRs. In addition, both optimized codes considerably outperform the P2P optimal codes at high SNRs. Similar to the previous example, the performance of the LDPC codes computed with SIC is better than that obtained with SUD, however, both are inferior to the performance of trellis-based codes.

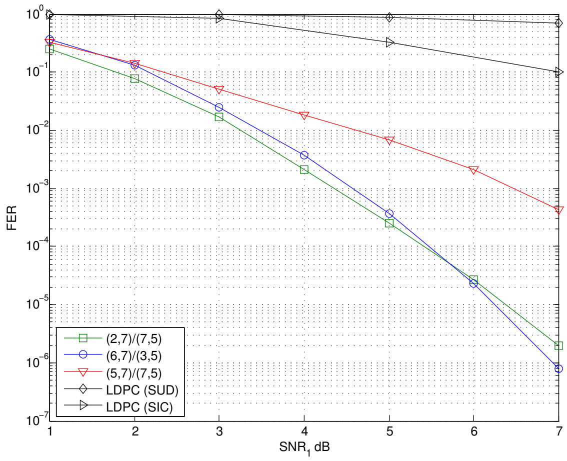


Fig. 3: Simulated overall frame error rates of LDPC codes and trellis-based codes employed for a GIC with strong interference, $SNR_1 - SNR_2 = 1$ dB, $INR_1 - SNR_2 = 2$ dB, and $INR_2 - SNR_1 = 1.5$ dB.

### B. Weak Interference Case

Consider a GIC with $SNR_1 - SNR_2 = 0.5$ dB, $INR_1 - SNR_2 = -1$ dB, and $INR_2 - SNR_1 = -1.5$ dB where the SNR and INR constraints satisfy the weak interference condition. The code design is pursued by minimizing (11) at $SNR = 20$ dB over codes with 4 states. The optimization results in the code pair $(4,5)/(5,7)$. The performance of the optimized codes is compared against that of the P2P optimal codes and the off-the-shelf LDPC codes. Fig. 4 shows the decoding results of the codes employed. For comparison, the performance of the trellis based codes are obtained for JML and SU decoding. It is shown that, under SU decoding where the interfering signal is treated as noise, the performance of the optimized code pair is similar to that of the P2P optimal codes. However, the optimized codes offer better performance than the P2P optimal codes under the JML decoding. Moreover, the performance of the LDPC codes with SU decoding is better than both the optimized codes and the P2P optimal codes, however, they are inferior to the trellis based codes under JML. The poor performance of the considered SU decoding can be attributed to the level of the interference at the receivers which is comparable to the power of the desired signals.

As another example, code optimization is carried out for a GIC with $SNR_1 - SNR_2 = -0.75$ dB, $INR_1 - SNR_2 =$

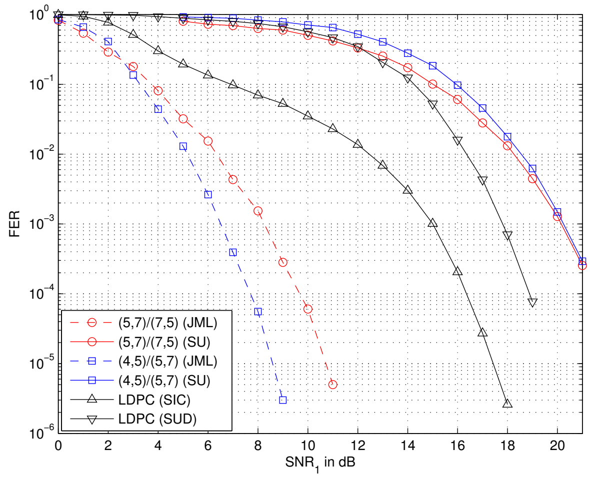


Fig. 4: Simulated overall frame error rates of LDPC codes and trellis-based codes employed for a GIC with weak interference $SNR_1 - SNR_2 = 0.5$ dB, $INR_1 - SNR_2 = -1$ dB, and $INR_2 - SNR_1 = -1.5$ dB.

$-1.5$ dB, and $INR_2 - SNR_1 = -0.5$ dB. For this example, P2P optimal codes achieve the minimum of the expression in (11) considering all the codes with 4 states where the bounds are computed at $SNR = 20$ dB. For comparison, we also consider the codes ranked second in the minimization process which are $(5,7)/(6,7)$. Fig. 5 demonstrates the performance of the employed codes for the considered GIC. Under the SU decoding, the P2P optimal codes and the optimized codes have comparable performance both are outperformed by LDPC codes. For the case of JML, however, LDPC codes are inferior to the trellis based codes and the optimized codes provide better performance than the P2P optimal codes.

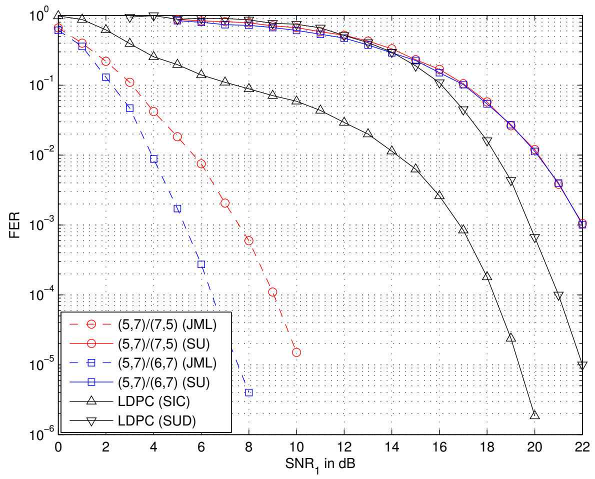


Fig. 5: Simulated overall frame error rates of LDPC codes and trellis-based codes employed for a GIC with weak interference $SNR_1 - SNR_2 = -0.75$ dB, $INR_1 - SNR_2 = -1.5$ dB, and $INR_2 - SNR_1 = -0.5$ dB.

## V. Conclusion

A code design method for the two-user GIC with short block lengths is proposed. The existing performance bounds for the two-user GMAC and the two-user GBC are exploited towards developing performance bounds for this case for different interference levels. These are then utilized in designing trellis-based codes. It is shown that under strong interference, the optimized trellis-based codes offer better performance than the P2P optimal codes, both outperforming the LDPC codes. For the case of weak interference, we notice that the optimized codes and the P2P optimal codes have similar performance, and both are inferior to LDPC codes under SU decoding. However, under the JML decoding the optimized codes outperform both LDPC and P2P optimal codes significantly.

## References


[1] T. Han and K. Kobayashi, “A new achievable rate region for the interference channel,” *IEEE Trans. Inf. Theory,* vol. 27, no. 1, pp. 49–60, Jan. 1981.

[2] G. Kramer, “Review of rate regions for interference channels,” in *Proc. Int. Zurich Seminar,* Zurich, 2006, pp. 162-165.

[3] A. Bennatan, S. Shamai, and A. Calderbank, “Soft-decoding-based strategies for relay and interference channels: analysis and achievable rates using LDPC codes,” *IEEE Trans. Inf. Theory,* vol. 60, no. 4, pp. 1977-2009, Apr. 2014.

[4] S. Sharifi, A. K. Tanc, and T. M. Duman, “On LDPC codes for Gaussian interference channels,” in *Proc. IEEE Int. Symp. Information Theory (ISIT),* Honolulu, HI, 2014, pp. 1992-1996.

[5] S. Sharifi, A. K. Tanc, and T. M. Duman, “Implementing the Han-Kobayashi scheme using low density parity check codes over Gaussian interference channels,” *IEEE Trans. Commun.,* vol. 63, no. 2, pp. 337-350, Feb. 2015.

[6] Y. Mao and A. Banihashemi, “A heuristic search for good low-density parity-check codes at short block lengths,” in *Proc. IEEE Int. Conf. Communications,* Helsinki, 2001, PP. 41–44.

[7] X. Hu, E. Eleftheriou, and D.-M. Arnold, “Progressive edge-growth Tanner graphs,” in *Proc. IEEE GLOBECOM,* San Antonio, TX, vol. 2, 2001, pp. 995-2001.

[8] T. Tian, C.R. Jones, J.D. Villasenor, R.D. Wesel, “Selective avoidance of cycles in irregular LDPC code construction,” *IEEE Trans. Commun.,* vol. 52, no. 8, pp. 1242–1247, Aug. 2004.

[9] A. Ramamoorthy, R. D. Wesel, “Construction of short block length irregular low-density parity-check codes,” in *Proc. IEEE Int. Conf. Communications,* Paris, 2004, pp. 410–414.

[10] A. Ozcelikkale and T. M. Duman, “Short length trellis-based codes for Gaussian multiple-access channels,” *IEEE Signal Process. Lett.,* vol. 21, no. 10, pp. 1177–1181, Oct. 2014.

[11] V. Tralli, A. Conti, and M. Chiani, “Pragmatic space-time trellis codes: GTF-based design for block fading channels,” *IEEE Trans. Signal Process.,* vol. 59, pp. 2809-2823, Jun. 2011.

[12] A. E. Gamal and K. Young-Han, *Network Information Theory.* New York: Cambridge University Press, 2011.

[13] U. Bhat, D. Fertonani, and T. M. Duman, “Approximate performance analysis for linear codes in superposition schemes over Gaussian broadcast channels,” *IEEE Trans. Commun.,* vol. 58, no. 8, pp. 2177–2182, Aug. 2010.

[14] A. Stefanov and T. M. Duman, “Performance bounds for space-time trellis codes,” *IEEE Trans. Inf. Theory,* vol. 49, no. 9, pp. 2134–2140, Sep. 2003.

[15] W. Ryan and S. Lin, *Channel Codes: Classical and Modern.* Cambridge, U.K.: Cambridge University Press, 2009.

[16] D. J. C. MacKay, “Encyclopedia of Sparse Graph Codes,” 1999, [Online]. Available: http://www.inference.phy.cam.ac.uk/mackay/Codes-Files.html

[17] P. Berlin and D. Tuninetti, “LDPC codes for fading Gaussian broadcast channels,” *IEEE Trans. Inf. Theory,* vol. 51, no. 6, pp. 2173–2182, Jun. 2005.