TSG-RAN Working Group 1 meeting #9 **TSGR1#9(99)j52**
Dresden, Germany
30 Nov – 3 Dec 1999

Agenda item: Ad Hoc #4
Source: LGIC
Title: Codeword Mapping Rule of TFCI Coding Scheme for FDD
Document for: Discussion & Decision

Introduction

In this contribution we propose two new improved mapping rules for the TFCI encoding. The proposed mapping rules provide the significant gain when the number of TFCI bit is between 2 and 5.

In current spec, if the TFCI consist of less than 10 bits, it is padded with zeros to 10 bits by setting the most significant bits to zero. The resulted 10 bits TFCI is encoded by the (32,10) sub-code of second order Reed-Muller code. The transmitted codewords are linear combination of 10 basis sequences: { M_0, M_1, …, M_9 }. If the TFCI consist of n bits less than 10 bits, the transmitted codewords are linear combination of n basis sequence: { M_0, M_1, …, M_{n-1} }.

One we propose here is to change the order of basis sequences as following: { M_0=(All 1's), M_1=$C_{32,16}$, M_2=$C_{32,8}$, M_3=$C_{32,4}$, M_4=$C_{32,2}$, M_5=$C_{32,1}$, M_6=Mask1, M_7= Mask2, M_8= Mask3, M_9= Mask4 }. We call this proposed method 1. The proposed method 1 is minor modification of the current one, which can be accomplished without the increase of hardware complexity and the big change of hardware structure. But, this proposal makes the TFCI coding scheme for FDD achieve more diversity gain in fading channel, which results in 0.5-2.5 dB gain in case of 2-5 bits length TFCI.

The other we propose here is to change the order of basis sequence as following : { M_0= $C_{32,16}$, M_1= $C_{32,8}$, M_2= $C_{32,4}$, M_3= $C_{32,2}$, M_4= $C_{32,1}$, M_5=(All 1's), M_6=Mask1, M_7= Mask2, M_8= Mask3, M_9= Mask4 }. We call this proposed method 2. The proposed method 2 achieves almost the same diversity gain as method 1. Additionally, the proposed method 2 gives more hardware flexibility using the information of punctured bit positions.

Discussion of proposed method 1

Although the current spec about TFCI coding in 25.212 has some editorial mistakes, it can be interpreted as following. The code words of TFCI coding are linear combination of 10 basis sequences given by,

M_0 = (1111 1111 1111 1111 1111 1111 1111 1111)
M_1 = (0000 0000 0000 0000 1111 1111 1111 1111)
M_2 = (0000 0000 1111 1111 0000 0000 1111 1111)
M_3 = (0000 1111 0000 1111 0000 1111 0000 1111)
M_4 = (0011 0011 0011 0011 0011 0011 0011 0011)
M_5 = (0101 0101 0101 0101 0101 0101 0101 0101)
M_6 = (0010 1000 0110 0011 1111 0000 0111 0111)
M_7 = (0000 0001 1100 1101 0110 1101 1100 0111)
M_8 = (0000 1010 1111 1001 0001 1011 0010 1011)
M_9 = (0001 1100 0011 0111 0010 1111 0101 0001)

If the TFCI consist of less than 10 bits, it is padded with zeros to 10 bits by setting the most significant bits to zero. That is, if the TFCI consists n bit where n is less than 10, n basis sequences from M_0 to M_{n-1} are used. Figure 1 show that encoder structure for TFCI coding.

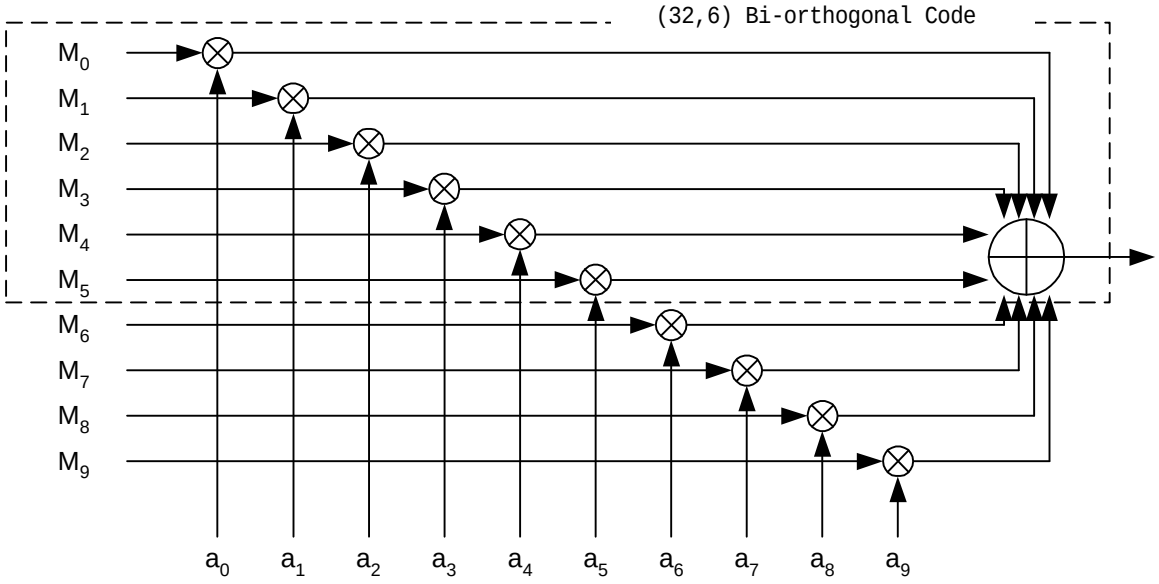


Figure 1. Encoder structure for TFCI coding.

The proposed method 1 is to change the order of basis sequences. The resulted 10 basis sequences are given by,

M_0 = (1111 1111 1111 1111 1111 1111 1111 1111)
M_1 = (0101 0101 0101 0101 0101 0101 0101 0101)
M_2 = (0011 0011 0011 0011 0011 0011 0011 0011)
M_3 = (0000 1111 0000 1111 0000 1111 0000 1111)
M_4 = (0000 0000 1111 1111 0000 0000 1111 1111)
M_5 = (0000 0000 0000 0000 1111 1111 1111 1111)
M_6 = (0010 1000 0110 0011 1111 0000 0111 0111)
M_7 = (0000 0001 1100 1101 0110 1101 1100 0111)
M_8 = (0000 1010 1111 1001 0001 1011 0010 1011)
M_9 = (0001 1100 0011 0111 0010 1111 0101 0001)

It is noticed that the positions from basis sequence M1 to M5 are only reversed comparing to the current method.

The objective of such a change can be illustrated by following simple example. In the current spec, if the TFCI consist of 2 bits, the transmitted codeword prior to puncturing is defined as following:

$(x_0, x_0, x_0, x_0, x_0, x_0, x_0, x_0, x_0, x_0, x_0, x_0, x_0, x_0, x_0, x_0, x_1, x_1, x_1, x_1, x_1, x_1, x_1, x_1, x_1, x_1, x_1, x_1, x_1, x_1)$
where x_0, x_1= 0 or 1.

In the proposed method where the order of basis sequences is changed, the transmitted codeword is defined as following:

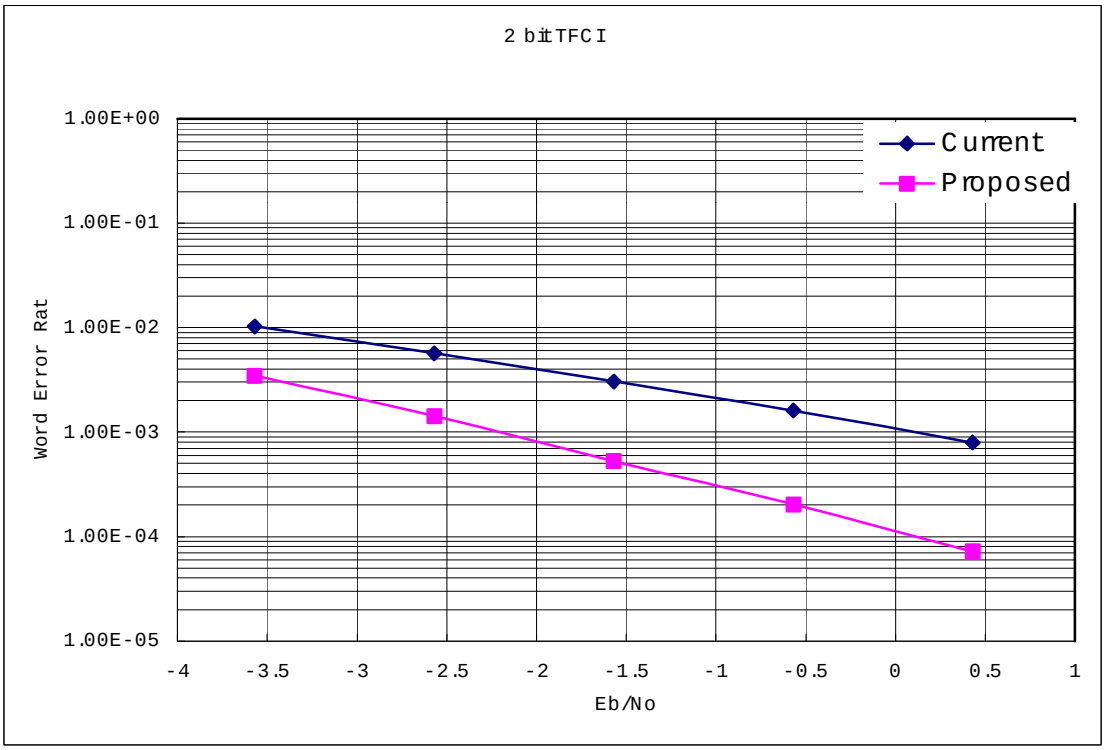
$(x_0, x_1, x_0, x_1, x_0, x_1, x_0, x_1, x_0, x_1, x_0, x_1, x_0, x_1, x_0, x_1, x_0, x_1, x_0, x_1, x_0, x_1, x_0, x_1, x_0, x_1, x_0, x_1, x_0, x_1)$

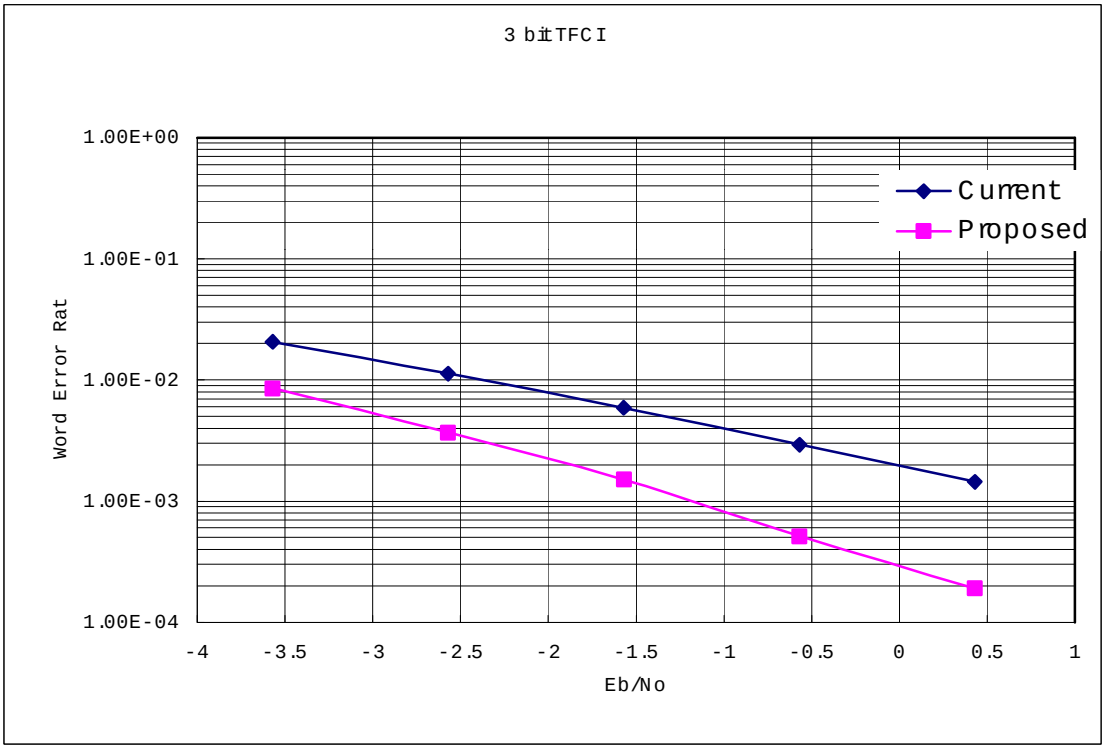
Comparing above two sequence, it is noticed that the proposed method achieve more diversity gain than current method, due to the reason which is described in ETRI Tdoc R1-99G30(Simulation results of TFCI repetition and its text proposal). It is also expected that the proposed method could achieve performance gain due to the same reason in case of 3-5 bits length TFCI.

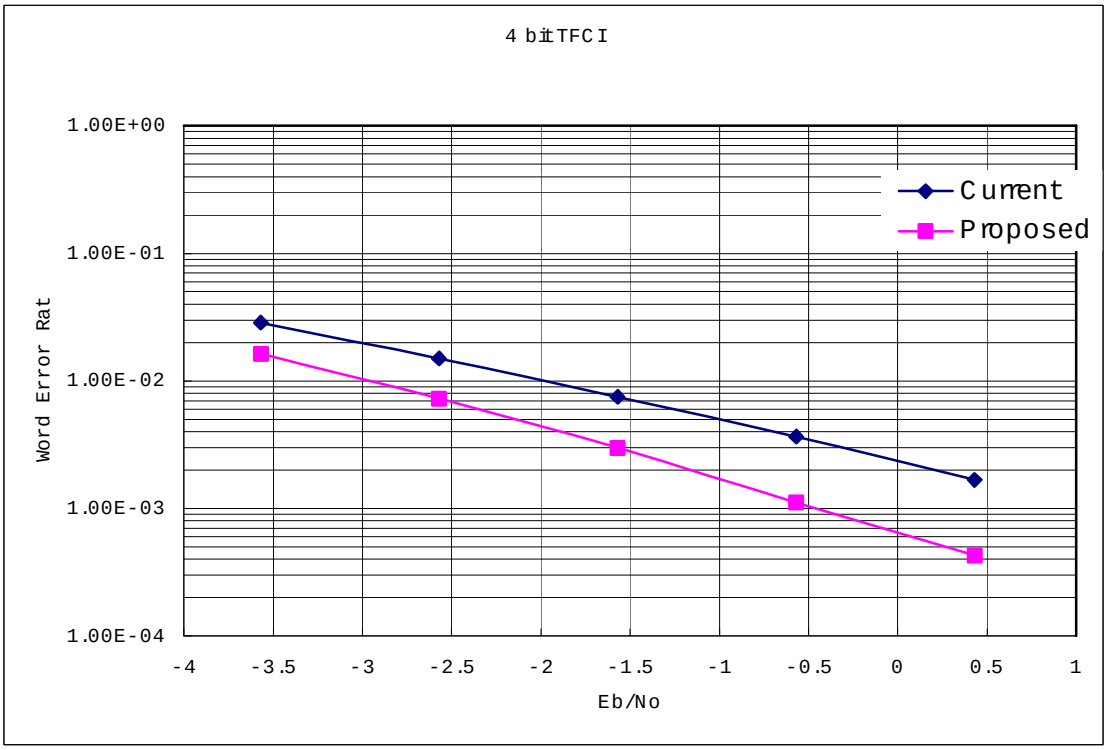
Simulation result

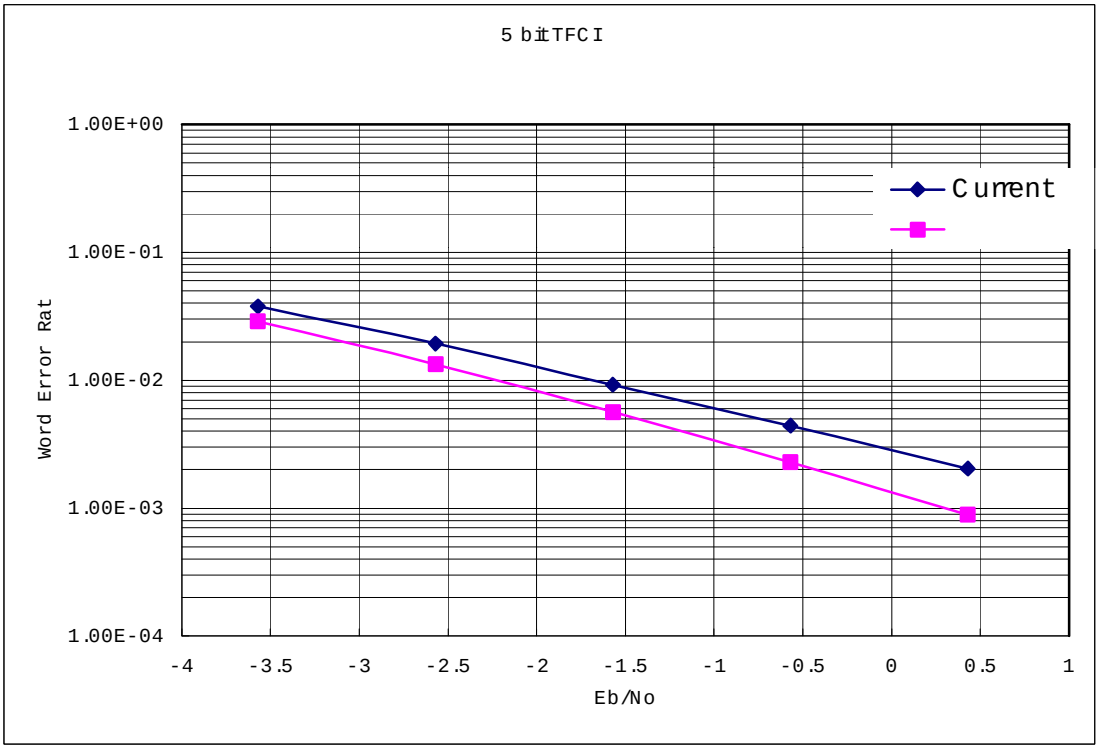
Result 1)

- Rayleigh Fading Channel, 1 Antenna, 1 path.
- 2GHz carrier frequency, 30km/hr Velocity
- Perfect channel Estimation
- Perfect SIR estimation, 1slot Power control loop delay
- Power control step size: 1dB
- Power control Command Error: 10%
- UP Link DPCCH frame structure
- Eb: Energy per DPCCH bit
- Using the TFCI bit length information in receiver




3 bit TFCI
1.00E+00
1.00E-01
1.00E-02
1.00E-03
1.00E-04
Word Error Rat
-4
-3.5
-3
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
Eb/No
Current
Proposed

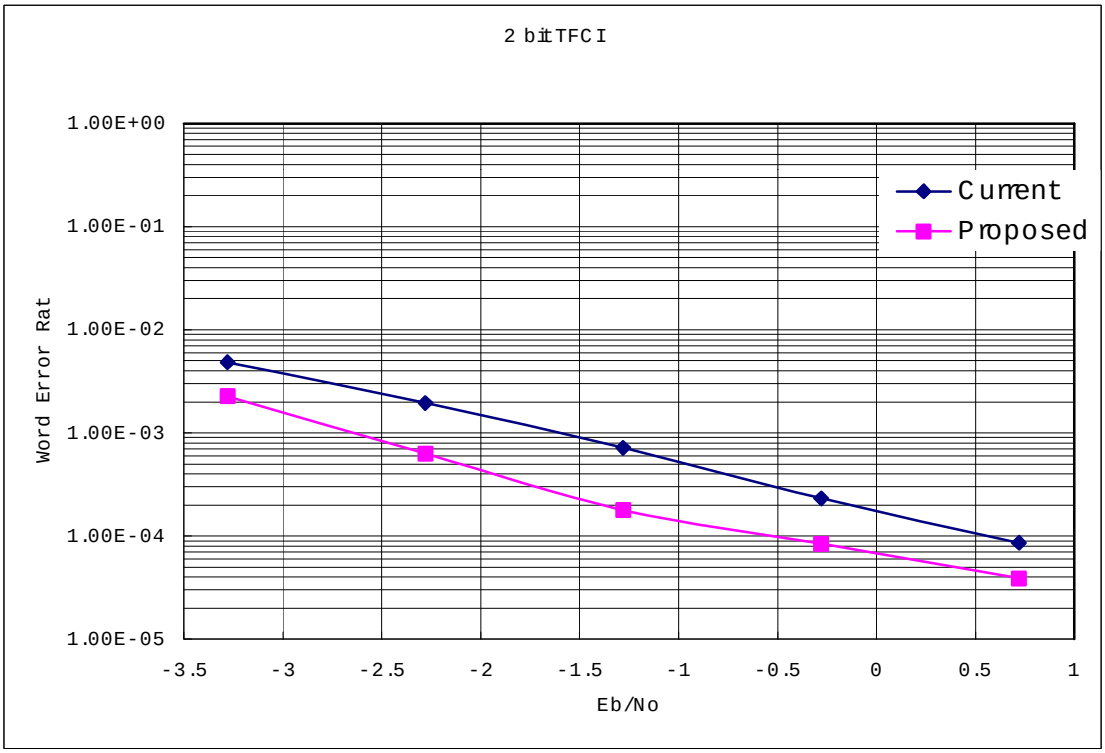

4 bit TFCI
1.00E+00
1.00E-01
1.00E-02
1.00E-03
1.00E-04
Word Error Rat
-4
-3.5
-3
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
Eb/No
Current
Proposed

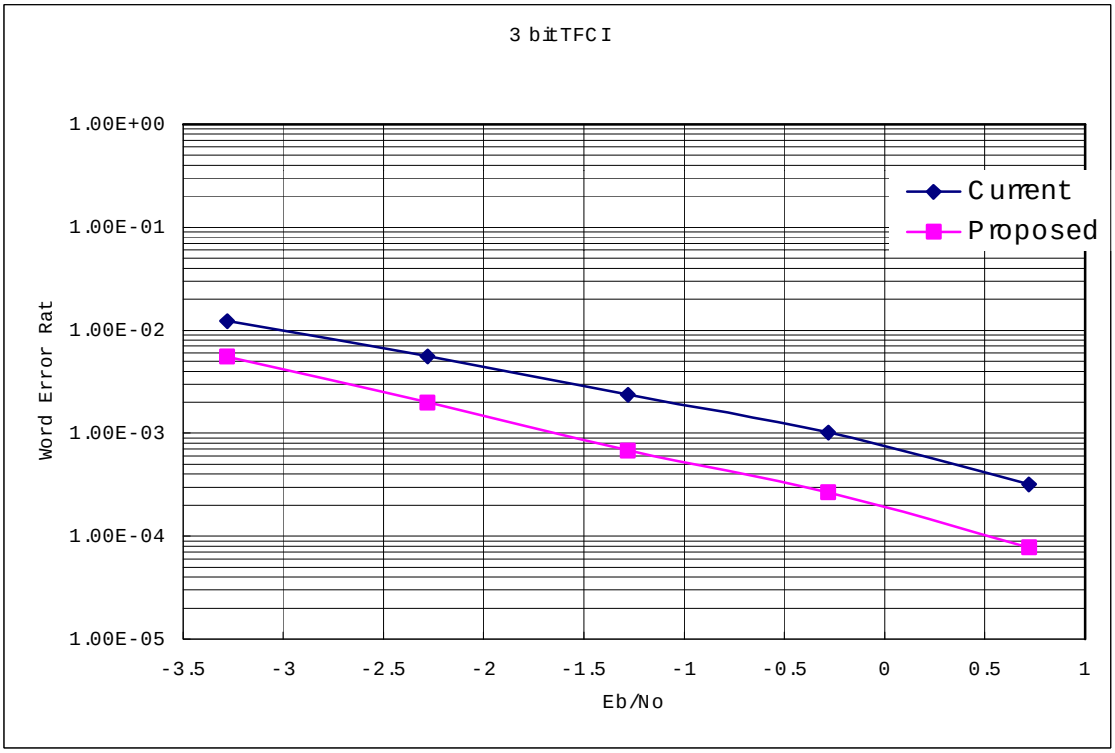

5 bit TFCI
1.00E+00
1.00E-01
1.00E-02
1.00E-03
1.00E-04
Word Error Rat
-4
-3.5
-3
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
Eb/No
Current

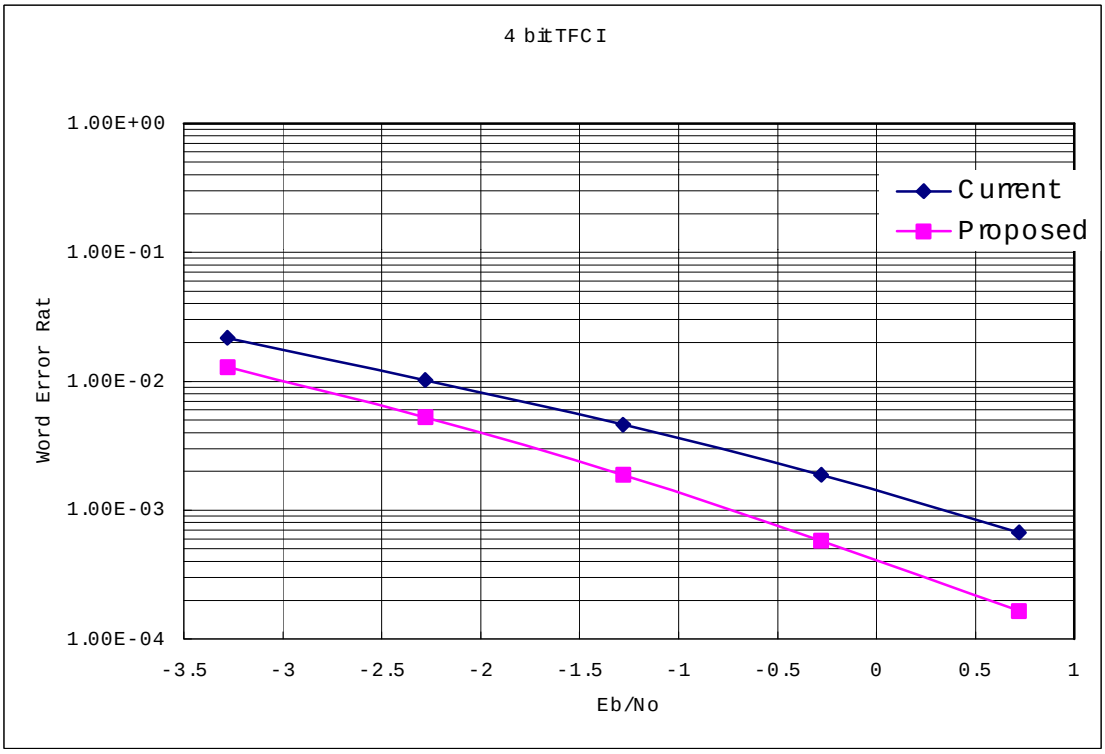
TFCI Simulation Result

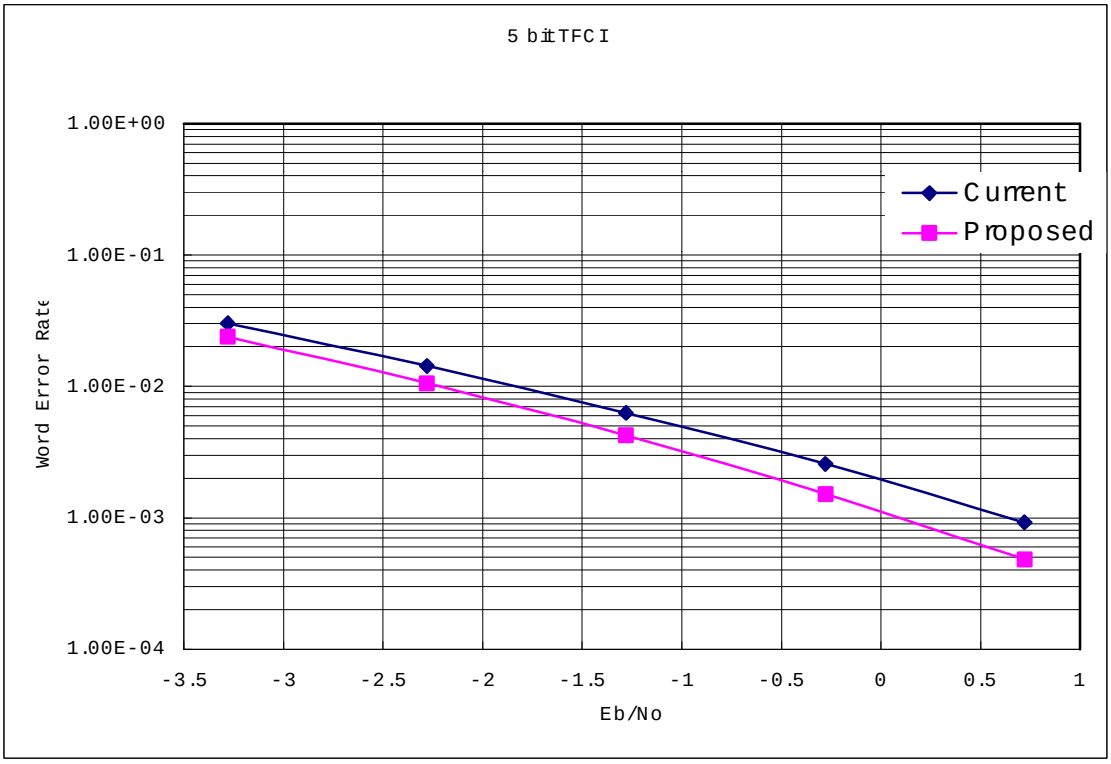
Result 2)

- Rayleigh Fading Channel, 1 Antenna, 1 path.
- 2GHz carrier frequency, 100km/hr Velocity
- Perfect channel Estimation
- Perfect SIR estimation, 1slot Power control loop delay
- Power control step size: 1dB
- Power control Command Error: 10%
- UP Link DPCCH frame structure
- Eb: Energy per DPCCH bit
- Using the TFCI bit length information in receiver




3 bit TFCI
1.00E+00
1.00E-01
1.00E-02
1.00E-03
1.00E-04
1.00E-05
Word Error Rat
-3.5
-3
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
Eb/No
Current
Proposed


4 bit TFCI
1.00E+00
1.00E-01
1.00E-02
1.00E-03
1.00E-04
Word Error Rat
-3.5
-3
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
Eb/No
Current
Proposed


5 bit TFCI
1.00E+00
1.00E-01
1.00E-02
1.00E-03
1.00E-04
Word Error Rate
-3.5
-3
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
Eb/No
Current
Proposed

Discussion of proposed method 2

Since the basis of OVSF codes $C_{32,1}$, $C_{32,2}$, $C_{32,4}$, $C_{32,8}$, $C_{32,16}$ correspond to that of Hadamard codes $H_{5,16}$, $H_{5,8}$, $H_{5,4}$, $H_{5,2}$, $H_{5,1}$ of length 2^5 = 32, optimizing the input pattern is equivalent to exchanging the basis codes from (M_0=all 1's, M_1=$C_{32,1}$, M_2=$C_{32,2}$, M_3=$C_{32,4}$, M_4=$C_{32,8}$, M_5=$C_{32,16}$, M_6, M_7, M_8, M_9) to (M_0=$H_{5,1}$= $C_{32,16}$, M_1=$H_{5,2}$= $C_{32,8}$, M_2=$H_{5,4}$= $C_{32,4}$, M_3=$H_{5,8}$= $C_{32,2}$, M_4=$H_{5,16}$= $C_{32,1}$, M_5=all 1's, M_6, M_7, M_8, M_9).
The proposed method 2 is to change the order of basis sequences. The resulted 10 basis sequences are given by,

M_0 = (0101 0101 0101 0101 0101 0101 0101 0101)
M_1 = (0011 0011 0011 0011 0011 0011 0011 0011)
M_2 = (0000 1111 0000 1111 0000 1111 0000 1111)
M_3 = (0000 0000 1111 1111 0000 0000 1111 1111)
M_4 = (0000 0000 0000 0000 1111 1111 1111 1111)
M_5 = (1111 1111 1111 1111 1111 1111 1111 1111)
M_6 = (0010 1000 0110 0011 1111 0000 0111 0111)
M_7 = (0000 0001 1100 1101 0110 1101 1100 0111)
M_8 = (0000 1010 1111 1001 0001 1011 0010 1011)
M_9 = (0001 1100 0011 0111 0010 1111 0101 0001)

Thus we can easily find that the (32,10) sub-code of second order Reed-Muller code becomes: let denote Z as the number of zeros padded to the TFCI encoder.

(1) Hadamard code if Z=5,6,7,8,9
(2) Biorthogonal code if Z=4
(3) Sub-code of second order Reed-Muller code if Z=0,1,2,3.

According to the current specification, the 1st and 17th bits of (32,10) sub-code of second order Reed-Muller codeword are punctured into (30,10) codeword. The following is shown that the 1st and 17th bits of Hadamard codes of length 32.

1^{st} bit	17^{th} bit
H_0	0 - - - - - - - - - - 0 - - - - - - - - - -
H_1	0 - - - - - - - - - - 0 - - - - - - - - - -
-	-
-	-
-	-
H_{15}	0 - - - - - - - - - - 0 - - - - - - - - - -
H_{16}	0 - - - - - - - - - - 1 - - - - - - - - - -
H_{17}	0 - - - - - - - - - - 1 - - - - - - - - - -
-	-
-	-
-	-
H_{31}	0 - - - - - - - - - - 1 - - - - - - - - - -

When Z equals to 5, the 1^{st} bit of TFCI codewords always becomes '0'. Similarly, when Z is larger than 5, the 1^{st} and 17^{th} bits of codewords always become '0'. This implies that the receiver exactly knows the 1^{st} and 17^{th} bits when the number of zeros padded to the TFCI encoder is larger than 5, and similarly knows the 1^{st} bit when the number of zeros padded is 5, respectively. This interesting property gives us not only performance gain but also hardware flexibility at the side of TFCI decoder since we achieve a gain by simply inserting the known punctured bit or bits at the side of receiver without changing the (32,10) TFCI decoder.
The performance of the proposed method 2 is almost the same as the proposed method 1.

Conclusion

It is shown from simulation results that the proposed method 1 provides the 0.5-2.5 dB performance gain compared to the current method. This gain is achieved without the increase of hardware complexity and the big change of hardware structure.
The proposed method 2 achieves the same performance as the method 1. Furthermore, the method 2 gives more hardware flexibility than the current one at the receiver side using the information of the positions of the punctured bit.