Multiple Access with Distance Preserving Mappings for Permutation Codes

Thokozani Shongwe
Department of Electrical and Electronic Engineering Science,
University of Johannesburg,
P.O. Box 524, Auckland Park, 2006,
Johannesburg, South Africa
Email: tshongwe@uj.ac.za

Theo G. Swart
Department of Electrical and Electronic Engineering Science,
University of Johannesburg,
P.O. Box 524, Auckland Park, 2006,
Johannesburg, South Africa
Email: tgswart@uj.ac.za

Hendrik C. Ferreira
Department of Electrical and Electronic Engineering Science,
University of Johannesburg,
P.O. Box 524, Auckland Park, 2006,
Johannesburg, South Africa
Email: hcferreira@uj.ac.za

***Abstract*—We present results for Distance Preserving Mappings (DPMs) for permutation codes that can be used for multiple access communication even under frequency jamming. We give examples of multiple codebooks that are DPMs such that each DPM can be assigned to a user for communication over a multiple access channel. We only consider one type of DPM called Distance Increasing Mappings (DIMs). The multiple access channel of interest is Time Division Multiple Access (TDMA). We show that it is possible to give a construction for DIMs that can be uniquely decoded even in the presence of frequency jamming. The DPMs are permutation codes found by mapping binary sequences to permutation sequences. The permutation codes have codewords of length M with symbols taken from an alphabet whose cardinality is M, where M is any integer. Each symbol may be seen as representing one out of the M frequencies in an M-ary Frequency Shift keying modulation scheme, for example.**

I. Introduction

A combination of Permutation coding and Multiple Frequency Shift Keying modulation (MFSK) has been shown in [1], [2], [3], [4] to be a robust way of communicating over the power line communications (PLC) channel which is characterised by severe noise such as impulse and narrowband interference. Since the introduction of the idea of combined permutation coding and MFSK for PLC by Vinck [1] in 2000, more research has been undertaken on permutation codes for PLC. Vinck and Häring [2] continued the original work by Vinck [1] and paid attention to the MFSK non-coherent demodulator using a threshold detector. Shum [5] demonstrated that permutation coding exploits the inherent frequency diversity in MFSK to provide robustness in frequency selective fading channels. Papilaya *et al.* [6] also presented a coded modulation scheme for OFDM transmission, where permutation codes were proposed as an alternative to convolutional codes when communicating over a PLC channel with frequency disturbances and frequency selective fading.

Another aspect of permutation coding that has been recently addressed in the literature, is that of synchronization. Cheng *et al.* [7] presented a synchronization coding scheme which uses single insertion/deletion error correcting permutation codes. Cheng *et al.* [8] continued their contribution in the synchronization of permutation codes, where they gave a scheme that can allow for the resynchronization of permutation trellis codes using a Viterbi-like algorithm. Heymann and Ferreira [9] made use of tree structures to resynchronize permutation codes. Their scheme can correct substitution, insertion or deletion errors. Heymann *et al.* [10] continued the work in [9] and showed that adjacent errors can be corrected using permutation code trees. One of the most recent work on synchronization of permutation codes is by Shongwe *et al.* [11] and Heymann *et al.* [12].

Earlier on in the introduction of permutation codes, it was realised that permutation block codes were lacking efficient encoding and decoding methods. To tackle that problem, a method for mapping binary convolutional code outputs to permutation sequences, called Distance Preserving Mapping (DPM), was used in [3] to construct permutation trellis codes. Permutation trellis codes can be decoded using the well known Viterbi algorithm. The DPMs for permutation trellis codes introduced in [3] were further classified, according to Hamming distances, into Distance Conserving Mapping (DCM), Distance Reducing Mapping (DRM) and Distance Increasing Mapping (DIM). Swart and Ferreira [13] continued the work on DPMs for permutation trellis codes and gave a general *mapping algorithm* for creating DPMs for permutation sequences from binary sequences. We give a brief description of the mapping algorithm in Section III-A, a more detailed description can be found in [14].

Another advancement in permutation codes, which is independent of the work on DPMs, is that of using permutation codes for multiple access communication. Balakirsky and Vinck [15] gave a construction for *uniquely decodable* permutation codes for a multiple access OR channel where the inputs originate from a Fast Frequency Hopping / Multiple Frequency Shift Keying modulation (MFSK). In [15], T independent senders were permitted to transmit M-ary codewords of length M and T receivers receive a vector of sets of length M. The i-th receiver tries to determine the message of the i-th sender. The receiver consists of M frequency detectors to identify which symbol is present or absent.

The construction in [15] is termed an (M, T, L, d) scheme, where M is the length of the permutation codeword, T is the number of pairs (sender receiver), L is the number of codewords per sender (user), and $(d-1)$ is the maximum number of jamming signals the code can handle and still allow for unique decodability. The uniquely decodable permutation codes are constructed based on the so-called, *individual entry* which allowed for unique decodability. The individual entry allows a codeword, for the case $L = 1$, to be uniquely identified by having a symbol that is unique to that particular user in a specific time slot. In our previous work [16] which was inspired by the work in [15], we gave a performance comparison of different uniquely decodable permutation codes for multiuser communication. We were interested in the effect of Hamming distances in different codebooks of permutation codes.

The previous work on permutation codes for multiple access in [15] and [16] considered permutation codes as block codes. In this work we consider permutation codes that are constructed by mappings from binary sequences, and we shall loosely call them DPMs. We find *uniquely identifiable* DPMs for multiple access. The multiple access scheme considered in this work is Time Division Multiple Access (TDMA). We give examples of uniquely identifiable DPMs that can be used for multiple users over a multiple access channel (TDMA), and then give one generic construction for the uniquely identifiable DPMs for multiple access. For the DPMs to be uniquely identifiable, we make use of *unique-symbols* such that each user can identify its codewords among codewords from different users. The use of unique-symbols for decoding has similarities to the individual entries introduced in [15]. The unique-symbols are basically used to differentiate codewords from different users, hence allowing the receiver to be able to identify its codewords. Throughout this work, we will use the terminology of an (M, T, L, d) scheme to describe the uniquely identifiable DPMs.

II. System Model

The communications system model is that of data transmission over a multiple access channel (MAC) as shown in Fig. 1, see [15] and [17]. The multiple access channel employs the well known TDMA scheme. The i-th user (sender), out of T users, is given a time slot to send its M-symbol codeword X_i. The i-th user's codeword is taken from one of L codewords in the user's codebook. The output of the MAC is an M-symbol codeword Y which is common to all receivers. The i-th receiver compares Y with its codewords X_i. In the presence of channel noise, it is possible that Y is decoded wrongly. The channel noise we focus on is impulse noise and frequency disturbances. Assuming that an impulse lasts for the duration of a symbol, the effect of impulse noise is that it sends energy on all frequencies in a time instant. The effect of a single frequency disturbance is that it sends energy on one frequency for the duration of a codeword.

To illustrate the effect of impulse noise and frequency disturbance, let us take the transmitted codeword to be $X_i = \{3, 1, 4, 2\}$. After passing through a channel with impulse

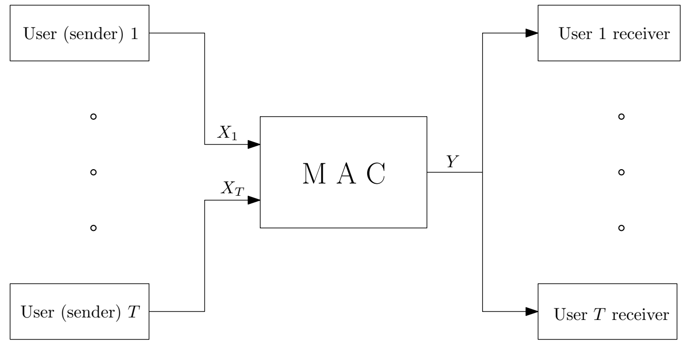

Fig. 1. Multiple access communications system model

noise and/or frequency disturbance, X_i is received as Y. We adopt the noise model described in [1]:

- we consider an impulse duration equal to the symbol time slot, where a single impulse occurs in a symbol time slot. The effect of an impulse at time slot 2 produces a $Y_{\text{impulse}} = \{(3), (1, 2, 3, 4), (4), (2)\}$.
- the effect of a frequency disturbance on frequency 3 results in $Y_{\text{freq. disturbance}} = \{(3), (1, 3), (4, 3), (2, 3)\}$. The frequency disturbance is assumed to last for at least the duration of a codeword transmission.

We consider either impulse noise or frequency disturbance as jamming, because the receiver deals with them in a similar manner. When an impulse is detected, the position where it occurred can be marked as an erasure and minimum distance decoding will be performed. Also, when a frequency disturbance has occurred, the detected frequency that is disturbed is marked as an erasure and minimum distance decoding is performed.

III. DPMs for Multiple Access

We start this section by giving a brief description of the algorithm for mapping binary sequences to permutation sequences such that we get Distance Preserving Mappings. Then we give DIMs for which multiple access communication is possible for a TDMA scheme.

A. Mapping Algorithm

To illustrate the mapping algorithm for creating DPMs we use the following example taken from [14]. Given a binary sequence $\mathcal{X} = (x_1, x_2, x_3, x_4, x_5, x_6)$, of length $n = 6$, being mapped to the permutation sequence $\mathcal{Y} =$

$(y_1, y_2, y_3, y_4, y_5, y_6)$, of length $M = 6$.

Input: $(x_1, x_2, x_3, x_4, x_5, x_6)$
Output: $(y_1, y_2, y_3, y_4, y_5, y_6)$
begin
  $(y_1, y_2, y_3, y_4, y_5, y_6) \leftarrow (1, 2, 3, 4, 5, 6)$
  if $x_3 = 1$ then swap(y_1, y_2)
  if $x_2 = 1$ then swap(y_3, y_4)
  if $x_1 = 1$ then swap(y_5, y_6)
  if $x_5 = 1$ then swap(y_1, y_3)
  if $x_4 = 1$ then swap(y_2, y_4)
  if $x_6 = 1$ then swap$(y_1, y_5)(y_2, y_6)$
end.

The algorithm takes in a binary sequence $\mathcal{X}$, of length n and produces a permutation of the sequence $\mathcal{Y}$, by swapping the positions of the elements of the sequence $\mathcal{Y}$ according to the bit position of $\mathcal{X}$ that is a 1. For example, let $\mathcal{X} = (100010)$ and $\mathcal{Y} = (123456)$, then the output of the algorithm will be a sequence (321465), where positions 5 and 6 are swapped, and positions 1 and 3 are swapped in $\mathcal{Y}$. This is done for all the 2^n binary sequences, producing 2^n permutation sequences of length M. Note that permutation sequence, 123456 is the basis for all the swaps and it corresponds to the all-zero binary sequence $\mathcal{X} = (00 \dots 0)$. We shall call this permutation the "seed-permutation". The result of the mapping algorithm is such that the corresponding Hamming distances between the binary sequences and between the permutation sequences is preserved, hence the term Distance Preserving Mapping. The three types of DPMs presented in [3], DCMs, DRMs and DIMs, were so named according to the corresponding Hamming distances between the binary sequences and the permutation sequences. Corresponding Hamming distances between permutation and binary sequences: is kept the same–DCM, is reduced–DRM and is increased–DIM. Formally, a mapping is represented as $\mathcal{M}(n, M, \delta)$, where n is the length of the binary sequence, $\mathcal{X}$, M is length of the permutation sequence, $\mathcal{Y}$. $\delta = 0$ indicates a Distance Conserving Mapping (DCM), $\delta < 0$ indicates a Distance Reducing Mapping (DRM) and $\delta > 0$ indicates a Distance Increasing Mapping (DIM).

B. Uniquely Identifiable DIMs

We give a group of DIMs which can be used together for multiple access communications under a TDMA scheme. We call this group of DIMs for multiple access, uniquely identifiable DIMs, and they are characterised by the parameters M, T, L, d as defined in Section I. For DPMs, $\mathcal{M}(n, M, \delta)$, $L = 2^n$ for each user codebook. Next we present results for uniquely identifiable DIMs in Tables I–III. Note that the first permutation sequence in each DIM is the seed-permutation. The underlined symbols (unique symbols) are responsible for uniquely identifying codewords from each DIM.

The DIMs in Table I were generated using the following combination of swaps. $x_1 = 1$ then swap $(y_3,\ y_4)$, $x_2 = 1$ then swap $(y_1,\ y_2)$, $x_3 = 1$ then swap $(y_2,\ y_4)$. For DIM 1, the seed-permutation is $(y_1, y_2, y_3, y_4) \leftarrow (1, 2, 3, 4)$. For DIM 2, the seed-permutation is $(y_1, y_2, y_3, y_4) \leftarrow (4, 3, 2, 1)$. The last two symbols of the seed-permutations for DIM 1 and DIM 2 were chosen such that they are different in order to have the uniquely decodable code in Table I.

TABLE I
UNIQUELY DECODABLE CODE FOR
$\mathcal{M}(3, 4, 1) : M = 4,\ T = 2,\ L = 8,\ d = 1$, FOR TDMA

DIM 1	DIM 2
1 2 **3** 4	4 3 **2** 1
1 4 **3** 2	4 1 **2** 3
2 1 **3** 4	3 4 **2** 1
4 1 **3** 2	1 4 **2** 3
1 2 **4** 3	4 3 **1** 2
1 3 **4** 2	4 2 **1** 3
2 1 **4** 3	3 4 **1** 2
3 1 **4** 2	2 4 **1** 3

TABLE II
UNIQUELY DECODABLE CODE FOR
$\mathcal{M}(4, 5, 1) : M = 5,\ T = 3,\ L = 16,\ d = 1$, FOR TDMA

DIM 1	DIM 2	DIM 3
1 2 **3** 4 5	1 2 **4** 3 5	1 2 **5** 3 4
1 2 **3** 5 4	1 2 **4** 5 3	1 2 **5** 4 3
3 2 1 4 5	**4** 2 1 3 5	**5** 2 1 3 4
3 2 1 5 4	**4** 2 1 5 3	**5** 2 1 4 3
1 **4** **3** 2 5	1 3 **4** 2 5	1 3 **5** 2 4
1 **4** **3** 5 2	1 3 **4** 5 2	1 3 **5** 4 2
3 **4** 1 2 5	**4** 3 1 2 5	**5** 3 1 2 4
3 **4** 1 5 2	**4** 3 1 5 2	**5** 3 1 4 2
2 1 **3** 4 5	2 1 **4** 3 5	2 1 **5** 3 4
2 1 **3** 5 4	2 1 **4** 5 3	2 1 **5** 4 3
3 1 2 4 5	**4** 1 2 3 5	**5** 1 2 3 4
3 1 2 5 4	**4** 1 2 5 3	**5** 1 2 4 3
2 **4** **3** 1 5	2 3 **4** 1 5	2 3 **5** 1 4
2 **4** **3** 5 1	2 3 **4** 5 1	2 3 **5** 4 1
3 **4** 2 1 5	**4** 3 2 1 5	**5** 3 2 1 4
3 **4** 2 5 1	**4** 3 2 5 1	**5** 3 2 4 1

The DIMs in Table II were generated using the following combination of swaps. $x_1 = 1$ then swap $(y_1,\ y_2)$, $x_2 = 1$ then swap $(y_2,\ y_4)$, $x_3 = 1$ then swap $(y_1,\ y_3)$, $x_4 = 1$ then swap $(y_4,\ y_5)$. For DIM 1, the seed-permutation is $(y_1, y_2, y_3, y_4, y_5) \leftarrow (1, 2, 3, 4, 5)$. For DIM 2, the seed-permutation is $(y_1, y_2, y_3, y_4, y_5) \leftarrow (1, 2, 4, 3, 5)$. For DIM 3, the seed-permutation is $(y_1, y_2, y_3, y_4, y_5) \leftarrow (1, 2, 5, 3, 4)$. The last three symbols of the seed-permutations for DIM 1, DIM 2 and DIM 3 were permuted to get the seed-permutations of each DIM. It can be seen that the middle symbol of the seed-permutation of each DIM is responsible for making the code in Table II uniquely decodable.

The DIMs in Table III were generated using the same combination of swaps as used for the DIMs in Table II. $x_1 = 1$ then swap $(y_1,\ y_2)$, $x_2 = 1$ then swap $(y_2,\ y_4)$, $x_3 = 1$ then swap $(y_1,\ y_3)$, $x_4 = 1$ then swap $(y_4,\ y_5)$. For DIM 1, the seed-permutation is $(y_1, y_2, y_3, y_4, y_5) \leftarrow (1, 2, 3, 4, 5)$. For DIM

TABLE III
UNIQUELY DECODABLE CODE FOR
$\mathcal{M}(4,5,1): M=5,\ T=2,\ L=16,\ d=2$, FOR TDMA

DIM 1	DIM 2
1 2 **3** **4** 5	1 2 **4** **3** 5
1 2 **3** 5 **4**	1 2 **4** 5 **3**
3 2 1 **4** 5	**4** 2 1 **3** 5
3 2 1 5 **4**	**4** 2 1 5 **3**
1 **4** **3** 2 5	1 **3** **4** 2 5
1 **4** **3** 5 2	1 **3** **4** 5 2
3 **4** 1 2 5	**4** **3** 1 2 5
3 **4** 1 5 2	**4** **3** 1 5 2
2 1 **3** **4** 5	2 1 **4** **3** 5
2 1 **3** 5 **4**	2 1 **4** 5 **3**
3 1 2 **4** 5	**4** 1 2 **3** 5
3 1 2 5 **4**	**4** 1 2 5 **3**
2 **4** **3** 1 5	2 **3** **4** 1 5
2 **4** **3** 5 1	2 **3** **4** 5 1
3 **4** 2 1 5	**4** **3** 2 1 5
3 **4** 2 5 1	**4** **3** 2 5 1

2, the seed-permutation is $(y_1, y_2, y_3, y_4, y_5) \leftarrow (1, 2, 4, 3, 5)$. DIM 1 and DIM 2 in Table III are the same as DIM 1 and DIM 2 in Table II, respectively. The only difference in Table III is that the absence of DIM 3 makes the code have $d = 2$.

To obtain the results in I–III, we searched for combinations of swaps that will work with the chosen seed-permutations such that we find uniquely identifiable DIMs. However, not all combinations of swaps produce uniquely identifiable DIMs. So we performed a trial-and-error search for combinations swaps and seed-permutations that will give us uniquely identifiable DIMs. The first step was to pick any combination of swaps and any seed-permutation that result into a DIM. Then observe the arrangement of the symbols in the entire DIM in relation to the seed-permutation symbols arrangement. We found that when creating a DIM, if a combination of swaps arranged the symbols of the seed-permutation such that in one column, only a few of the $\{1, 2 \ldots M\}$ symbols are present, then that combination of swaps may be tried to generate more DIMs such that we form uniquely identifiable DIMs.

We also found that for a given M, if we perform the following order of swaps, $x_i = 1$ then swap $(i,\ i+1)$, where $1 \le i < M$, and keep changing the seed-permutation for each DIM, we can find uniquely identifiable DIMs for the TDMA scheme. We give an example to illustrate this point.

Example 1: $\mathcal{X} = (x_1, x_2, x_3)$.
$\mathcal{Y} = (y_1, y_2, y_3, y_4)$.
$\mathcal{M}(3,4,1)$: $x_1 = 1$ then swap $(y_1,\ y_2)$, $x_2 = 1$ then swap $(y_2,\ y_3)$, $x_3 = 1$ then swap $(y_3,\ y_4)$
For DIM 1: $(y_1, y_2, y_3, y_4) \leftarrow (4, 3, 2, 1)$. For DIM 2: $(y_1, y_2, y_3, y_4) \leftarrow (1, 2, 3, 4)$.

It can be seen that the first two symbols of the seed-permutations form the unique symbols for each DIM, which allows the DIMs to be uniquely identifiable. There can only be $T \le 2$ users, and to achieve $T = 2$, the condition will be to make sure that the first two symbols for each seed-permutation are not the same. There are only two seed-

TABLE IV
UNIQUELY DECODABLE CODE FOR
$\mathcal{M}(3,4,1): M=4,\ T=2,\ L=8,\ d=1$, FOR TDMA

DIM 1	DIM 2
4 3 2 1	**1** 2 3 4
4 3 1 2	**1** 2 4 3
4 2 3 1	**1** 3 2 4
4 2 1 3	**1** 3 4 2
3 4 2 1	**2** 1 4 3
3 4 1 2	**2** 1 3 4
3 2 4 1	**2** 3 1 4
3 2 1 4	**2** 3 4 1

permutations meeting this condition, hence $T \le 2$.

IV. CONCLUSION

Distance Preserving Mappings (DPMs) generated by mapping binary sequences to permutation sequences were discussed. We focused on one type of DPM, DIM, and showed that for a TDMA scheme, it is possible to have multiple DIM codebooks that can be uniquely identified. The uniquely identifiable DIMs can be used as codebooks for multiple users wishing to communicate over a multiple access channel. Lastly, we gave a general simple construction for uniquely identifiable DIMs for a TDMA scheme.

REFERENCES

[1] A. J. H. Vinck, "Coded modulation for powerline communications," *AEÜ International Journal of Electronics and Communications*, vol. 54, no. 1, pp. 45–49, Jan. 2000.

[2] A. J. H. Vinck and J. Häring, "Coding and modulation for power-line communications," in *Proceedings of the 2000 International Symposium on Power Line Communications*, Limerick, Ireland, Apr. 5–7, 2000, pp. 265–272.

[3] 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, USA, Sept. 24–28, 2000, pp. 2401–2407.

[4] H. C. Ferreira, A. J. H. Vinck, T. G. Swart, and I. de Beer, "Permutation trellis codes," *IEEE Transactions on Communications*, vol. 53, no. 11, pp. 1782–1789, Nov. 2005.

[5] K. W. Shum, "Permutation coding and M-FSK modulation for frequency selective channel," in *Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Lisbon, Portugal, Sept. 15–18, 2002, pp. 2063–2066.

[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, China, Mar. 27–30, 2012, pp. 200–205.

[7] L. Cheng, T. G. Swart, and H. C. Ferreira, "Synchronization using insertion/deletion correcting permutation codes," in *Proceedings of the 2008 IEEE International Symposium on Power Line Communications and its Applications*, Jeju Island, Korea, Apr.2–4 2008, pp. 135–140.

[8] ——, "Re-synchronization of permutation codes with viterbi-like decoding," in *Proceedings of the 2009 IEEE International Symposium on Power Line Communications and its Applications*, Dresden, Germany, Mar. 29–Apr. 1, 2009, pp. 36–40.

[9] R. Heymann and H. C. Ferreira, "Using tree structures to resynchronize permutation codes," in *Proceedings of the 2010 IEEE International Symposium on Power Line Communications and its Applications*, Rio de Janeiro, Brazil, Mar.28–31 2010, pp. 108–113.

[10] R. Heymann, T. G. Swart, and H. C. Ferreira, "Correcting adjacent errors using permutation code trees," in *Proceedings of the IEEE Africon*, Livingstone, Zambia, Sept.13–15 2011, pp. 1–5.

[11] T. Shongwe, T. G. Swart, H. C. Ferreira, and T. van Trung, "Good synchronization sequences for permutation codes," *IEEE Transactions on Communications*, vol. 60, no. 5, pp. 1204–1208, May 2012.

[12] R. Heymann, H. C. Ferreira, and T. G. Swart, "Combined permutation codes for synchronization," in *Proceedings of the Information Theory and its Applications (ISITA)*, Honolulu, Hawaii, USA, Oct.28–31 2012, pp. 230–234.

[13] T. G. Swart and H. C. Ferreira, "A multilevel construction for mappings from binary sequences to permutation sequences," in *Proceedings of the 2006 IEEE International Symposium on Information Theory*, Seattle, WA, USA, July 9–14, 2006, pp. 1895–1899.

[14] T. G. Swart, "Distance-preserving mappings and trellis codes with permutation sequences," Ph.D. dissertation, University of Johannesburg, Johannesburg, South Africa, Aug. 2006.

[15] V. B. Balakirsky and A. J. H. Vinck, "On the performance of permutation codes for multi-user communication," *Problems of Information Transmission*, vol. 39, no. 3, pp. 239–254, 2003.

[16] T. Shongwe, U. Speidel, T. G. Swart, and H. C. Ferreira, "The effect of hamming distances on permutation codes for multiuser communication in the power line communications channel," in *Proceedings of the IEEE Africon*, Livingstone, Zambia, Sept.13–15 2011, pp. 1–5.

[17] S. Chang and J. K. Wolf, "On the T-user M-frequency noiseless multiple access channel with and without intensity information," *IEEE Transactions on Information Theory*, vol. 27, no. 1, pp. 41–48, 1981.