

A RANKING METHOD FOR RATING THE PERFORMANCES OF PERMUTATION CODES

K. Ogunyanda* and T.G. Swart*

* *Department of Electrical and Electronic Engineering Science, University of Johannesburg, Auckland Park, 2006, South Africa. Email: ogunyanda@gmail.com and tgswart@uj.ac.za*

Abstract: Minimum Hamming distance, d_m , has been widely used as the yardstick for the performance of permutation codes (PCs). However, a number of PCs with the same d_m and cardinality can have different performances, even if they have the same distance optimality. Since PC is a robust channel coding scheme in power line communications applications, we present a simple and fast ranking method that predicts the relative performance of PCs, by using the information extracted from their Hamming distance distributions. This tool is useful for selecting an efficient PC codebook out of a number of similar ones.

Key words: Channel coding, Hamming distance distribution, Permutation coding, Power line communications, Ranking method.

1 INTRODUCTION

Power line communications (PLC), which involves information transmission through the existing power network, is now a popular technology that may be used for home internet access, data distribution, networking, smart metering and electric vehicle-to-charging stations. The major setback of this technology is the performance degradation posed by the various noise types present in the communication channel (i.e. power line). These noise types include background noise, impulsive noise and narrowband interference [1, 2]. These noise types are inherent in the channel, because the main purpose of the power line network is not for communication applications. In order to make the channel conducive for communications, it may be helpful to use higher transmission power or distortion-free frequencies in the communication system. There are, however PLC communication standards that enforce constraints on the range of power and frequencies that are useable [3–5]. As such, in order to achieve effective PLC systems, a robust channel coding scheme is a crucial component to consider.

The channel coding scheme of interest in this work is permutation coding (PC). This is motivated based on the foundational work on PCs reported by Vinck, who was the first to suggest its usage for PLC applications [1]. According to Vinck, PCs in combination with MFSK are capable of handling the types of noise associated with narrowband PLC channels. This then gave inspiration for more research works on PCs, some of which are reported in [6–16].

There have been some approaches to the generation of PC codebooks with different minimum Hamming distances and cardinalities. The term d_m refers to the minimum Hamming distance between any two distinct codewords in a codebook, C , while $|C|$ is the cardinality, which is the total number of possible codewords in the codebook. However, how one selects the best codebook out of

a number of competitive PC codebooks with similar characteristics, is a challenging problem that we have attempted to solve in this study. The quality of a communication system can be greatly impaired if a poor code design is used.

In coding, d_m is usually used to determine the strength of a code in handling errors. According to [2], a PC is able to detect t symbol errors provided that:

$$t \leq (d_m - 1). \quad (1)$$

Moreover, in terms of error correcting capability, a PC is able to correct $t/2$ number of errors. However, apart from the d_m in a given codebook, other distances greater than d_m also feature, when the distance relationships between every distinct codeword are computed. All of these distances have a cumulative contribution in the performance of any given codebook. This notion was used by Viterbi in [19], when he demonstrated that various possible distances, resulting from every remerging path in the trellis-code representation of a convolutional code, contribute to its error probability. A similar approach was used in [20–22] to determine the distance spectrum used for comparing the performances of convolutional codes with similar constraint lengths and decoding complexities. We thus employ a similar approach in the ranking method presented in this study. However, instead of using it in the context of convolutional codes, it is used in the context of non-binary PC schemes. Also, Hamming distance distribution is what we have used in this study, instead of the distance spectrum used in convolutional code, which is a binary code. As such, this study is centered on determining the cumulative effects of all the possible distances, including d_m , on the error detecting capability of any given PC.

Other work done in respect of PCs' performance tools include the one reported in [23], where a probabilistic approach was used to determine PC decoders' performances. However, this is not related to the work presented here, in

that our approach is not decoder-dependent, as opposed to the work in [23], which is based on the type of decoding algorithm used in the design.

In addition, instead of using cardinality and Hamming weight distribution, Hamming distance distribution (HDD), which takes into consideration the contribution of every possible distance, is employed in our work. The reason for this is discussed later in this paper. To the best of our knowledge, this is the first time a method such as this is employed for PCs.

It is however worth noting that the ranking method presented here is only evaluated based on combined background noise (modeled as AWGN) and impulsive noise channel conditions. As such, a representation of AWGN+IN channel model is involved in this work. In order to study the performance of data transmission over a PLC channel, the authors in [24] used Middleton's model to model the transmission channel using a binary symmetric channel, whose transmission error probability ϵ is dependent on the impulsive noise variance. In our proposed ranking method, we represent the channel model as a composite channel, comprising of the modulator, demodulator and the PLC channel. This composite setup is appropriate, in order to be able to analyse channel codings [25]. The ϵ of this channel setup is derived from Middleton's noise model for impulsive noise.

Our contribution in this work therefore further promotes the use of PCs in PLC-related applications, by providing a good ranking tool that can be used to rate the performances of PC schemes of similar characteristics, without the need to perform performance simulations. Having to perform performance simulations for a multitude of codebooks, where each simulation could take a considerable time to complete, would mean that the process of choosing the best codebook will be extremely time consuming. In practical terms, this tool can be handy, when the best codebook is to be chosen out of a number of competitive codebooks. In [17] and [18], various PC codebooks of the same $|C|$ and d_m were presented which had the same distance optimality, but their performances are seen to be slightly different when simulated. The method proposed here is therefore able to determine the best PC codebook in such situations. Hence, this work can be viewed as an extension of the work reported in [17] and [18].

We briefly describe what PC entails in Section 2. Section 3 gives a brief background to the HDD of a codebook, which is then used to propose the PC ranking method in Section 4. In Section 5, we present the HDDs of some known PCs and use the proposed method to analyse and predict their performances. Some simulation results are presented in Section 6 to validate the analysis performed in Section 5, after which a concluding note is given in Section 7.

2 INTRODUCTION TO PERMUTATION CODING

As stated in [9], a PC mapping maps binary data sequences to codewords of non-binary permutation sequences. A PC codebook is usually denoted as $C(n, M, \eta)$, where n is the

number of bits being mapped onto M symbols, while η determines the mapping type. Each codeword is of length M , and its symbols are chosen from $\{0, 1, \dots, M-1\}$. An example of a PC, with codeword length $M=4$, where $n=2$ bits are mapped to the PC symbols [9], is:

$$\{00, 01, 10, 11\} \rightarrow \{2130, 2103, 3120, 0213\}. \quad (2)$$

Another example with $M=4$ and $n=4$ is [18]:

$$\left\{ \begin{array}{l} 0000, 0001, 0010, 0011, \\ 0100, 0101, 0110, 0111, \\ 1000, 1001, 1010, 1011, \\ 1100, 1101, 1110, 1111 \end{array} \right\} \rightarrow \left\{ \begin{array}{l} 1230, 1203, 1320, 1302, \\ 2130, 2103, 3120, 3102, \\ 0231, 0213, 0321, 0312, \\ 2031, 2013, 3021, 3012 \end{array} \right\}. \quad (3)$$

If the distances between the binary and permutation sequences are compared, three types of PC mappings can be defined, namely distance-conserving mappings (DCMs), distance-increasing mappings (DIMs) and distance-reducing mappings (DRMs). For instance, the binary sequences in (2) have $d_m = 1$, while their corresponding permutation sequences have $d_m = 2$. With this, $\eta = 1$. Hence, such a PC mapping is a DIM. By definition, a DIM ensures that the distances of the permutation sequences have some increase above those of the binary sequences; a DCM ensures that the distances of the binary sequences are at least maintained in their corresponding permutation sequences; and a DRM ensures a controlled distance loss between distances. These three mapping types are collectively referred to as distance-preserving mappings [17, 26].

A number of PC mapping and decoding algorithms are available in the literature. However, since we are only focusing on analysing PCs' relative performance, the simulations done in this work have considered the simple PC mapping and decoding algorithm presented in [9]. Therein, information bits are grouped into n bits, and these grouped bits are in turn mapped to corresponding codewords of length M , based on the grouped bit sequences. On the decoding side, a codeword having the closest Hamming distance to the received codeword is selected as the decoded codeword.

3 HAMMING DISTANCE DISTRIBUTION

We define HDD as the amount of contribution each possible Hamming distance has in the performance of a given PC codebook. Let Hamming distance k be represented by d_k . Given a codebook C , with M and d_m , all the possible distances d_k in C are given by:

$$k \in \{0, 1, \dots, M\}. \quad (4)$$

If the minimum distance is d_m , (4) therefore reduces to

$$k \in \{d_m, d_m + 1, \dots, d_M\}, \quad (5)$$

If the distance relationship between every distinct

codeword in \mathcal{C} is computed, a $|\mathcal{C}| \times |\mathcal{C}|$ dimensional distance matrix \mathbf{E} is generated. This matrix \mathbf{E} consists of elements $e_{i,j}$ that represent the distance between every pair of distinct codewords x_i and x_j in the codebook, where $i, j = 1, 2, \dots, |\mathcal{C}|$ and is defined as:

$$\mathbf{E} = \begin{matrix} & x_1 & x_2 & \cdot & \cdot & \cdot & x_{|\mathcal{C}|} \\ \begin{matrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_{|\mathcal{C}|} \end{matrix} & \begin{pmatrix} e_{1,1} & e_{1,2} & \cdot & \cdot & \cdot & e_{1,|\mathcal{C}|} \\ e_{2,1} & e_{2,2} & \cdot & \cdot & \cdot & e_{2,|\mathcal{C}|} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ e_{|\mathcal{C}|,1} & e_{|\mathcal{C}|,2} & \cdot & \cdot & \cdot & e_{|\mathcal{C}|,|\mathcal{C}|} \end{pmatrix} \end{matrix}, \quad (6)$$

where $e_{i,j} = 0$, for $i = j$.

The total number of occurrences of each d_k can then be determined from the elements in matrix \mathbf{E} . For instance, the distance matrix for the codebook in (2) is given by:

$$\mathbf{E} = \begin{matrix} & 2130 & 2103 & 3120 & 0213 \\ \begin{matrix} 2130 \\ 2103 \\ 3120 \\ 0213 \end{matrix} & \begin{pmatrix} 0 & 2 & 2 & 4 \\ 2 & 0 & 3 & 3 \\ 2 & 3 & 0 & 4 \\ 4 & 3 & 4 & 0 \end{pmatrix} \end{matrix}. \quad (7)$$

Let δ_k denote the number of times that d_k appears in the distance matrix. We can then express the HDD, Δ , of all the possible d_k as

$$\Delta = \{\delta_1, \dots, \delta_{m-1}, \delta_m, \delta_{m+1}, \dots, \delta_{M-1}, \delta_M\}, \quad (8)$$

with $\delta_1, \dots, \delta_{m-1} = 0$.

Using a probabilistic approach, the contribution of each d_k , given its δ_k , is given by:

$$P_k = \frac{\delta_k}{\sum_{k=m}^M \delta_k}, \quad m \leq k \leq M. \quad (9)$$

To illustrate this, the HDD and distance probabilities of the example codebook given in (3) are respectively given by:

$$\Delta = \{0, 64, 64, 112\}, \quad (10)$$

and

$$P_2 = P_3 = 0.2667 \text{ and } P_4 = 0.4667. \quad (11)$$

According to (11), it is clear that the codebook's performance mainly depends on d_4 , since it has the greatest probability. The significance of the δ_k values (i.e., the HDD) in determining the strength of a PC can be understood by considering Viterbi's findings in [19]. In the trellis diagram of a convolutional code, any arbitrary path leading to a given node is associated with a distance,

which is computed by determining the bitwise differences between the code sequences along such a path and the expected code sequence. In order to detect the optimum path, all the distance information, which are dependent on the channel transition probability, need to be computed [19].

Similar to convolutional codes where numerous different codes, with varying performance, can be obtained for the same parameters, various mappings from binary sequences to permutation sequences can be obtained, all having the same d_m and satisfying the distance preserving property. Thus a method was needed to determine which of these mappings would perform the best. The work in [17] and [18] addressed this by introducing the concept of distance optimality derived from distance matrix \mathbf{E} like the one defined in (6). By denoting the sum of all distances in matrix \mathbf{E} by $|\mathbf{E}|$, a PC is considered distance optimal if $|\mathbf{E}|$ is maximised. It was shown that an upper bound on $|\mathbf{E}|$ exists, and that any PC codebook attaining the bound would perform better than those that do not.

This clearly showed that all the distance contributions affect the performance of a given PC codebook. However, this approach is unable to distinguish between some codebooks with the same $|\mathcal{C}|$, d_m and maximised $|\mathbf{E}|$. Hence, we investigate the proposed PC ranking method that goes further in analysing the HDD, to enable us to distinguish between such codebooks.

In HDD, the diagonal elements of \mathbf{E} is not considered, because this would amount to comparing a codeword with itself. Hamming weight distribution on the other hand, is the number of codewords with Hamming distance k from an all 0's reference codeword. This however holds for binary codes. For non-binary codes such as PCs, a reference origin codeword, which may or may not be in \mathcal{C} , is compared with each codeword, in order to determine the weight distribution [23]. One could state that HDD and weight distribution are the same, if the comparison is done with reference to an origin codeword. However, this is only valid for a linear code. Since PC is a non-linear code, to get accurate results every distinct codeword must be compared with all other codewords.

4 RANKING METHOD

In the context of this work, the term *ranking* refers to the order of performances of a given set of codewords, when being compared. The ranking method proposed in this study entails using the HDD discussed above to determine the probability of undetected symbol errors of a given codebook \mathcal{C} . Since the PCs considered in this study are non-binary codes, this necessitates the use of non-binary modulation, such as phase shift keying (PSK) modulation. In order to evaluate the error probability of a PC system using such modulations, we employ the composite channel setup in Fig. 1 as the transmission channel.

The PLC channel block is a combination of impulsive noise and additive white Gaussian noise (AWGN). Since the

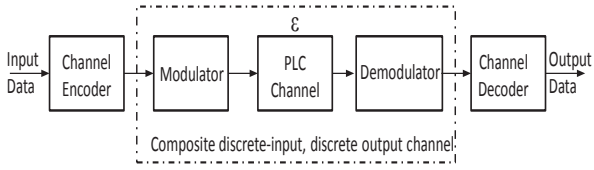


Figure 1: Channel model for ranking method.

modulator employs non-binary waveforms and the detector makes hard decisions, we can say that the composite channel has a discrete-time non-binary input sequence $X = \{0, 1, \dots, M_{\text{DP}} - 1\}$, where M_{DP} is a modulation order whose value is a power of 2, and a discrete-time non-binary output sequence. Each input symbol has a probability $1/M_{\text{DP}}$ of being received at the output. The probability of receiving an incorrect symbol is $\xi = \epsilon/(M_{\text{DP}} - 1)$, while $(1 - \epsilon)$ is the probability of correctly receiving a symbol at the detector's output, where ϵ is the transmission error probability.

If φ and α respectively denote a modulated and a demodulated symbol from the system, where $\varphi \in (0, 1, \dots, M - 1)$ and $\alpha \in (0, 1, \dots, M_{\text{DP}} - 1)$, the following expression therefore holds, based on the above composite channel setup:

$$P(Y = \alpha | X = \varphi) = \begin{cases} 1 - \epsilon, & \text{if } \varphi = \alpha, \\ \xi, & \text{if } \varphi \neq \alpha. \end{cases} \quad (12)$$

Also, as stated in (1), a PC has a symbol error detecting capability of $(d_m - 1)$, which can be generalised for each d_k as

$$t_k = (d_k - 1) = (k - 1). \quad (13)$$

Since each d_k has its number of appearances δ_k , the probability of a PC detecting t_k number of errors is dependent on the value of δ_k . The contribution of d_k in the probability P_u of undetected errors is based on the cumulative error probabilities of other distances $> d_k$ to yield a correct symbol and on the probabilities of d_k and those of other distances $< d_k$ to yield an incorrect symbol (as defined in (12)). This is mathematically expressed as:

$$P_{|d_k} = (\xi)_{|d_M} \times (\xi)_{|d_{M-1}} \times \dots \times (\xi)_{|d_{k+1}} \times \delta_k (1 - \epsilon)_{|d_k} \times (1 - \epsilon)_{|d_{k-1}} \times \dots \times (1 - \epsilon)_{|d_m}, \quad (14)$$

where $(\cdot)_{|d_k}$ is the computation of (\cdot) due to distance d_k .

By computing all the $P_{|d_k}$ values for every possible d_k defined in (5), the expression in (14) reduces to a generalised expression for the probability of undetected errors P_u in a PC system as:

$$P_u = \sum_{k=m}^M \delta_k \xi^k (1 - \epsilon)^{M-k}. \quad (15)$$

The expression in (15) can work for every type of channel, provided that ϵ , associated with such a channel, can be computed. The computation of ϵ is what gives the characteristics of the communication channel. If ϵ for a PLC channel can be obtained, it can be substituted into (15), thereby providing a means of analysing the performance of any PC codebook $\mathcal{C}(n, M, \eta)$ over a PLC channel, provided its HDD is known. This shall therefore lead us to the computation of ϵ for a PLC channel impaired with impulsive noise, which is one of the most notorious noise types in PLC.

Impulsive noise has a broadband power spectral density (PSD), which sometimes affects frequency components of the transmitted data for a particular length of time. According to the approximated form of Middleton's class A noise model, it has a strength Γ , which is the relationship between its variance σ_I^2 and that of additive white Gaussian noise (AWGN) σ_g^2 , given by $\Gamma = \sigma_g^2 / \sigma_I^2$ [6, 27]. The smaller Γ is, the more severe the impulsive noise effect becomes. If A is the probability of impulsive noise affecting a symbol, the general form of error probability for an MPSK system is modified to account for the combination of AWGN and impulsive noise effects as follows:

$$\epsilon_{\text{MPSK}} = \frac{(1-A)}{\beta} Q \left(\sqrt{\frac{\beta E_b}{\sigma_g^2}} \sin(\pi/M_{\text{DP}}) \right) + \frac{A}{\beta} Q \left(\sqrt{\frac{\beta E_b}{\sigma_g^2 + \sigma_g^2/(A\Gamma)}} \sin(\pi/M_{\text{DP}}) \right), \quad (16)$$

where $\beta = \gamma(\log_2(M))$ and $Q(\cdot)$ is a Q function.

For a PLC system impaired with impulsive and background noise, ϵ in (15) is substituted with ϵ_{MPSK} . The value of γ , given by $\gamma = n/M$ is for coding rate compensation.

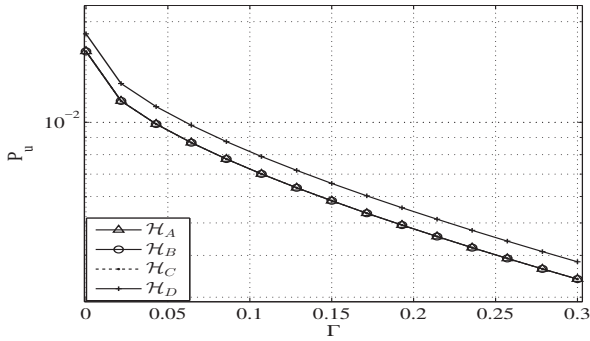
5 HDDs OF SOME KNOWN PCs AND THEIR PERFORMANCE RANKINGS

In order to validate the authenticity of the proposed ranking method, we adopt a number of codebooks from the available literature on PCs. As a form of evaluation, the expression in (16) was used to generate transmission errors for the expression in (15), which in turn, was used to simulate performance ranking curves for each codebook evaluated. An impulsive noise probability $A = 0.01$ was used, with varying strength Γ from 0 to 0.3.

According to [18], (3) and two other codebooks were proven to be distance optimal. We therefore consider

Table 1: Comparison of $M = 4$ PC mappings

Code	δ_2	δ_3	δ_4	Source
\mathcal{H}_A	64	64	112	(3), [18, (9)]
\mathcal{H}_B	64	64	112	[18, (8)]
\mathcal{H}_C	64	64	112	[18, (10)]
\mathcal{H}_D	72	84	84	[18, (11)]

Figure 2: Performance ranking curves for Schemes \mathcal{H}_A to \mathcal{H}_D .

another non-optimal PC of $M = 4$ and $|C| = 16$ (i.e., (8) from [18]), in comparison with the optimal ones. Table 1 gives their HDDs, while their performance ranking curves, simulated from (15) and (16), are as shown in Figure 2.

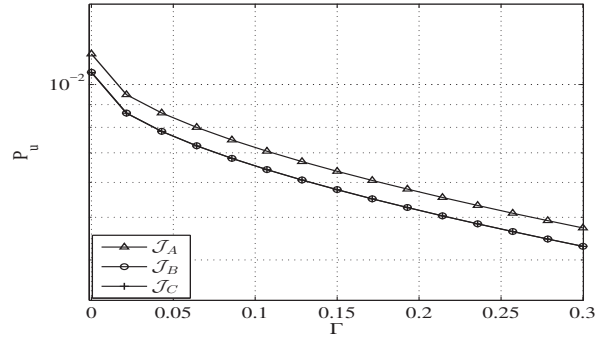
According to Figure 2, code \mathcal{H}_D , has the worst performance. Of course, \mathcal{H}_A , \mathcal{H}_B and \mathcal{H}_C are expected to have overlapping performances, as their ranking curves clearly show, although their codewords are different. This agrees with the findings in [18]. This shall also be validated in our simulation section.

We proceed to consider PCs of $M = 5$. Here we adopt the codebook in (12) of [18] and two other similar codebooks $M(5,5,0)$ and $M(5,5,0)^*$ from [17, Appendix B.1] and denote these three codebooks as \mathcal{J}_A , \mathcal{J}_B and \mathcal{J}_C . These codebooks all have $M = 5$ and $|C| = 32$. According to the distance optimality approach in [17] and [18], none of these codebooks are optimal, but \mathcal{J}_C is near optimal. However, what makes \mathcal{J}_B and \mathcal{J}_C different from \mathcal{J}_A is the fact that they have some repeating permutation symbols in some of their codewords. The HDDs of these codebooks are as shown in Table 2, with their ranking curves in Figure 3.

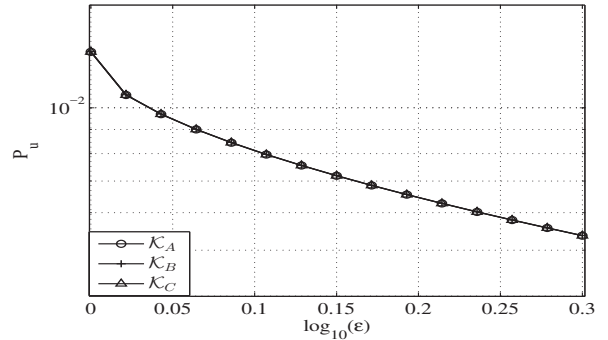
The curves in Figure 3 show that \mathcal{J}_A should have the worst

Table 2: Comparison of $M = 5$ PC mappings

Code	δ_2	δ_3	δ_4	δ_5	Source
\mathcal{J}_A	176	224	368	224	[18, (12)]
\mathcal{J}_B	160	192	384	256	[17, $M(5,5,0)$]
\mathcal{J}_C	160	176	352	304	[17, $M(5,5,0)^*$]

Figure 3: Performance ranking curves for Schemes \mathcal{J}_A to \mathcal{J}_C .Table 3: Comparison of $M = 6$ PC mappings

Code	δ_2	δ_3	δ_4	δ_5	δ_6
\mathcal{K}_A	320	256	896	896	1664
\mathcal{K}_B	320	320	736	1024	1632
\mathcal{K}_C	320	256	960	768	1728

Figure 4: Performance ranking curves for Schemes \mathcal{K}_A to \mathcal{K}_C .

performance compared to the others, which agrees with the claims in [17] and [18]. Although the findings in [17] show that \mathcal{J}_C is closer to optimal than \mathcal{J}_B , this analysis however shows that \mathcal{J}_B and \mathcal{J}_C should have overlapping behaviours, which shall later be validated in the simulation section.

Next, we consider the three near-optimal codebooks, with $M = 6$, $d_m = 2$ and $|C| = 64$, presented in [17]. As observed in [17], these codebooks have the same level of optimality. Their HDDs are presented in Table 3, as schemes \mathcal{K}_A , \mathcal{K}_B and \mathcal{K}_C , while Figure 4 presents their ranking curves.

Based on the curves in Figure 4, all the three codebooks \mathcal{K}_A , \mathcal{K}_B and \mathcal{K}_C should have relatively overlapping performances. This is because their P_u values in Figure 4 are the same. This finding actually corroborates the notion of the optimum distance approach presented in [17].

Another set of codebooks to be considered are the four optimal codebooks, with $M = 8$, $d_m = 2$ and $|C| = 256$

Table 4: Comparison of $M = 8$ PC mappings

Code	δ_2	δ_3	δ_4	δ_5	δ_6	δ_7	δ_8
\mathcal{L}_A	1536	1024	3584	4096	7680	7168	40192
\mathcal{L}_B	1024	0	5632	0	17408	0	41216
\mathcal{L}_C	1024	512	3072	3072	10240	12800	34560
\mathcal{L}_D	1536	1024	4096	4096	6656	7168	40704

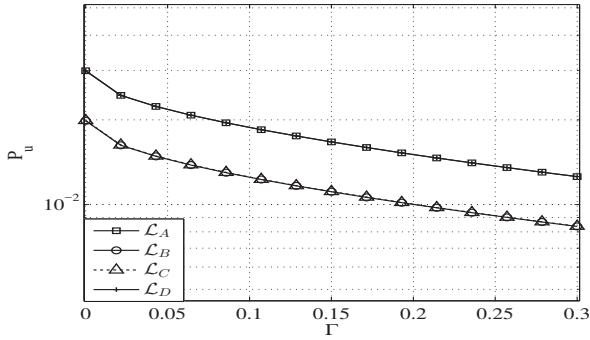
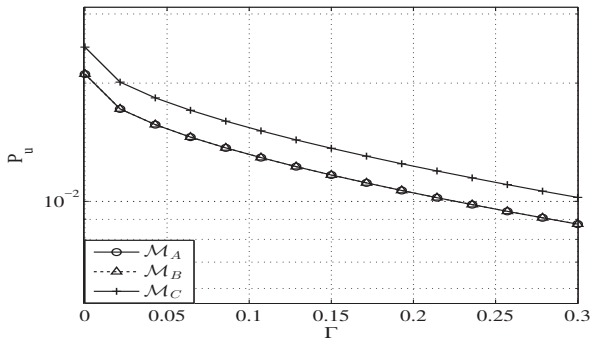
Figure 5: Performance ranking curves for Schemes \mathcal{L}_A to \mathcal{L}_D .

Table 5: HDDs for other mappings from [17]

Code	δ_2	δ_3	δ_4	δ_5	δ_6	δ_7
\mathcal{M}_A	768	640	1920	1536	4352	7040
\mathcal{M}_B	768	640	2432	3072	5888	3456
\mathcal{M}_C	896	1280	3456	4864	4224	1536

Figure 6: Performance ranking curves for Schemes \mathcal{M}_A to \mathcal{M}_C .

in [17]. Their HDDs and ranking curves are presented in Table 4 and Figure 5, respectively.

According to the ranking curves of these codebooks, we see that \mathcal{L}_A and \mathcal{L}_D have overlapping curves, hence their real performances should follow a similar trend. Also, for \mathcal{L}_B and \mathcal{L}_C , the same applies because of their overlapping performance ranking curves. Despite the fact that the four codebooks are declared optimal, our ranking method is able to determine the best from these sets of codebooks of the same class.

Further analyses carried out on some other mappings adopted from [17], are presented in Table 5 and Figure 6, based on the proposed ranking method.

Scheme \mathcal{M}_A is the $M(7,7,0)$ DPM with $M = 7$, $d_m = 2$ and $|C| = 128$ from [17, Appendix B.1]. Schemes \mathcal{M}_B and \mathcal{M}_C from [17, Appendix C.1] are also of the same class with this $M(7,7,0)$ mapping. According to [17], \mathcal{M}_A is more optimal than the rest of these mappings, while \mathcal{M}_C is the least optimal. However, according to their ranking curves presented in Figure 6, \mathcal{M}_A and \mathcal{M}_B should have overlapping performances in reality, because their curves overlap, while \mathcal{M}_C should perform worse than the rest.

6 SIMULATION RESULTS AND DISCUSSION

With a view to validating the proposed ranking method, all the schemes whose HDDs and ranking curves have been presented in Tables 1 to 5 and Figures 2 to 6, were simulated under combined AWGN and impulsive noise associated with PLC channels. The input data is composed of 9600 random bits. In order to model the effect of impulsive noise on the transmitted information, we assumed a Gilbert-Elliot model, where probabilities of moving from the bad state to good state and from the good state to bad state are defined. The probability that a transmitted symbol is hit by impulsive noise, in a bad state was assumed to be $IN = 1/32$, while a good state was assumed to be free of impulsive noise. Details about this model is available in [6]. The E_b/N_o considered in the channel ranges from 0 to 25 dB.

The results shown in Figures 7 to 10 are for all the codebooks considered in our evaluations. As displayed in these figures, all the simulation results agree with the predicted performance rankings presented in Section 5. As predicted in Figure 2, Scheme \mathcal{H}_D has the worst curve, which is the reason why it has the worst performance according to Figure 7. Also, the performance curves of Schemes \mathcal{H}_A , \mathcal{H}_B and \mathcal{H}_C are seen to overlap, which is in line with the prediction of the proposed ranking method. The fact that these four schemes have the same d_m does not mean they have the same performance. With the use of the proposed ranking method, we are able to know that Scheme \mathcal{H}_D is not to be used in a transceiver design when codes such as \mathcal{H}_A , \mathcal{H}_B and \mathcal{H}_C are available.

The behaviours of Schemes \mathcal{J}_A to \mathcal{J}_C are also in the order of their ranking curves, with \mathcal{J}_C and \mathcal{J}_B being the best, while \mathcal{J}_A is the worst, as the results in Figure 8 show. More so, as shown in Figure 9 the similarities in the performances of Schemes \mathcal{K}_A to \mathcal{K}_C also obey the predicted rankings presented in Figure 4.

In line with the predicted rankings in Figure 5, Schemes \mathcal{L}_A and \mathcal{L}_D are seen to have closely overlapping curves, especially at $E_b/N_o > 15$ dB, as shown in Figure 10. Schemes \mathcal{L}_B and \mathcal{L}_C also have similar patterns of behaviour, as predicted by the ranking method.

The results of the simulations done on Schemes \mathcal{M}_A , \mathcal{M}_B and \mathcal{M}_C are presented in Figure 11. According to this

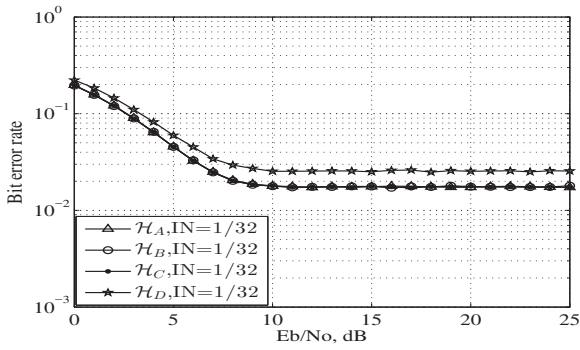


Figure 7: Bit error rate curves for Schemes \mathcal{H}_A to \mathcal{H}_D , under AWGN+impulsive noise channel.

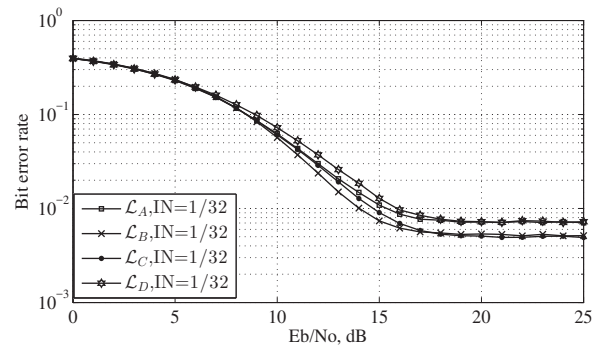


Figure 10: Bit error rate curves for Schemes \mathcal{L}_A to \mathcal{L}_D , under AWGN+impulsive noise channel.

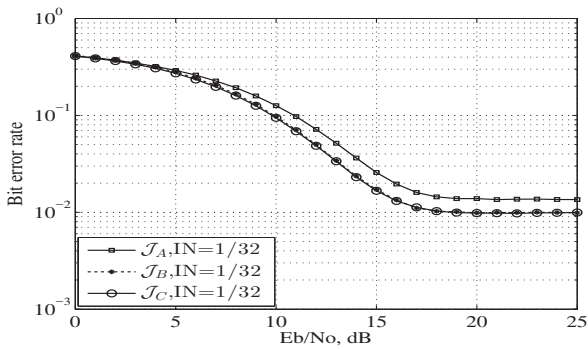


Figure 8: Bit error rate curves for Schemes \mathcal{J}_A to \mathcal{J}_C , under AWGN+impulsive noise channel.

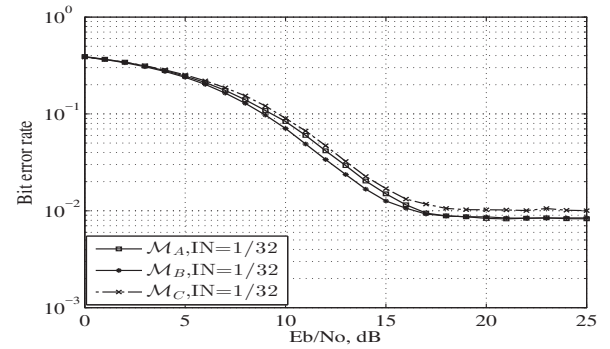


Figure 11: Bit error rate curves for Schemes \mathcal{M}_A to \mathcal{M}_C , under AWGN+impulsive noise channel.

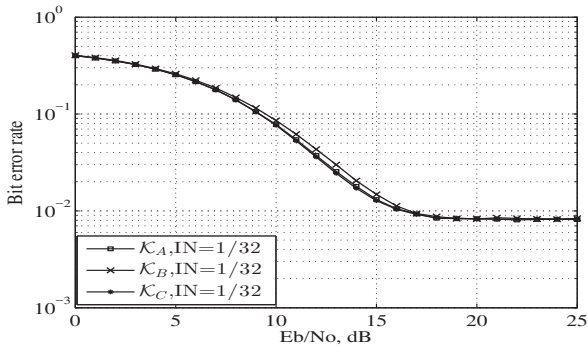


Figure 9: Bit error rate curves for Schemes \mathcal{K}_A to \mathcal{K}_C , under AWGN+impulsive noise channel.

figure, \mathcal{M}_A is seen to overlap with \mathcal{M}_B , but at high E_b/N_o values, while \mathcal{M}_C is seen to be the worst performing scheme. This is in agreement with the predicted rankings presented in Figure 6, based on the proposed ranking method.

In the above simulation results, it is worth noting that the probability IN used in the impulsive noise model is considered very high. That is why the performances of all the codebooks presented are relatively poor. With the aid of interleaving and concatenated outer codes, such as Reed-Solomon codes, the performances will definitely improve. Although the codebooks evaluated have a $d_m =$

2, it does not mean that the proposed ranking method is limited to such cases. The method can be applied to any codebook category. Also, the significance of the different behaviours of all the schemes considered, are best observed at high E_b/N_o values.

7 CONCLUSION

Permutation coding has emerged as a promising channel coding scheme in PLC related applications. With a view to contributing to this field, we have presented a fast and simple way of ranking the relative performance of PC codebooks with similar properties. This method computes the Hamming distance distribution of the PC codebook and uses the information therein to compute the probability P_u of undetected error, which can, in turn, be used to compare the codebook to other codebooks with similar properties. A codebook with lower P_u values has better performance than any codebook with higher P_u values.

The method was validated using various known PC codebooks adopted from literature. The results of the simulations carried out agree with our claims, when the proposed method was used as a performance ranking tool. This tool is useful, when an efficient codebook is to be selected out of a number of similar codebooks. More so, it can be incorporated when exhaustive search for codebooks is done. It should however be noted that the computation of P_u considers only AWGN and impulsive noise effects.

An extended work in this area of study can be centered on including the effects of other types of PLC noise, such as narrowband noise in the computation of probability P_u . This, however remains a challenge as there is no link found between its PSD and that of AWGN.

ACKNOWLEDGEMENT

This work is based on research supported in part by the National Research Foundation of South Africa (UID 77596).

REFERENCES

- [1] A.J.H. Vinck and J. Häring, "Coding and modulation for power-line communications," in *Proceedings of the 2000 IEEE International Symposium on Power Line Communications and its Applications*, Limerick, Apr. 2000, pp. 265–71.
- [2] H.C. Ferreira, H.M. Grové, O. Hooijen and A.J.H. Vinck, "Power line communication," *Wiley Encyclopedia of Electrical and Electronics Engineering*, 2001.
- [3] eRDF, *PLC G3 Physical Layer Specification*, 2013.
- [4] CENELEC, *50065 part 1: Signalling on low voltage electrical installations in the frequency range 3 kHz to 148.5 kHz, general requirements, frequency bands and electromagnetic disturbances*, 1992.
- [5] PRIME, *PRIME Technology, whitepaper: PHY, MAC and Convergence layers*, 21st Ed., 2008.
- [6] V.N. Papilaya, T. Shongwe, A.J.H. Vinck and H.C. Ferreira, "Selected subcarriers QPSK-OFDM transmission schemes to combat frequency disturbances," in *Proceedings of the 2012 IEEE International Symposium on Power Line Communications and its Applications*, Beijing, Mar. 2012, pp. 200–5.
- [7] H.C. Ferreira, A.J.H. Vinck, T.G. Swart and I. de Beer, "Permutation trellis codes," *IEEE Transactions on Communications*, Vol. 53, pp. 1782–9, Nov. 2005.
- [8] T.G. Swart and H.C. Ferreira, "Decoding distance-preserving permutation codes for power-line communications," in *Proceedings of IEEE AFRICON*, Windhoek, Sept. 2007, pp. 1–7.
- [9] K. Ogunyanda, A.D. Familua, T.G. Swart, H.C. Ferreira and L. Cheng, "Evaluation and implementation of cyclic permutation coding for power line communications," in *Proceedings of the 2014 IEEE International Conference on Adaptive Science & Technology (ICAST)*, Ota, Oct. 2014, pp. 1–7.
- [10] P.J. Dukes, "Coding with injections," *Designs, Codes and Cryptography*, Vol. 65, pp. 213–22, Dec. 2012.
- [11] F.H. Hunt, S. Perkins and D.H. Smith, "Decoding mixed errors and erasures in permutation codes," *Designs, Codes and Cryptography*, Vol. 74, pp. 481–93, Feb. 2015.
- [12] S. Huczynska and G.L. Mullen, "Frequency permutation arrays," *Journal of Combinatorial Designs*, Vol. 14, pp. 463–78, Jan. 2006.
- [13] R.F. Bailey, "Error-correcting codes from permutation groups," *Discrete Mathematics*, Vol. 309, pp. 4253–65, 2009.
- [14] K. Ogunyanda, A.D. Familua, T.G. Swart, H.C. Ferreira and L. Cheng, "Permutation coding with differential quinary phase shift keying for power line communication," in *Proceedings of the 2014 IEEE PES Innovative Smart Grid Technologies European Conference*, Istanbul, Oct. 2014, pp. 1–6.
- [15] H.C. Ferreira and A.J.H. Vinck, "Interference cancellation with permutation trellis codes," in *Proceedings of the 2000 IEEE Vehicular Technology Conference*, Boston, MA, Sep. 2000, pp. 2401-7.
- [16] Y.M. Chee, H.M. Kia, P. Purkayastha and C. Wang, "Importance of symbol equity in coded modulation for power line communications," in *Proceedings of the 2012 IEEE International Symposium on Information Theory*, Cambridge, Jul. 2012, pp. 661–5.
- [17] T.G. Swart, "Distance-preserving mappings and trellis codes with permutation sequences," Ph.D. dissertation, University of Johannesburg, 2006.
- [18] T.G. Swart, I. de Beer and H.C. Ferreira, "On the distance optimality of permutation mappings," in *Proceedings of the 2005 IEEE International Symposium on Information Theory*, Adelaide, Sept. 2005, pp. 1068–72.
- [19] A.J. Viterbi, "Convolutional codes and their performance in communication systems," *IEEE Transactions on Communications Technology*, Vol. 19, pp. 751–72, Oct. 1971.
- [20] G.A. Alexandre, M. Guido and B. Sergio, "A new approach to the construction of high-rate convolutional codes," *IEEE Communications Letters*, Vol. 5, pp. 453–5, Nov. 2001.
- [21] B.F. Ucha-Filho, R.D. Souza, C. Pimentel and M. Jar, "Further results on convolutional codes based on a minimal trellis complexity measure," in *Proceedings of the 2006 IEEE International Telecommunication Symposium*, Fortaleza, Sept. 2006, pp. 123–8.
- [22] Y. Bian, A. Popplewell and J.J. O'Reilly, "New very high rate punctured convolutional codes," *Electronics Letters*, Vol. 30, pp. 1119–20, Jul. 1994.

- [23] T. Wadayama and M. Hagiwara, "LP-decodable permutation codes based on linearly constrained permutation matrices," *IEEE Transactions on Information Theory*, Vol. 58, pp. 5454–70, Aug. 2012.
- [24] V.B. Balakirsky and A.J.H. Vinck, "Potential limits on power-line communication over impulsive noise channels," in *Proceedings of the 2003 IEEE International Symposium on Power Line Communications and its Applications*, Kyoto, Mar. 2003, pp. 32–7.
- [25] J.G. Proakis, *Digital Communications*, McGraw-Hill, 4th edition, Chapter 7, pp. 364–412, 2001.
- [26] T.G. Swart and H.C. Ferreira, "A generalised upper bound and a multilevel construction for distance-preserving mappings," *IEEE Transactions on Information Theory*, Vol. 52, pp. 3685–95, Aug. 2006.
- [27] T. Shongwe, A.J.H. Vinck and H.C. Ferreira, "On impulse noise and its models," *Proceedings of the 2014 IEEE International Symposium on Power Line Communications and its Applications*, Glasgow, Mar. 2014, pp. 12–7.

APPENDIX

Codes for Schemes \mathcal{H}_A to \mathcal{H}_D :

$$\mathcal{H}_A = \left\{ \begin{array}{l} 1230, 1203, 1320, 1302, 2130, 2103, 3120, 3102 \\ 0231, 0213, 0321, 0312, 2031, 2013, 3021, 3012 \end{array} \right\}$$

$$\mathcal{H}_B = \left\{ \begin{array}{l} 1230, 1203, 1320, 1302, 0231, 0213, 0321, 0312 \\ 2130, 2103, 3120, 3102, 2031, 2013, 3021, 3012 \end{array} \right\}$$

$$\mathcal{H}_C = \left\{ \begin{array}{l} 1230, 1203, 2130, 2103, 1320, 1302, 3120, 3102 \\ 0231, 0213, 2031, 2013, 0321, 0312, 3021, 3012 \end{array} \right\}$$

$$\mathcal{H}_D = \left\{ \begin{array}{l} 1230, 1203, 1320, 1302, 1023, 1032, 2130, 2103 \\ 3210, 3201, 2310, 2301, 3021, 3012, 3120, 3102 \end{array} \right\}$$

Codes for Schemes \mathcal{J}_A to \mathcal{J}_C :

$$\mathcal{J}_A = \left\{ \begin{array}{l} 01234, 01243, 01324, 01342, 01423, 01432 \\ 02134, 02143, 03214, 03241, 02314, 02341 \\ 03421, 03412, 03124, 03142, 41230, 41203 \\ 41320, 41302, 41023, 41032, 42130, 42103 \\ 43210, 43201, 42310, 42301, 43021, 43012 \\ 43120, 43102 \end{array} \right\}$$

$$\mathcal{J}_B = \left\{ \begin{array}{l} 12340, 21340, 12430, 21430, 32140, 31240 \\ 42130, 41230, 14320, 24310, 13420, 23410 \\ 34120, 34210, 43120, 43210, 02341, 01342 \\ 02431, 01432, 02143, 01243, 02134, 01234 \\ 04321, 04312, 03421, 03412, 04123, 04213 \\ 03124, 03214 \end{array} \right\}$$

$$\mathcal{J}_C = \left\{ \begin{array}{l} 04321, 03421, 04312, 03412, 40321, 30421 \\ 40312, 30412, 02341, 02431, 01342, 01432 \\ 42301, 32401, 41302, 31402, 04123, 03124 \\ 04213, 03214, 40123, 30124, 40213, 30214 \\ 02143, 02134, 01243, 01234, 42103, 32104 \\ 41203, 31204 \end{array} \right\}$$

Codes for Schemes \mathcal{K}_A to \mathcal{K}_C :

$$\mathcal{K}_A = \left\{ \begin{array}{l} 123450, 213450, 124350, 214350, 123405 \\ 213405, 124305, 214305, 321450, 312450 \\ 421350, 412350, 321405, 312405, 421305 \\ 412305, 143250, 243150, 134250, 234150 \\ 143205, 243105, 134205, 234105, 341250 \\ 342150, 431250, 432150, 341205, 342105 \\ 431205, 432105, 503412, 503421, 504312 \\ 504321, 053412, 053421, 054312, 054321 \\ 501432, 502431, 501342, 502341, 051432 \\ 052431, 051342, 052341, 503214, 503124 \\ 504213, 504123, 053214, 053124, 054213 \\ 054123, 501234, 502134, 501243, 502143 \\ 051234, 052134, 051243, 052143 \end{array} \right\}$$

$$\mathcal{K}_B = \left\{ \begin{array}{l} 123450, 213450, 124305, 214305, 321450 \\ 312450, 421305, 412305, 143250, 243150 \\ 134205, 234105, 341250, 342150, 431205 \\ 432105, 523410, 513420, 024315, 014325 \\ 521430, 512430, 021345, 012345, 543210 \\ 543120, 034215, 034125, 541230, 542130 \\ 031245, 032145, 103452, 203451, 154302 \\ 254301, 301452, 302451, 451302, 452301 \\ 103254, 203154, 154203, 254103, 301254 \\ 302154, 451203, 452103, 503412, 503421 \\ 054312, 054321, 501432, 502431, 051342 \\ 052341, 503214, 503124, 054213, 054123 \\ 501234, 502134, 051243, 052143 \end{array} \right\}$$

$$\mathcal{K}_C = \left\{ \begin{array}{l} 123450, 213450, 124350, 214350, 123405 \\ 213405, 124305, 214305, 341250, 342150 \\ 431250, 432150, 341205, 342105, 431205 \\ 432105, 523410, 513420, 524310, 514320 \\ 023415, 013425, 024315, 014325, 541230 \\ 542130, 531240, 532140, 041235, 042135 \\ 031245, 032145, 103452, 203451, 104352 \\ 204351, 153402, 253401, 154302, 254301 \\ 301254, 302154, 401253, 402153, 351204 \\ 352104, 451203, 452103, 503412, 503421 \\ 504312, 504321, 053412, 053421, 054312 \\ 054321, 501234, 502134, 501243, 502143 \\ 051234, 052134, 051243, 052143 \end{array} \right\}$$

Codes for Schemes \mathcal{L}_A to \mathcal{L}_D :

$$\mathcal{L}_A = \left\{ \begin{array}{l} 12345670, 12345607, 12346570, 12346507, 12435670, 12435607, 12436570, 12436507, 21345670 \\ 21345607, 21346570, 21346507, 21435670, 21435607, 21436570, 21436507, 12345076, 12345706 \\ 12346075, 12346705, 12435076, 12435706, 12436075, 12436705, 21345076, 21345706, 21346075 \\ 21346705, 21435076, 21435706, 21436075, 21436705, 14325670, 14325607, 14326570, 14326507 \\ 13425670, 13425607, 13426570, 13426507, 24315670, 24315607, 24316570, 24316507, 23415670 \\ 23415607, 23416570, 23416507, 14325076, 14325706, 14326075, 14326705, 13425076, 13425706 \\ 13426075, 13426705, 24315076, 24315706, 24316075, 24316705, 23415076, 23415706, 23416075 \\ 23416705, 32147650, 32140657, 32147560, 32140567, 42137650, 42130657, 42137560, 42130567 \\ 31247650, 31240657, 31247560, 31240567, 41237650, 41230657, 41237560, 41230567, 32147056 \\ 32140756, 32147065, 32140765, 42137056, 42130756, 42137065, 42130765, 31247056, 31240756 \\ 31247065, 31240765, 41237056, 41230756, 41237065, 41230765, 34127650, 34120657, 34127560 \\ 34120567, 43127650, 43120657, 43127560, 43120567, 34217650, 34210657, 34217560, 34210567 \\ 43217650, 43210657, 43217560, 43210567, 34127056, 34120756, 34127065, 34120765, 43127056 \\ 43120756, 43127065, 43120765, 34217056, 34210756, 34217065, 34210765, 43217056, 43210756 \\ 43217065, 43210765, 56701234, 56701234, 65701234, 65701234, 56701243, 56701243, 65701243 \\ 65701243, 56702134, 56702134, 65702134, 65702134, 56702143, 56702143, 65702143, 65702143 \\ 50761234, 57061234, 60751234, 67051234, 50761243, 57061243, 60751243, 67051243, 50762134 \\ 57062134, 60752134, 67052134, 50762143, 57062143, 60752143, 67052143, 56701432, 56701432 \\ 65701432, 65701432, 56701342, 56701342, 65701342, 65701342, 56702431, 56702431, 65702431 \\ 65702431, 56702341, 56702341, 65702341, 65702341, 50761432, 57061432, 60751432, 67051432 \\ 50761342, 57061342, 60751342, 67051342, 50762431, 57062431, 60752431, 67052431, 50762341 \\ 57062341, 60752341, 67052341, 76503214, 06573214, 75603214, 05673214, 76504213, 06574213 \\ 75604213, 05674213, 76503124, 06573124, 75603124, 05673124, 76504123, 06574123, 75604123 \\ 05674123, 70563214, 07563214, 70653214, 07653214, 70564213, 07564213, 70654213, 07654213 \\ 70563124, 07563124, 70653124, 07653124, 70564123, 07564123, 70654123, 07654123, 76503412 \\ 06573412, 75603412, 05673412, 76504312, 06574312, 75604312, 05674312, 76503421, 06573421 \\ 75603421, 05673421, 76504321, 06574321, 75604321, 05674321, 70563412, 07563412, 70653412 \\ 07653412, 70564312, 07564312, 70654312, 07654312, 70563421, 07563421, 70653421, 07653421 \\ 70564321, 07564321, 70654321, 07654321 \end{array} \right.$$

$$\mathcal{L}_B = \left\{ \begin{array}{l} 12345670, 12345607, 12346570, 12346507, 12435670, 12435607, 12436570, 12436507, 21345670 \\ 21345607, 21346570, 21346507, 21435670, 21435607, 21436570, 21436507, 12347056, 12340756 \\ 12347065, 12340765, 12437056, 12430756, 12437065, 12430765, 21347056, 21340756, 21347065 \\ 21340765, 21437056, 21430756, 21437065, 21430765, 34125670, 34125607, 34126570, 34126507 \\ 43125670, 43125607, 43126570, 43126507, 34215670, 34215607, 34216570, 34216507, 43215670 \\ 43215607, 43216570, 43216507, 34127056, 34120756, 34127065, 34120765, 43127056, 43120756 \\ 43127065, 43120765, 34217056, 34210756, 34217065, 34210765, 43217056, 43210756, 43217065 \\ 43210765, 12705634, 12075634, 12706534, 12076534, 12705643, 12075643, 12706543, 12076543 \\ 21705634, 21075634, 21706534, 21076534, 21705643, 21075643, 21706543, 21076543, 12567034 \\ 12560734, 12657034, 12650734, 12567043, 12560743, 12657043, 12650743, 21567034, 21560734 \\ 21657034, 21650734, 21567043, 21560743, 21657043, 21650743, 34705612, 34075612, 34706512 \\ 34076512, 43705612, 43075612, 43706512, 43076512, 34705621, 34075621, 34706521, 34076521 \\ 43705621, 43075621, 43706521, 43076521, 34567012, 34560712, 34657012, 34650712, 43567012 \\ 43560712, 43657012, 43650712, 34567021, 34560721, 34657021, 34650721, 43567021, 43560721 \\ 43657021, 43650721, 56341270, 56341207, 65341270, 65341207, 56431270, 56431207, 65431270 \\ 65431207, 56342170, 56342107, 65342170, 65342107, 56432170, 56432107, 65432170, 65432107 \\ 70341256, 07341256, 70341265, 07341265, 70431256, 07431256, 70431265, 07431265, 70342156 \\ 07342156, 70342165, 07342165, 70432156, 07432156, 70432165, 07432165, 56123470, 56123407 \\ 65123470, 65123407, 56124370, 56124307, 65124370, 65124307, 56213470, 56213407, 65213470 \\ 65213407, 56214370, 56214307, 65214370, 65214307, 70123456, 07123456, 70123465, 07123465 \\ 70124356, 07124356, 70124365, 07124365, 70213456, 07213456, 70213465, 07213465, 70214356 \\ 07214356, 70214365, 07214365, 56701234, 56701234, 65701234, 65701234, 56701243, 56701243, 56701243 \\ 65701243, 65701243, 56702134, 56702134, 65702134, 65702134, 56702143, 56702143, 65702143 \\ 65702143, 70561234, 07561234, 70651234, 07651234, 70561243, 07561243, 70651243, 07651243 \\ 70562134, 07562134, 70652134, 07652134, 70562143, 07562143, 70652143, 07652143, 56703412 \\ 56703412, 65703412, 65703412, 56704312, 56704312, 65704312, 65704312, 56703421, 56703421 \\ 65703421, 65703421, 56704321, 56704321, 65704321, 65704321, 70563412, 07563412, 70653412 \\ 07653412, 70564312, 07564312, 70654312, 07654312, 70563421, 07563421, 70653421, 07653421 \\ 70564321, 07564321, 70654321, 07654321 \end{array} \right.$$

$$\mathcal{L}_C = \left\{ \begin{array}{l} 12345670, 12346507, 21435670, 21436507, 12345076, 12346705, 21435076, 21436705, 14325670 \\ 14326507, 23415670, 23416507, 14325076, 14326705, 23415076, 23416705, 32147650, 32140567 \\ 41237650, 41230567, 32147056, 32140765, 41237056, 41230765, 34127650, 34120567, 43217650 \\ 43210567, 34127056, 34120765, 43217056, 43210765, 12305674, 12376504, 21405673, 21476503 \\ 12365074, 12356704, 21465073, 21456703, 14305672, 14376502, 23405671, 23476501, 14365072 \\ 14356702, 23465071, 23456701, 32107654, 32170564, 41207653, 41270563, 32167054, 32150764 \\ 41267053, 41250763, 34107652, 34170562, 43207651, 43270561, 34167052, 34150762, 43267051 \\ 43250761, 12745630, 12046537, 21735640, 21036547, 12745036, 12046735, 21735046, 21036745 \\ 14725630, 14026537, 23715640, 23016547, 14725036, 14026735, 23715046, 23016745, 32547610 \\ 32640517, 41537620, 41630527, 32547016, 32640715, 41537026, 41630725, 34527610, 34620517 \\ 43517620, 43610527, 34527016, 34620715, 43517026, 43610725, 12705634, 12076534, 21705643 \\ 21076543, 12765034, 12056734, 21765043, 21056743, 14705632, 14076532, 23705641, 23076541 \\ 14765032, 14056732, 23765041, 23056741, 32507614, 32670514, 41507623, 41670523, 32567014 \\ 32650714, 41567023, 41650723, 34507612, 34670512, 43507621, 43670521, 34567012, 34650712 \\ 43567021, 43650721, 56341270, 65341207, 56432170, 65432107, 50341276, 67341205, 50432176 \\ 67432105, 56321470, 65321407, 56412370, 65412307, 50321476, 67321405, 50412376, 67412305 \\ 76143250, 05143267, 76234150, 05234167, 70143256, 07143265, 70234156, 07234165, 76123450 \\ 05123467, 76214350, 05214367, 70123456, 07123465, 70214356, 07214365, 56301274, 65371204 \\ 56402173, 65472103, 50361274, 67351204, 50462173, 67452103, 56301472, 65371402, 56402371 \\ 65472301, 50361472, 67351402, 50462371, 67452301, 76103254, 05173264, 76204153, 05274163 \\ 70163254, 07153264, 70264153, 07254163, 76103452, 05173462, 76204351, 05274361, 70163452 \\ 07153462, 70264351, 07254361, 56741230, 65041237, 56732140, 65032147, 50741236, 67041235 \\ 50732146, 67032145, 56721430, 65021437, 56712340, 65012347, 50721436, 67021435, 50712346 \\ 67012345, 76543210, 05643217, 76534120, 05634127, 70543216, 07643215, 70534126, 07634125 \\ 76523410, 05623417, 76514320, 05614327, 70523416, 07623415, 70514326, 07614325, 56701234 \\ 65071234, 56702143, 65072143, 50761234, 67051234, 50762143, 67052143, 56701432, 65071432 \\ 56702341, 65072341, 50761432, 67051432, 50762341, 67052341, 76503214, 05673214, 76504123 \\ 05674123, 70563214, 07653214, 70564123, 07654123, 76503412, 05673412, 76504321, 05674321 \\ 70563412, 07653412, 70564321, 07654321 \end{array} \right.$$

$$\mathcal{L}_D = \left\{ \begin{array}{l} 12345670, 21436507, 12345076, 21436705, 14325670, 23416507, 14325076, 23416705, 32147650 \\ 41230567, 32147056, 41230765, 34127650, 43210567, 34127056, 43210765, 12305674, 21476503 \\ 12365074, 21456703, 14305672, 23476501, 14365072, 23456701, 32107654, 41270563, 32167054 \\ 41250763, 34107652, 43270561, 34167052, 43250761, 12745630, 21036547, 12745036, 21036745 \\ 14725630, 23016547, 14725036, 23016745, 32547610, 41630527, 32547016, 41630725, 34527610 \\ 43610527, 34527016, 43610725, 12705634, 21076543, 12765034, 21056743, 14705632, 23076541 \\ 14765032, 23056741, 32507614, 41670523, 32567014, 41650723, 34507612, 43670521, 34567012 \\ 43650721, 15342670, 26431507, 15342076, 26431705, 15324670, 26413507, 15324076, 26413705 \\ 37142650, 40231567, 37142056, 40231765, 37124650, 40213567, 37124056, 40213765, 15302674 \\ 26471503, 15362074, 26451703, 15304672, 26473501, 15364072, 26453701, 37102654, 40271563 \\ 37162054, 40251763, 37104652, 40273561, 37164052, 40253761, 15742630, 26031547, 15742036 \\ 26031745, 15724630, 26013547, 15724036, 26013745, 37542610, 40631527, 37542016, 40631725 \\ 37524610, 40613527, 37524016, 40613725, 15702634, 26071543, 15762034, 26051743, 15704632 \\ 26073541, 15764032, 26053741, 37502614, 40671523, 37562014, 40651723, 37504612, 40673521 \\ 37564012, 40653721, 62345170, 51436207, 02345176, 71436205, 64325170, 53416207, 04325176 \\ 73416205, 62147350, 51230467, 02147356, 71230465, 64127350, 53210467, 04127356, 73210465 \\ 62305174, 51476203, 02365174, 71456203, 64305172, 53476201, 04365172, 73456201, 62107354 \\ 51270463, 02167354, 71250463, 64107352, 53270461, 04167352, 73250461, 62745130, 51036247 \\ 02745136, 71036245, 64725130, 53016247, 04725136, 73016245, 62547310, 51630427, 02547316 \\ 71630425, 64527310, 53610427, 04527316, 73610425, 62705134, 51076243, 02765134, 71056243 \\ 64705132, 53076241, 04765132, 73056241, 62507314, 51670423, 02567314, 71650423, 64507312 \\ 53670421, 04567312, 73650421, 65342170, 56431207, 05342176, 76431205, 65324170, 56413207 \\ 05324176, 76413205, 67142350, 50231467, 07142356, 70231465, 67124350, 50213467, 07124356 \\ 70213465, 65302174, 56471203, 05362174, 76451203, 65304172, 56473201, 05364172, 76453201 \\ 67102354, 50271463, 07162354, 70251463, 67104352, 50273461, 07164352, 70253461, 65742130 \\ 56031247, 05742136, 76031245, 65724130, 56013247, 05724136, 76013245, 67542310, 50631427 \\ 07542316, 70631425, 67524310, 50613427, 07524316, 70613425, 65702134, 56071243, 05762134 \\ 76051243, 65704132, 56073241, 05764132, 76053241, 67502314, 50671423, 07562314, 70651423 \\ 67504312, 50673421, 07564312, 70653421 \end{array} \right.$$

Codes for Schemes \mathcal{M}_A to \mathcal{M}_D :

$$\mathcal{M}_A = \left\{ \begin{array}{l} 1234567, 2134567, 1243567, 2143567, 1234657, 2134657, 1243657, 2143657, 3214567, 3124567 \\ 4213567, 4123567, 3214657, 3124657, 4213657, 4123657, 1432567, 2431567, 1342567, 2341567 \\ 1432657, 2431657, 1342657, 2341657, 3412567, 3421567, 4312567, 4321567, 3412657, 3421657 \\ 4312657, 4321657, 1234765, 2134765, 1243765, 2143765, 1234756, 2134756, 1243756, 2143756 \\ 3214765, 3124765, 4213765, 4123765, 3214756, 3124756, 4213756, 4123756, 1432765, 2431765 \\ 1342765, 2341765, 1432756, 2431756, 1342756, 2341756, 3412765, 3421765, 4312765, 4321765 \\ 3412756, 3421756, 4312756, 4321756, 5674123, 5674213, 5673124, 5673214, 6574123, 6574213 \\ 6573124, 6573214, 5674321, 5674312, 5673421, 5673412, 6574321, 6574312, 5673421, 6573412 \\ 5672143, 5671243, 5672134, 5671234, 6572143, 6571243, 6572134, 6571234, 5672341, 5671342 \\ 5672431, 5671432, 6572341, 6571342, 6572431, 6571432, 7654123, 7654213, 7653124, 7653214 \\ 7564123, 7564213, 7563124, 7563214, 7654321, 7654312, 7653421, 7653412, 7564321, 7564312 \\ 7563421, 7563412, 7652143, 7651243, 7652134, 7651234, 7562143, 7561243, 7562134, 7561234 \\ 7652341, 7651342, 7652431, 7651432, 7562341, 7561342, 7562431, 7561432 \end{array} \right\}$$

$$\mathcal{M}_B = \left\{ \begin{array}{l} 1234566, 2134566, 1243566, 2143566, 1234656, 2134656, 1243656, 2143656, 3412566, 3421566 \\ 4312566, 4321566, 3412656, 3421656, 4312656, 4321656, 5234166, 5134266, 5243166, 5143266 \\ 6234156, 6134256, 6243156, 6143256, 5412366, 5421366, 5312466, 5321466, 6412356, 6421356 \\ 6312456, 6321456, 1634526, 2634516, 1643526, 2643516, 1534626, 2534616, 1543626, 2543616 \\ 3612546, 3621546, 4612536, 4621536, 3512646, 3521646, 4512636, 4521636, 5634126, 5634216 \\ 5643126, 5643216, 6534126, 6534216, 6543126, 6543216, 5612346, 5621346, 5612436, 5621436 \\ 6512346, 6521346, 6512436, 6521436, 1264563, 2164563, 1263564, 2163564, 1264653, 2164653 \\ 1263654, 2163654, 3462561, 3461562, 4362561, 4361562, 3462651, 3461652, 4362651, 4361652 \\ 5264163, 5164263, 5263164, 5163264, 6264153, 6164253, 6263154, 6163254, 5462361, 5461362 \\ 5362461, 5361462, 6462351, 6461352, 6362451, 6361452, 1664523, 2664513, 1663524, 2663514 \\ 1564623, 2564613, 1563624, 2563614, 3662541, 3661542, 4662531, 4661532, 3562641, 3561642 \\ 4562631, 4561632, 5664123, 5664213, 5663124, 5663214, 6564123, 6564213, 6563124, 6563214 \\ 5662341, 5661342, 5662431, 5661432, 6562341, 6561342, 6562431, 6561432 \end{array} \right\}$$

$$\mathcal{M}_C = \left\{ \begin{array}{l} 1234555, 2134555, 1243555, 2143555, 1254553, 2154553, 1253554, 2153554, 3214555, 3124555 \\ 4213555, 4123555, 3254551, 3154552, 4253551, 4153552, 1432555, 2431555, 1342555, 2341555 \\ 1452553, 2451553, 1352554, 2351554, 3412555, 3421555, 4312555, 4321555, 3452551, 3451552 \\ 4352551, 4351552, 5234155, 5134255, 5243155, 5143255, 5254153, 5154253, 5253154, 5153254 \\ 5214355, 5124355, 5213455, 5123455, 5254351, 5154352, 5253451, 5153452, 5432155, 5431255 \\ 5342155, 5341255, 5452153, 5451253, 5352154, 5351254, 5412355, 5421355, 5312455, 5321455 \\ 5452351, 5451352, 5352451, 5351452, 1534525, 2534515, 1543525, 2543515, 1554523, 2554513 \\ 1553524, 2553514, 3514525, 3524515, 4513525, 4523515, 3554521, 3554512, 4553521, 4553512 \\ 1532545, 2531545, 1542535, 2541535, 1552543, 2551543, 1552534, 2551534, 3512545, 3521545 \\ 4512535, 4521535, 3552541, 3551542, 4552531, 4551532, 5534125, 5534215, 5543125, 5543215 \\ 5554123, 5554213, 5553124, 5553214, 5514325, 5524315, 5513425, 5523415, 5554321, 5554312 \\ 5553421, 5553412, 5532145, 5531245, 5542135, 5541235, 5552143, 5551243, 5552134, 5551234 \\ 5512345, 5521345, 5512435, 5521435, 5552341, 5551342, 5552431, 5551432 \end{array} \right\}$$