

## **New Spreading Transforms for Fading Channels**

Jonathan S. Yedidia, Karunakar Pedagani, Andreas F. Molisch

TR2004-120 September 2004

### **Abstract**

We introduce a new class of symbol-spreading transforms that are well-suited for use in error-control coding systems operating in fading channels. These unitary transforms have the property that they allow for the unambiguous determination of input symbols from just a single correctly received output symbol. The transforms can be represented using factor graphs that allow for "fast" encoding. Our simulations show that on channels with fading or interference, systems based on these new transforms can greatly outperform previously proposed systems that use Walsh-Hadamard transforms.

*Allerton Conference on Communications, Control, and Computing*

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.



**Publication History:–**

1. First printing, TR-2004-120, September 2004

# New Spreading Transforms for Fading Channels

Jonathan S. Yedidia, Karunakar Pedagani, and Andreas F. Molisch

Mitsubishi Electric Research Laboratories  
201 Broadway, 8th floor  
Cambridge, MA 02139  
{yedidia,pedagani,molisch}@merl.com

## Abstract

We introduce a new class of symbol-spreading transforms that are well-suited for use in error-control coding systems operating in fading channels. These unitary transforms have the property that they allow for the unambiguous determination of input symbols from just a single correctly received output symbol. The transforms can be represented using factor graphs that allow for “fast” encoding. Our simulations show that on channels with fading or interference, systems based on these new transforms can greatly outperform previously proposed systems that use Walsh-Hadamard transforms.

## 1 Introduction

The two major complicating factors in achieving reliable wireless communication are the phenomena of fading and interference between users [1]. To model these phenomena, we consider the following point-to-point channel model: assume that a transmitter emits a block of  $L$  complex symbols  $x_l[t]$  at each discrete time  $t$ , and that the receiver receives the corrupted complex numbers  $y_l[t]$ , where

$$y_l[t] = H_l[t]x_l[t] + n_l[t]. \quad (1)$$

Here the noise variables  $n_l[t]$  are taken to be complex variables independently chosen from a zero-mean complex Gaussian distribution. The complex fading parameters  $H_l[t]$  are also chosen from a probability distribution; we will restrict our attention here to two different choices for that distribution. For the first choice, the  $H_l[t]$  are independently chosen from a zero-mean complex Gaussian distribution (i.e. a fast-fading Rayleigh channel model). For the second choice, which we call an “erasures and noise” model, we take the  $H_l[t]$  to be independently chosen to be 0 with probability  $p_e$ , and 1 with probability  $1 - p_e$ . In all cases, we assume that the receiver has perfect knowledge of the fading parameters (this is the case of “coherent” communication; such knowledge can be obtained in practice using pilot tones [1]). The transmitter is assumed to *not* have any channel state information.

This model is a general one, and the  $L$  different sub-channels can actually be thought of as arising from frequency, spatial (multiple antenna), or temporal diversity. The erasure model is a highly simplified model of interference; to motivate it, one can imagine a case where multiple users randomly hop between frequency sub-channels, and have a good signal as long as there is no clash, but suffer from interference if they choose the same sub-channel as another user.

## 1.1 OFDM

Our primary motivation for considering this model is as an effective channel model for orthogonal frequency division multiplexing (OFDM) systems [2]. OFDM is used in a variety of state-of-the-art high data-rate wireless systems, including digital video broadcasting, wireless local area networks (covered by the IEEE 802.11a, 802.11g, and 802.11n standards), fixed wireless access systems (802.16), and high data-rate, high mobility systems (802.20). Anticipated fourth generation cellular telephone systems are also expected to use OFDM.

The basic principle of OFDM is to convert a frequency-selective wide-band channel into a number (on the order of 50-2000) of parallel narrow-band sub-channels (also called “sub-carriers” or “tones”) each of which is frequency-nonselective (i.e. non-delay-dispersive).

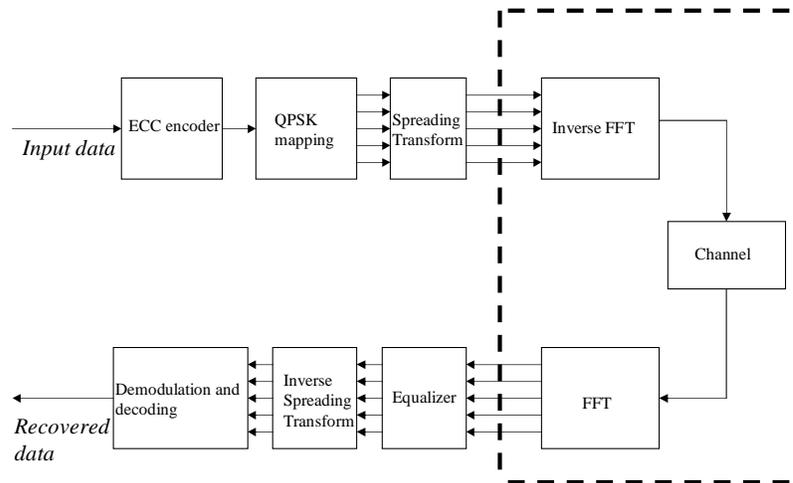


Figure 1: A coded OFDM system using spreading transforms, including both the transmitter and the receiver. The elements inside the dashed box, considered together, form an effective channel.

A simplified block diagram for a (coded) OFDM system is shown in figure 1. At the transmitter, data bits are first encoded using an error-correcting code. The resulting code-words are mapped into complex symbols using some QAM modulation scheme (in this paper, we only consider QPSK). A block of complex symbols is then spread over many sub-carriers by a spreading transform. These spreading transforms will be the main focus of this paper. Finally, an inverse FFT is performed, a cyclic prefix is added, and a continuous-time signal is created, which is transmitted over the channel.

In the standard receiver for such a system, also shown in figure 1, the received signal is first processed using an FFT, and the cyclic prefix is stripped out. The resulting complex symbols are equalized, then a “detector” performs an inverse spreading transform, and finally, the resulting corrupted complex symbols are fed into a decoder for the error-correcting code to recover the original data symbols.

One can treat the combination of the inverse FFT at the transmitter, followed by the

channel, followed by the FFT at the receiver (see figure 1) as an effective channel. Doing so, one obtains the channel model of equation 1, where the  $L$  different diversity branches correspond to  $L$  different sub-carriers.

The “equalization” of OFDM systems is extremely simple, as each sub-channel is effectively narrow-band, and thus the signal received for this tone needs only be divided by the complex number  $H_l[t]$ .

## 1.2 Spreading Transforms

Unfortunately, because each sub-channel in an OFDM system is narrowband, OFDM by itself does not give any frequency diversity—if a sub-carrier is in a fading dip or an erasure, when  $H_l[t]$  has a small magnitude, then its signal-to-noise-ratio (SNR) will be bad, and the probability of error in decoding the corresponding received symbol will be high. Of course, the error-correcting code attached to the OFDM system is there to help deal with this problem, but by itself (i.e. without a good spreading transform to obtain diversity over the sub-carriers), it cannot effectively deal with fading. The problem is that to achieve a high spectral efficiency without too much complexity in the modulation scheme, one often needs to use high-rate codes, which are not capable of dealing with large numbers of erasures.

Therefore, an approach known as “multi-carrier CDMA” or “spread OFDM” was developed over the last decade [2, 3, 4]. In this approach, the block (over different sub-carriers) of mapped QPSK symbols  $u_l[t]$  is first multiplied by an  $L$  by  $L$  Walsh-Hadamard transform (WHT) matrix  $W^L$ , to obtain the “spread” block of complex symbols  $x_l[t]$ :

$$x_l[t] = \sum_{m=1}^L W_{lm}^L u_m[t]. \quad (2)$$

This multiplication spreads the information for each symbol over all available tones. Since the number of symbols is the same as the number of tones, no additional redundancy is added. The idea is to exploit the frequency diversity over the sub-carriers, or viewed in another way, to “soften” the channel that the error-correcting code sees. We note, however, that spreading transforms are neither necessary nor helpful when there is no fading.

There are two major problems with receivers for coded OFDM systems that use Walsh-Hadamard spreading transforms. The first is associated with commonly-used “linear” detectors for the inverse WHT, which simply multiply the received signal with the inverse of WHT matrix after the equalization of each tone, but before demodulation and decoding. Unfortunately, this approach leads to noise enhancement, as the noise from the “bad” tones (with low SNR) are amplified by the equalization, and then distributed over all tones. This problem can in principle be solved by replacing the linear detectors with maximum-likelihood detectors, but the complexity of maximum-likelihood detectors limits their use to very small WHT matrices.

A second important problem is connected to the WHT matrices themselves. Recall that an  $N$  by  $N$  WHT matrix  $W^N$ , where  $N$  is a power of two, is formed as a Kronecker product of 2 by 2 WHT matrices  $W^2$ , where

$$W^2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}. \quad (3)$$

Thus,  $W^4 = W^2 \otimes W^2$ ,  $W^8 = W^2 \otimes W^2 \otimes W^2$ , etc., and for example

$$W^4 = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 \end{pmatrix}. \quad (4)$$

WHT matrices have the nice property of unitarity, which implies that the distance between two vectors of symbols is not changed under a WHT transform. There also exist “fast” algorithms for doing Walsh-Hadamard transforms. However, there is no strong *a priori* argument why Walsh-Hadamard transforms should be used as spreading transforms instead of other transforms that share these properties. In fact an important objective of this paper is to show that they should *not* be used, and to suggest another set of transforms that perform better.

The key problem with WHT matrices is revealed in conditions when there are many sub-carrier erasures (which can be caused either by fading dips or interference). Suppose, to take a very simple illustrative example, that we use a 2 by 2 WHT matrix, and the first output symbol has been erased, while the second output symbol is detected (to high precision) as a 0.0. Suppose further that the input symbols have been mapped to QPSK constellation points at  $(\pm 1 \pm i)/\sqrt{2}$ . Unfortunately, from the single output, the two inputs symbols cannot be resolved. We can infer that they must be equal, so that their difference is zero, but we cannot determine their values.

The lesson of this simple example extends to larger Walsh-Hadamard transforms. In general, from a small number of surviving output symbols, one has no way of disambiguating the possible input symbols.

To deal with this problem, at least in the case of 2 by 2 transforms, an interesting suggestion was recently made by Ojard [5]. He suggested using an asymmetric transform of the form

$$V = \frac{1}{\sqrt{5}} \begin{pmatrix} 2 & 1 \\ -1 & 2 \end{pmatrix}. \quad (5)$$

Now again suppose that the first output is erased, but the second output is received to high precision—say for example, that it is received as a  $(1 + 3i)/\sqrt{10}$ . In this case it is possible to infer the value of the pair of inputs—the only possibility that would give that value for the second output is when the first input equals  $(1 - i)/\sqrt{2}$  and the second input equals  $(1 + i)/\sqrt{2}$ . More generally, every pair of inputs maps to a different output, for both the first and the second output, so that even if only a single output is received correctly, the inputs can be recovered unambiguously.

Ojard’s idea was developed independently to the related idea of “permutation codes” recently proposed by Tavildar and Viswanath, primarily in the context of MIMO systems [6]. Tavildar and Viswanath’s interesting work approaches the problem from a different perspective, but they also limit themselves to “codes” that amount in our context to transforms using only two sub-carriers.

The goal of our work is to give an explicit construction for disambiguating unitary spreading transforms (DUSTs) for larger  $L$ . We will construct these transforms using factor graphs that can be used to give a “fast” encoding. We will also show by simulations that these transforms indeed give an important performance advantage.

## 2 Factor Graphs for DUSTs

Fast Fourier transforms and fast Walsh-Hadamard transforms can be constructed using “normal” factor graphs [7, 8]. The basic building blocks for these factor graphs are “butterfly” factor nodes [8] with two input and two output variables. Such nodes are drawn using a square with four constants (known as “twiddle factors”) inside, representing the two constraints on the input and output variables (see figure 2).

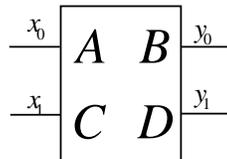


Figure 2: A prototypical butterfly factor node, which implements the two constraints  $y_0 = Ax_0 + Bx_1$  and  $y_1 = Cx_0 + Dx_1$ .

To construct factor graphs for our DUSTs, we entirely copy the “wiring” for a standard FFT or fast WHT, and just modify the twiddle factors in the butterfly factor nodes. An example factor graph for an 8 by 8 DUST is shown in figure 3. Our construction assumes that the DUST has a length  $N = 2^I$ , where  $I$  is an integer greater than or equal to one. We arrange the butterfly factor nodes in  $I$  stacks, and for the  $i$ th stack, the twiddle factors are chosen to be  $A = D = 2^i$ ,  $B = 1$ ,  $C = -1$ . For convenience, we create a final stack of normalization factor nodes to the right of the butterfly factor nodes. The normalization constant  $Z$ , which is necessary to ensure that the transform is unitary, is given by  $Z = \sqrt{\sum_{i=0}^{2^I-1} 2^{2i}}$ .

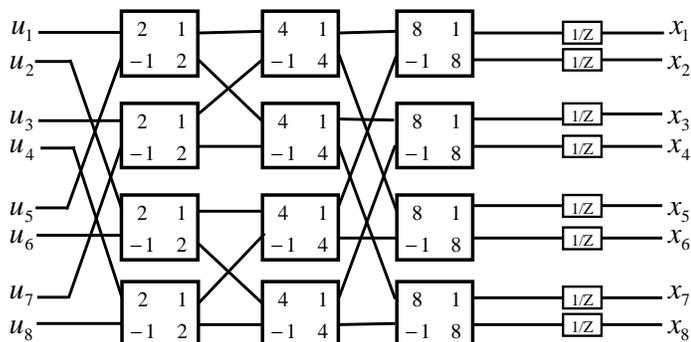


Figure 3: A factor graph for an 8 by 8 DUST implementing the transform  $x_l[t] = \sum_{m=1}^8 V_{lm}^8 u_m[t]$ .

These factor graphs provide a fast encoding method for DUSTs. The inputs  $u_m[t]$  enter on the right, and then one processes one stack of butterfly nodes at a time, followed by the stack of normalization nodes, to finally obtain the outputs  $x_l[t]$ , using a total number of computations of order  $L \log L$ .

It is easy to compute the DUST matrix corresponding to these factor graphs. The output results obtained when the  $i$ th input in the DUST factor graph is a 1 and all other

inputs are zeros corresponds to the  $i$ th row of the DUST matrix. For the 2 by 2 case, we just recover Ojard’s matrix. For the 4 by 4 case, we obtain the matrix

$$V = \frac{1}{\sqrt{85}} \begin{pmatrix} 8 & 2 & 4 & 1 \\ -2 & 8 & -1 & 4 \\ -4 & -1 & 8 & 2 \\ 1 & -4 & -2 & 8 \end{pmatrix}. \quad (6)$$

It is not hard to verify that the transpose of a DUST matrix is equal to its inverse, which means that the DUST matrices are unitary. However, the most attractive feature of DUSTs is their *disambiguating* property. Given  $L$  input QPSK symbols, each of the  $L$  outputs can take on  $4^L$  distinct values, one for each possible combination of inputs. This means that if even a single output is received with perfect reliability, all the inputs can be recovered.

For QPSK inputs, the constellation of possible values for each output will form a  $2^L$  by  $2^L$  square lattice grid. The mapping of inputs to constellation points for each output will be different from each other. Inputs that map to neighboring constellation points for one output will map to distant constellation points for other outputs. As emphasized by Tavildar and Viswanath, this is precisely the property desired in a good spreading transform [6]. To illustrate these facts, we show in figure 4 the constellations of possible output values for the 2 by 2 case, with QPSK inputs.

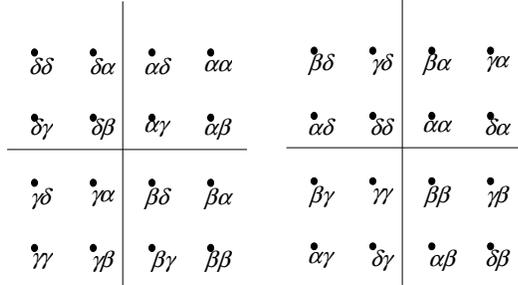


Figure 4: The constellations of possible output values for the two outputs of a 2 by 2 DUST matrix, with QPSK inputs  $\alpha = (1+i)/\sqrt{2}$ ,  $\beta = (1-i)/\sqrt{2}$ ,  $\gamma = (-1-i)/\sqrt{2}$ , and  $\delta = (-1+i)/\sqrt{2}$ . The labels on each constellation point are the values of the two inputs that give that output. The positions of the constellation points are  $+3/\sqrt{10}$ ,  $+1/\sqrt{10}$ ,  $-1/\sqrt{10}$ , and  $-3/\sqrt{10}$  on both axes.

### 3 Simulation Results

#### 3.1 Set-up

To simulate the DUSTs, in combination with error-correcting codes, we use the following set-up, illustrated by the factor graph shown in figure 5. We first encode  $k$  information bits, using an  $[N, k]$  block error-correcting code, into a codeword of length  $N$ . We then map the resulting codewords onto  $L = N/2$  QPSK symbols. The resulting block of  $L$  QPSK symbols is partitioned into  $P$  sub-blocks of length  $M = 2^I$ , where  $I$  is a non-negative integer, and then each of these sub-blocks are transformed using a  $M$  by  $M$

spreading transform (WHT or DUST). The resulting  $L$  complex symbols are concatenated and transmitted over the channel which fades them and adds noise as described by equation 1.

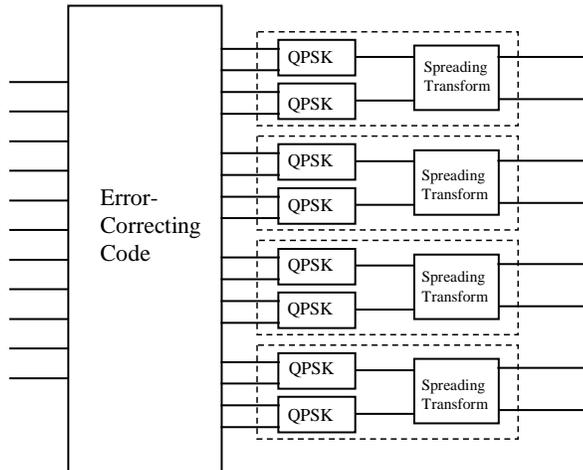


Figure 5: A factor graph that describes both the transmitter and receiver of our spread coding system. This particular factor graph corresponds to a  $[N = 16, k = 11]$  code, using 2 by 2 spreading transforms. The dashed boxes indicate the exact detectors described in the text; the inputs to a detector are the complex symbols on the right, and the outputs are *a posteriori* likelihood values for codeword bits exiting on the left.

Our receivers work as follows. For each  $M$  by  $M$  WHT or DUST, we use a detector which computes the exact *a posteriori* marginal likelihoods of the  $2M$  corresponding codeword bits, given the received complex symbols. This is done by summing over the likelihoods (given the received complex symbols) of every possible configuration of the  $2M$  codeword bits. This detector is thus also a demodulator and an equalizer, and is denoted by the dashed boxes in figure 5. The  $N$  marginal likelihoods that are output by the detector are then fed into a soft-input soft-decision decoder for the error-correcting code. Of course, using such an exact detector is only possible if  $M$  is small. We emphasize, however, that our receiver is not an optimal receiver even when the detector and error-correcting code decoder are separately optimal.

## 3.2 Results

The first set of simulations used a  $[N = 16, k = 11]$  extended Hamming code. One advantage of using this code is that we can compare our receiver to an overall optimal receiver, which computes the marginal probabilities of the codeword bits by summing over the probabilities of each of the  $2^{11} = 2048$  possible codewords.

In figure 6, we show our results, for the fast-fading Rayleigh channel, and using an overall-optimal decoder, for both WHTs and DUSTs. The DUSTs completely outperform the WHTs. Notice that even the 2 by 2 DUST outperforms the 8 by 8 WHT. Note also that the advantage of using an 8 by 8 DUST compared to a 4 by 4 DUST is not significant.

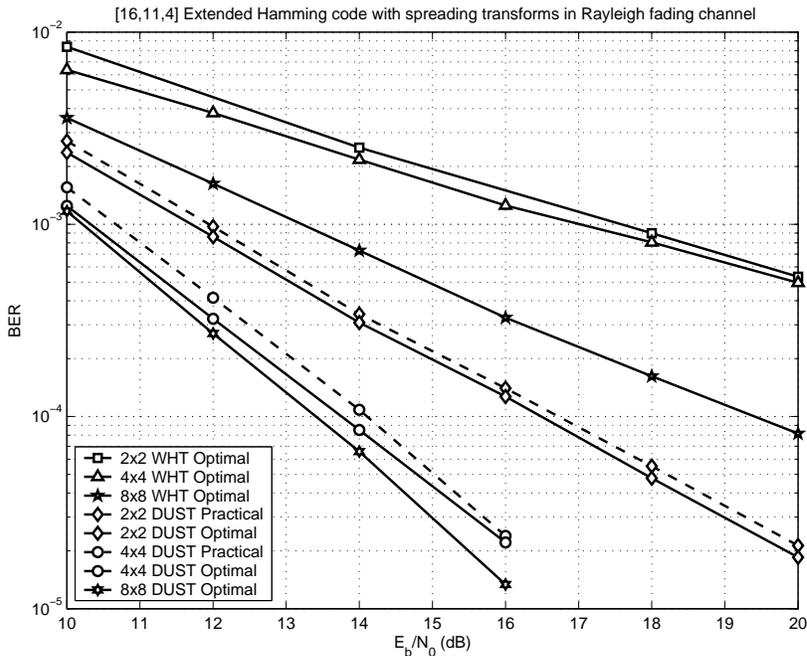


Figure 6: Performance of spreading transforms combined with [16,11,4] extended Hamming code on Rayleigh fading channel. The performance of practical decoders is given by the dashed lines. The definition of “practical” and “optimal” decoding is given in the main text.

In this figure, we also plot, for the 2 by 2 and the 4 by 4 DUSTs, the performance of “practical” receivers that function as described above, by separately doing exact detection of the DUSTs, followed by bit-optimal decoding of the extended Hamming code. Note that there is very little performance loss compared to an overall-optimal receiver. An exact 4 by 4 DUST detector must sum over  $2^8 = 256$  possible combinations of codeword bits, so it is within the realm of practical systems, but an exact 8 by 8 DUST detector would need to sum over  $2^{16} = 65536$  possibilities, and so is not very realistic.

In figure 7, we switch to the ‘erasures and noise’ channel, and use an  $[N = 3584, k = 3140]$  LDPC code obtained from MacKay’s database [9]. We used a high rate code because, for a weak code, the importance of the spreading transform is relatively greater. We use an exact detector as described above for the DUST or WHT transforms, and a standard sum-product decoder for the LDPC code, with 100 iterations.

As the figure shows, the disambiguating property of the DUST allows the error-correcting decoder to function, whereas the WHT can sometimes perform badly no matter how low the noise becomes. An intuitive explanation for the behavior at low noise is that a DUST detector will only fail if *all* the  $M$  inputs are erased. Thus, a DUST detector effectively turns an erasure rate of  $p_e$  into an erasure rate of  $p_e^M$ . Because a single correctly received output for a WHT is not normally useful, the error-correcting code attached to a WHT must deal with a much higher effective erasure rate.

As can be seen in figure 8, the performance difference between systems using WHTs and DUSTs is not so great on a Rayleigh channel, but is still noticeable. For both the Rayleigh and the “erasures and noise” channels, we are still far from the Shannon limit, even when we use DUSTs.

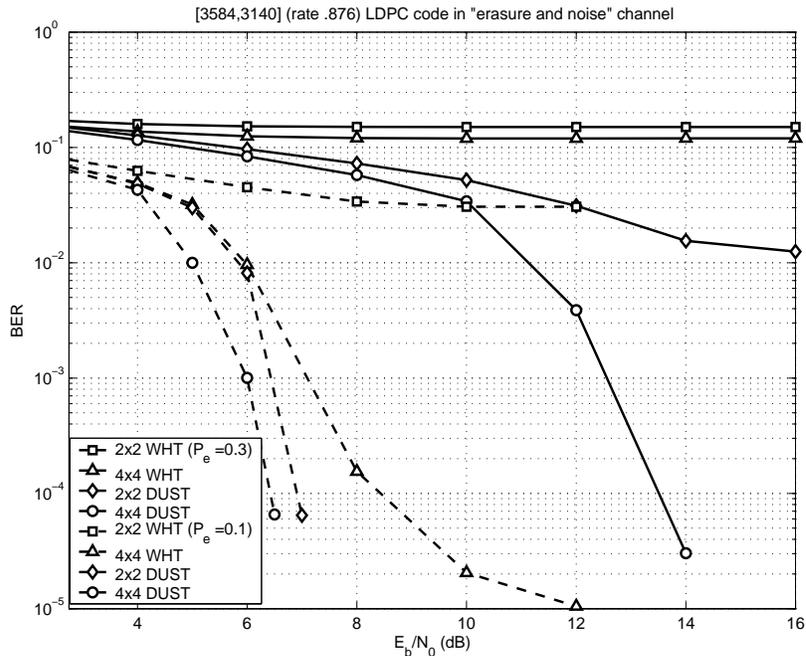


Figure 7: Performance of spreading transforms combined with an  $[N = 3584, k = 3140]$  LDPC code in “noise and erasures” channel. Dashed curves are for  $p_e = 0.1$ , and solid curves are for  $p_e = 0.3$ . Note that systems based on DUSTs function well at low enough noise, but those based on WHTs can fail completely. The Shannon limit (with no restriction on the signalling) for  $p_e = 0.1$  is 1.66 dB, and for  $p_e = 0.3$  it is 2.70 dB.

## 4 Discussion

We have restricted ourselves in the simulations to exact detectors, which naturally limits us to small DUSTs. Thus the fact that DUSTs have fast encoders for larger sizes becomes largely irrelevant. However, the factor graph representation of DUSTs gives some hope that good approximate detectors for larger DUSTs can be constructed. On the other hand, it only really makes sense to pursue this path if larger DUSTs significantly outperform the 4 by 4 DUSTs used here—the results shown in figure 6 cast some doubt on this notion.

## Acknowledgements

We thank Ben Vigoda for stimulating discussions, and Drs. Jin Zhang and Joe Marks for their support and encouragement. The valuable insights of Dr. Eric Ojard, from Broadcom Corporation, are gratefully acknowledged.

## References

- [1] D. Tse and P. Viswanath, *Fundamentals of Wireless Communications*, Lecture notes 2004.

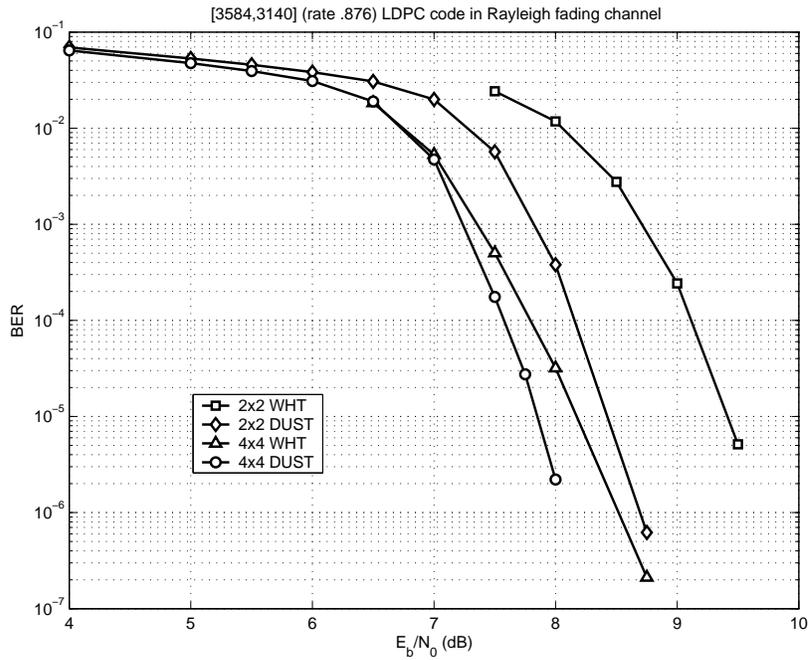


Figure 8: Performance of spreading transforms combined with an LDPC code in Rayleigh fading channel. The Shannon limit (with no restriction on the signalling) is 2.74 dB.

- [2] L. Hanzo, M. Munster, B.J. Choi, and T. Keller, *OFDM and MC-CDMA for Broadband Multi-User Communications, WLANs, and Broadcasting*, John Wiley & Sons, 2003.
- [3] K. Fazel, "Performance of CDMA/OFDM for Mobile Communications Systems," ICUPC 1993, vol. 2, pp. 975-979, 1993.
- [4] S. Kaiser, "Trade-off Between Channel Coding and Spreading in Multi-carrier CDMA Systems," Proc. ISSSTA 1996, pp. 1366-1370, 1996.
- [5] E. Ojard, (Broadcom), private communication 2004.
- [6] S. Tavildar and P. Viswanath, "Permutation Codes for the Parallel Fading Channel: Achieving the Diversity-Multiplexing Tradeoff," Proc. of CISS, pp. 526-531, 2004.
- [7] G.D. Forney, Jr., "Codes on Graphs: Normal Realizations," *IEEE Trans. on Info. Theory*, vol. 47, pp. 520-548, Feb. 2001.
- [8] J.S. Yedidia, "Sparse Factor Graphs Representations of Reed-Solomon and Related Codes," MERL Technical Report TR2003-135, available online at <http://www.merl.com/papers/TR2003-135/>, 2003.
- [9] D.J.C. Mackay, online database of codes. The code we used is available at <http://www.inference.phy.cam.ac.uk/mackay/codes/data.html#l130>.