International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:1, No:3, 2007

On the Symbol Based Decision Feedback Equalizer

Mohammed Nafie

***Abstract*—**Decision Feedback equalizers (DFEs) usually outperform linear equalizers for channels with intersymbol interference. However, the DFE performance is highly dependent on the availability of reliable past decisions. Hence, in coded systems, where reliable decisions are only available after decoding the full block, the performance of the DFE will be affected. A symbol based DFE is a DFE that only uses the decision after the block is decoded. In this paper we derive the optimal settings of both the feedforward and feedback taps of the symbol based equalizer. We present a novel symbol based DFE filterbank, and derive its taps optimal settings. We also show that it outperforms the classic DFE in terms of complexity and/or performance.

***Keywords*—**Coding, DFE, Equalization, Exponential Channel models.

I. Introduction

DECISION Feedback Equalizers (DFEs)[1][2][3] have long been used in digital communication systems. In channels with high intersymbol interference they outperform linear equalizers such as the zero forcing (ZF) [4] or minimum mean squared error (MMSE) equalizers [5]. Theoretical derivations of optimal DFE taps as well as its performance assume correct decisions at its output, and hence its estimated performance will be close to the actual performance only when this is the case. This will be true in non-coded systems and in high signal to noise ratio (SNR) regions. However, in coded systems, and since there might be a high disparity between the error performance at the chip level, and the error performance at the block level obtained after decoding the full block, the performance of classical DFEs, or chip based DFEs might suffer. The concept of a symbol based DFE was introduced in [6]. In a symbol based DFE, only reliable *decoded* bits are fedback into the feedback section of the equalizer and zeros would be fedback in between decoding blocks. In [6], symbol based DFE taps were derived based on least squares (LS) or least mean squares (LMS) training. However, no theoretical derivation of the optimal taps was presented. In this paper we derive the optimal taps of the symbol based DFE. Moreover, we introduce the novel symbol based DFE filterbank (SDFEF), and we theoretically derive the optimal taps of such a filterbank which can be shown to outperform the symbol based DFE of [6].

Manuscript received March 10, 2007

Mohammed Nafie is with the Department of Communications and Electronics, Cairo University, Egypt (phone: +2-010-6044766, e-mail: mnafie@eng.cu.edu.eg).

This paper is organized as follows. In the next section we briefly overview the classic DFEs. In section III, we briefly re-introduce the concept of symbol based DFEs and we derive the optimal settings of both the feedforward and feedback sections of the equalizer. In section 0, we introduce the concept of the symbol based DFE filterbank and explain that it would outperform the symbol based DFE. In section V we present the performance of our SDFEF over two different coded systems, one using a classic BCH [7] code, and another using a synthetic code utilizing Hadamard sequences [8]. We show that the performance of the SDFEF outperforms the chip based DFE over some regions of SNR while it always has lower complexity. We end with the conclusion in section VI.

II. Chip Decision Feedback Equalizers

DFEs have been studied extensively in the literature. A block diagram of a DFE is shown in Fig. 1.

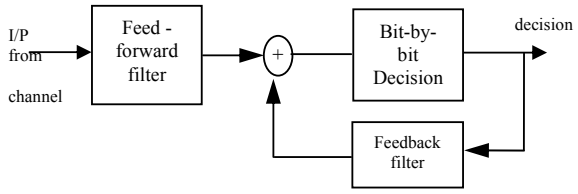

Fig. 1 Decision Feedback Equalizer

In this DFE, the input from the channel is passed through a feedforward filter. The output of the feedback filter is then subtracted from the output of the feedforward filter. The past decisions of the bits are fedback into this feedback filter. Hence, the feedforward filter is used to equalize part of the channel while the remaining inter-symbol interference is subtracted from its output using the feedback filter. Optimal filters for DFEs are infinite in length [8], and their taps depend on the channel, and signal and noise correlations. Since, we usually oversample the input from the channel; a fractional DFE was derived in [9]. Since infinite DFEs are not used in practice, optimal setting for a finite impulse response DFE was derived in [1]. Based on this derivation we will later derive our symbol based DFE and symbol based DFE filterbank.

DFEs are used in a variety of systems, and hence a multi-input-multi-output (MIMO) DFE was presented in [10]. Since

the noise at the output of optimal DFEs can be shown to be white, DFEs have also been shown to equalize the noise correlation and enhance the performance of frequency shift keying modulation as the one used in the Bluetooth system in [11].

III. Symbol Based DFEs

Symbol based DFEs are DFEs that get decisions fedback into them only after a whole symbol, or block, is decoded. In between symbols, zeros are fedback into the feedback section of the DFE. A block diagram of a symbol based DFE is shown in Fig. 2.

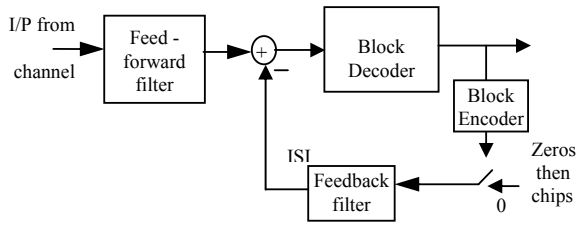

Fig. 2 Symbol DFE Block Diagram

Note that what gets fedback into the feedback filter is not direct decision on the output of the summer, but rather these outputs are first decoded using the block decoder to produce higher reliability bits, then these bits are then re-encoded to produce the bits, or chips, that can be used to subtract the ISI from the output of the feedforward filter. The encoder will output zeros for duration of one less than the length of the encoded block. Then at the instant the block is decoded and re-encoded, the full block will replace these zeros in the feedback filter. In [6], this DFE was used, and its taps were estimated using training techniques. In this paper, we will derive the optimal DFE taps from the point of view of minimizing the mean squared error at the output of the decision device.

Now, and following the terminology of [1], let the sampled input from the channel at time k be y_k. Then, we could say

$$y_k = Hx_k + n_k \tag{1}$$

where H is a matrix containing the channel coefficients, x_k is a vector with the transmitted bits, and n_k is the noise vector. Now, assume that the feedforward filter is the row vector w^* and that the feedback filter is the row vector b^*, and define $\tilde{b}^* = [0\ 0\ 0\ 0\ \ldots 1\ b^*]$ where the number of zeros is a variable to be optimized for. If the number of zeros is termed Δ, define the vector

$$x_k^i = [x_k(1)\ x_k(2)\ x_k(3)\ \ldots.\ x_k(\Delta)\ x_k(\Delta+1)\ 0\ 0\ \ldots\ x_k(\Delta+i+2)\ \ x_k(\Delta+i+2)\ \ldots\ ]^{\mathrm{T}}, \tag{2}$$

where $x_k(j)$ is the jth element of the vector x_k. Note that the number of zeros after the element $x_k(\Delta+1)$ are i zeros, and i is an index that takes values from 0 up to L-1, where L is the length of the code word, or the full block length. This index i would be 0 at the instant a block is decoded and is ready to be fedback into the feedback filter, and then it will be increased by 1 each time a new chip is decided upon, but its block is not yet complete for decoding and re-encoding.

Our mean squared error at a certain instant can hence be written as

$$MSE^i = \tilde{b}^*\ R_{xx}^i\ \tilde{b} - \tilde{b}^*\ R_{xy}^i w - w^* R_{yx}^i\ \tilde{b} + w^* R_{yy} w \tag{3}$$

where,

$R_{xx}^i = E(x_k^i x_k^{i*})$, $R_{xy}^i = E(x_k^i y_k^*)$, and $R_{yy} = E(y_k y_k^*)$. If we define, $R_{x^i x} = E(x_k^i x_k^*)$ and $R_{nn} = E(n_k n_k^*)$, then we could say $R_{xy}^i = R_{x^i x} H$, and $R_{yy} = H R_{xx}^0 H^* + R_{nn}$

Note that our equation (3) is equivalent to a similar one in [1], but only when the index i is set equal to 0. Our overall MSE now becomes, $\mathrm{MSE} = \sum_{i=0}^{i=L-1} MSE^i$, which using equation (3) can be written as

$$\mathrm{MSE} = \tilde{b}^*\ \mathrm{R}_{\mathrm{xx}}\ \tilde{b} - \tilde{b}^*\ \mathrm{R}_{\mathrm{xy}} w - w^* \mathrm{R}_{\mathrm{yx}}\ \tilde{b} + w^* R_{yy} w \tag{4}$$

where,

$$\mathrm{R}_{\mathrm{xx}} = \frac{1}{L}\sum_{i=0}^{L-1} R_{xx}^i,\ \mathrm{R}_{\mathrm{xy}} = \frac{1}{L}\sum_{i=0}^{L-1} R_{xy}^i, \text{ and } \mathrm{R}_{\mathrm{yx}} = \frac{1}{L}\sum_{i=0}^{L-1} R_{yx}^i.$$

We can then follow the derivation in [1] to obtain the optimal setting of the feedforward and feedback filters, but substituting with our matrices, R_{xx}, R_{xy}, .and R_{yx}. If we follow the same assumption as in [1], we would arrive at the following setting for the filters,

$$\tilde{b} = L e_{\Delta_{opt}} \tag{5}$$

where, $\mathrm{R}_{\mathrm{xx}}^{-1} + H^* R_{nn}^{-1} H = LDL^*$ and $e_{\Delta_{opt}}$ is a vector with zeros everywhere except at the element Δ_{opt} where it is equal to 1, and,

$$w^* = d_{\Delta_{opt}}^{-1}\ e_{\Delta_{opt}}^*\ L^{-1} H^* R_{nn}^{-1} \tag{6}$$

where $d_{\Delta_{opt}}$ is the Δ_{opt}-*th* element of the diagonal of the matrix D.

IV. Symbol Based DFE Filterbank

The motivation behind the symbol based DFE filterbank (SDFEF) to be introduced in this section stems from the fact that different vectors are filtered with the feedback section of

the DFE for each index, *i*. Specifically, a different number of zeros are fedback, or in other words, different positions of the signal vector x_k are not yet decided upon and hence are not filtered by the DFE filter at each index. Hence, one could envision a filterbank composed of several filters, both a feedforward section and a feedback section, such that we have a distinct filter for each index *i*. Hence, one would obtain a better MSE^i at each *i* than the case of the same symbol based DFE, and consequently a better overall total MSE. A block diagram of the SDFEF is shown in Fig. 3. There are *L* filters numbered from *0* to *L-1*. Note that at each index *i*, only filters numbered *i* are active, and hence the feedforward filter *i* gets its input from the channel and the feedback filter *i* gets its input from the block decoder. The summer *i* feeds the block decoder at index *i* as well. One has to note that the block decoder would produce a block, or symbol, decision when it gets the estimate of the last bit, or chip, of the symbol, indexed *L-1*. At this instant, it is able to produce a codeword estimate which gets re-encoded using the block encoder into *L* chips. These *L* chips can then be fedback into filter number 0. For subsequent indices, and since no codeword decision is yet ready, zeros are fedback into the filter similar to the symbol based DFE. The derivation of each feedforward and feedback filter taps can be achieved via optimization of equation (3) for each index *i*. Hence we would have *L-1* optimization problems to solve. One can also assume that the feedback filter taps are set to zero in the positions corresponding to unknown chips, and hence be able to use matrices R_{xx}^0 .and R_{xy}^0 through all the optimization without having to compute all *L-1* versions of these matrices. In this way one can derive the optimal taps of the feedforward and feedback filters as follows. Using the matrix defined in [1], $R_{x/y}^{\perp} = R_{xx} - R_{xy}^0 R_{yy}^{-1} R_{yx}^0$, we define the matrix, R_{Δ}^i, where,

$$R_{\Delta}^i = [0_{(N-\Delta-i-1)X\Delta} \; e_1 * e_1^{\mathrm{T}} \; 0_{(N-\Delta-i-1)X\, i} \; I_{(N-\Delta-i-1)X(N-\Delta-i-1)}]$$

$$* R_{x/y}^{\perp} * \begin{bmatrix} 0_{\Delta X(N-\Delta-i-1)} \\ e_1^{\mathrm{T}} * e_1 \\ 0_{i\,X(N-\Delta-i-1)} \\ I_{(N-\Delta-i-1)X(N-\Delta-i-1)} \end{bmatrix}$$

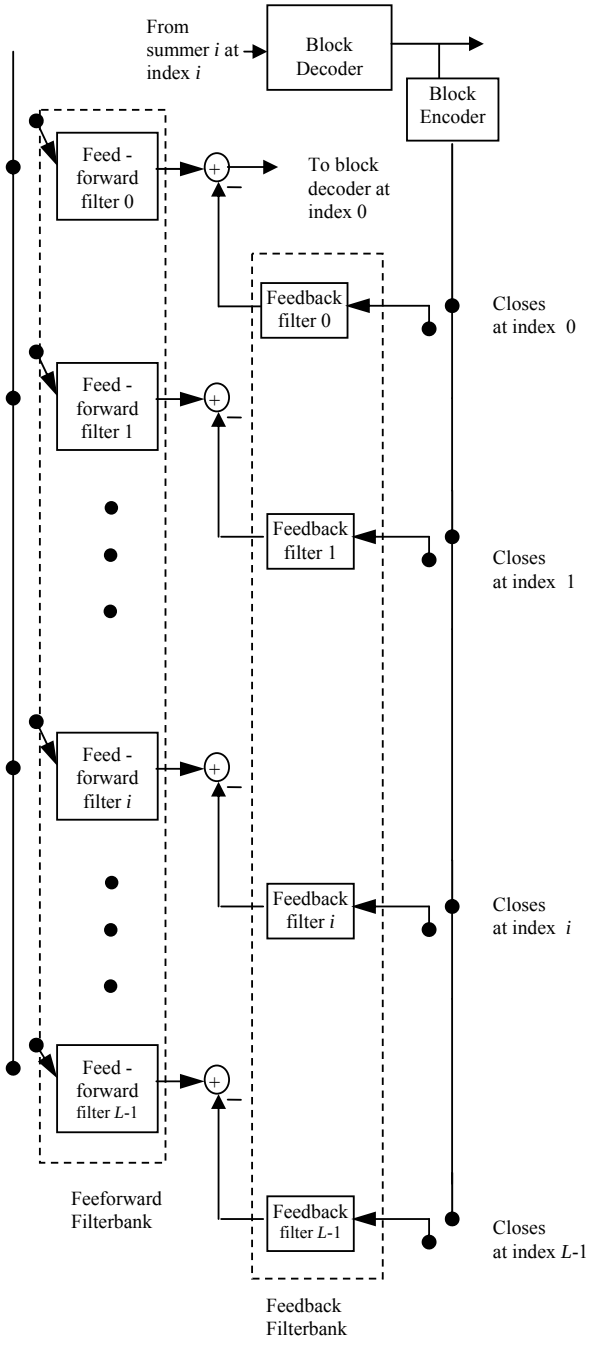

Fig. 3 Symbol-based DFE Filterbank (SDFEF)

Then we could say,

$$MSE^i = [1 \; b^{i*}] * R_{\Delta}^i * \begin{bmatrix} 1 \\ b^i \end{bmatrix}. \qquad (7)$$

We can then minimize each MSE^i to obtain the *L-1* feedback filters, and can then substitute in

$w^{i*} = [1\ b^{i*}]R_{xy}^{0}R_{yy}^{-1}$ to obtain the *L-1* feedforward filters. Note that the value of the delay, Δ, should be optimized for each *i* and we need not assume a constant Δ throughout. Our simulations actually show that the value of optimal Δ, that which achieves a minimum *MSE,* does in fact differ with the values of *i*. Note that also, we cannot here assume the same assumption as in [1], namely that the number of feedback taps is equal to the channel memory. Here, we allow the number of feedback taps to change to use the maximum possible number of feedback taps to obtain the optimal settings minimizing the *MSE*, and hence we use a number equal to $N-\Delta^i-i-1$ where Δ^i is the optimal value of Δ at index *i*. One can see that if the value of Δ^i is the same for all *i*'s, then the number of feedback taps will decrease by 1 for each increase in *i*. However, our simulations show that the optimal delay at larger *i*'s is smaller than that at lower *i*'s, and hence the number of feedback taps at higher *i*'s need not always be smaller than number of feedback taps used at lower *i*'s.

V. Simulation Results

In this section, we will present some results showing the performance of our symbol based DFEs. In these results we use the exponential Rayleigh fading channel represented by a number of taps, each of which has a Rayleigh distribution. The zero delayed path is at the highest average power and the average power of the following paths decreases exponentially. In this channel model, the total power of all the paths is normalized to one for every channel realization. This channel is represented by

$$h_k = N(0, \tfrac{1}{2}\sigma_k^2) + jN(0, \tfrac{1}{2}\sigma_k^2), \sigma_k^2 = \sigma_0^2 e^{-kT_s}$$

$$\sigma_0^2 = 1 - e^{-T_s}, \sum \sigma_k^2 = 1$$

where h_k is the *i*th tap, and T_s is the ratio between the sampling interval and the delay spread of the channel. The number of paths depends on T_s, and in the simulation results shown below we use a T_s value of approximately 0.45 and a number of taps of 23. We also use QPSK modulation.

Fig. 4 compares the average probability of error performance of both the chip DFE where we still correct the decisions after each symbol, and the performance of our SDFEF, for a communications system utilizing the BCH(15,5) code[7]. The BCH decoder uses hard decisions of the bits constituting the BCH codeword, obtained via applying a threshold to the output of the summer of the DFE. The "EbNo" shown in our simulation results are chip signal to noise ratio.

Fig. 4 shows that our proposed symbol based DFE filterbank outperforms the classical chip DFE. Both techniques use the same length feedforward filter and our SDFEF uses the same length of the chip DFE feedback filter at index 0, but shorter filters for other indices. Our average feedback filter length was calculated to be 9.68, while the length of the chip DFE feedback filter was 12, which means that it achieves better performance at lower complexity.

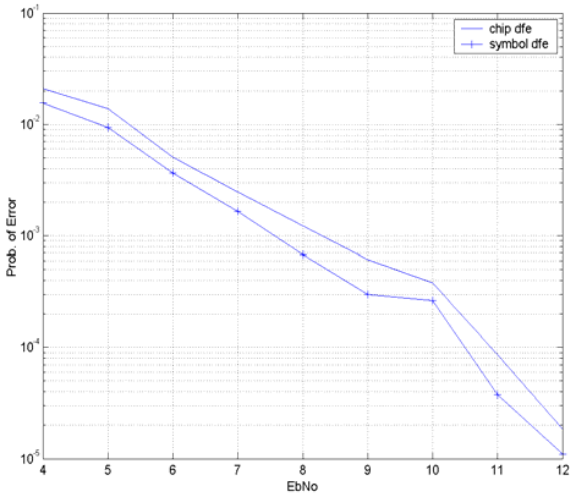

Fig. 4 Probability of error for BCH (15,5)

In Fig. 5, we show the ninetieth percentile probability of error performance, which means that 90% of the channels achieve better performance than the shown performance. Our SDFEF still outperforms the chip DFE.

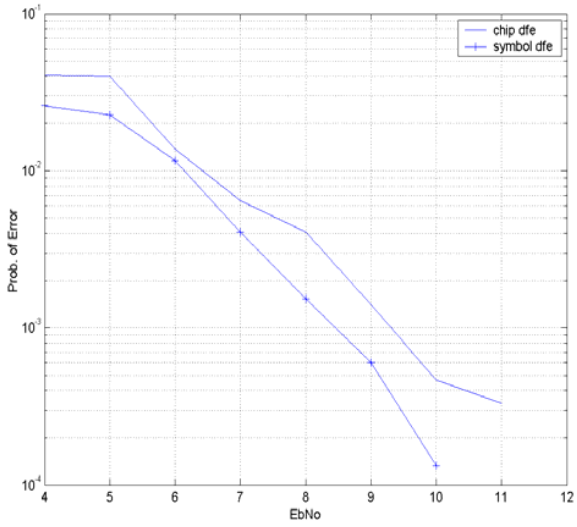

Fig. 5 Ninetieth-percentile probability of error for BCH (15,5)

One more important advantage of our SDFEF is that we can more accurately estimates the achievable MSE at the output of the DFE. One can see this from Fig. 6, where we show the estimated mean squared error calculated as a sum of $MSE^{(i)}$ using equation (7) for the SDFEF, and using the mean squared error equation in [1] for the chip DFE. We can see that we can

almost perfectly estimate the mean squared error of the SDFEF, while there is a gap between the estimated and actual errors in case of the chip DFE. This would have implications if, for example, soft decisions from this DFE are used to feed a decoder in another stage which makes us of the value of the noise variance.

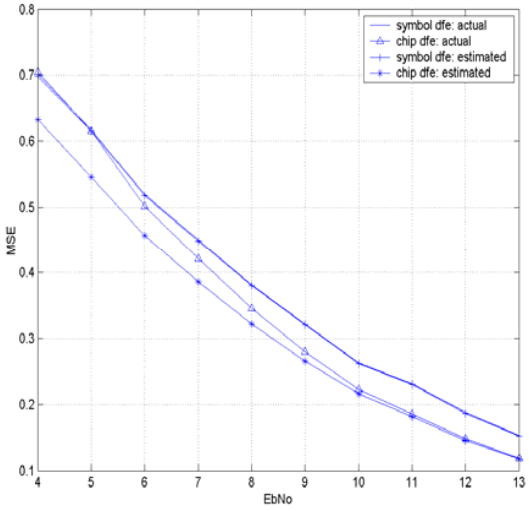

Fig. 6 MSE for BCH(15,5)

We have also tested our SDFEF in a scenario where 4 bits are mapped into one of 16 complex Hadamard sequences [8], each of length 16. Again, the SDFEF has lower complexity than the chip DFE as the average feedback filter of the SDFEF was calculated to be 9.16, while that of the chip DFE is 13. We used two decoding techniques, the first utilizing correlation between the possible 16 Hadamard sequences, and the vector composed of 16 values from the summer of the DFE. The decoded vector is then the vector achieving the highest correlation value. For this technique, the average probability of error is shown in Fig. 7, and the ninetieth percentile probability of error is shown in Fig. 8. We can see that, at higher SNRs, our SDFEF outperforms the chip DFE. Also, and similar to the BCH case, using the SDFEF allows a more accurate estimate of the mean squared error at the output of the DFE as shown in Fig. 9. Note that the decoder we used does not utilize this information, and in fact assumes a constant noise variance for each chip, which is not actually the case. One would expect that if we did utilize the mean squared error information, we could obtain even better performance.

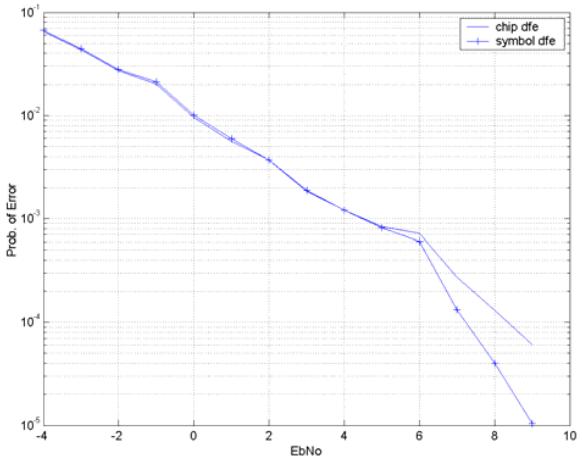

Fig. 7 Probability of error for Hadamard sequences

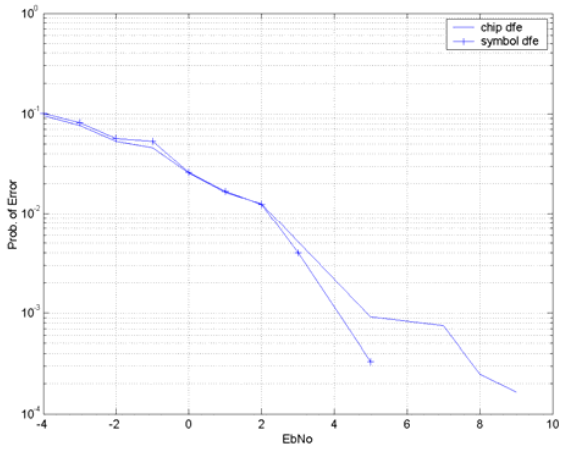

Fig. 8 Ninetieth-percentile probability of error for Hadamard sequences

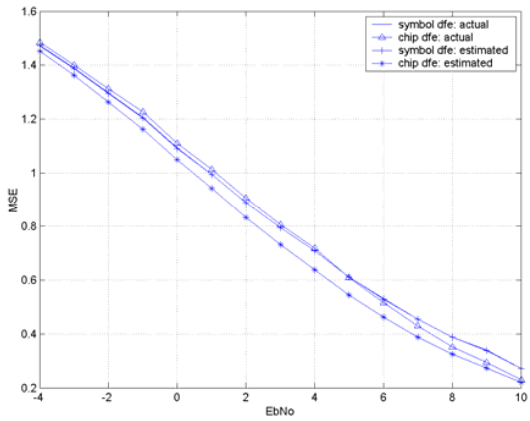

Fig. 9 MSE for Hadamard Sequences

The second decoding technique uses the hard decisions of the output of the summer, and feeds these hard decisions to the decoder utilizing the correlation. Again, our SDFEF achieves better performance than the chip DFE. The average probability of error is shown in Fig. 10. The SDFEF is shown to outperform the chip DFE.

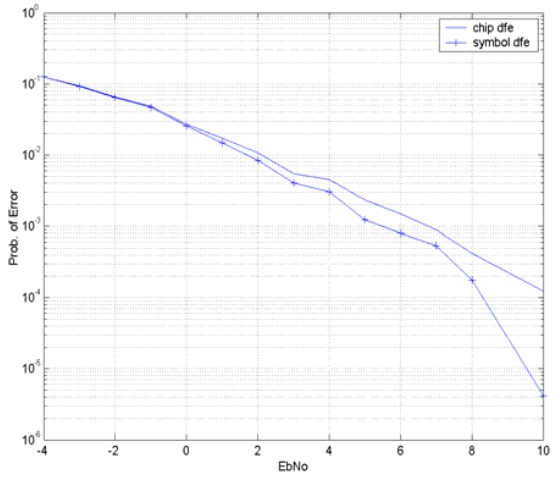

Fig. 10 Probability of error for Hadamard sequences using hard decisions

VI. Conclusion

In this paper, we presented the theoretical derivation of the optimal settings of the symbol DFE equalizer. The symbol DFE equalizer is fedback with zeros when a symbol is not yet decoded rather than being fed with the unreliable estimates of the undecoded chips We also presented the new concept of the symbol based DFE filterbank, a DFE that uses a different filter for each position within the symbol, such that it has a filter that is optimized for the number of zeros that it is fed. This allows this SDFEF to obtain lower overall mean squared error, and hence lower expected probability of error. We presented equations to obtain the optimal settings of the filters of the SDFEF at each position within the codeword.

Results showing the performance of the SDFEF were also shown. The results indicates that the SDFEF outperforms the chip DFE for the test cases we shown, where one of them utilized a BCH(15,5) code and the other utilized Hadamard sequences of length 16 as the codewords. We showed that the SDFEF outperforms the chip DFE in both cases of soft and hard decision decoding for the Hadamard case. Our test cases were simulated using the exponential channel model, but we can expect that the SDFEF will still outperform the chip DFE in other types of channel models and other codes as well. We also showed that the estimate of the mean squared error at the output of the SDFEF is more accurate than that at the output of the chip DFE, which allows a better use of this estimate in subsequent stages of the receiver.

References

[1] N. Al-Dhahir, and J. Cioffi, "MMSE Decision-Feedback Equalizers: Finite-Length Results," IEEE Trans. On Info. Theory, vol. 41, No. 4, pp. 961-975, July 1995.

[2] D. George, R. Bowen, and J. Storey, "An Adaptive Decision-Feedback Equalizer," IEEE Trans. On Commun. Tech., vol. 19, pp. 283-293, June 1971.

[3] P. Monsen, "Feedback Equalization for Fading Dispersive Channels," IEEE Trans. On Info. Theory, vol. 17, pp. 56-64, Jan. 1971.

[4] R. Lucky, "Techniques for Adaptive Equalization of Digital Communications," Bell Systems Technical Journal, Vol. 45, pp. 255-286, April 1965.

[5] B. Widrow, "Adaptive Filters, I: Fundamentals," Technical Report No. 6764-6, Stanford University, CA, Dec. 1966.

[6] M. Nafie, M. Khairy and K. Shawky, "Equalizer Training in IEEE 802.11b Standard," IEEE Midwest Symposium on Circuits and Systems, Cairo, Egypt, Dec. 2003.

[7] S. Lin, and D. Costello, "Error Control Coding, Fundamentals and Applications," Prentice Hall, 1983.

[8] J. Proakis, *Digital Communications*, 4th Edition, McGraw-Hill, New York, 1999.

[9] A. Gatherer, "The Effect of Microreflections on Decision Feedback Equalization," IEEE Trans. On Sig. Proc, vol. 45, No. 1, pp. 228-231, Jan. 1997.

[10] A. Duel-Hallen, "Equalizers for Multiple Input/Multiple Output Channels and PAM Systems with Cyclostationary Input Sequences," *IEEE J-SAC*, vol. 10, pp. 630-639, April 1992.

[11] M. Nafie, A. Gatherer and A. Dabak, "Decision Feedback Equalization for Bluetooth Systems," Proc. Int. Conf. on Acoustics, Speech and Sig. Proc., Salt Lake City, UT, May 2001.