Evaluation of mixed permutation codes in PLC channels, using Hamming distance profile

Kehinde Ogunyanda · Ayokunle D. Familua · Theo G. Swart · Hendrik C. Ferreira · Ling Cheng

Received: date / Accepted: date

Abstract We report a new concept involving an adaptive mixture of different sets of permutation codes (PC) in a single DPSK-OFDM modulation scheme. Since this scheme is robust and the algorithms involved are simple, it is a good candidate for implementation for OFDM-based power line communication (PLC) systems. By using a special and easy concept called Hamming distance profile, as a comparison tool, we are able to showcase the strength of the new PC scheme over other schemes reported in literature, in handling the incessant noise types associated with PLC channels. This prediction tool is also useful for selecting an efficient PC codebook out of a number of similar ones.

Keywords Channel coding · Digital modulation · G3-PLC · Hybrid permutation coding · OFDM · Power line communications

K. Ogunyanda . T. G. Swart . H. C. Ferreira
Department of Electrical and Electronic Engineering Science, PO Box 524, University of Johannesburg, Auckland Park, 2006, South Africa
Tel.: +27731368796
E-mail: ogunyanda@gmail.com

T.G. Swart
E-mail: tgswart@uj.ac.za

H.C. Ferreira
E-mail: hcferreira@uj.ac.za

A. D. Familua . L. Cheng
School of Electrical and Information Engineering, University of the Witwatersrand, Private Bag 3, WITS. 2050, South Africa
E-mail: familuakunle@gmail.com

L. Cheng
E-mail: ling.cheng@wits.ac.za

1 Introduction

Narrowband power line communications (PLC) is plagued with a number of noise types which include background noise (usually modelled as additive white Gaussian noise (AWGN)), impulsive noise (IN) and narrowband interference (NBI) [1–3]. Based on the noisy nature of the PLC channel, one approach would have been to transmit at high power, or at frequencies that are free from the distortions associated with the channel. However, these are not practicable, due to the frequency and power limitations from the regulatory standards [4–6]. Another approach is to clip the impulsive noise at the receiver side, such as reported in [7–9]. Although narrowband PLC is employed for low speed communication applications such as load management and automatic meter reading [10–12], it is however needed to ensure robustness and simplicity in the PLC transmission scheme. As such, low-rate but robust error correction codes and modulation schemes are necessary measures.

The two narrowband PLC standards have identified orthogonal frequency division multiplexing (OFDM) as a robust modulation scheme, using M-ary differential phase shift keying (MDPSK) as the constellation mapper [5,6]. In G3-PLC, concatenation of a Reed-Solomon (RS) code with a convolutional code (CC) is used in an MDPSK-OFDM system (i.e., RS-CC-MDPSK-OFDM) [6]. From henceforth, we shall refer to this as Scheme A, which is used as the platform of comparison in this study.

The interest in permutation coding (PC) got revived by Vinck [3], when he suggested its use for PLC purposes. Afterwards, a number of work has been reported on PC, which proved its robustness in PLC applications [2, 3, 13–22]. In [21], a differential quinary PSK modulator (i.e., DQuiPSK) which helps improve the performance of an RS-PC-DPSK-OFDM scheme was reported. However, this scheme (henceforth called Scheme B), only has appreciable performance gain over Scheme A at lower signal power to noise spectral density (E_b/N_o) values. Also, a modified PC scheme called injection code (henceforth called Scheme C) was reported by Dukes in [16]. Its disadvantage is the complexity involved in the codeword generation.

In the current study, we report a hybrid PC (HPC) scheme (henceforth called Scheme D), which helps to improve the performance of an RS-PC-DPSK-OFDM scheme, at both low and high E_b/N_o values, over Scheme A. It adaptively maps RS bits onto a hybrid of two sets of PCs, one of which is mathematically derived from the other. This thus reduces the complexity of generating the entire codebook. It also has competitive performance with Scheme C. Another Scheme E which offers a less complex decoding algorithm, but with a little performance degradation, is further developed from Scheme D. As far as we know, this is the first time PC schemes such as these are reported. With a view to comparing the performances of all the various PC schemes considered, we devised a concept called Hamming distance profile (HDP), which evaluates the contributions of each posible Hamming distance (HD) in each code. Hence, the scheme with better HDP yields better performance.

Section 2 gives a brief description of PC, while Section 3 describes the HDP concept, together with Schemes B and C, before describing the new HPC concept and other comparative schemes in Section 4. We look into PC mappings with very large possible HDs in Section 5, where the usefulness of HDP is further explored. Simulation setup and results obtained are presented in Section 6. Section 7 concludes the paper.

2 General Background

Permutation coding entails mapping binary data onto non-binary sequences, with each codeword having M non-repetitive symbols. Several distance parameters such as Hamming, Euclidean, Kendall tau, Chebyshev and Ulam distance are used in the literature to classify permutation codes and its combinations [23–28]. In this paper, we focus on Hamming distance. The cardinality $|C|$ of a PC, which is the number of codewords in the codebook, is upper bounded by $d_{\min}$ as [23, 24]

$$|C| \leq \frac{M!}{(d_{\min} - 1)}. \tag{1}$$

More so, in [15] the authors presented a concept of a distance-preserving mapping, where a PC can be regarded as a distance-conserving mapping, distance-increasing mapping or distance-reducing mapping, based on the distance relationship between the information bits and the corresponding codewords. Swart in [29] used the distance properties of PCs to determine their optimalities. However, this approach has a shortcomming, which is addressed in the HDP approach presented in Section 5 of this work. As such, a part of this work can be seen as an extension of the work done in [29].

In this work, we shall adapt PC to D8PSK modulation, whose number of constellation points is $M_{\text{DP}} = 8$. In order to minimize the encoding and decoding complexities, let us consider a PC of codeword length $M = 5$. Better performance is expected if $M > 5$, but at the expense of complexity. Hence, a suitable PC mapping can map 5 bits onto 5 PC symbols. As such, 32 codewords are needed. An example is [30]:

$$\left\{ \begin{array}{c} 12034, 21430, 13204, 24103, 21304, 12340, 23014 \\ 23140, 10243, 01423, 20134, 03241, 41320, 21043 \\ 31024, 30142, 14230, 12403, 34201, 04132, 42013 \\ 32410, 34012, 43102, 04321, 02431, 40231, 30421 \\ 02314, 40312, 43021, 03412 \end{array} \right\} \tag{2}$$

Using these codewords in a D8PSK modulator, only 5 out of the 8 available constellation points will be used by the modulators, which gives room for 3 other symbols (i.e., 5, 6 and 7) to feature as *foreign symbol errors* (FSE), at the receiving end. We define the term FSE as an incorrect symbol which, ideally, is not expected to be found in the defined codewords. To some extent, Schemes B [21], C [16] and the proposed Scheme D are able to address this issue, with C and D having excellent performances.

3 Hamming Distance Profile

In coding, it is generally known that $d_{\min}$ is a measure of the strength of a code in correcting errors [22, 31]. By $d_{\min}$, we refer to the minimum HD, which is the minimum (least) possible HD between any two codewords in a codebook. In CC, a term called distance spectrum is generally employed to compare the performances of CCs with similar constraint lengths and decoding complexities [32–34]. This distance spectrum makes use of the amount of error events for every distance between the current and previous d_{free} spectrum, which in turn is the least amount of errors that produces an error event in the encoded sequence. A code with the best distance spectrum is considered the best in the competition. Also, in [35], Viterbi demonstrated that various possible distances contributed by every remerging path in a trellis-code representation contribute to the error probability of a CC. In this study, we however demonstrate that PC codebooks of the same $d_{\min}$ can have varying performances, based on their HDPs. This is because given a particular codebook, apart from the $d_{\min}$, other distances also in one way or another, contribute to its performance. Hence, HDP is defined as the amount of contributions each possible HD offers in the oveall performance of a whole set of codewords in a codebook. This approach is less intricate than the approach of the distance spectrum in CC [32–34], where computation of variable separate vectors is involved [36].

Definition 1 Given a permutation codebook $\mathcal{C}$ with M and $d_{\min}$, the following describes the possible HDs in the codebook:

$$d_{\min} = d_{M-u} = M - u, \quad \text{where} \quad 0 \leq u \leq M, \quad M \geq 2$$
$$\text{and} \quad d_{M-0} > d_{M-1} > d_{M-2} > \cdots > d_2. \tag{3}$$

For example, if the $d_{\min} = d_{M-2}$, this implies that other HDs d_{M-0} and d_{M-1} are also possible in the codebook, which, by values, are higher than the $d_{\min}$. This example fits with the codebook in (4)

$$\left\{ \begin{matrix} 0123, 2013, 1230, 0231, 1023, 3021, 2310, 2130 \\ 0213, 1032, 2031, 3201, 3012, 2103, 2301, 1320 \end{matrix} \right\}, \tag{4}$$

where $M = 4$, and d_2 (i.e., $d_{\min}$), d_3 and d_4 are the possible HDs.

To make comparisons between codebooks, we will use the fractional contribution of the sum of each of these possible HDs between permutation sequences in a given codebook. The fraction of distance d_{M-u} in a codebook is given by $\mathcal{F}_{d_{M-u}}$ as:

$$\mathcal{F}_{M-u} = N_{M-u}/d_t, \tag{5}$$

where N_{M-u} is the number of times distance d_{M-u} appears and d_t is the total number of appearance of all possible HDs.

Proposition 1 The performance of any permutation mapping is said to be independent of any HD, whose fractional contribution is very low, as compared to those of other distances.

Proof: Assumming PC is regarded as a probabilistic algorithm, whose codewords are probabilistically selected from the entire codebook $\mathcal{C}$, we can prove the above proposition using the negligible probability approach usually adopted in cryptography [37–40]. If the contribution of a minor $\mathcal{F}_{M-u}$ is to be analyzed, we assume such fraction to be the probability of failure P_{fail}, while the composite contribution of all other fractions is considered to be the probability of success P_{succ}. For instance, assumming the possible HDs in $\mathcal{C}$ are distances 2, 3, 4 and 5, and we want to analyze the contribution of $\mathcal{F}_3$, which is assumed to be minor, we thus have

$$P_{\text{fail}} = \mathcal{F}_3 \quad \text{and} \quad P_{\text{succ}} = \mathcal{F}_2 + \mathcal{F}_4 + \mathcal{F}_5. \tag{6}$$

In general, if the probability that an event succeeds is p, its failure probability will be $q = 1-p$, provided that $p > 0$. Hence, to render q negligible, its value should be below some threshold, such that $1-p \approx 0$. This implies that $p \gg q$. Hence, we can generalize that, if a certain d_{M-u} has a minimal fraction $\mathcal{F}_{M-u}$, its contribution will also be negligible in the overall performance of the given codebook.

Proposition 2 The performance that any permutation mapping can attain is mostly dependent on the Hamming distance whose fractional contribution (or number of occurence) is the largest.

Proof: The same proof employed above is employed here as well. The larger HD fractions fall in the P_{succ} category. Hence, if P_{fail} is negligible, the codebook performance tends to depend on P_{succ}.

Proposition 3 The difference in the various contributions of P_{fail} and P_{succ} becomes more significant, as L_n increases.

Proof: We employ the concept of statistical significance [41, 42] in this claim. If the mapped message bits are not a definite sequence, PC is considered an algorithm which takes an input parameter z and Pics codewords from $\mathcal{C}$ up to $L_n = L/n$ number of times. Here, the input parameter z is the grouped n bits symbol to be mapped onto a PC codeword from $\mathcal{C}$, while L is the total number of message bits that are being mapped onto the PC codewords. In this regard, $z = 1$, since a set of n bits is mapped at a time. Hence, we assume L_n to be the sample size in a statistical random process. Statistical testing is usually done, with the purpose of revealing a significant difference, if at all it exists. Large sample size aids the chance of achieving a statistical significant difference. In other words, when large sample size is involved, minute differences becomes significant. Hence, it is certain that the difference is real.

For example, let us assume we are conducting a statistical trial to see if there will be significant difference between the number of appearance of a distance d_x and that of distance d_y, when the codebook is called 1,000 times (for scenario 1) and 25 times (for scenario 2). Assumming the mean of the number of appearances of d_x is 97 and that of d_y is 100, in the two scenarios, the difference between the two means 97 and 100, based on this trial is somewhat small. We can use a distribution curve to determine how likely the difference is significant in each scenario, as shown in Fig. 1.

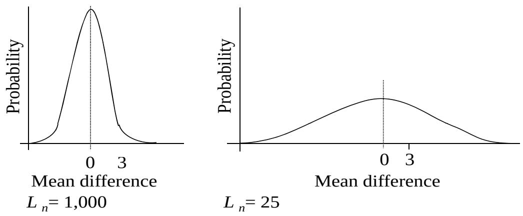

Fig. 1 Illustrating statistical significance, using two sample sizes 1,000 and 25

We define a null hypothesis Υ_0, such that there is no difference in the number of appearances of d_x and d_y, and an alternative hypothesis Υ_A, such that a difference exists [43]. The two curves are centred on 0, to indicate Υ_0 (i.e., when no difference occurs in the two means). Hence, in scenario 1, where the sample size is large and the distribution curve narrower, the difference becomes significant and a mean difference of 3 is more extreme to it than the case in scenario 2. This thus indicates that the difference is very real and not just a coincidence. The curve for scenario 1 is spikier, because of the standard error of the mean. Hence, the trial case becomes more accurate as the sample size (i.e., L_n) gets larger. From the curve for scenario 2, we can infer that the two-point difference between the appearances of d_x and d_y is insignificant, due to the smaller sample size, and there is no solid assurance that the difference is not just a mere coincidence.

Now, considering the case where the difference is large (i.e., not minute), the larger size of L_n makes the contributions of the distance with a huge fraction (i.e., P_{succ}) to be significant, while rendering the ones with a very small fraction (i.e., P_{fail}) to be nearly negligible. This thus establishes the fact in our proposition.

3.1 Scheme B

This uses a conventional PC system, such as in (2), together with a DQuiPSK modulator. For fair comparison, the codewords in (2) shall be used in this scheme. They are more optimal than those used in [21], due to better $d_{\min}$, which is now 3. Using (3), d_3, d_4 and d_5 exist, and according to (5), $N_3 = 288$, $N_4 = 294$, $N_5 = 410$ and $d_t = 992$. It should be noted that distances between each codeword and itself are not considered. Hence, $\mathcal{F}_3 = 288/992$, $\mathcal{F}_4 = 294/992$ and $\mathcal{F}_5 = 410/992$, respectively.

The DQuiPSK modulator involves constraining the modulator's output to M, thereby reducing the chances of having FSEs at the receiving end.

3.2 Scheme C

This is an injection code, which is constructed from dispersed sets of symbols selected from the entire PC symbol set [16]. Here, alphabets of larger sizes than M are defined, thereby giving room for more possible codeword combinations. As defined in [16], a $Q(8;5;4)$ injection code has 56 possible codewords, out of which 32 can be selected and used in this scheme, where the alphabet size is 8, $M = 5$ and $d_{\min} = 4$. Using (3) and (5), the HDP of these 32 codewords thus gives $\mathcal{F}'_4 = 610/992$ and $\mathcal{F}'_5 = 382/992$.

4 Hybrid Permutation

As done in [16], the proposed HPC also makes use of all the D8PSK constellation points, but in a simplified sequence. Fig. 2 depicts the schematic of the proposed scheme in a DPSK-OFDM system. The upper section is the transmitter and the lower section is the receiver.

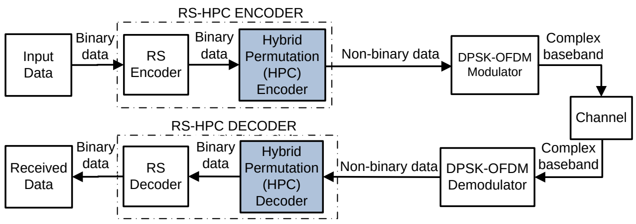

Fig. 2 A complete RS-HPC-DPSK-OFDM transceiver

The strength of PC, in PLC systems, has been demonstrated to have competitive performance with its CC counterpart [2]. It has also been demonstrated both by simulation and practical experimentation in [44, 45], that a PC scheme outperforms a conventional CC scheme when used with differential modulation under severe PLC channel conditions. Also, RS code is known to be robust in the presence of burst errors [31]. Hence, in order to further strengthen the communication system, a concatenated RS-HPC scheme is proposed in this work, as shown in Fig. 2.

4.1 Scheme D

Although there are a number of works reported on HPC [46, 47], the approaches in such works are quite different from what is reported in this work. For instance, the work in [46] combines different permutation sequences consisting of squaring permutations and cyclic permutations, with the former being inferior to the latter, due to its smaller size of permutation group. Since different sequences and sizes are involved, complexity in the decoding algorithm is inevitable, unlike the case of our proposed HPC, where the PCs combined are of similar sequences and sizes. Also in [47], the authors used a steady-state genetic algorithm to generate sequences which are further modified by successive applications of an adjacent pairwise permutation procedure. As such, this approach is more or less a double permutation algorithm, which is different from what is proposed here.

In the HPC proposed in this work, we define two classes of permutation codewords of similar sequences, denoted as PC1 and PC2, each having $|C| = 16$ and $d_{\min} = 4$. Both PC1 and PC2 contain symbols chosen from a universal set, U whose elements are non-binary symbols between 0 and $M_{\mathrm{DP}} - 1$. These properties are mathematically expressed as

$$\begin{aligned} &u_{\mathrm{pc1}} \subset U, \quad u_{\mathrm{pc2}} \subset U, \\ &u_{\mathrm{pc1}} \cup u_{\mathrm{pc2}} = U \quad \text{and} \quad |\mathrm{PC1}| = |\mathrm{PC2}|, \end{aligned} \tag{7}$$

where u_{pc1} and u_{pc2} denote the sets of symbols to be permuted in order to obtain all the codewords needed in PC1 and PC2 respectively, while $|\mathrm{PC1}|$ and $|\mathrm{PC2}|$ denote the respective cardinalities of PC1 and PC2.

In order to obey the property in (7), the individual symbol s_{pc2_l} contained in u_{pc2} can be derived from the individual symbol s_{pc1_l} contained in u_{pc1}, using the expression:

$$s_{\mathrm{pc2}_l} = s_{\mathrm{pc1}_l} + (M_{\mathrm{DP}} - M), \quad i = 1, \ldots, M, \tag{8}$$

where in this case, $M = 5$ for each PC class, and $M_{\mathrm{DP}} - M$ is the number of redundant constellation points which is 3. This number is added to each element s_{pc1_l} of the set u_{pc1}, in order to obtain elements s_{pc2_l} in the set u_{pc2}. For this particular scheme, U and u_{pc1} are given by $U = \{0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7\}$ and $u_{\mathrm{pc1}} = \{0,\ 1,\ 2,\ 3,\ 4\}$. Hence, for example, $s_{\mathrm{pc1}_4} = 3$, for $i = 4$. Using (8), $s_{\mathrm{pc2}_4} = 6$. Hence, all the elements in u_{pc2} are given by $\{3,\ 4,\ 5,\ 6,\ 7\}$.

Based on the above description, the following gives examples of the codewords in PC1 and PC2 for the two PC classes:

$$\begin{aligned} \mathrm{PC1} &= \left\{ \begin{array}{c} 12340, 13402, 14023, 10234, 23014, 20143 \\ 21430, 24301, 31042, 32104, 34210, 30421 \\ 02413, 03241, 01324, 04132 \end{array} \right\} \\ \mathrm{PC2} &= \left\{ \begin{array}{c} 45673, 46735, 47356, 43567, 56347, 53476 \\ 54763, 57634, 64375, 65437, 67543, 63754 \\ 35746, 36574, 34657, 37465 \end{array} \right\} \end{aligned} \tag{9}$$

If PC1 and PC2 are combined, the following PC symbol mapping algorithm can be used to encode data, consisting of L bits, onto HPC symbols.

Symbol mapping algorithm

1. Bits grouping: Group the input L bits into sets of n. Here, $n = M$.
2. Binary to decimal conversion: Convert each set of M bits into their corresponding decimal values, and denote them as D_i, where $i = \{1, 2, \ldots, L/n\}$.
3. Bits adaptation: Select a codeword, whose index corresponds to $D_i + 1$, from the mixed PC1 and PC2 classes.

The following detection algorithm applies as well.

Codeword detection algorithm

1. Codeword assembly: Arrange the received symbols, composed of L rows, into M columns, with each row representing a prospective PC codeword x_i.
2. Metric computation: Compute the distance, r_k between x_i and every possible codeword in the two PC classes, where $k = \{1, 2, \ldots, 2\ |C|\}$.
3. Codeword declaration: The codeword with the least distance r_k to x_i is selected as the detected codeword.
4. Confusion error declaration: If more than one least r_k exists, declare a confusion error, and make a random guess between the codewords having the same least r_k.

The PC decoder may get confused at some points, where repeating symbols feature in x_i, due to channel errors. For convenience, we shall term this kind of phenomenon a *repetitive symbol error* (RSE). FSEs, discussed in Section 2, can also cause confusion in x_i. In such cases, more than one codeword are bound to have the same least r_k to x_i, as stated in step 4 above. These algorithms also hold for general PC schemes, but $|C|$ will be used instead of $2\ |C|$ in step 2 of the *Codeword detection algorithm.*

Using (3) and (5), the combined HDP for Scheme D gives $\mathcal{F}_3'' = 24/992$, $\mathcal{F}_4'' = 536/992$ and $\mathcal{F}_5'' = 432/992$. Since $\mathcal{F}_4''$ and $\mathcal{F}_5''$ have the larger share of the HD Fraction, with $\mathcal{F}_3''$ being insignificant (Propositions 1 to 3), Scheme D's error correcting capability tends to rely mostly on d_4 and d_5. The essence of HPC is therefore to reduce the contribution of $d_{\min}$ in the codebook's performance. Hence, it will theoretically perform better than the conventional PC in Scheme B, whose capability relies mostly on d_3, d_4 and d_5. Although these Schemes B and D have the same $d_{\min} = 3$, the HDP approach is able to predict which one performs better.

Also, because the distance in Scheme C is dominated by d_4 and d_5, it should have average performance similar to D, since $\mathcal{F}_3''$ is minimal. Since the above analysis shows that HPC has better performance than ordinary PC, which has also been demonstrated to be more robust than CC [44, 45], we can therefore infer that the proposed HPC performs better than the conventional CC system (as in Scheme A). This shall be later revealed in our simulation results.

Example 1: We assume the correct codewords from (9) are to be detected for the following assembled codewords $x_1 = (1\ 6\ 5\ 2\ 2)$ and $x_2 = (3\ 1\ 4\ 0\ 5)$. Here, x_1 has two least r_k, which correspond to (1 3 4 0 2) and (1 4 0 2 3). Symbols 6 and 5 are featuring as a result of FSEs (based on PC1), while 2 and 2 as RSEs. In this regard, a confusion error is declared, and a random guess between the two codewords has to be made. For x_2, only one least r_k exists, which corresponds to (3 7 4 6 5). This is selected as the output codeword. Here, only FSEs are present (based on PC2). If we denote the index of the selected codeword as v, the original encoded RS bits can be obtained from the detected PC codewords, by computing the binary equivalence of the index, $v - 1$. In the case of (3 7 4 6 5), its index number is $v = 31$ from the combined PC classes. Hence, the original input bits are (1 1 1 1 1).

4.2 Schemes E and F

If PC word length is reduced, decoding becomes less complex, since the number of codewords will invariably reduce. Hence, for Scheme E, we map 4 bits onto 4 PC symbols, using the same algorithm used in D. Its two sets of codewords are given by

$$\begin{aligned} &\text{PC1} = \{1230, 1302, 1023, 3201, 0132, 2310, 2031, 2103\} \\ &\text{PC2} = \text{PC1} + (M_{\text{DP}} - M) = \text{PC1} + 4. \end{aligned} \tag{10}$$

However, since $M = 4$ here, one could have suggested the use of DQPSK, with 4 constellation points. We thus use this modulator in another conventional Scheme F as a platform of comparison for E. In Scheme F, 16 codewords, with $d_{\min} = 2$ are possible, examples of which have been presented in (4).

The HD profile for Scheme E gives $\mathcal{F}_3''' = 80/240$ and $\mathcal{F}_4''' = 160/240$, while that of F gives $\mathcal{F}_2'''' = 64/240$, $\mathcal{F}_3'''' = 84/240$ and $\mathcal{F}_4'''' = 92/240$. Hence, Scheme E should theoretically outperform Scheme F.

5 PC mappings with very large possible Hamming distances

So far, we have considered various PCs whose largest HD is $< d_6$. At a glance, it is very easy to predict the performances of such PCs, by examining their HDPs. However, in a situation where there are HDs $> d_5$, a critical look into the HDP will be required, before predicting the PC's performance, because each possible HD contributes to the performance (Section 3). In [29], various similar mappings, with HD $> d_5$ were presented. Therein, it was discovered by simulation that similar mappings/codebooks, with the same $|C|$ and distance optimality actually exhibit slightly different performances, without a solid explanation of these disparities. In this section, we therefore use the HDP approach to explain the disparities. For the sake of emphasis, we shall briefly describe the optimality approach described in [29].

Two matrices $\mathbf{E}$ and $\mathbf{E}^{(m)}$ are used to establish the distance optimality of a PC. Matrix $\mathbf{E}$ consists of elements $e_{i,j}$ that represent the HD between codewords x_i, where $i = 1, 2, \ldots, |C|$. For example, according to (2), $x_1 = \{1\ 2\ 0\ 3\ 4\}$, and $x_2 = \{2\ 1\ 4\ 3\ 0\}$. Hence by computation, the distance between x_1 and x_2 gives the $e_{i,j}$ element as $e_{1,2} = 4$. By doing this for all the sequences in (2), $\mathbf{E}$ can be generated as a $|C| \times |C|$ matrix. Similarly, $\mathbf{E}^{(m)}$ is the distance matrix (with elements $e_{i,j}^{(m)}$) that is generated by the symbols in position m, $1 \leq m \leq M$. Consequently, when the distances between codewords are computed, if there is a different symbol in position m for the symbol being considered, then that would contribute a 1 to $\mathbf{E}^{(m)}$. Invariably, M number of $\mathbf{E}^{(m)}$ matrices, whose dimensions are $|C| \times |C|$, are generated for a codebook. The magnitudes of these matrices, denoted by $|\mathbf{E}|$ and $|\mathbf{E}^{(m)}|$, can be represented by [29]:

$$|\mathbf{E}| = \sum_{i=1}^{|C|} \sum_{j=1}^{|C|} e_{i,j} \quad \text{and} \quad |\mathbf{E}^{(m)}| = \sum_{i=1}^{|C|} \sum_{j=1}^{|C|} e_{i,j}^{(m)}. \tag{11}$$

A codebook is said to be distance optimal, if $|\mathbf{E}|$ is maximized, but to achieve that, all $|\mathbf{E}^{(m)}|$ need to be maximized. According to (2), $|\mathbf{E}| = 4090$, and $|\mathbf{E}^{(1)}| = |\mathbf{E}^{(2)}| = |\mathbf{E}^{(3)}| = |\mathbf{E}^{(4)}| = |\mathbf{E}^{(5)}| = 818$, which is exactly the maxima obtainable. Hence, the PC is said to be optimal.

Based on the above distance optimality approach, three $M(6, 6, 0)$ mappings (i.e., G, H and I), and four $M(8, 8, 0)$ mappings (i.e., J, K, L and M) were discovered, using a multilevel approach of codeword generation in [29]. Each of these mappings have slightly different performance from its counterparts. Their HDPs are presented in Table 1.

From this table, $\mathcal{F}_{Y_i}$ is the fraction of distance i from codebook Y. As established in Section 3, larger fractions of weaker HDs negatively affect a codebook's performance. Likewise, larger fractions of stronger HDs positively affects the codebook's performance. Here, we consider weaker distances to be $\leq d_3$, while stronger distances are $> d_3$. Based on this, we use the following algorithm to judge the performances of the codebooks.

HDP decision algorithm for large HDs

1. Input: ($\mathcal{F}_{Y_i}$ and $\mathcal{F}_{Y'_i}$, for $i = 2, 3, \ldots, M$), where $\mathcal{F}_{Y_i}$ is the HD fraction for distance i of codebook Y being compared with other fractions of distance i from the other codebooks Y'.
2. Descision based on weaker distances: While $i \leq 3$, compare $\mathcal{F}_{Y_i}$ with $\mathcal{F}_{Y'_i} \pm \delta$, where δ is the percentage of tolerable difference between $\mathcal{F}_{Y_i}$ and $\mathcal{F}_{Y'_i}$, within which we can say $\mathcal{F}_{Y_i} \approx F_{Y'_i}$. If $\mathcal{F}_{Y_i} < F_{Y'_i} \pm \delta$, Y is better than Y', else if $\mathcal{F}_{Y_i} > F_{Y'_i}$, Y is not better than Y', go to step 4. Otherwise, increment i and repeat this step until $i = 3$ and go to step 3.
3. Descision based on stronger distances: While $i > 3$, compare $\mathcal{F}_{Y_i}$ with $\mathcal{F}_{Y'_i} \pm \delta$. If $\mathcal{F}_{Y_i} > F_{Y'_i} \pm \delta$, Y is better than Y', else if $\mathcal{F}_{Y_i} < F_{Y'_i}$, Y is not better than Y', go to step 4. Otherwise, increment i and repeat this step until $i = M$. If at this stage, $\mathcal{F}_{Y_M} = F_{Y'_M}$ the codekooks have overlapping performances, go to step 4.
4. Repeat steps 1 to 3 until all codebooks are compared.

For the comparisons of the three $M(6,6,0)$ codebooks and the four $M(8,8,0)$ codebooks, we assume $\delta = 3\%$. From Table 1, $\mathcal{F}_{\mathrm{G}_2} = \mathcal{F}_{\mathrm{H}_2} = F_{\mathrm{I}_2}$. We then proceed to $i = 3$, according to step 2 above. At this stage, $\mathcal{F}_{\mathrm{G}_3} = \mathcal{F}_{\mathrm{I}_3} \approx F_{\mathrm{H}_3}$. We then proceed to step 3. At this stage, at $i = 4$, $\mathcal{F}_{\mathrm{H}_4} < F_{\mathrm{G}_4}$ and $\mathcal{F}_{\mathrm{I}_4}$, but $\mathcal{F}_{\mathrm{G}_4} \approx \mathcal{F}_{\mathrm{I}_4}$. Hence, H is not better than G and I. We then proceed to determine the better out of the remaining two. At $i = 5$, $\mathcal{F}_{\mathrm{G}_5} < \mathcal{F}_{\mathrm{I}_5}$. With this, we can conclude that the order of performance of the three $M(6,6,0)$ schemes is I, G and H, from best to worst. Following the same algorithm, the order of performance of the four $M(8,8,0)$ schemes is K, L, J and M, from best to worst. However, J and M should have overlapping performances, because all their probabilities are approximately equal.

Table 1 Comparisons of other PC mappings

Code	$\mathcal{F}_{Y_2}$	$\mathcal{F}_{Y_3}$	$\mathcal{F}_{Y_4}$	$\mathcal{F}_{Y_5}$	$\mathcal{F}_{Y_6}$	$\mathcal{F}_{Y_7}$	$\mathcal{F}_{Y_8}$
G	0.0794	0.0635	0.2222	0.02222	0.4127	–	–
H	0.0794	0.0794	0.1825	0.2540	0.4048	–	–
I	0.0794	0.0635	0.2381	0.1905	0.4286	–	–
J	0.0235	0.0157	0.0549	0.0627	0.1176	0.1098	0.6157
K	0.0157	0	0.0863	0	0.2667	0	0.6314
L	0.0157	0.0078	0.0471	0.0471	0.1569	0.1961	0.5294
M	0.0235	0.0157	0.0627	0.0627	0.1020	0.1098	0.6235

6 Simulation Setup and Results

6.1 Setup

Schemes A to F are simulated for the purpose of comparisons, using the setup presented in Fig. 2. Three noise models, namely AWGN, NBI and IN are used to represent the channel conditions in our evaluation. For NBI, various NBI probabilities, P (i.e., 1/32, 1/16, 1/8 and 1) are used, using the simplified NBI model described in [2]. Likewise, for IN, a parameter $\Gamma = 0.1$, which represents the ratio between the IN noise power spectrum density and that of AWGN have been considered, using the concept of the IN model described in [1]. For all these schemes, the effective ratio of their coding rates is $R_{\mathrm{A}} : R_{\mathrm{B}} : R_{\mathrm{C}} : R_{\mathrm{D}} : R_{\mathrm{E}} : R_{\mathrm{F}} = 1.5 : 1 : 1 : 1 : 1 : 1$. This has been put into consideration in the E_b/N_o computations, to ensure fair comparisons. In order to validate our claim in the *HDP decision algorithm for large HDs*, we also simulated the codebooks G to I and J to M under an AWGN+IN+NBI channel.

6.2 Results

The results of the simulated Schemes A to F, in the presence of only AWGN, are shown in Fig. 3. At $E_b/N_o >$ 7 dB, Scheme A has better performance than C, D, E and F, while B has the best performance all through. At BER of 2.5×10^{-4}, Scheme B has about 1.0 dB gain over A, 4.5 dB over C and D, 2.5 dB over E, and 5 dB over F. This is due to the optimized codewords used in Scheme B, coupled with the advantage inherent in the DQuiPSK algorithm [21]. Scheme F outperforms E, due to the DQPSK modulator used, which is naturally better than D8PSK, under AWGN. At $E_b/N_o < 12$, Scheme E has relatively the same performance as C and D, before getting slightly worse. Since the channel considered here is AWGN, which is purely a random distribution, the strength of the proposed HPC is not apparent, until we include other PLC channel noise, such as IN and NBI.

The curves shown in Figs. 4 and 5, are the results obtained when a combined AWGN+NBI and a combined AWGN+IN+NBI, are respectively considered, where P is the NBI parameter. One can notice the error floors in the curves due to the ever-presentIN and NBI contributions, while the E_b/N_o values for the AWGN is made variable. On average, Schemes C and D exhibit the same but better performances than A, B, E and F, with B having the best performance only at low E_b/N_o values. Scheme E now greatly outperforms F. As such, the purpose of inventing the HPC scheme, which is to assist the conventional PC schemes (as in B), at high E_b/N_o values, can be said to be accomplished. The effect of different HD profiles for Schemes C and F becomes obvious at high E_b/N_o values, where C outperforms F. Also, for all the schemes, their performances get worse as the NBI probability increases.

Confusion rates for the five PC decoders are displayed in Fig. 6. Due to RSEs and FSEs, Schemes B to F have different confusion patterns. FSEs may not yield as much confusions as RSEs, unless the codebooks have a very weak $d_{\min}$. Because the number of constellation points is constrained to M in Scheme B, it is only prone to RSEs, while C to F are open to mixtures of RSEs and FSEs. Hence, at high E_b/N_o, Schemes C, D and E have slimmer chances of having decoding errors than B. This is because the small amount of possible errors are distributed between RSEs and FSEs, while the errors in Scheme B are only RSEs. Scheme F has the worst confusion pattern due to its poor HD profile.

Fig. 7 compares the injection code (i.e., Scheme C) and the HPC mapping (i.e., Scheme D), when only one frequency is disturbed with an NBI probability of 1, in the presence of AWGN and IN. This was done with the NBI present on a different frequency position each time. Since all positions exhibit the same pattern of performance at various NBI positions, this is an indication that these two mappings attain the optimality requirement presented in [29]. Also, the essence of the HPC scheme is further strenghtened in this result, due to the fact that it has a $d_{\min}$ of 3, but it has overlapping performance with that of Scheme D whose $d_{\min}$ is 4, at all NBI positions. This is because HPC makes the contribution of $d_{\min}$ to be negligible, as shown in Propositions 1 and 2.

The results of the simulated Schemes G to I and J to M are presented in Figs. 8 to 10, respectively. According to Fig. 8, Scheme H performs worse than G and I, with I overlapping with G. Why these two curves have relatively similar performances is because of their $\mathcal{F}_{Y_6}$, which are relatively similar, despite the fact that their performances are theoretically judged by their $\mathcal{F}_{Y_5}$. However, the dissimilarity between Scheme G and the other two is not well pronounced, due to the RS code concatenated with the schemes. This helps to neutralize some of the errors that PC is unable to correct. In order to make the dissimilarity a little more pronounced, Fig. 9 shows the three schemes without the RS code, under an AWGN channel. Also, from Fig.

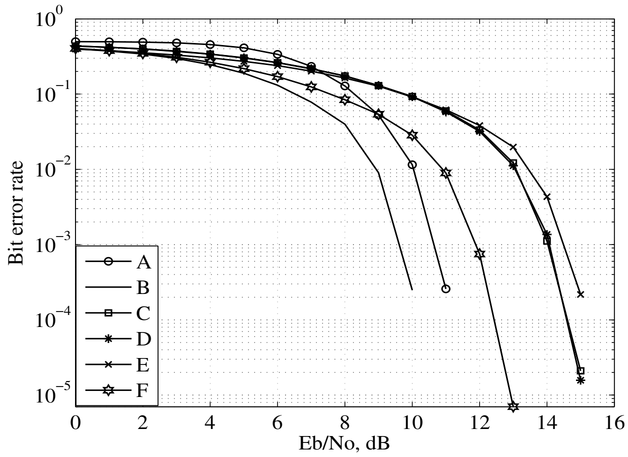

Fig. 3 Bit error rate curve for Schemes A to F, in the presence of AWGN

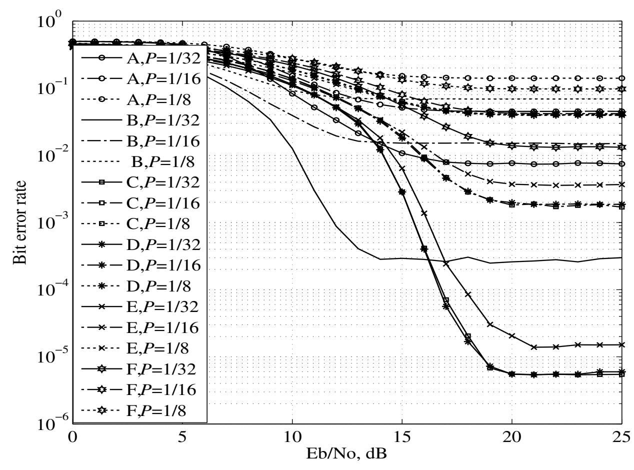

Fig. 4 Bit error rate curve for Schemes A to F, in the presence of AWGN+NBI

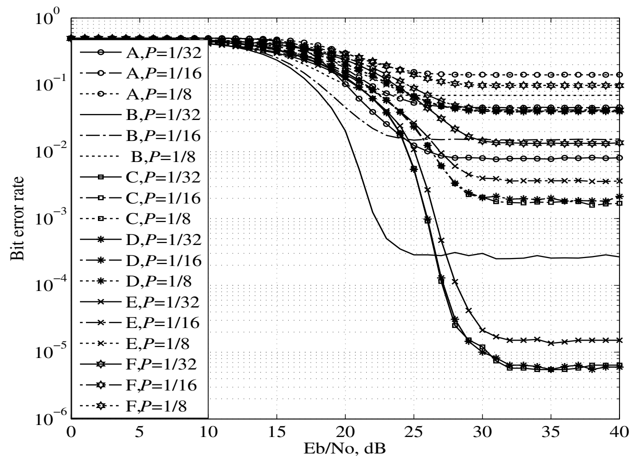

Fig. 5 Bit error rate curve for Schemes A to F, in the presence of AWGN+IN+NBI

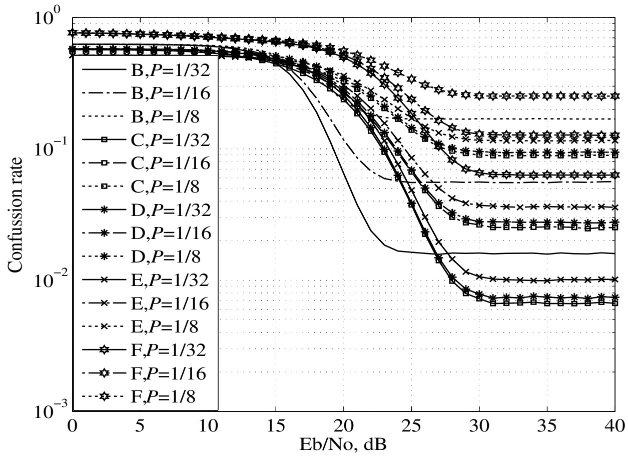

Fig. 6 Confusion rate curve for Schemes B to F, in the presence of AWGN+IN+NBI

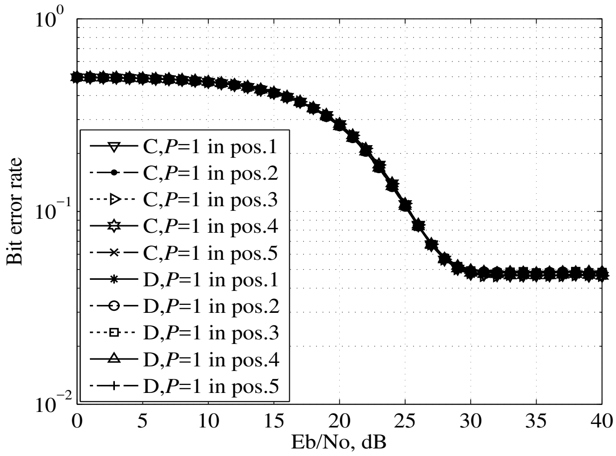

Fig. 7 Single NBI in different positions for Schemes C and D, in the presence of AWGN+IN

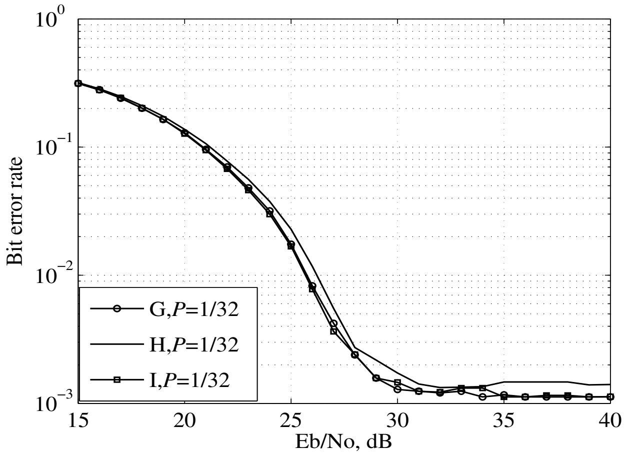

Fig. 8 Bit error rate curve for Schemes G to I, in the presence of AWGN+IN+NBI

10, Scheme K is the best performing scheme, followed by L, while J and M have relatively overlapping performances. This is in accordance to the judgement of the *HDP decision algorithm for large HDs* discussed in Section 5.

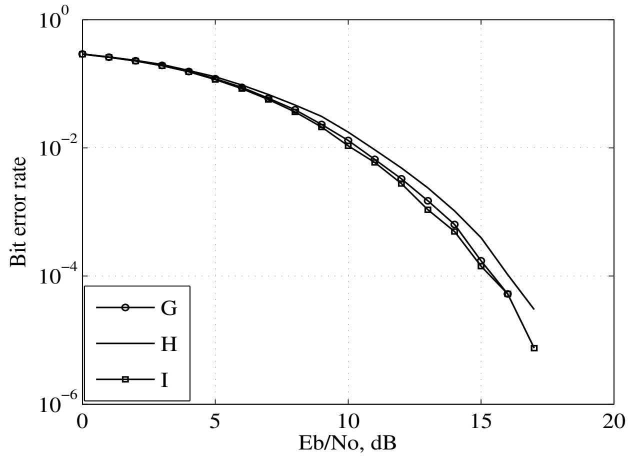

Fig. 9 Bit error rate curve for Schemes G to I, in the presence of AWGN (without RS code)

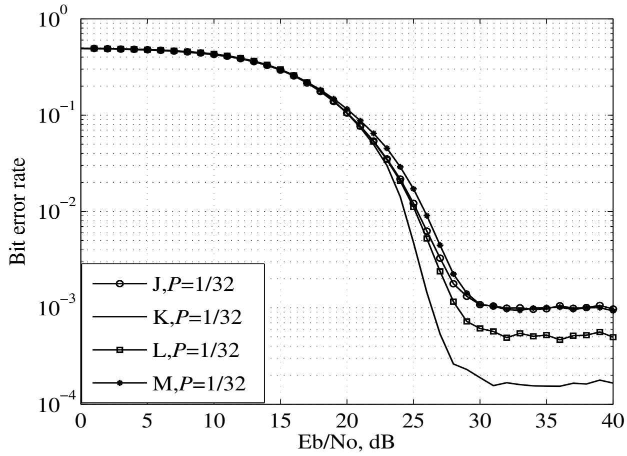

Fig. 10 Bit error rate curve for Schemes J to M, in the presence of AWGN+IN+NBI

On a general note, the proposed HDP tool is a good prediction tool. Despite the fact that the HPC in Scheme D has lower $d_{\min}$ than the injection code in C, we are able to predict their similar performances, by using the HDP tool. Also, despite the fact that G to I and J to M are optimal, according to [29], we are able to explain the disparities in their performances, using the information extracted from their respective HDPs.

7 Conclusion

A PC system, called HPC, that helps to improve the performance of an RS-PC-DPSK-OFDM scheme over the conventional RS-CC-DPSK-OFDM (specified in G3-PLC), has been presented, together with a simple PC comparison tool called HDP that computes the contributions of all possible HDs of a given codebook. The HPC scheme works by adaptively mapping the RS coded bits onto PC symbols, composed of multiple classes of permutation codeword structures of the same $d_{\min}$, while still maintaining the same rate, as though only one PC class were used. An advantage of this scheme is the possibility of reducing its decoding complexity by shortening M, as done in Scheme E. Since the algorithms behind this scheme are not as complex as those in the conventional CC, coupled with its better performance, it is a good candidate for implementation in an inexpensive OFDM-based PLC system.

The introduced HDP can be easily used to predict the performance of any given PC, before its actual usage in the desired design. It is however worth noting that the *HDP decision algorithm for large HDs* presented in this work is not precisely a definite approach for deciding the best performing scheme with HDs $> d_5$, but it gives a good idea of the performance of each code. Based on the introduced schemes and HDP tool, we have highlighted how the performances offered by standardized PLC solutions may be largely improved and easily evaluated. To achieve further improvement, combining impulsive noise clipping and PC mapping may be a good consideration.

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

References

1. Ferreira, H.C., Grové, H.M., Hooijen, O., & Vinck, A.J.H. (2001). *Power line communication*. Wiley Encyclopedia of Electrical and Electronics Engineering.
2. Papilaya, V.N., Shongwe, T., Vinck, A.J.H., & Ferreira, H.C. (2012). Selected subcarriers QPSK-OFDM transmission schemes to combat frequency disturbances, In *IEEE International Symposium on Power Line Communications and Its Application* (pp. 200–205).
3. Vinck, A.J.H., & Häring, J. (2000). Coding and modulation for power-line communications, In *IEEE International Symposium on Power Line Communications and Its Application* (pp. 265–271).
4. CENELEC (1992). *50065 part 1: Signalling on low voltage electrical installations in the frequency range 3 kHz to 148.5 kHz, general requirments, frequency bands and electromagnetic disturbances.*
5. _ (2008). *PRIME Technology, whitepaper: PHY, MAC and Convergence layers*, 21st Ed.
6. eRDF (2013). *PLC G3 Physical Layer Specification.*
7. Juwono, F.H., Guo, Q., Huang, D., & Wong, K.P. (2014). Deep clipping for impulsive noise mitigation in OFDM-based power-line communications. *IEEE Transactions on Power Delivery*, 29(3), 1335–1343.
8. Juwono, F.H., Guo, Q., Huang, D., & Wong, K.P (2013). Joint peak amplitude and impulsive noise clippings in OFDM-based power line communications. *Asia-Pacific Conference on Communications* (pp. 567–571).
9. Al-Mawali, K.S., & Hussain, Z.M. (2009). Adaptive-threshold clipping for impulsive noise reduction in OFDM-based power line communications. *International Conference on Advanced Technologies for Communications* (pp. 43–48).
10. Salehi, V., Mohamed, A., Mazloomzadeh, A., & Mohammed, O.A. (2012). Laboratory-based smart power system, part II: Control, monitoring, and protection. *IEEE Transactions on Smart Grid*, 3(3), 1405–1417.
11. Zhang, P., Li, F., & Bhatt, N. (2010). Next-generation monitoring, analysis, and control for the future smart control center. *IEEE Transactions on Smart Grid*, 1(2), 186–192.
12. Cheng, L., & Ferreira, H.C. (2012). Time-diversity permutation coding scheme for narrow-band power-line channels. In *IEEE International Symposium on Power Line Communications and Its Application* (pp. 120–125).
13. Cheng, L., Swart, T.G, & Ferreira, H.C. Adaptive rateless permutation coding scheme for OFDM-based PLC. (2013). In *IEEE International Symposium on Power Line Communications and Its Application* (pp. 242–246).
14. Swart, T.G., & Ferreira, H.C. (2007). Decoding distance-preserving permutation codes for power-line communications. In *IEEE AFRICON* (pp. 1–7).
15. Ferreira, H.C., Vinck, A.J.H., Swart, T.G., & de Beer, I. (2005) Permutation trellis codes. *IEEE Transactions on Communications*, 53(11), 1782–1789.
16. Dukes, P.J. (2012). Coding with injections. *Designs, Codes and Cryptography*, 65(3), 213–222.
17. Huczynska, S., & Mullen, G.L. (2006). Frequency permutation arrays. *Journal of Combinatorial Designs*, 14(6), 463–478.
18. Hunt, F.H., Perkins, S., & Smith, D.H. (2015). Decoding mixed errors and erasures in permutation codes. *Designs, Codes and Cryptography*, 74(2), 481–493.
19. Bailey, R.F. (2009). Error-correcting codes from permutation groups. *Discrete Mathematics*, 309(13), 4253–4265.
20. Chee, Y.M., Kiah, H.M., Purkayastha, P., & Wang, C. (2012). Importance of symbol equity in coded modulation for power line communications. In *IEEE International Symposium on Information Theory* (pp. 661–665).
21. Ogunyanda, K., Familua, A.D., Swart, T.G., Ferreira, H.C., & Cheng, L. (2014). Permutation coding with differential quinary phase shift keying for power line communication. In *IEEE PES Innovative Smart Grid Technology European Conference* (pp. 1–6).
22. Barta, J., Montemanni, R., & Smith, D.H. (2014). A branch and bound approach to permutation codes. In *IEEE International Conference on Infromation and Communication Technology* (pp. 187–192).
23. Vinck, A.J.H. (2000). Coded Modulation for Power Line Communications. *AEU International Journal of Electronics and Communications*, 54(1), 45–49.
24. Deza, M., & Vanstone, S.A. (1978). Bounds for permutation arrays. *Journal of Statistical Planning and Inference*, 2(2), 197–209.
25. Wadayama, T., & Hagiwara, M. (2012). LP-decodable permutation codes based on linearly constrained permutation matrices. *IEEE Transactions on Information Theory*, 58(8), 5454–5470.
26. Kong, J., & Hagiwara, M. (2012). Comparing Euclidean, Kendall tau metrics toward extending LP decoding. In *IEEE International Symposium on Information Theory and its Applications* (pp. 91–95).
27. Klove, T., Lin, T.T., Tsai, S.C., & Tzeng, W.G. (2010). Permutation arrays under the Chebyshev distance. *IEEE Trans. Informat. Theory*, 56(6), 2611–2617.
28. Gologlu, F., Lember, J., Riet, A.E., & Skachek, V. (2015). New bounds for permutation codes in Ulam metric. In *IEEE International Symposium on Information Theory and its Applications* (pp. 1726-1730).

29. Swart, T.G. (2006). Distance-preserving mappings and trellis codes with permutation sequences. Ph.D. dissertation, University of Johannesburg, Johannesburg, South Africa.
30. Swart, T.G., de Beer, I., & Ferreira, H.C. (2005). On the distance optimality of permutation mappings. In *IEEE International Symposium on Information Theory* (pp. 1068–1072).
31. Sklar, B., & Grant, P.M. (1988). *Digital Communications: Fundamentals and Applications*. Prentice Hall.
32. Alexandre, G.A., Guido, M., & Sergio, B. (2001). A new approach to the construction of high-rate convolutional codes. *IEEE Communications Letters*, 5(11), 453–455.
33. Ucha-Filho, B.F., Souza, R.D., Pimentel, C., & Jar, M. (2006). Further results on convolutional codes based on a minimal trellis complexity measure. In *IEEE International Telecommunications Symposium* (pp. 123–128).
34. Bian, Y., Popplewell, A., & O'Reilly, J.J. (1994). New very high rate punctured convolutional codes. *Electronics Letters*, 30(14), 1119–1120.
35. Viterbi, A.J. (1971). Convolutional codes and their performance in communication systems. *IEEE Transactions on Communications Technology*, 19(5), 751–772.
36. Guftaar, A.S.S. (2007). Computation of the distance spectrum of convolutional codes and the delay spectrum of a network. Semester Project, Jacobs University Bremen, Bremen, Germany.
37. Bellare, M. (1997). A note on negligible functions. Technical Report CS97-529, University of California, San Diego, CA.
38. MathWiki (2015). The negligible, the noticeable and the overwhelming. http://mathwiki.cs.ut.ee/asymptotics/06_the_negligible_the_noticeable_\and_the_overwhelming
39. Courant, J., Daubignard, M., Ene, C., Lafourcade, P., & Lakhnech, Y. (2011). Automated proofs for asymmetric encryption. *Journal of Automated Reasoning on Computer Security*, 46(3-4), 261–291.
40. Katz, J., & Lindell, Y. (2007). *Introduction to Modern Cryptography: Principles and Protocols*. illustrated (Ed). Taylor & Francis.
41. Riegelman, R. (2004). *Studying a Study and Testing a Test: How to Read the Medical Evidence*. 5th (Ed). Philadelphia, PA: Lippincott Williams & Wilkins.
42. Concept Stew. (2015). Statistics for the terrified: The importance of n (sample size) in statistics. http://www.conceptstew.co.uk/PAGES/nsamplesize.html
43. Johnson R., & Kuby, P. (2011). *Elementary Statistics*, student (Ed). Cengage Learning.
44. Ogunyanda, K., Familua, A.D., Swart, T.G., Ferreira, H.C., & Cheng, L. (2014). Adaptive permutation coded differential OFDM system for power line communications. In *IEEE International Conference on Adaptive Science & Technology* (pp. 1–7).
45. Ogunyanda, K., Familua, A.D., Swart, T.G., Ferreira, H.C., & Cheng, L. (2014). Evaluation and implementation of cyclic permutation coding for power line communications. In *IEEE International Conference on Adaptive Science & Technology* (pp. 1–7).
46. Oh, M., & Sweeney, P. (1999). Low complexity soft-decision sequential decoding using hybrid permutation for Reed-Solomon codes. In *IMA International Conference on Cryptography and Coding* (pp. 163–172).
47. Singh, A. (2010). A hybrid permutation-coded evolutionary algorithm for the early/tardy scheduling problem. *Asia-Pacific Journal of Operational Research*, 27(6), 713–725.