

RESEARCH

Open Access

# Delay tolerant (delto) distributed TAST codes for cooperative wireless networks

Zahoor Ahmed<sup>1\*</sup>, JP Cances<sup>2</sup> and V Meghdadi<sup>2</sup>

## Abstract

In a distributed cooperative communication system, as the distances between different relay nodes and the receiving nodes may be different, so the performances of distributed space time codes at receiving nodes may badly be degraded if timing synchronization is not assured. In this article, extending the work of Damen et al. we introduce the design of distributed threaded algebraic space-time (TAST) codes offering resistance to timing delay off-set. We present some new and useful techniques of constructing delay tolerant TAST code for distributed cooperative networks, which, like their brethren codes, are delay tolerant for any delay profile and achieve full diversity for arbitrary number of relays, transmit/receive antennas, and input alphabet size. Our proposed codes with minimum lengths achieve better performances than the existing codes retaining full rate and full diversity with or without use of guard bands. Simulations results confirm our claim of obtaining better performances.

**Keywords:** space-time coding system, space-time block codes, threaded algebraic space-time (TAST) code, timing offset, asynchronous transmission, cooperative relay networks, delay tolerant TAST codes.

## 1. Introduction

Wireless communication systems with multiple antennas have recently attracted considerable interests [1-3]. Perhaps, the reason is that the performance of a wireless system is often limited by fading and may significantly be improved by exploiting some sort of diversity, for example spatial diversity. But on other hand equipping the pocket size mobile handsets with additional radio-frequency (RF) hardware is not feasible. Therefore, thinking for alternate options, many researchers have proposed different solutions and proposals.

Sendonaris et al. [4] proposed the idea of cooperative diversity which enables the source/destination to use nearby nodes as virtual antennas. In other words, the nearby relay nodes may act as auxiliary receivers/transmitters for the original transmitter/receiver. But, again the main problem with cooperative terminals as indicated by Li and Xia [5,6] is the asynchronous nature of transmission which forces the traditionally designed space-time codes to lose their diversity and coding gain when used over distributed cooperative networks.

In fact in an unsynchronized cooperative network, the data from different relays reach the destination after different delays. In [7], it was shown that all the well-known codes lose their diversity at receiver. Mei et al. [8] proposed a technique of using of guard bands between blocks of symbols. The proposed scheme in [8] could achieve full-diversity but the main drawback of this technique is its limitation in the number of relays (only two are allowed) and a huge rate loss due to the insertion of guard bands.

The proposed delay tolerant codes for asynchronous cooperative network of Li and Xia [5,6] were further generalized and refined in [9] by including full-diversity delay tolerant space-time trellis codes of minimum constrained length. In [7], delay tolerant distributed space-time block codes based on threaded algebraic space-time (TAST) codes [10] are designed for unsynchronized cooperative network. The distributed TAST codes of [7] preserve the rank of the space-time codewords under arbitrary delays at the receiver. In a similar way, the authors in [7] further extend their study in [11] by introducing delay tolerant codes with minimum lengths. A lattice-based maximum likelihood detector is used for decoding, which is computationally more complex than the decoupled decoding of orthogonal space-time block codes.

\* Correspondence: Zahoor@buetk.edu.pk

<sup>1</sup>Balochistan University of Engineering and Technology, Khuzdar, Pakistan  
Full list of author information is available at the end of the article

Following the framework of [7,11], in this article we introduce some new and useful techniques for designing delay tolerant TAST codes. The proposed codes achieve full diversity and are flexible with respect to signalling constellation, transmission rate, and number of transmit/receive antennas. Our proposed codes achieve better performances over the existing codes particularly at high SNR. For ease to write, hereafter, we will use delto code as short form of **delay tolerant code**.

The rest of the article is organized as follows. The background and system model are given in Section 2. Some construction techniques of delay tolerant space-time codes are discussed in Section 3. Multiple threads packing are given in Section 4. Section 5 elaborates the construction methods for delto codes with minimum lengths. Some construction examples are presented in Section 6. Simulation results are given in Section 7 and the conclusion is given in Section 8.

## 2. Preliminaries

### 2.1. System model

In a cooperative communication system, the communication between source and destination is modelled in two phases.

In phase-I, the source sends information to the destination and at the same time this information is also received by the relays.

In phase-II, the relays help the source by forwarding or retransmitting the received information to destination.

Initially proposed in [12], the relays use different protocols for processing and re-transmitting the received signal from source to destination. In this article, we consider decode-and-forward processing strategy at the relays.

Since the relay nodes use common time slots and frequency bandwidth for retransmissions of their signals, the relays may expose to overlap both in time and frequency, i.e. each node transmits a distinct coded bit stream, the superposition of which forms a space-time code. In what follows, the design and performance analysis of such distributed STBC codes will be our main focus.

We assume the conventional MIMO system, modelled with  $N_t$  transmit antennas corresponding to  $N$  relays and  $N_r$  receive antennas at the destination. At time instant  $t$ , the received signal is expressed in vector notation as

$$\bar{r}_t = \mathbf{H}_t \bar{s}_t + \bar{n}_t \quad (1)$$

where  $\bar{r}_t \in \mathbb{C}^{N_r \times 1}$  is the received vector at time  $t$ ,  $\bar{s}_t \in \mathbb{C}^{N_t \times 1}$  is modulated signal vector transmitted during the  $t$ th symbol interval,  $\mathbf{H}_t \in \mathbb{C}^{N_r \times N_t}$  is the channel

matrix and  $\bar{n}_t \in \mathbb{C}^{N_r \times 1}$  denotes additive white Gaussian noise. The  $N_t \times T$  modulated space-time codeword matrix  $\mathbf{s}$  is transmitted over  $T$  symbol intervals by taking  $\bar{s}_t$  to be the  $t$ th column of  $\mathbf{s}$ . The channel is assumed to be quasi-static, i.e. the channel transfer matrix  $\mathbf{H}_t$  is constant over a codeword interval but is generated randomly and independently from codeword to codeword. We further assume that no error occurs between sources and relays.

The nature of processing strategies at relays greatly impact the code design and decoding complexity, and it is beyond the scope of this article to discuss this issue in detail. In the simplest case, the timing offset from different relays at reception may be avoided by use of guard bands or some sort of timing advance protocols [13].

If the maximum possible timing offset among the different relay's transmissions is  $L'$  symbol intervals and a pad of duration  $L'$  symbols is used by each relay between its coded transmissions, then the different composite space-time codewords never overlap in time. Each space-time codeword can be decoded individually. For short space-time block codes, the significant rate loss induced by the use of fill symbols or guard intervals can be mitigated by allowing the relays to transmit its coded streams one after another [13], but for long block size the code rate loss is an open problem. In what follows, we present the construction of space-time codes which are robust to arbitrary delays without insertion of guard bands.

### 2.2. Delay tolerance of space-time codes

For the sake of completeness we review some notations from [7]. Let  $\mathcal{S}$  be an STBC code with codeword of size  $N_t \times T$ . Assume  $\mathbf{s}_1$  and  $\mathbf{s}_2$  are two distinct codewords of  $\mathcal{S}$ . The diversity order of  $\mathcal{S}$  is the minimum rank of the difference matrix  $\mathbf{s}_1 - \mathbf{s}_2$  over all pairs of distinct codeword in  $\mathcal{S}$ . This condition is referred as rank criterion [14].

For our purposes, the transmitted symbols will finitely be generated from an underlying finite constellation using algebraic number field constructions. Let  $\mathcal{A}$  denote the two-dimensional constellation chosen from  $Z[i]$  or  $Z[j]$ , and let  $F = Q(i)$  or  $Q(j)$  denote the field of complex rational numbers and complex Eisenstein rational numbers, respectively. Let  $F(\theta)$  be an extension field of degree  $[F(\theta): F]$ . Then, the fundamental alphabet for our constructions is given by

$$\Omega = \left\{ x = \sum_{k=0}^{P-1} u_k \theta^k : u_k \in \mathcal{A} \right\} \quad (2)$$

where integer  $P \leq [F(\theta): F]$ .

Each transmitted symbol  $s \in \mathcal{S}$  is from  $\Omega$  or, more generally, is from its image  $f(\Omega)$  under some specified one-to-one mapping:  $f: \Omega \rightarrow \mathcal{C}$ .

A space time code  $\mathcal{S}$  is said to be  $\tau$ -delto for the quasi-synchronous cooperative diversity scenario, if the difference between every non-trivial pair of codewords in  $\mathcal{S}$  retains full rank even though its rows are transmitted with arbitrary delays of duration at most  $\tau$  symbols [15].

If all relays start transmitting all rows of a distributed STBC simultaneously, then different rows will reach destination with different delays  $\delta_i \leq \delta_{max}$ ,  $i \in \{1, 2, \dots, N\}$ . If all relays continuously transmit the rows of different distributed STBC at different blocks, then the data of two consecutively transmitted STBC can be overlapped due to the timing errors. All relays start transmitting the assigned rows of the codeword simultaneously and as the values of the relative delays are unknown, therefore each of them waits for  $\delta_{max}$  time interval after the transmission of the codeword is finished. Due to the delays in the reception, an  $N_t \times T$  transmitted STBC  $s$  is transmitted into an  $N_t \times (T + \delta_{max})$  codeword at the receiver as follows

$$s^\Delta = \begin{bmatrix} \mathbf{0}_{1 \times \delta_1} & s(1, :) & \mathbf{0}_{1 \times (\delta_{max} - \delta_1)} \\ \mathbf{0}_{1 \times \delta_2} & s(2, :) & \mathbf{0}_{1 \times (\delta_{max} - \delta_2)} \\ \vdots & \vdots & \vdots \\ \mathbf{0}_{1 \times \delta_{N_t}} & s(N_t, :) & \mathbf{0}_{1 \times (\delta_{max} - \delta_{N_t})} \end{bmatrix} \quad (3)$$

where a  $\mathbf{0}$  represents no transmission and  $\delta_{max}$  denotes the maximum of the relative delays. Let  $W$  symbols be encoded into the original STBC  $S \in \mathbb{C}^{N_t \times T}$ , then it can be seen from (3) that they take  $T + \delta_{max}$  time interval for transmitting  $s$ . Hence, the effective data rate in the asynchronous cooperative network is  $W/T + \delta_{max}$ , which is less than the data rate in a synchronous system  $W/T$  for which the STBC is traditionally designed.

Now, a space-time code  $\mathcal{S}$  is called  $\tau$ -delto if for all delay profiles  $\Delta$  with  $\delta_{max}(\Delta) \leq \tau$ , the effective space-time code  $s^\Delta$  achieves the spatial diversity as high as that of  $\mathcal{S}$ . A space-time code is fully delto if it is delto for any positive integer  $\tau$ . For more detail examples, the readers are referred to [7]. Furthermore, in [7], it has also been proved that the space-time codes constructed from cyclic division algebra [16], including the well-known Golden code, are also not delay tolerant.

### 3. Construction of delto codes

In this section, we try to develop two useful techniques for construction of STBC codes based on threads that are delay tolerant. The constructed codes achieve maximal spatial diversity and are fully delto. They are also flexible with respect to signalling constellation,

transmission rate, number of transmit/receive antennas and decoder complexity. For most of the cases, we use the fundamental signalling alphabet  $\Omega$  derived from constellation  $\mathcal{A}$  in accordance with (2).

A layer is a mapping strategy that assigns a particular transmit antenna to be used at each individual time interval of a code word [17]. A layer is called a thread when it spans in spatial and temporal dimensions in such a way that at each time instant:  $1 \leq t \leq T$  at most one antenna is used [18]. With a minor modification we relax the condition of antenna usage at each time interval and allow the signalling intervals to be empty, i.e. no symbol be transmitted from any antenna during certain signalling intervals.

We use a technique very similar to that of Huffman (HM) binary tree. We develop an HM binary tree of  $N_t + 1$  nodes. We assume that nodes 2 (node 1 is discarded) to  $N_t + 1$  of the HM binary trees represent the rows 1 to  $N_t$  of codeword matrices, respectively. We further assume that the weights or more precisely hamming weights of nodes  $m_i$ ,  $i = 1, 2, \dots, N_t + 1$  are such that  $m_i > m_{i+1} > \dots > m_{N_t+1}$  and  $m_{N_t-1} > m_{N_t} + m_{N_t+1}$ . With these assumptions, we may construct the HM tree in a straightforward way starting from bottom node coming up to top node.

As an example, consider the HM thread  $\Lambda$  defined for  $N_t$  transmit antennas, where

$$T_{N_t}^{HM} = N_t \times \left\lceil \frac{N_t}{2} \right\rceil + \left( \frac{1 + (-1)^{N_t}}{2} \right) \left\lceil \frac{N_t}{2} \right\rceil \quad (4)$$

For  $N_t = 3$  and 4, to obtain HM threads, we draw two HM binary trees in Figures 1 and 2, respectively.

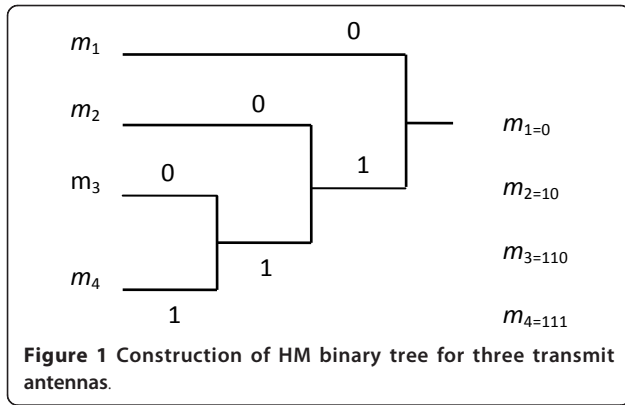
Putting the obtained numerical values in matrix form in a row end-to-start manner by non-zero elements, i.e. after discarding the first node (i.e.  $m_i$ ) of the tree, the  $m_{i+2}$  th row is started immediately from next column in which  $m_{i+1}$  th has its last non-zero element. The process is repeated till  $m_{N_t+1}$  th row. The empty positions are filled by zeros, we get

$$\Lambda_3^{HM} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \quad (5)$$

and, for  $N_t = 4$ , we have

$$\Lambda_4^{HM} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (6)$$

As an alternate method, we can develop such type of codeword matrices by using the following expression,



**Figure 1** Construction of HM binary tree for three transmit antennas.

where the thread  $\Lambda_{\text{HM}}$  has  $(i, j)$  entry defined as

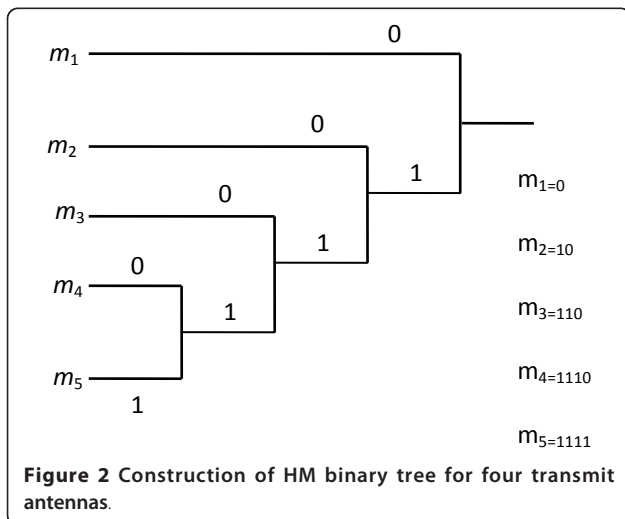
$$\Lambda(i, j) = \begin{cases} 1, & \text{if } i = (\lfloor \log_2 j + \delta_{j,7} \rfloor + 1) \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

where the Kronecker delta function is defined as

$$\delta_{j,7} = \begin{cases} 1 & \text{if } j = 7 \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

**Lemma 1:** Let  $\mathcal{S} = \Omega\Lambda_{\text{HM}}$  denote the space-time code in which the repetition code over alphabet  $\Omega$  is used over the thread  $\Lambda_{\text{HM}}$ , then  $\mathcal{S}$  achieves full spatial diversity and is fully delto.

*Proof:* One can see that  $\mathcal{S}$  encompasses multiples of  $\Lambda_{\text{HM}}$ , so all the differences between codeword in  $\mathcal{S}$  are multiples of  $\Lambda_{\text{HM}}^{\Delta}$ , hence it is easy to show that  $\Lambda_{\text{HM}}^{\Delta}$  is of full rank for all delay profiles. One can see that regardless of  $\Delta$  size, the  $i$ th row of  $\Lambda_{\text{HM}}^{\Delta}$  always contains the same number of ones as its index (i.e.  $m_i = i$ ), whereas the total number of non-zero elements in all



**Figure 2** Construction of HM binary tree for four transmit antennas.

lower numbered rows is  $i(i-1)/2$ . Hence, for each  $i$ , there is a column in  $\Lambda_{\text{HM}}^{\Delta}$  for which the entry in the  $i$ th row is 1 and all the elements above it are zeros. The set of these columns for  $i = 1$  to  $N_t$  forms an  $N_t \times N_t$  submatrix that is lower-triangular with ones on the diagonal. Since this submatrix has determinant 1, so we can say that  $\Lambda_{\text{HM}}^{\Delta}$  is of full rank.

One can see from the code structure that any permutation of rows or columns may be done in  $\Lambda_{\text{HM}}$  to produce an equivalent thread yet preserving the properties of its parent code, deletion of rows in  $\Lambda_{\text{HM}}$  also would not affect the delto property.

Now, generalizing the obtained results over DAST codes [19], for  $t = 1, 2, \dots, T_{N_t}^{\text{HM}}$ , consider  $f_t: \Omega \rightarrow C$  be a one to one function, and we derive the corresponding thread function matrix  $\mathbf{F}_{\Lambda}(x)$  for thread  $\Lambda_{\text{HM}}$  by replacing the non-zero elements in matrix  $\Lambda_{\text{HM}}$  by the function  $\mathbf{F}_{\Lambda}(x)$ . For example, for  $N_t = 4$ , we have

$$\mathbf{F}_{\Lambda}(x) = \begin{bmatrix} f_1(x) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & f_2(x) & f_3(x) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & f_4(x) & f_5(x) & f_6(x) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & f_7(x) & f_8(x) & f_9(x) & f_{10}(x) \end{bmatrix}$$

**Lemma 2:** let  $\mathcal{S}$  denote space-time code of form  $\mathbf{s}_a = \mathbf{F}_{\Lambda}(a)$  for some  $a \in \Omega$ . Then,  $\mathcal{S}$  achieves full diversity and is delto.

*Proof:* If  $a$  and  $b$  are two distinct codewords of  $\mathcal{S}$ , then the difference codeword matrix  $f_{\Lambda}(a)$  and  $f_{\Lambda}(b)$  will adopt the same form as  $\Lambda_{\text{HM}}$  by replacing 1 for  $t = 1, 2, \dots, T_{N_t}^{\text{HM}}$  by the difference matrix  $f_t(a) - f_t(b)$ .

Let an arbitrary delay profile  $\Delta$  be applied to the difference matrix  $f_{\Lambda}(a) - f_{\Lambda}(b)$  to produce the matrix  $\mathbf{F}^{\Delta}$ , then, as proved before, the columns  $t_1, t_2, \dots, t_{N_t}$  in  $\mathbf{F}^{\Delta}$  form a lower triangular matrix with diagonal entries equal to  $f_t(a) - f_t(b)$  for  $i = 1, 2, \dots, N_t$  and this matrix has determinant:

$$D = \prod_{i=1}^{N_t} (f_t(a) - f_t(b)) \quad (9)$$

Since all the functions  $f_t$  are one to one, so the determinant  $D$  will be zero subject to condition if  $a = b$ , likewise  $\mathbf{F}_{\Lambda}(a)$  will be equal to  $\mathbf{F}_{\Lambda}(b)$  if  $a = b$ . Therefore, the matrix  $\mathbf{F}^{\Delta}$  is full rank.

#### Another method

The HM method for construction of codeword matrices may be lethargic for large value of  $N_t$ , as one can see from the code structure that there is a large disparity or unevenness in usage of antennas. Here, we develop another method of thread matrix construction in which each antenna is used for the same number of time. We

call this technique of thread construction as uniform use (UU) threads matrices. In this particular scheme, each transmit antenna is used twice per codeword. The two non-zero elements of the codeword matrix in row  $i$  ( $i = 1, 2, \dots, N_t$ ) are spanned by  $u_i = \left\lfloor 2^{i-2} + \frac{i}{N_t} \right\rfloor$ ,<sup>a</sup> where  $u$  is the number of zeros between two non-zero elements in row  $i$ .

For example, for  $N_t = 3$  and 4, we have the following two matrices, where  $T_{N_t}^{UU} = 2N_t$

$$\Lambda_{UU} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (10)$$

$$\Lambda_{UU} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (11)$$

The first non-zero element in row  $i$  lies in columns  $j$  according to Table 1.

The UU design is fully delto and offers full diversity.

Equivalently, we can construct such codeword matrices for UU threads as follows, where  $(i, j)$  th entry is defined as

$$\Lambda_{(i,j)} = \begin{cases} 1, & \text{if } i_{\text{del mod } N_t} = \lfloor \log_2 j + \delta_j \rfloor \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

where *delmod* is ordinary modulo  $N_t$  function, and is not taken into account when Kronecker delta function  $\delta_j$  is active, and the Kronecker delta function is defined as

$$\delta_j = \begin{cases} P_{(i+2)}, & \text{if } P_i = j \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

where  $P$  is a vector of first  $T$  elements of safe prime numbers.<sup>b</sup>

*Lemma 3:* Let  $S = \Omega \Lambda_{UU}$  be the  $N_t \times T_{N_t}^{UU}$  space-time code in which the repetition code is used over the thread  $\Lambda_{UU}$ . Then,  $S$  achieves full spatial diversity and is fully delto.

**Table 1 Positions of leading non-zero elements in respective UU threads matrices**

| $i/j$ | $N_t = 3$ | $N_t = 4$ | $N_t = 5$ |
|-------|-----------|-----------|-----------|
| 1     | 2         | 2         | 2         |
| 2     | 4         | 4         | 6         |
| 3     | 1         | 5         | 4         |
| 4     | -         | 1         | 5         |
| 5     | -         | -         | 1         |

*Proof:* As one can see from code structure, it is easy to show that for any delay profile  $\Delta$ , the  $i$ th row of the thread matrix cannot be expressed as a linear combination of rows 1 through  $N_t-1$ .

The two non-zero elements in  $i$ th row are separated by  $u$  zero elements, where  $u$  is given by

$$u_i = \begin{cases} 0 & \text{if } i = 1 \\ 2^{i-2} + 1 & \text{if } i = N_t \\ 2^{i-2} & \text{if } 2 \leq i < N_t \end{cases} \quad (14)$$

or, more precisely

$$u_i = \left\lfloor 2^{i-2} + \frac{i}{N_t} \right\rfloor \quad (15)$$

Furthermore, the leading non-zero element in row  $i_1$  and  $i_{N_t}$  always starts from column  $j_2$  and  $j_1$ , respectively, whereas the second non-zero element of same rows lies in  $j_2$  and  $j_{2N_t-1}$ , respectively. Likewise, for the rest of the rows, the second non-zero elements lay in position  $j_i + \varpi$ , where  $\varpi$  is the position of leading non-zero element in that row.

We know that in the linear combination of even weight rows, if the leading non-zero element in row  $N_t$  lies in column  $j_i$ , then there must be an odd number of rows having a non-zero element in column  $j_i$  [7]. Therefore, we say that our proposed codeword is fully delto.

#### 4. Multiple thread delto codes

In previous section, we discussed different techniques for construction of single thread delto codes. To improve the rate of these codes, we combine multiple delto threads in single codeword matrices. There is more than one way of packing such threads. Here, we discuss two methods as follows.

##### 4.1. Cyclic shift

This method has a very simple and interesting structure. We use to shift each column of thread matrix  $\Lambda_k$  ( $k = 1, 2, \dots, N_t$ ) by one element in thread matrix  $\Lambda_{k+1}$ . We repeat the process till the last thread  $\Lambda_M$ .

Let  $\Lambda_k$  be thread  $k$  for  $N_t$  transmit antennas and  $T$  vector channel uses. Then, for  $N_t = 4$  and  $T_{N_t}^{HM} = N_t \times \left\lfloor \frac{N_t}{2} \right\rfloor + \left( \frac{1 + (-1)^{N_t}}{2} \right) \left\lfloor \frac{N_t}{2} \right\rfloor$ , for thread matrix  $\Lambda_1^{HM}$ , we get

$$\Lambda_1^{HM} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (16)$$

For ease to understand, we replace the non-zero elements in their respective locations by alphabets  $a, b, c$  and  $d$  in  $\Lambda_1$  to  $\Lambda_4$ , respectively. So, the above codeword matrix is reproduced as

$$\Lambda_1^{\text{HM}} = \begin{bmatrix} a_{1,1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & a_{2,2} & a_{2,3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{3,4} & a_{3,5} & a_{3,6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{4,7} & a_{4,8} & a_{4,9} & a_{4,10} \end{bmatrix} \quad (17)$$

After making a shift by one element in each column in above codeword matrix, we get

$$\Lambda_2^{\text{HM}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & b_{1,7} & b_{1,8} & b_{1,9} & b_{1,10} \\ b_{2,1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & b_{3,2} & b_{3,3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b_{4,4} & b_{4,5} & b_{4,6} & 0 & 0 & 0 & 0 \end{bmatrix} \quad (18)$$

After making a shift by one element in each column in above codeword matrix, we get

$$\Lambda_3^{\text{HM}} = \begin{bmatrix} 0 & 0 & 0 & c_{1,4} & c_{1,5} & c_{1,6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & c_{2,7} & c_{2,8} & c_{2,9} & c_{2,10} \\ c_{3,1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_{4,2} & c_{4,3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (19)$$

After making a shift by one element in each column in above codeword matrix, we get

$$\Lambda_4^{\text{HM}} = \begin{bmatrix} 0 & d_{1,2} & d_{1,3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & d_{2,4} & d_{2,5} & d_{2,6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & d_{3,7} & d_{3,8} & d_{3,9} & d_{3,10} \\ d_{4,1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (20)$$

Equivalently, we can construct such a codeword threads matrix  $\Lambda_{N_t}^{\text{HM}}$  by following expression

$$(a_{i,j})_k = \begin{cases} 1, & \text{if } i = X \bmod_{N_t} (\lfloor \log_2 j + \delta_{j,7} \rfloor + k) \\ 0, & \text{otherwise} \end{cases} \quad (21)$$

where  $X \bmod$  is ordinary modulo  $N_t$  function with a small difference that it replaces the output zero by  $N_t$ , and the Kronecker delta function is defined as

$$\delta_{j,7} = \begin{cases} 1 & \text{if } j = 7 \\ 0, & \text{otherwise} \end{cases} \quad (22)$$

From this packing of threads, we get an  $N_t \times T$  space-time code  $\mathcal{S}$  which transmits  $N_t$  repetition codes simultaneously, one per thread by selecting the code code-words [7].

$$\mathbf{s} = a_1 \Lambda_1 + a_2 \Lambda_2 + \dots + a_{N_t} \Lambda_{N_t} \quad (23)$$

For  $a_1, a_2, \dots, a_{N_t} \in \Omega$  arbitrary.

(Note that the notations  $a, b, c, d, \dots$  used in above codewords matrices are replaced by  $a_1, a_1, \dots, a_{N_t}$ ).

For HM thread structure when  $N_t = 4$  and

$$T = N_t \times \left\lceil \frac{N_t}{2} \right\rceil + \left( \frac{1 + (-1)^{N_t}}{2} \right) \left\lceil \frac{N_t}{2} \right\rceil$$

by packing threads  $\Lambda_1^{\text{HM}}$  to  $\Lambda_4^{\text{HM}}$ , we get

$$\mathbf{s} = \begin{bmatrix} a & d & d & c & c & c & b & b & b & b \\ b & a & a & d & d & d & c & c & c & c \\ c & b & b & a & a & a & d & d & d & d \\ d & c & c & b & b & b & a & a & a & a \end{bmatrix} \quad (24)$$

Similarly for  $N_t = 3$ , we have

$$\mathbf{s} = \begin{bmatrix} a & c & c & b & b & b \\ b & a & a & c & c & c \\ c & b & b & a & a & a \end{bmatrix} \quad (25)$$

and when  $N_t = 2$ , we have

$$\mathbf{s} = \begin{bmatrix} a & b & b \\ b & a & a \end{bmatrix} \quad (26)$$

#### In case of UU threads

UU threads can also be packed in the same way as we did above for HM threads.

For  $N_t = 4$  and  $T = 2N_t$ , we have

$$\Lambda_1^{\text{UU}} = \begin{bmatrix} 0 & a_{1,2} & a_{1,3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{2,4} & 0 & a_{2,6} & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{3,5} & 0 & 0 & a_{3,8} \\ a_{4,1} & 0 & 0 & 0 & 0 & 0 & a_{4,7} & 0 \end{bmatrix} \quad (27)$$

$$\Lambda_2^{\text{UU}} = \begin{bmatrix} b_{1,1} & 0 & 0 & 0 & 0 & 0 & b_{1,7} & 0 \\ 0 & b_{2,2} & b_{2,3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b_{3,4} & 0 & b_{3,6} & 0 & 0 \\ 0 & 0 & 0 & 0 & b_{4,5} & 0 & 0 & b_{4,8} \end{bmatrix} \quad (28)$$

$$\Lambda_3^{\text{UU}} = \begin{bmatrix} 0 & 0 & 0 & 0 & c_{1,5} & 0 & 0 & c_{1,8} \\ c_{2,1} & 0 & 0 & 0 & 0 & 0 & c_{2,7} & 0 \\ 0 & c_{3,2} & c_{3,3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{4,4} & 0 & c_{4,6} & 0 & 0 \end{bmatrix} \quad (29)$$

$$\Lambda_4^{\text{UU}} = \begin{bmatrix} 0 & 0 & 0 & d_{1,4} & 0 & d_{1,6} & 0 & 0 \\ 0 & 0 & 0 & 0 & d_{2,5} & 0 & 0 & d_{2,8} \\ d_{3,1} & 0 & 0 & 0 & 0 & 0 & d_{3,7} & 0 \\ 0 & d_{4,2} & d_{4,3} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (30)$$

and packing all the four threads into a single code-word matrix, we get

$$\mathbf{s} = \begin{bmatrix} b & a & a & d & c & d & b & c \\ c & b & b & a & d & a & c & d \\ d & c & c & b & a & b & d & a \\ a & d & d & c & b & c & a & b \end{bmatrix} \quad (31)$$

For  $N_t = 3$ , we get

$$\mathbf{s} = \begin{bmatrix} b & a & a & c & b & c \\ c & b & b & a & c & a \\ a & c & c & b & a & b \end{bmatrix} \quad (32)$$

and for  $N_t = 2$ , get

$$\mathbf{s} = \begin{bmatrix} b & a & a & b \\ a & b & b & a \end{bmatrix} \quad (33)$$

#### 4.2. Algebraically packed multiple-threads

The codes constructed in Section 3 are individually delto and fully diverse, but when they are packed together in a single codeword matrix in a way as we did above, it is not guaranteed that they are delto and fully diverse because the threads may interact in a detrimental way [7]. The remarkable work of El Gamal and Damen [10] can be used to make it sure that the packed codewords are delto and fully diverse.

Let  $\Lambda$  be the HM thread for  $N_t$  transmit antennas and

$$T = N_t \times \left\lceil \frac{N_t}{2} \right\rceil + \left( \frac{1 + (-1)^{N_t}}{2} \right) \left\lceil \frac{N_t}{2} \right\rceil \text{ vector channel}$$

uses. Let  $f_i, j: \Omega \rightarrow C$  be a one-to-one function for each choice  $i = 1, 2, \dots, N_t$  and  $j = 1, 2, \dots, T$ . For each thread,  $\Lambda_k$  derived from  $\Lambda$  in accordance with (6), form the threaded matrix function  $\mathbf{F}_k(x)$  whose  $(i, j)$  th entry is  $f_i(x) \cdot \Lambda_k(i, j)$  [[7], th: 12].

Consider the  $N_t \times T$  space-time code  $\mathcal{S}$  with  $L \leq N_t$  active threads consisting of all modulated codewords of the form:

$$\mathbf{s}(a_1, a_2, \dots, a_L) = \mathbf{F}_1(a_1) + \phi \mathbf{F}_2(a_2) + \dots + \phi^{L-1} \mathbf{F}_L(a_L)$$

for  $a_1, a_2, \dots, a_L \in \Omega$  arbitrary.

Then,  $\mathcal{S}$  achieve full spatial diversity and is fully delto.

*Proof:* Assume  $a$  and  $b$  are two distinct codewords and are subject to the delay profile  $\Delta$ .

Then,  $\mathbf{s}^\Delta$  is given as

$$\begin{aligned} \mathbf{s}^\Delta &= \mathbf{s}^\Delta(a_1, a_2, \dots, a_L) - \mathbf{s}^\Delta(b_1, b_2, \dots, b_L) \\ &= \sum_{i=1}^L \phi^{i-1} \mathbf{F}_i^\Delta(a_i, b_i) \end{aligned}$$

where

$$\mathbf{F}_i^\Delta(a_i, b_i) = \mathbf{F}_i^\Delta(a_i) - \mathbf{F}_i^\Delta(b_i)$$

Let  $m$  denote the largest index for which  $a_m \neq b_m$  but  $a_i = b_i$  for  $i > m$ . Then,

$$\mathbf{s}^\Delta = \sum_{i=1}^m \phi^{i-1} \mathbf{F}_i^\Delta(a_i, b_i) \quad (34)$$

The non-zero elements in main diagonal form a sub-matrix, and are given by

$$\phi^{m-1} [f_{i,j_i}(a_m) - f_{i,j_i}(b_m)] \quad \text{for } i = 1, 2, \dots, N_t$$

This submatrix has determinant

$$D(\phi) = G(\phi) + \phi^{N_t(m-1)} \prod_{i=1}^{N_t} (f_{i,j_i}(a_m) - f_{i,j_i}(b_m)) \quad (35)$$

where  $G(\phi)$  is a polynomial in  $\phi$  over  $\mathbb{F}(\theta)$  of degree  $< N_t(m-1)$ . Since the functions  $f_{i,j_i}$  are all one-to-one and  $a_m \neq b_m$ , Equation (35) is a non-trivial polynomial in  $\phi$  of degree  $N_t(m-1)$  over  $\mathbb{F}(\theta)$ .

By design choice,  $\phi$  is not the root of any non-trivial polynomial of degree  $N_t(m-1)$  over  $\mathbb{F}(\theta)$ . Hence,  $D(\phi) \neq 0$ , so the matrix is of full rank. We conclude that  $\mathcal{S}$  achieves full spatial diversity and is fully delto.

#### Code rate

In the multiple thread code construction, the rate of the space-time code  $\mathcal{S}$  is given [7] as

$$R = \frac{PL}{T} \log_2 \mathcal{A} \quad \text{bpcu}$$

Thus, we can make  $\mathcal{S}$  full rate by proper selection of parameters  $L$  and  $P$  for a given set of code parameters  $N_b, N_r$  and  $T$ . In other words, we make the modulation parameters flexible to match the specified spatio-temporal structure. This selection of modulation parameters can be done in different ways, a natural choice is to take  $L = \min(N_b, N_r)$  and  $P = T$ .

#### 4.3. Packing of threads (when $L < N_t$ )

In previous section, we developed a technique of packing the single thread codeword matrices into  $L = N_t$  threads codeword matrices. Selecting fewer threads than  $N_t$  may increase the spectral efficiency of the code without increasing the constellation size by reducing the code interval length, but for that we have to relax the condition of antenna usage per time unit within each thread. In this section, we pack the threads in such a way that we allow the usage of more than one antenna per time unit within each thread.

##### 4.3.1. HM thread

We denote the *smallest* code length for transmission of  $L$  threads from  $N_t$  transmit antennas by  $T_{N_t, L}^{\text{SHM}}$ . From Section 3, we know that for HM threads the total

number of channel usage is  $n_1 + n_2 + \dots + n_{N_t} = N_t(N_t + 1)/2$ , where  $n_i = i$ .

Now let  $\eta_1, \eta_2, \dots, \eta_L$  denote permutation assigning the values  $n_1 + n_2 + \dots + n_{N_t}$  to the transmit antennas  $1, 2, \dots, N_t$ . Then, according to [7], we may write

$$T_{N_t, L}^{\text{SHM}} = \min_{\eta_1, \eta_2, \dots, \eta_L} \max_{m=1, 2, \dots, N_t} \left\{ \sum_{i=1}^L \eta_i(m) \right\} \quad (36)$$

From (36), for  $L = 1$ , we have

$$T_{N_t, 1}^{\text{SHM}} = N_t \quad (37)$$

and for  $L > 1$ , we have

$$T_{N_t, L}^{\text{SHM}} = \left\lceil \frac{L \left( N_t \times \left\lceil \frac{N_t}{2} \right\rceil + \left( \frac{1 + (-1)^{N_t}}{2} \right) \left\lceil \frac{N_t}{2} \right\rceil \right)}{N_t} \right\rceil \quad (38)$$

For  $N_t = 4$  and  $L = 1$  and  $2$ , we have for example

$$s_{4,1}^{\text{SHM}} = \begin{bmatrix} a & 0 & 0 & 0 \\ a & a & 0 & 0 \\ a & a & a & 0 \\ a & a & a & a \end{bmatrix} \quad (39)$$

$$s_{4,2}^{\text{SHM}} = \begin{bmatrix} a & b & b & b \\ a & a & b & b \\ a & a & a & b \\ a & a & a & a \end{bmatrix} \quad (40)$$

#### 4.3.2. UU thread

We denote the smallest code length for transmission of  $L$  threads from  $N_t$  transmit antennas by  $T_{N_t, L}^{\text{UU}}$ . From Section 3, we saw that for UU thread, the maximum expansion between two channel uses is  $2N_t - 1$ . So, we may deduce that

$$T_{N_t, L}^{\text{UU}} = \left\lceil \left\lfloor \frac{L}{N_t} \right\rfloor + (2N_t - 1) \right\rceil \quad (41)$$

For example, for  $N_t = 4$  and  $L = 1$ , we have

$$s_{4,1}^{\text{UU}} = \begin{bmatrix} 0 & a & a & 0 & 0 & 0 & 0 \\ a & 0 & a & 0 & 0 & 0 & 0 \\ a & 0 & 0 & a & 0 & 0 & 0 \\ a & 0 & 0 & 0 & 0 & 0 & a \end{bmatrix} \quad (42)$$

For  $N_t = 4, L = 2$

$$s_{4,2}^{\text{UU}} = \begin{bmatrix} b & a & a & 0 & 0 & 0 & b \\ a & 0 & a & b & 0 & 0 & b \\ a & 0 & 0 & a & b & 0 & b \\ a & b & b & 0 & 0 & 0 & a \end{bmatrix} \quad (43)$$

when  $N_t = 4$  and  $L = 3$

$$s_{4,4}^{\text{UU}} = \begin{bmatrix} d & a & c & c & a & b & d & b \\ c & b & d & d & b & a & c & a \\ b & c & a & a & c & d & b & d \\ a & d & b & b & d & c & a & c \end{bmatrix} \quad (44)$$

For  $N_t = 3$  and  $L = 3$  we have for example

$$s_{3,3}^{\text{UU}} = \begin{bmatrix} b & a & a & c & b \\ c & b & b & a & c \\ a & c & c & b & a \end{bmatrix} \quad (45)$$

Such type of codes will work efficiently for larger value of  $L$ . For smaller value of  $L$ , we can delete zero columns in (17) and (18), even after amputation of these columns the obtained codes still retain their properties of full diversity and delto.

### 5. Minimum length delto codes

The delto codes discussed in the previous sections have the codes length  $T > N_t$ ; therefore, for large size  $N_t$  their performances may decrease. In this section, we extend our work and propose a technique for constructing delto codes with minimum delay length  $T = N_t$ . Our construction method is based on tight packing of the HM threads as developed in Section 3.

In fact, an  $N_t \times T$  MIMO codeword matrix is a strand of algebraic SISO codes separated by Diophantine numbers  $\varphi$ , and the difference between distinct  $N_t \times N_t$  submatrices is the diversity order of the codeword matrix [[10], th: 4].

In [11], the authors show that the minimum length delto STBC code  $s$  can be constructed by multiplying the designed thread codeword matrix with an  $N_t \times T$  matrix  $C$ , whose entries are re-arrangement of  $\bar{c} \in \mathcal{C}$  ( $\mathcal{C}$  being a full diversity one dimensional block code of length  $N_t T$ ).

Of course the main problem in designing such type of codes is the design of thread codeword matrix. In [11], the authors have proposed two types of such matrices for two and three relays. In what follows, we discuss a new technique for construction of thread codeword matrices, and we claim that our proposed code get better performance over [11], particularly at high SNRs.

#### 5.1. Construction of thread codeword matrix

Recall from Section 3 for HM generalized thread construction, here we develop a simple construction method for  $T_{N_t}^{\text{ML}} = N_t$  as follows:

- For row  $i$ , ( $i = 2, \dots, N_t$ ), define a complex number  $\varphi$  whose power of 2 is simple addition of non-zero elements of row  $i$  in HM single thread codeword matrix.



- In the first row of HM single thread codeword matrix, the  $i$ -tuples of zeros above the non-zero elements in  $i$ th row ( $i = 2, \dots, N_t$ ) are replaced by  $\varphi^{i-1}$ .
- Fill the empty positions by 1.

For example for  $N_t = 3$ , the  $\Lambda_3^{\text{HM}}$  matrix from (3) can be represented as

$$\Lambda_3^{\text{ML}} = \begin{bmatrix} 1 & \phi & \phi^2 \\ 1 & \phi^4 & 1 \\ 1 & 1 & \phi^8 \end{bmatrix} \quad (46)$$

For ease to understand, let  $\alpha_{i,j}$  represent the location of  $\varphi$  in (46).

We show that by an appropriate selection of parameters  $\varphi$  and one-dimensional code  $\mathcal{C}$ , the resulted space-time code  $\mathcal{S}$  is delto for every delay profile.

Now let  $\Xi$  denote those  $N_t$ -tuples of  $\alpha_{i,j}$  in  $\Lambda_{\text{ML}}$ , which are taken from different rows, and let  $\phi^{\alpha_{\text{max}}}$  be the highest number used, where  $\alpha_{\text{max}} = 2^{N_t}$ .

*Lemma 4:* Let  $S = \Omega \Lambda_{\text{ML}}$  denote the space-time code in which the repetition code (with codewords of length  $N_t^2$ ) over alphabet  $\Omega$  is used as one-dimensional SISO code in conjunction with thread  $\Lambda_{\text{ML}}$ , then  $\mathcal{S}$  achieves full spatial diversity and is fully delto, if the following conditions are satisfied.

- $\varphi$  is chosen as an algebraic or transcendental number such that the numbers  $\{1, \phi, \dots, \phi^{\alpha_{\text{max}}}\}$  are algebraically independent over the field  $\mathbb{F}(\theta)$  that contains  $\Omega$  [20].
- The parameters  $\alpha_{i,j}$  are chosen such that the summation of the entries of every  $N_t$ -tuples in  $\Xi$  is unique.

Since the one-dimensional code  $\mathcal{C}$  is a repetition code, it is sufficient to show that  $\Lambda_{\text{ML}}^{\Delta}$  is full rank for every arbitrary delay profiles  $\Delta = (\delta_1, \delta_2, \dots, \delta_{N_t})$ . To verify the diversity order of the code, we need to find out the largest square submatrix in  $\Lambda_{\text{ML}}^{\Delta}$  which is full rank.

- First column is chosen such that it contains a non-zero element in row  $N_t$ .
- $j$ th Column is chosen such that it contains a power of  $\varphi$  at  $i$ th row ( $i, j = 2, \dots, N_t-1$ )
- As a last step chose  $N_t$ th column (for which we have only one choice)

As a result, the obtained  $N_t \times N_t$  submatrix has at least one thread  $L$  with all non-zero elements containing  $N_t$  elements of power  $\varphi$ .

If the sum of the powers of  $\varphi$  in  $L$  threads is  $m$ , then the determinant of the submatrix is given by

$$\det(\mathbf{D}) = g(\varphi) + \varphi^m \quad (47)$$

where  $g(\varphi)$  is a polynomial of  $\varphi$  with degree less than or equal to  $\alpha_{\text{max}}$ . Since  $m$  is unique,  $g(\varphi)$  does not contain any term in  $\varphi^m$ . Therefore, if the number  $\{1, \phi, \dots, \phi^{\alpha_{\text{max}}}\}$  are algebraically independent over  $\mathbb{F}(\theta)$ ,  $\det(\mathbf{D})$  is not zero and the code achieves full diversity for every delay profile. Due to the nice structure of our codes, we may use more than one method to verify the determinant of the largest  $N_t \times N_t$  submatrix, for example we can use (35) or the same proof as used for Lemma 1.

## 6. Examples

In this section, we lay down some examples of delto distributed TAST codes. Similar to of TAST codes [10], the construction of delto codes are carried out by appropriate selection of the SISO codes and the numbers  $\varphi$ . Full-diversity SISO codes over fading channels can be constructed by applying full-diversity unitary transformations to input signals drawn from lattices or multidimensional constellations carved from a ring. Damen et al. [20] provided a systematic way of constructing  $N_t \times N_t$  fully diverse unitary transformations over the field that contains the elements of information symbols, as

$$\mathbf{X} = \mathbf{R}\mathbf{U} \quad (48)$$

where  $\mathbf{R} = \mathbf{W}^H \cdot \text{diag}(\mathbf{D})$ ,  $\mathbf{W}$  being a discrete Fourier transform matrix built from the transmit QAM symbols vector of size  $P$ ; we have

$$W(k, l) = 1/\sqrt{P} \cdot \exp(-2j\pi \cdot (k-1) \cdot (l-1)/P), \quad k, l = 1, 2, \dots, P$$

and  $\mathbf{D}$  is a vector of the following form

$$\mathbf{D} = \left[ 1, \theta^{1/P}, \theta^{2/P}, \dots, \theta^{(P-1)/P} \right] \quad (49)$$

where  $\theta$  is a transcendental or an algebraic numbers of suitable degree to guarantee the full diversity of the rotation [20].

For  $N_t = N_r = L = 2$ ,  $P = T = 3$ , using HM thread construction guideline from Section 3, we get delto distributed TAST code as follows:

$$\begin{bmatrix} x_1 & \phi y_2 & \phi y_3 \\ \phi y_1 & x_2 & x_3 \end{bmatrix} \quad (50)$$

where  $\mathbf{X} = (x_1, x_2, x_3)^T = \mathbf{R}\mathbf{U}$  and  $\mathbf{Y} = (y_1, y_2, y_3)^T = \mathbf{R}\mathbf{V}$ ,  $\mathbf{U}$ ,  $\mathbf{V}$  are two  $3 \times 1$  vectors of QAM symbols and  $\mathbf{R}_3$  is optimal  $3 \times 3$  complex rotation according to (49). By

setting  $\varphi = \exp(2\pi i/15)$ , this code provides the rate of 2-QAM symbols per channel use and achieves a transmit diversity of 2 regardless of the delay profile.

Another example is the well-known Alamouti code, which is not delto, but its extension to HM threads makes it delto, as shown in Equation (51)

$$\begin{bmatrix} x_1 & -x_2^* & -x_2^* \\ x_2 & x_1^* & x_1^* \end{bmatrix} \quad (51)$$

For  $N_t = 3$ ,  $N_r = L = 2$ ,  $P = T = 5$ , we have the SUU STBC code (45) with codeword matrices

$$\begin{bmatrix} x_1 & \phi\gamma_2 & \phi\gamma_3 & 0 & x_5 \\ 0 & x_2 & x_3 & \phi\gamma_4 & 0 \\ \phi\gamma_1 & 0 & 0 & x_4 & \phi\gamma_5 \end{bmatrix} \quad (52)$$

where  $X = (x_1, x_2, x_3, x_4, x_5)^T = \mathbf{R}\mathbf{U}$  and  $Y = (y_1, y_2, y_3, y_4, y_5)^T = \mathbf{R}\mathbf{V}$ ,  $\mathbf{U}$ ,  $\mathbf{V}$  are two  $5 \times 1$  vectors of QAM symbols and  $\mathbf{R}_5$  is optimal  $5 \times 5$  complex rotation according to (48). By setting  $\varphi = \exp(2\pi i/25)$ , this code provides the rate of 2-QAM symbols per channel use and achieves a transmit diversity of 3 regardless of the delay profile.

In (52), the number of active threads  $L$  is less than the number of transmit antennas  $N_t$ . One can re-construct (52) to get a delto distributed TAST code of smaller latency by reducing the number of zeros in transmission.

Thus, for  $N_t = 3$ ,  $N_r = L = 2$ ,  $P = 5$  and  $T = 4$ , one has the STBC code with codeword matrix

$$\begin{bmatrix} x_1 & \phi\gamma_2 & \phi\gamma_3 & 0 \\ 0 & x_2 & \phi\gamma_4 & x_5 \\ \phi\gamma_1 & x_3 & x_4 & \phi\gamma_5 \end{bmatrix} \quad (53)$$

In this case, by setting  $\varphi = \exp(2\pi i/36)$ , Equation (53) guarantees full diversity irrespective of the delay profile. This code provides the rate of 2.5-QAM symbols per channel use.

Although the above examples are independently derived from the thread construction techniques discussed in Sections 3 and 4, but they resemble to that of Damens' codes designed in [7], and it was also confirmed by the simulation results that they have exactly same performances as that of [7], but we hope that the simplicity in construction techniques of our codes may reduce hardware complexity.

In Section 5, we introduced a technique for  $T = N_t$  codeword matrix construction, where the information symbols are chosen from  $Z[i]$  for  $N_t = 2$  with a required full-diversity rotations of  $4 \times 4$ , and  $Z[j]$  when  $N_t = 3$ , with a required full-diversity rotations of size  $9 \times 9$ .

For  $N_t = N_r = T = 2$ , we get a delto STBC codeword matrix of the form

$$\begin{bmatrix} x_1 & \phi x_3 \\ x_2 & \phi^4 x_4 \end{bmatrix} \quad (54)$$

where  $X = (x_1, x_2, x_3, x_4)^T = \mathbf{R}\mathbf{U}$ ,  $\mathbf{U}$  is  $4 \times 1$  vectors of QAM symbols and  $\mathbf{R}_4$  is optimal  $4 \times 4$  complex rotation according to (48). By setting  $\varphi = \exp(2\pi i/3)$ , this code provides a rate of 2-QAM symbols per channel use and achieves a transmit diversity of 2 regardless of the delay profile among its rows.

The noiseless received signal of (54) can be written as

$$\mathbf{Z}_{N_r \times 2} = \mathbf{H}_{N \times 2} \cdot \begin{bmatrix} x_1 & \phi \cdot x_3 \\ x_2 & \phi^4 \cdot x_4 \end{bmatrix}_{2 \times 2} \quad (55)$$

We remark that  $x_1 = \mathbf{R}(1, :)_1 \times 4 \cdot \mathbf{U}_4 \times 1$ ,  $x_2 = \mathbf{R}(2, :)_1 \times 4 \cdot \mathbf{U}_4 \times 1$ ,  $x_3 = \mathbf{R}(3, :)_1 \times 4 \cdot \mathbf{U}_4 \times 1$  and  $x_4 = \mathbf{R}(4, :)_1 \times 4 \cdot \mathbf{U}_4 \times 1$  with  $\mathbf{U}_4 \times 1 = [u_1, u_2, u_3, u_4]^T$  denoting the vector of transmitted QAM symbols. Hence, we have

$$\begin{aligned} \mathbf{Z}_{N_r \times 2} &= \mathbf{H}_{N \times 2} \cdot \begin{pmatrix} \mathbf{R}(1, :)_1 \times 4 \cdot \mathbf{U}_4 \times 1 & \phi \cdot \mathbf{R}(3, :)_1 \times 4 \cdot \mathbf{U}_4 \times 1 \\ \mathbf{R}(2, :)_1 \times 4 \cdot \mathbf{U}_4 \times 1 & \phi^4 \cdot \mathbf{R}(4, :)_1 \times 4 \cdot \mathbf{U}_4 \times 1 \end{pmatrix}_{2 \times 2} \\ &= \begin{pmatrix} \mathbf{H}(1, 1) & \mathbf{H}(1, 2) \\ \mathbf{H}(2, 1) & \mathbf{H}(2, 2) \\ \vdots & \vdots \\ \mathbf{H}(N, 1) & \mathbf{H}(N, 2) \end{pmatrix}_{N_r \times 2} \cdot \begin{pmatrix} \mathbf{R}(1, :)_1 \times 4 \cdot \mathbf{U}_4 \times 1 & \phi \cdot \mathbf{R}(3, :)_1 \times 4 \cdot \mathbf{U}_4 \times 1 \\ \mathbf{R}(2, :)_1 \times 4 \cdot \mathbf{U}_4 \times 1 & \phi^4 \cdot \mathbf{R}(4, :)_1 \times 4 \cdot \mathbf{U}_4 \times 1 \end{pmatrix}_{2 \times 2} \end{aligned} \quad (56)$$

This can be equivalently written in columns as

$$\begin{aligned} \mathbf{Z}(1, 1) &= \mathbf{H}(1, 1) \cdot \mathbf{R}(1, :)_1 \times 4 \cdot \mathbf{U}_4 \times 1 + \mathbf{H}(1, 2) \cdot \mathbf{R}(2, :)_1 \times 4 \cdot \mathbf{U}_4 \times 1 \\ \mathbf{Z}(1, 2) &= \mathbf{H}(1, 1) \cdot \phi \cdot \mathbf{R}(3, :)_1 \times 4 \cdot \mathbf{U}_4 \times 1 + \mathbf{H}(1, 2) \cdot \phi^4 \cdot \mathbf{R}(4, :)_1 \times 4 \cdot \mathbf{U}_4 \times 1 \\ &\vdots \\ \mathbf{Z}(N, 1) &= \mathbf{H}(N_r, 1) \cdot \mathbf{R}(1, :)_1 \times 4 \cdot \mathbf{U}_4 \times 1 + \mathbf{H}(N_r, 2) \cdot \mathbf{R}(2, :)_1 \times 4 \cdot \mathbf{U}_4 \times 1 \\ \mathbf{Z}(N, 2) &= \mathbf{H}(N_r, 1) \cdot \phi \cdot \mathbf{R}(3, :)_1 \times 4 \cdot \mathbf{U}_4 \times 1 + \mathbf{H}(N_r, 2) \cdot \phi^4 \cdot \mathbf{R}(4, :)_1 \times 4 \cdot \mathbf{U}_4 \times 1 \end{aligned} \quad (57)$$

This can be summarized in matrix form as

$$\begin{aligned} \mathbf{Z}(1, 1) &= (\mathbf{H}(1, 1) \cdot \mathbf{R}(1, :)_1 \times 4 + \mathbf{H}(1, 2) \cdot \mathbf{R}(2, :)_1 \times 4) \cdot \mathbf{U}_4 \times 1 \\ \mathbf{Z}(1, 2) &= (\mathbf{H}(1, 1) \cdot \phi \cdot \mathbf{R}(3, :)_1 \times 4 + \mathbf{H}(1, 2) \cdot \phi^4 \cdot \mathbf{R}(4, :)_1 \times 4) \cdot \mathbf{U}_4 \times 1 \\ &\vdots \\ \mathbf{Z}(N, 1) &= (\mathbf{H}(N_r, 1) \cdot \mathbf{R}(1, :)_1 \times 4 + \mathbf{H}(N_r, 2) \cdot \mathbf{R}(2, :)_1 \times 4) \cdot \mathbf{U}_4 \times 1 \\ \mathbf{Z}(N, 2) &= (\mathbf{H}(N_r, 1) \cdot \phi \cdot \mathbf{R}(3, :)_1 \times 4 + \mathbf{H}(N_r, 2) \cdot \phi^4 \cdot \mathbf{R}(4, :)_1 \times 4) \cdot \mathbf{U}_4 \times 1 \end{aligned} \quad (58)$$

$$\begin{pmatrix} \mathbf{Z}(1, 1) \\ \mathbf{Z}(1, 2) \\ \vdots \\ \mathbf{Z}(N_r, 1) \\ \mathbf{Z}(N_r, 2) \end{pmatrix}_{2N_r \times 1} = \begin{pmatrix} \mathbf{H}(1, 1) \cdot \mathbf{R}(1, :)_1 \times 4 + \mathbf{H}(1, 2) \cdot \mathbf{R}(2, :)_1 \times 4 \\ \mathbf{H}(1, 1) \cdot \phi \cdot \mathbf{R}(3, :)_1 \times 4 + \mathbf{H}(1, 2) \cdot \phi^4 \cdot \mathbf{R}(4, :)_1 \times 4 \\ \vdots \\ \mathbf{H}(N_r, 1) \cdot \mathbf{R}(1, :)_1 \times 4 + \mathbf{H}(N_r, 2) \cdot \mathbf{R}(2, :)_1 \times 4 \\ \mathbf{H}(N, 1) \cdot \phi \cdot \mathbf{R}(3, :)_1 \times 4 + \mathbf{H}(N_r, 2) \cdot \phi^4 \cdot \mathbf{R}(4, :)_1 \times 4 \end{pmatrix}_{2N_r \times 4} \cdot \mathbf{U}_4 \times 1 \quad (59)$$

So, the transformation of the transmit constellation is obtained by multiplication with the equivalent matrix:

$$\mathbf{B} = \begin{pmatrix} \mathbf{H}(1, 1) \cdot \mathbf{R}(1, :)_1 \times 4 + \mathbf{H}(1, 2) \cdot \mathbf{R}(2, :)_1 \times 4 \\ \mathbf{H}(1, 1) \cdot \phi \cdot \mathbf{R}(3, :)_1 \times 4 + \mathbf{H}(1, 2) \cdot \phi^4 \cdot \mathbf{R}(4, :)_1 \times 4 \\ \vdots \\ \mathbf{H}(N_r, 1) \cdot \mathbf{R}(1, :)_1 \times 4 + \mathbf{H}(N_r, 2) \cdot \mathbf{R}(2, :)_1 \times 4 \\ \mathbf{H}(N_r, 1) \cdot \phi \cdot \mathbf{R}(3, :)_1 \times 4 + \mathbf{H}(N_r, 2) \cdot \phi^4 \cdot \mathbf{R}(4, :)_1 \times 4 \end{pmatrix} \quad (60)$$

Note that when the number of equations is less than the number of unknowns it is necessary to use a

decision feedback equalizer (DFE) to help the sphere decoding to converge. For example it is possible to proceed like as following.

At each time instant the first  $n_m$  transmitted symbols in a packet correspond to last  $n_m$  decoded symbols in the last packet. The matrix  $\mathbf{B}$  can be partitioned in the following way

$$\mathbf{B} = \left[ \begin{array}{c} [\mathbf{B}_1]_{2N_r \times n_m} \\ [\mathbf{B}_2]_{2N_r \times (4-n_m)} \end{array} \right]_{2N_r \times 4} \quad (61)$$

and the transmitted symbol vector can be partitioned as:  $\mathbf{U} = [\mathbf{C}_{n_m \times 1}; \mathbf{D}_{4-n_m \times 1}]$ ; where  $\mathbf{C}_{n_m \times 1}$  is the last  $n_m$  decoded symbols in the last packet, thus we have

$$\mathbf{Z} = \mathbf{B}_1 \cdot \mathbf{C} + \mathbf{B}_2 \cdot \mathbf{D} \quad (62)$$

and we can run the sphere decoding algorithm with the following transformation:  $\mathbf{Z}' \rightarrow \mathbf{Z} - \mathbf{B}_1 \cdot \mathbf{C}$  and  $\mathbf{U} \rightarrow \mathbf{D}$ . The new system involves the calculation of vector  $\mathbf{D}$  of lower size and this can be done with the classical sphere decoding algorithm.

#### In the case of one delay symbol period

We suppose that first row is delayed by one symbol period. In this case, the new space time code can be written as

$$\mathbf{N}_t = \begin{bmatrix} 0 & x_1 & \phi \cdot x_3 \\ x_2 & \phi^4 \cdot x_4 & 0 \end{bmatrix} \quad (63)$$

The noiseless received signal can then be written

$$\mathbf{Z}_{N_r \times 3} = \mathbf{H}_{N_r \times 2} \cdot \begin{bmatrix} 0 & x_1 & \phi \cdot x_3 \\ x_2 & \phi^4 \cdot x_4 & 0 \end{bmatrix}_{2 \times 3} \quad (64)$$

We remark that  $x_1 = \mathbf{R}(1,:)_1 \times 4 \cdot \mathbf{U}_{4 \times 1}$ ,  $x_2 = \mathbf{R}(2,:)_1 \times 4 \cdot \mathbf{U}_{4 \times 1}$ ,  $x_3 = \mathbf{R}(3,:)_1 \times 4 \cdot \mathbf{U}_{4 \times 1}$  and  $x_4 = \mathbf{R}(4,:)_1 \times 4 \cdot \mathbf{U}_{4 \times 1}$ . Hence, we have

$$\mathbf{Z}_{N_r \times 3} = \mathbf{H}_{N_r \times 2} \cdot \begin{bmatrix} 0 & \mathbf{R}(1,:)_1 \times 4 \cdot \mathbf{U}_{4 \times 1} & \phi \cdot \mathbf{R}(3,:)_1 \times 4 \cdot \mathbf{U}_{4 \times 1} \\ \mathbf{R}(2,:)_1 \times 4 \cdot \mathbf{U}_{4 \times 1} & \phi^4 \cdot \mathbf{R}(4,:)_1 \times 4 \cdot \mathbf{U}_{4 \times 1} & 0 \end{bmatrix}_{2 \times 3} \quad (65)$$

$$= \begin{bmatrix} \mathbf{H}(1,1) & \mathbf{H}(1,2) \\ \mathbf{H}(2,1) & \mathbf{H}(2,2) \\ \vdots & \vdots \\ \mathbf{H}(N_r,1) & \mathbf{H}(N_r,2) \end{bmatrix}_{N_r \times 2} \cdot \begin{bmatrix} 0 & \mathbf{R}(1,:)_1 \times 4 \cdot \mathbf{U}_{4 \times 1} & \phi \cdot \mathbf{R}(3,:)_1 \times 4 \cdot \mathbf{U}_{4 \times 1} \\ \mathbf{R}(2,:)_1 \times 4 \cdot \mathbf{U}_{4 \times 1} & \phi^4 \cdot \mathbf{R}(4,:)_1 \times 4 \cdot \mathbf{U}_{4 \times 1} & 0 \end{bmatrix}_{2 \times 3}$$

This can be equivalently written in column

$$\begin{aligned} \mathbf{Z}(1,1) &= \mathbf{H}(1,2) \cdot \mathbf{R}(2,:)_1 \times 4 \cdot \mathbf{U}_{4 \times 1} \\ \mathbf{Z}(1,2) &= \mathbf{H}(1,1) \cdot \mathbf{R}(1,:)_1 \times 4 \cdot \mathbf{U}_{4 \times 1} + \mathbf{H}(1,2) \cdot \phi^4 \cdot \mathbf{R}(4,:)_1 \times 4 \cdot \mathbf{U}_{4 \times 1} \\ \mathbf{Z}(1,3) &= \mathbf{H}(1,1) \cdot \phi \cdot \mathbf{R}(3,:)_1 \times 4 \cdot \mathbf{U}_{4 \times 1} \\ &\vdots \\ \mathbf{Z}(N_r,1) &= \mathbf{H}(N_r,2) \cdot \mathbf{R}(2,:)_1 \times 4 \cdot \mathbf{U}_{4 \times 1} \\ \mathbf{Z}(N_r,2) &= \mathbf{H}(N_r,1) \cdot \mathbf{R}(1,:)_1 \times 4 \cdot \mathbf{U}_{4 \times 1} + \mathbf{H}(N_r,2) \cdot \phi^4 \cdot \mathbf{R}(4,:)_1 \times 4 \cdot \mathbf{U}_{4 \times 1} \\ \mathbf{Z}(N_r,3) &= \mathbf{H}(N_r,1) \cdot \phi \cdot \mathbf{R}(3,:)_1 \times 4 \cdot \mathbf{U}_{4 \times 1} \end{aligned} \quad (66)$$

This can be summarized in matrix form as

$$\begin{bmatrix} \mathbf{Z}(1,1) \\ \mathbf{Z}(1,2) \\ \mathbf{Z}(1,3) \\ \vdots \\ \mathbf{Z}(N_r,1) \\ \mathbf{Z}(N_r,2) \\ \mathbf{Z}(N_r,3) \end{bmatrix}_{3N_r \times 1} = \begin{bmatrix} \mathbf{H}(1,2) \cdot \mathbf{R}(2,:)_1 \times 4 \\ \mathbf{H}(1,1) \cdot \mathbf{R}(1,:)_1 \times 4 + \mathbf{H}(1,2) \cdot \phi^4 \cdot \mathbf{R}(4,:)_1 \times 4 \\ \mathbf{H}(1,1) \cdot \phi \cdot \mathbf{R}(3,:)_1 \times 4 \\ \vdots \\ \mathbf{H}(N_r,2) \cdot \mathbf{R}(2,:)_1 \times 4 \\ \mathbf{H}(N_r,1) \cdot \mathbf{R}(1,:)_1 \times 4 + \mathbf{H}(N_r,2) \cdot \phi^4 \cdot \mathbf{R}(4,:)_1 \times 4 \\ \mathbf{H}(N_r,1) \cdot \phi \cdot \mathbf{R}(3,:)_1 \times 4 \end{bmatrix}_{3N_r \times 1} \cdot \mathbf{U}_{4 \times 1} \quad (67)$$

For  $N_t = T = 3$ , and  $N_r = 2$ , we get a delto distributed TAST codeword of the form

$$\begin{bmatrix} x_1 & \phi x_4 & \phi^2 x_7 \\ x_2 & \phi^4 x_5 & x_8 \\ x_3 & x_6 & \phi^8 x_9 \end{bmatrix} \quad (68)$$

where  $\mathbf{X} = (x_1, x_2, \dots, x_9)^T = \mathbf{R} \cdot \mathbf{U}$ ,  $\mathbf{U}$  is a  $9 \times 1$  vector of information symbols belonging to a 4-array constellation in  $\mathbb{Z}[j]$  and  $\mathbf{R}_9$  is optimal  $9 \times 9$  complex rotation according to (48). By setting  $\phi = \exp(\pi i/12)$ , this code provides the rate of 3 symbols per channel use and achieves a transmit diversity of 3 regardless of the delay profile.

## 7. Simulation

Similar to that of TAST codes, we use sphere decoder for decoding our delto codes. In case of delay profiles where the received signals may contain some unknown equations are dealt by the use of minimum mean square error-decision feedback equalization (MMSE-DFE) processing, as explained in previous section and can originally be found in [21,22].

The simulation figures illustrated below show bit and symbol error rates (SERs) as function of  $E_b/N_0$  in decibels, which is adjusted as follows.

$$\left. \frac{E_b}{N_0} \right|_{dB} = \left. \frac{E_s}{N_0} \right|_{dB} - 10 \log_{10} R$$

where  $E_s$  is the average signal energy per receive antenna and  $R$  is the code rate in bit per channel use (bpcu).

Figure 3 shows the bit error rate (BER) and SER of the delto distributed TAST code (50) with and without delay. We repeat that in case of code (50), the code parameters are  $N_t = N_r = L = 2$  and  $P = T = 3$ . In case of delay, the first row is shifted by one symbol right to the second row.

In Figure 4, we simulated the BER and SER performances of delto distributed TAST codes (52) with and without delay. The code parameters of (52) are  $N_t = 3$ ,  $N_r = L = 2$ , and  $P = T = 5$ . For delay profile, the first row is shifted by one symbol right to the second row.

In Figure 5, we considered the BER and SER performances of delto distributed TAST code (53) with and without delay. The code parameters of (53) are  $N_t = 3$ ,

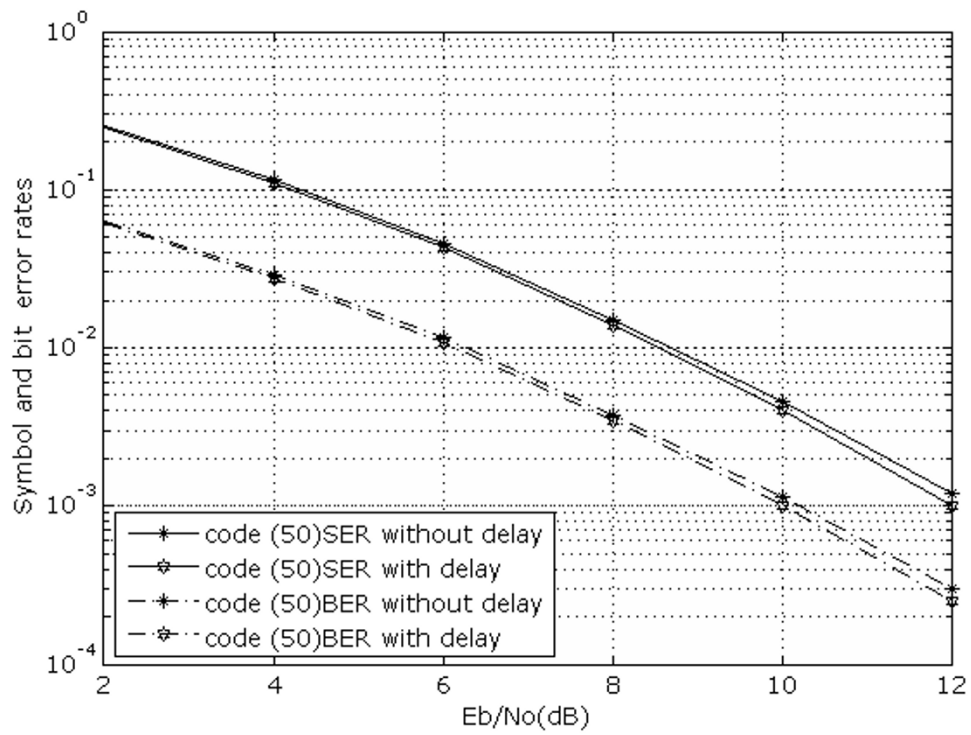


Figure 3 Performance of the delta codeword matrix (50) with and without delay.

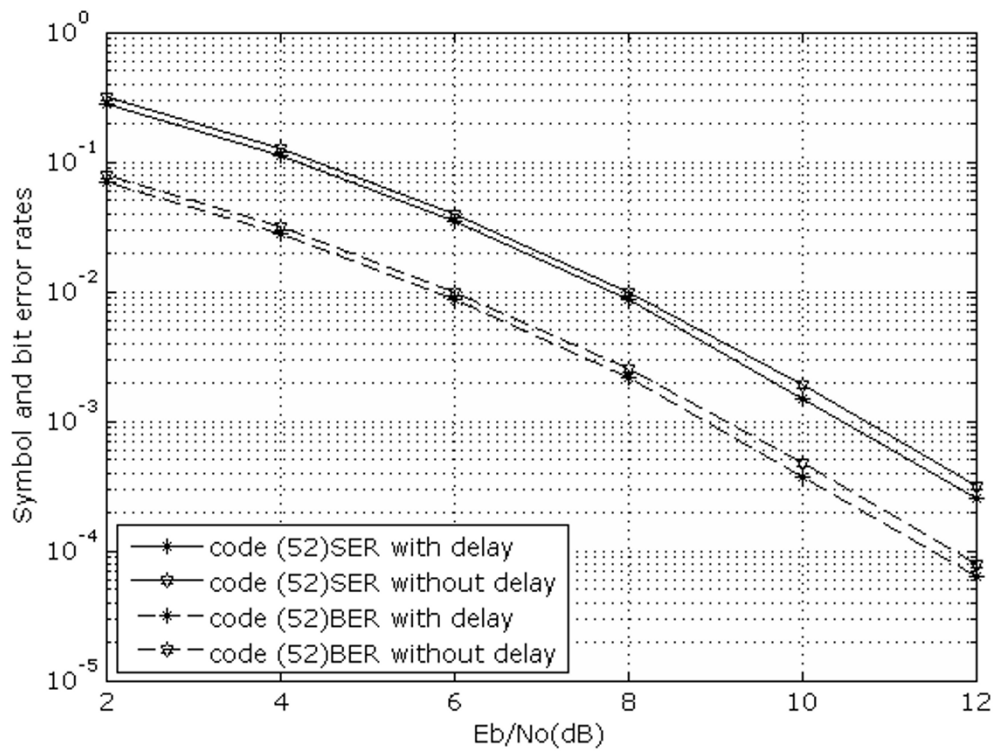


Figure 4 Performance of delta codeword matrix (52) with and without delay.

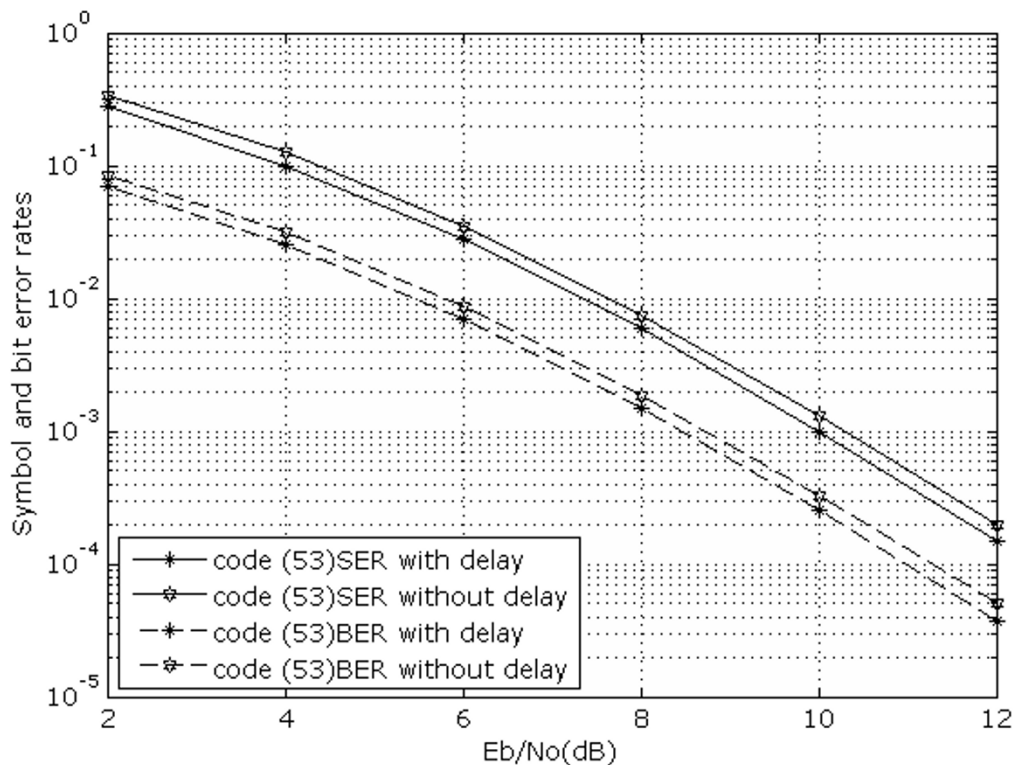


Figure 5 Performance of delto codeword matrix (53) with and without delay.

$N_r = L = 2$ ,  $P = 5$  and  $T = 4$ . For delay profile, the first row is shifted by one symbol right to the second row.

Figure 6 shows the BER and SER performances for codeword matrix (54) without delay. The results are compared with the result of the well-known golden code [23] and the code proposed in [11]. The associated code parameters of (54) are  $N_t = N_r = T = 2$ . One can see that at high SNRs our proposed code (54) gets better performances.

Figure 7 shows the BER and SER performances for codeword matrix (54) with delay. For delay case, we shifted the first row by one symbol interval as shown in (63). The results are compared with the result of well-known golden code [23] and the code given in [11].

Figure 8 shows the BER and SER performances for codeword matrix (68) without delay. The associated code parameters (68) are  $N_t = T = 3$ , and  $N_r = 2$ . The results are compared with the result of the code given in [11].

From this figure, one can observe that our proposed code gets better performance by 0.5 dB at the BER of  $10^{-3}$ . The proposed code gets better performances at high SNRs.

Figure 9 shows the BER and SER performances for codeword matrix (68) with delay. The code parameters for (68) are  $N_t = T = 3$ , and  $N_r = 2$ . For delay case, we

shifted the first row by one symbol interval. The results are compared with that of the code given in [11].

Figure 10 shows the BER and SER performances for codeword matrix (32) with delay. Delay profile is obtained by shifting the first row in (68) by one symbol to the right of other rows. The associated code (68) parameters are  $N_t = N_r = T = 3$ . The results are compared with that of the code in [11]. Our proposed delto distributed TAST code (68) gets better performances particularly at high SNRs.

Figure 11 shows the BER and SER performances for codeword matrix (68) without delay. In this case, the associated code (68) parameters are  $N_t = N_r = T = 3$ . The results are compared with that of the code in [11]. One can see that the error performance of our proposed delto distributed TAST code (68) is improved by about 2 dB at the BER of  $10^{-5}$ .

## 8. Conclusion

Within the same framework developed in [7,11], we introduced some easy and useful techniques for the construction of delay tolerant distributed STBC codes having full diversity and full rate. Like their brethren codes, our proposed codes are flexible with respect to constellation size, number of receive/transmit antennas. We introduced two useful techniques for constructing

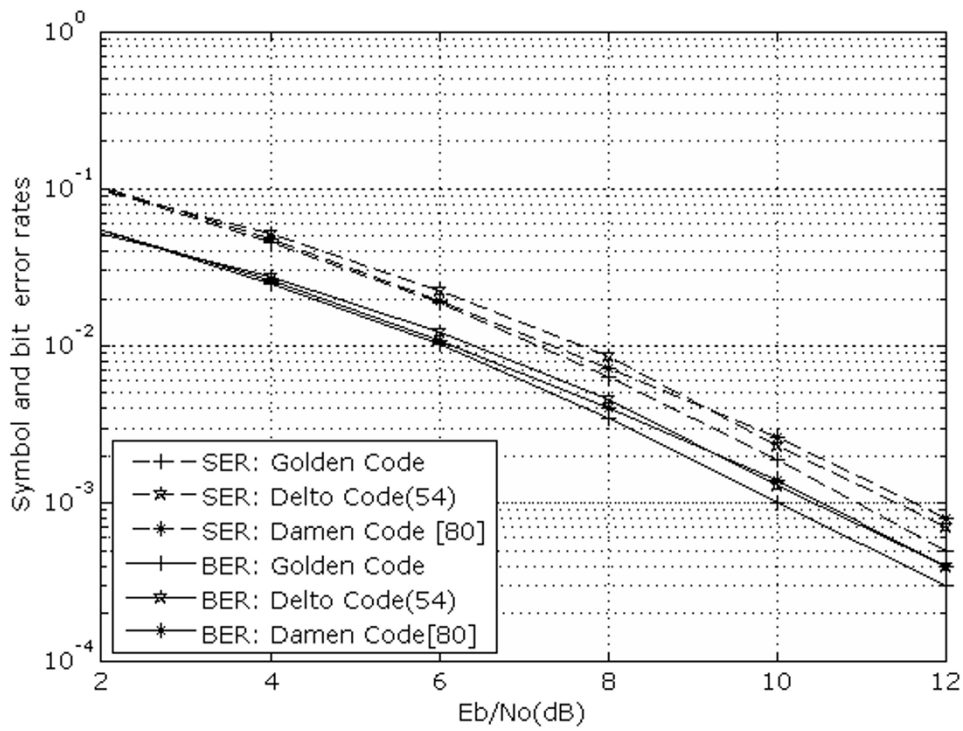


Figure 6 Performances of delto codeword matrix (54) without delay (perfect synchronization) with two transmit two receive antennas and 4-QAM (4 bpcu).

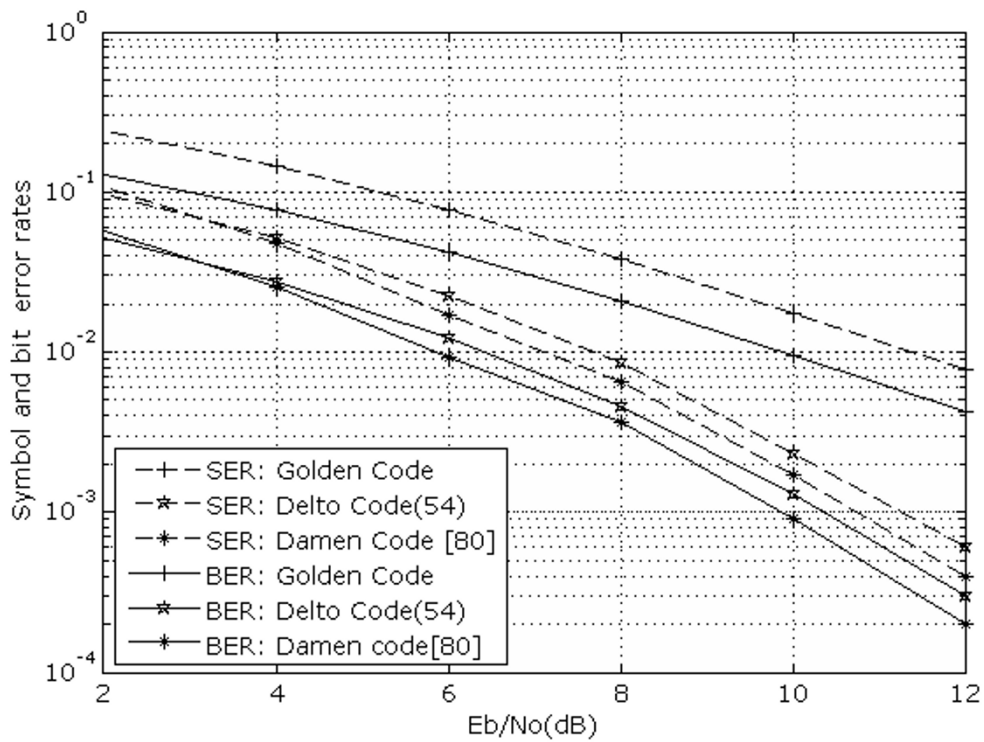
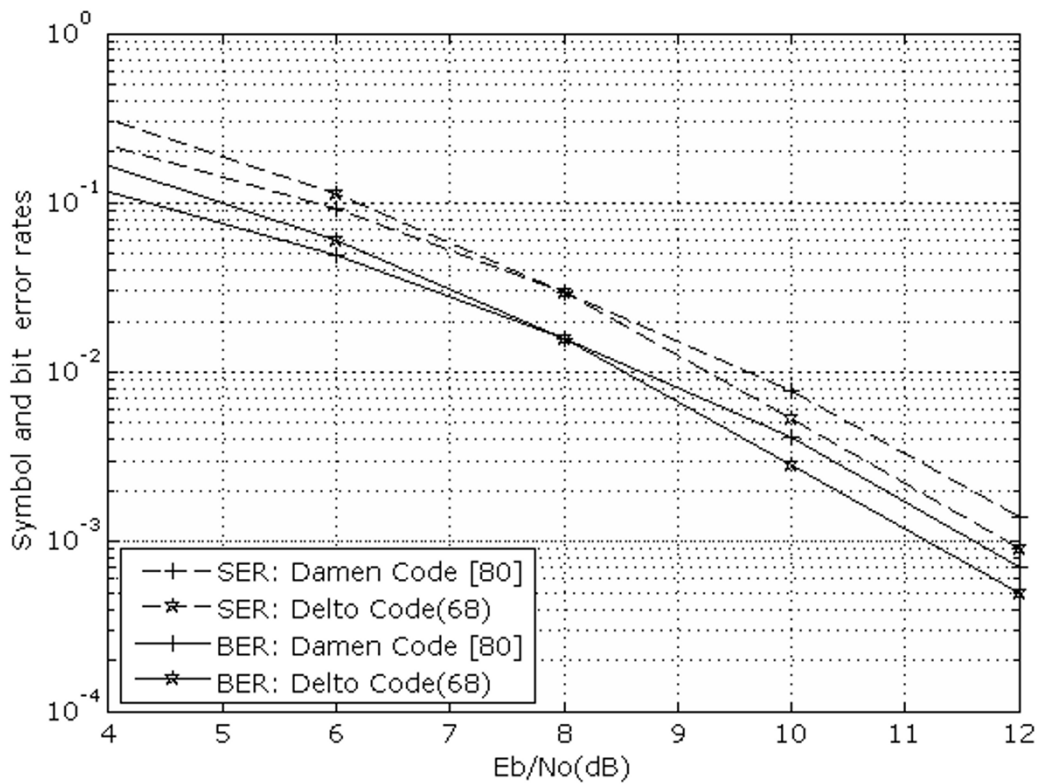
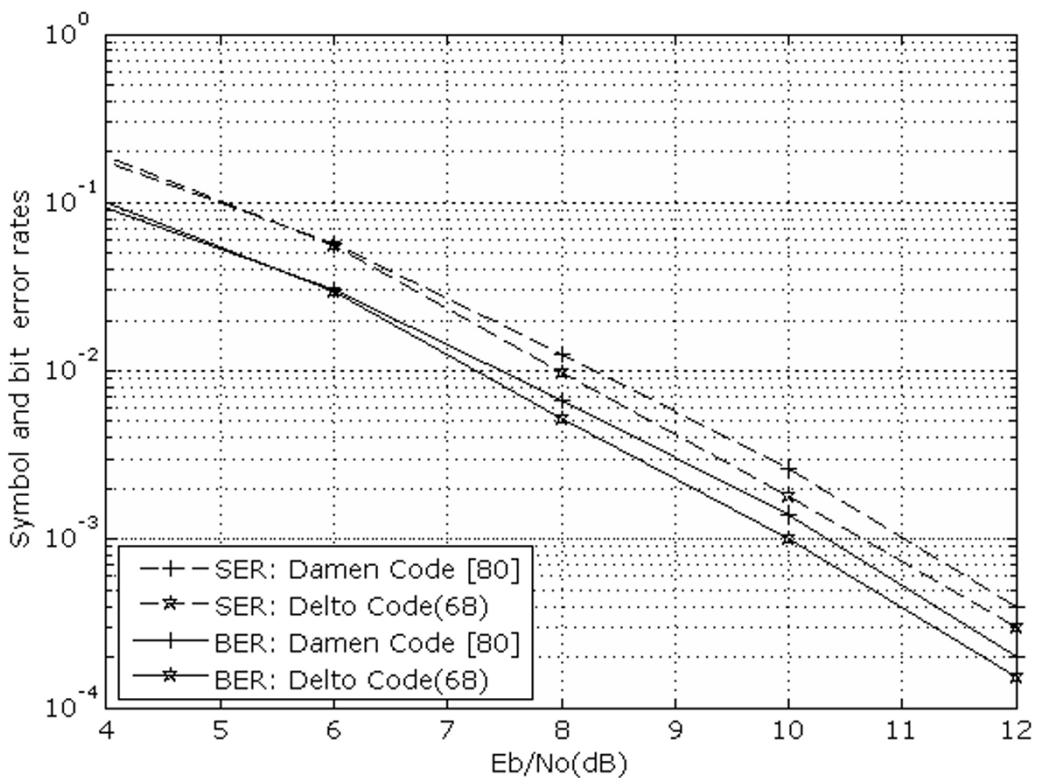


Figure 7 Performances of delto codeword matrix (54) with delay (asynchronous relays) with two transmit two receive antennas and 4-QAM (8/3 bpcu).



**Figure 8** Performances of delto codeword matrix (68) without delay (perfect synchronization) with three transmit two receive antennas and 4-QAM (4 bpcu due to DFE equalization).



**Figure 9** Performances of delto codeword matrix (68) with delay (asynchronous relays) with three transmit two receive antennas and 4-QAM (3 bpcu due to DFE equalization).

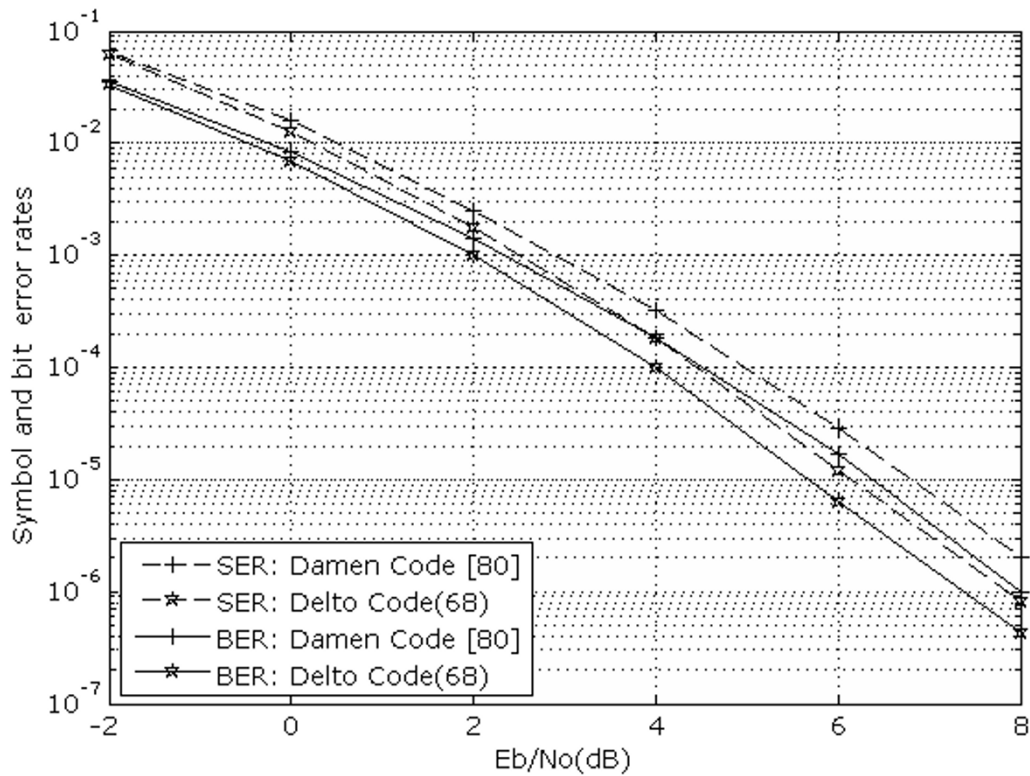


Figure 10 Performances of delto codeword matrix (68) with delay (asynchronous relays) with three transmit three receive antennas and 4-QAM (9/2 bpcu).

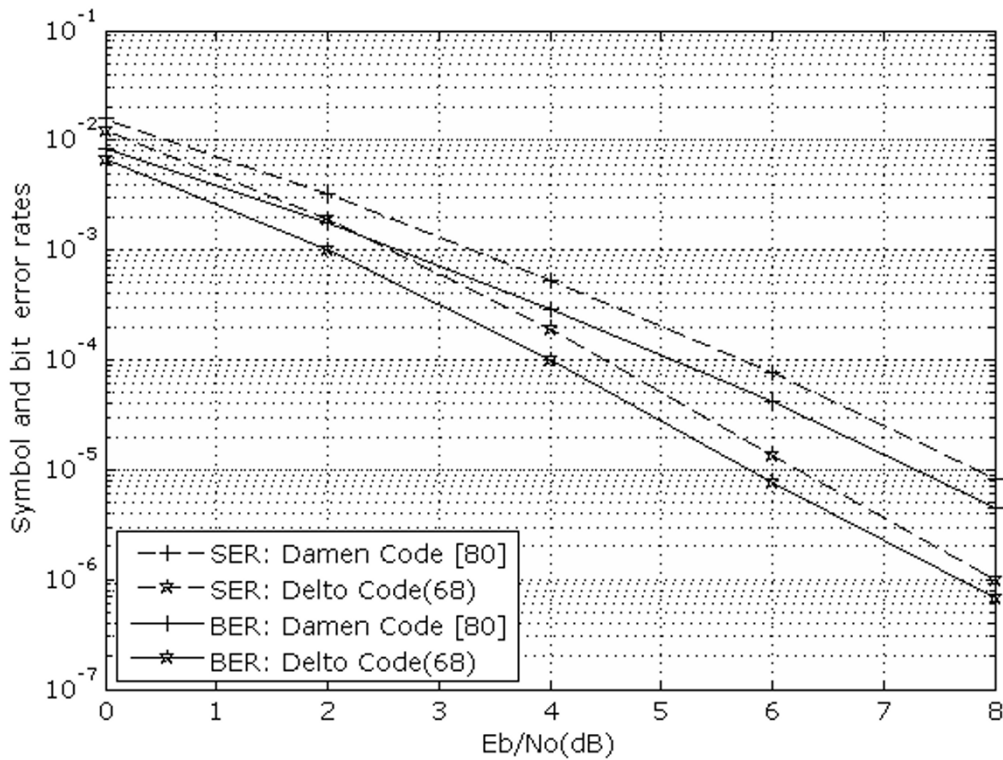


Figure 11 Performances of delto codeword matrix (68) without delay (perfect synchronization) with three transmit three receive antennas and 4-QAM (6 bpcu).



threads codewords matrices. The packing of different threads into a single codeword matrix provides different code structures to be used over cooperative networks with different setup of relays and antennas. In term of error rates, the codes with  $T > N_t$  developed in (50) to (53) do not outperform the codes introduced by Damen but we hope that their simple structures may reduce the hardware complexity. The codes with  $T = N_t$  developed in (54) and (68) outperform the existing codes without sacrificing decoding complexity and other nice characteristics. For example, the error performance of the code proposed in (68) is improved by about 2 dB at the BER of  $10^{-5}$  when  $N_t = N_r = 3$ , and 0.5 dB at  $10^{-3}$  when  $N_t = 2$  and  $N_r = 2$ .

### Endnotes

<sup>a</sup>For  $N_t > 4$ , plus sign is replaced by minus sign. <sup>b</sup>A safe prime is a prime number of the form  $2p + 1$ , where  $p$  is also a prime. For example, first seven safe prime numbers are [5,7,11,23,47,59,83].

### Author details

<sup>1</sup>Balochistan University of Engineering and Technology, Khuzdar, Pakistan  
<sup>2</sup>Ecole Nationale Supérieure d'Ingénieurs de Limoges (ENSIL), XLIM-Dept. C2S2,UMR CNRS 6172, 16, Rue Atlantis - Parc ESTER - BP 6804- 87068 Limoges cedex, France

### Competing interests

The authors declare that they have no competing interests.

Received: 20 August 2011 Accepted: 5 March 2012

Published: 5 March 2012

### References

1. E Biglieri, R Calderbank, A Goldsmith, A Yaswami, A Paulraj, H Vincent Poor, *MIMO Wireless Communications* (Cambridge University Press, New York, 2007)
2. G Tsoulos, *MIMO System Technology for Wireless Communications* (CRC Press, New York, 2006)
3. D Gesbert, M Shafi, D Shiu, PJ Smith, A Naguib, From theory to practice: an overview of MIMO space time coded wireless system. *IEEE J Sel Commun.* **21**(3), 281–302 (2003). doi:10.1109/JSAC.2003.809458
4. A Sendonaris, E Erkip, B Aazhang, User cooperative diversity. Part-I. System description. *IEEE Trans Commun.* **51**(11), 1927–1938 (2003). doi:10.1109/TCOMM.2003.818096
5. Y Li, X-G Xia, Full diversity distributed space time trellis codes for asynchronous cooperative communication in *ISIT*, Adelaide, SA 911–915 (2005)
6. Y Li, X-G Xia, A family of distributed space-time trellis codes with asynchronous cooperative diversity. *IEEE Trans Commun.* **55**(4), 790–800 (2007)
7. MO Damen, AR Hammons, Delay tolerant distributed TAST codes for cooperative diversity. *IEEE Trans Inf Theory.* **53**(10), 3755–3773 (2007)
8. Y Mei, Y Hua, A Swami, B Daneshrad, Combating synchronization errors in cooperative relays, *Proc IEEE International Conference on Acoustic, Speech, and Signal Processing*, Philadelphia, PA, USA, (1) 1–6 (2005)
9. Y Shang, X-G Xia, Limited-shift-full-rank matrices with applications in asynchronous cooperative communications. *IEEE Trans Inf Theory.* **53**(11), 4119–4126 (2007)
10. H El Gamal, MO Damen, Universal space-time coding. *IEEE Trans Inf Theory.* **49**(5), 1097–1119 (2003). doi:10.1109/TIT.2003.810644

11. M Torbatian, M Oussama, On the design of delay tolerant distributed space time codes with minimum length. *IEEE Trans Wirel Commun.* **8**(2), 931–939 (2009)
12. TM Cover, AA El Gamal, Capacity theorems for the relay channel. *IEEE Trans Inf Theory.* **25**(5), 572–584 (1979). doi:10.1109/TIT.1979.1056084
13. AR Hammons Jr, RE Conklin Jr, Space-time block codes for quasi-synchronous cooperative diversity, in *Proc Military Communications Conf (MILCOM)*, Washington, DC 1–7 (October 2006)
14. V Tarokh, N Seshadri, AR Calderbank, Space-time codes for high data rate wireless communication: performance criterion and code construction. *IEEE Trans Inf Theory.* **44**(2), 744–765 (1998). doi:10.1109/18.661517
15. AR Hammons Jr, Algebraic space-time codes for quasi-synchronous cooperative diversity. in *Proc IEEE Int Conf Wireless Networks, Communications and Mobile Computing (WirelessCom 2005)* vol. **1**, 11–15 (June 2005)
16. F Oggier, J-C Belfiore, E Viterbo, *Cyclic Division Algebras: A Tool for Space-Time Coding now* (Boston 2007)
17. GJ Foschini, Layered space time architecture for wireless communication in fading environment when using multiple antennas. *Bell Labs Technol* 41–59 (1996)
18. H Gamal, AR Hammon, A new approach to layered space time coding and signal processing. *IEEE Trans Inf Theory.* **47**(6), 2321–2334 (2001). doi:10.1109/18.945250
19. MO Damen, KA Meraim, JC Belfiore, Diagonal algebraic space time block codes. *IEEE Trans Inf Theory.* **48**(3), 628–636 (2002). doi:10.1109/18.985979
20. MO Damen, HE Gamal, NC Beaulieu, Systematic construction of full-diversity algebraic constellations. *IEEE Trans Inf Theory.* **49**(12), 3344–3349 (2003). doi:10.1109/TIT.2003.820024
21. AD Murugan, H El Gamal, MO Damen, G Caire, A unified framework for tree search decoding: rediscovering the sequential decoder. *IEEE Trans Inf Theory.* **52**(3), 933–953 (2006)
22. MO Damen, H El Gamal, G Caire, On maximum-likelihood detection and the search for the closest lattice point. *IEEE Trans Inf Theory.* **49**(10), 2389–2401 (2003). doi:10.1109/TIT.2003.817444
23. JC Belfiore, G Rekaya, E Viterbo, The golden code: a  $2 \times 2$  full-rate space-time code with nonvanishing determinants. *IEEE Trans Inf Theory.* **51**(4), 310–310 (2005)

doi:10.1186/1687-1499-2012-82

**Cite this article as:** Ahmed et al.: Delay tolerant ( $\delta$ ) distributed TAST codes for cooperative wireless networks. *EURASIP Journal on Wireless Communications and Networking* 2012 **2012:82**.

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)