

Multi-Directional Self-Iterating Soft Equalization for 2D Intersymbol Interference

Jaehyeong No and Jaekyun Moon

Department of Electrical Engineering

KAIST

Daejeon, Korea

Email: eee4u@kaist.ac.kr, jmoon@kaist.edu

Abstract—This paper focuses on two-dimensional (2D) soft-input soft-output (SISO) equalization to mitigate intersymbol interference (ISI) among symbols that arise within a 2D array of data cells. The proposed method is based on arranging multiple component equalizers to exchange soft information with one another to enhance decision quality in an iterative manner. The component equalizers are simple one-dimensional linear equalizers running in different directions and do not perform well enough individually in the challenging 2D ISI environment, but working together, they consistently reach high-quality decisions. Performance comparison is made with the reduced-state trellis-based equalizers as well as the conceptually straightforward 2D linear equalizers. The results indicate excellent complexity/performance trade-off options for the proposed scheme.

I. INTRODUCTION

Inter-symbol interference (ISI) in communications and storage channels often presents major obstacles in reliable data reception/recovery. ISI that arises in all directions in 2D array of cells has been a serious recent concern for flash memory devices [1], [2] as well as next-generation high-density disk drive channels [3]-[5].

Much work has been done on 2D ISI equalization [6]-[9], and achieving desirable trade-off between equalizer performance and efficiency of the algorithm continues to be a major issue. The basic idea underlying our present research is to concatenate multiple one-dimensional (1D) SISO equalizers that are simple yet, when working together, would produce reliable decisions. The idea of constructing a strong self-iterating equalizer based on multiple, relatively weak equalizers, has been suggested for severe 1D ISI channels in [10]. Here, we also consider forming a high-performing equalizer based on more than one component equalizer but our target is 2D ISI and we specifically employ 1D linear equalizers running in different radial directions as component equalizers. Since our equalizer is made up of multiple component equalizers among which soft information is iterated, a natural view is that self-iteration goes on within our equalizer. Given this view as well as the fact that the component equalizers run in multiple directions, we will call the proposed equalizer the multi-directional self-iterating soft equalizer (MD-SISE). The concept of combining multiple equalizers running in different directions has already been considered for 2D ISI applications in [8] but there each component equalizer is a trellis-based scheme with substantial complexity.

In order to validate performance of the proposed method, we make comparisons with the Bahl-Cocke-Jelinek-Raviv (BCJR) equalizer [11] running on a reduced-state trellis representation of the 2D ISI channel as well as the 2D linear equalizer with taps deployed in all radial directions. The simulation results show that the proposed scheme is superior. While the MD-SISE can be used in conjunction with an outer soft decoder (i.e., in turbo equalization setting), we will consider only the equalization side in this paper.

In Section II, a 2D ISI channel with a hexagonal ISI mask pattern is defined. In Section III, 2D equalizers based on the linear minimum mean-squared-error (MMSE) method with taps positioned in all radial directions and the BCJR algorithm running on a reduced-state trellis representation of 2D ISI are discussed. The proposed MD-SISE method is also presented in this section. Section IV shows bit error rate (BER) simulation results. Section V draws conclusions.

II. CHANNEL ASSUMPTION

Consider 2D data symbols with an ISI mask shown in Fig. 1. The islands represent cells where the data is stored. The ISI mask is assumed to be of a hexagonal shape here, although the algorithms we consider in this paper can be applied to any ISI pattern. In the read process, the ISI mask scans the 2D array of data cells, and the read signal corresponding to a given position of the mask is simply a linear combination of the stored values for the cells captured within the mask. For the particular ISI mask shown in the figure, there are six interfering edge cells affecting the center cell. In our present ISI model, the amount of interference is simply characterized by the single parameter ‘ c ’ that controls the weights on the interference. More precisely, the noise-corrupted read signal is written as

$$y_k = \sum_{i=0}^l h_i x_{k-i+l/2} + n_k \quad (1)$$

where l represents the size of interference mask (or the number of cells captured by the mask at a time) and fixed to 6 through out this paper. The coefficients h_i 's are the weights on the cells as shown in Fig. 1. The normalization factor s is given by $1/\sqrt{6c^2 + 1}$, as the total energy in h_i 's is constrained to be one. The symbol $x_i \in \{\pm 1\}$ represents the stored binary data and n_k is the sample of zero-mean

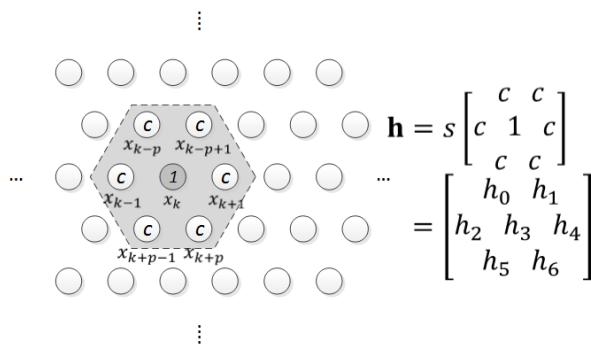


Fig. 1. 2D array of data symbols with ISI mask and matrix

additive white Gaussian noise (AWGN). Note that the single subscript for x indicating the cell position contains both the row position information and column position information with an understanding that p is the size of each row. This single-subscript position representation makes it convenient to take advantage of known 1D equalizer designs. We assume that the time-invariant “channel response” weights h_i ’s are known by the equalizer.

III. 2D EQUALIZATION

A. 2D MMSE Equalizer

A conceptually straightforward 2D linear MMSE equalizer puts the taps on the cell positions directly. In fact, we can make a direct use of the classical 1D MMSE method by rewriting the channel matrix \mathbf{h} in Fig. 1 as a 1D vector:

$$\mathbf{h} = s [c \quad \cdots \quad c \quad 1 \quad c \quad \cdots \quad c]^T. \quad (2)$$

We can also rewrite (1) using (2):

$$\mathbf{y}_k = \mathbf{x}_k^T \mathbf{h} + n_k \quad (3)$$

where $\mathbf{x}_k^T = [x_{k-p} \quad x_{k-p+1} \quad \cdots \quad x_{k+p}]$ represents the channel input vector. Let z_k denote the 2D linear equalizer output and write:

$$z_k = \mathbf{y}_k^T \mathbf{g} \quad (4)$$

where $\mathbf{y}_k^T = [y_{k-p} \quad y_{k-p+1} \quad \cdots \quad y_{k+p}]$ is the channel output vector and \mathbf{g} is a vector of equalizer filter coefficients.

Using a channel matrix notation with zeros inserted as in

$$\mathbf{H}^T \triangleq s \begin{bmatrix} c & c & 0 & c & 1 & c & 0 & 0 & c & c & 0 & 0 & \cdots & 0 \\ 0 & c & c & 0 & c & 1 & c & 0 & 0 & c & c & 0 & \cdots & 0 \\ \vdots & & & & \ddots & & & & & & & & & \vdots \\ 0 & \cdots & 0 & c & c & 0 & 0 & 0 & c & 1 & c & 0 & c & c \end{bmatrix}$$

and filling both sides of the channel input vector to match the dimensions so that $\mathbf{x}_k^T = [x_{k-2p} \quad x_{k-2p+1} \quad \cdots \quad x_{k+2p}]$, we arrive at a more compact channel output equation:

$$\mathbf{y}_k^T = \mathbf{x}_k^T \mathbf{H} + \mathbf{n}^T \quad (5)$$

where $\mathbf{n}^T = [n_{k-p} \quad n_{k-p+1} \quad \cdots \quad n_{k+p}]$ is a vector of the AWGN samples. The dimension of the channel matrix \mathbf{H}

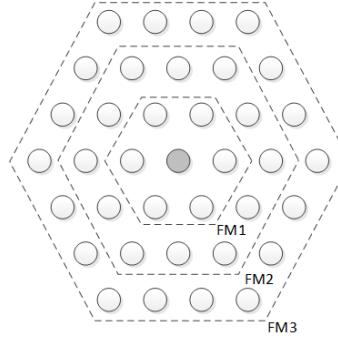


Fig. 2. Filter masks for 2D linear equalization

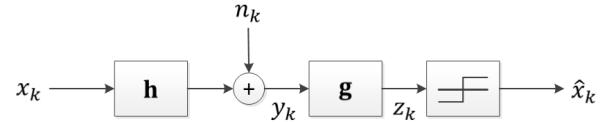


Fig. 3. ISI channel with 2D linear equalizer

is $\{N + \sqrt{3(4N - 1)} + 3\} \times N$, where N is the number of equalizer taps used (or the size of the filter mask). Zeros should be inserted appropriately into the channel matrix \mathbf{H} in consideration of the dimensions of \mathbf{x}_k and \mathbf{y}_k , for the chosen size of the filter mask. The channel input/output relationship as written in (5) is no different from that of a standard 1D ISI channel and thus all known 1D equalizer designs can be applied directly.

In particular, classical linear MMSE design yields the filter coefficients [12]:

$$\mathbf{g} = (\mathbf{H}^T \mathbf{R}_{xx,k} \mathbf{H} + \mathbf{R}_{nn,k})^{-1} \mathbf{H}^T \mathbf{R}_{xx,k} \mathbf{e} \quad (6)$$

where $\mathbf{R}_{xx,k}$ and $\mathbf{R}_{nn,k}$ are covariance matrices for the channel input and noise, respectively:

$$\mathbf{R}_{xx,k} = \text{diag} [v_{k-2p} \quad \cdots \quad v_k \quad \cdots \quad v_{k+2p}]$$

$$\mathbf{R}_{nn,k} = \text{diag} [w_{k-p} \quad \cdots \quad w_k \quad \cdots \quad w_{k+p}]$$

with v_k and w_k denoting the variances for the input symbol and noise; and $\mathbf{e} = [0 \quad \cdots \quad 0 \quad 1 \quad 0 \quad \cdots \quad 0]^T$. Note that while $\mathbf{R}_{xx,k}$ and $\mathbf{R}_{nn,k}$ are expressed as time-varying matrices in general, for classical linear MMSE design, they are set to $\mathbf{R}_{xx,k} = \text{diag} [1 \quad \cdots \quad 1 \quad \cdots \quad 1]$ and $\mathbf{R}_{nn,k} = \text{diag} [\sigma_n^2 \quad \cdots \quad \sigma_n^2 \quad \cdots \quad \sigma_n^2]$, where σ_n^2 is the variance of the noise that is assumed to be stationary. We also use these time-invariant covariance matrices for our 2D linear MMSE equalizer simulations. As will be discussed shortly, though, for the MD-SISE development $\mathbf{R}_{xx,k}$ will in principle assume a time-varying form.

We set three different sizes for the filter mask (to set the number of taps to be utilized) for 2D linear MMSE equalizer simulation in the present paper. See Fig. 2. Filter masks FM1/FM2/FM3 consists of 7/19/37 taps, in comparison with the 7-cell ISI mask assumed. Fig. 3 shows a block diagram

model for the noisy 2D ISI channel and the 2D linear MMSE equalizer. It is assumed that the ISI channel coefficients and the AWGN samples are real-valued, and the equalizer knows the noise variance in addition to the channel response \mathbf{h} .

B. 2D BCJR Equalizer

We now wish to discuss the BCJR algorithm running on a trellis diagram representation of 2D ISI. There is a major difficulty here, however. The 2D ISI we are considering is not decomposable (i.e., the D-transformation of the 2D channel impulse response cannot be expressed as the product of two D-transformations along orthogonal directions). Because of this, a finite-state machine description is not available that completely characterizes the 2D ISI channel. The consequence is that no finite-state trellis diagram exists on which the maximum-likelihood sequence detection or the maximum *a posteriori* probability detection can be executed. Another way of looking at this difficulty is that neither the maximum likelihood function nor the *a posteriori* probability can be computed in a recursive fashion with complexity independent of the growing input data size.

The only option seems to be to force a suboptimal, reduced-state finite-state-machine description. To this end, we force another mask that defines the number of states in the assumed trellis. See the “state masks” shown in Fig. 4. Three different sizes of state masks are shown. The white islands correspond to the inputs symbols captured within a single state at a given time while the black islands represent new input symbols. The state at a given time along with the set of input symbols define a set of noiseless channel outputs (just u_k for state mask 1; u_{k-p}, u_k for state mask 2; and u_{k-p}, u_k, u_{k+p} for state mask 3) corresponding to the ISI mask position k . As the mask moves in the horizontal direction, state transitions occur, corresponding to the walk along a particular path in the trellis made up of all possible states and state transitions. It is easy to see that the number of states depends exponentially on the number of white islands ($2^4 = 16$, $2^6 = 64$ and $2^8 = 256$ for the three state masks shown in the figure) while the number of competing branches arriving at each state node is an exponential function of the number of black islands (8, 16 and 32, respectively, for the three state masks shown). The state masks shown in Fig. 4 are by no means the only possibilities, but the choices taken here for the state mask shapes and the white/black partitions are a reasonable one.

The number of partially overlapping ISI masks within the state mask defines the number of channel observation samples needed to compute each branch metric. The noiseless channel output u_k is given by

$$u_k = \mathbf{x}'_k^T \mathbf{h} \quad (7)$$

where \mathbf{x}' is the vector of appropriate input symbols determined by the given state mask. The branch metrics are calculated using the observations y_k and the noiseless channel outputs u_k as

$$\sum_{i=0}^{m-1} (y_{k_i} - u_{k_i})^2 \quad (8)$$

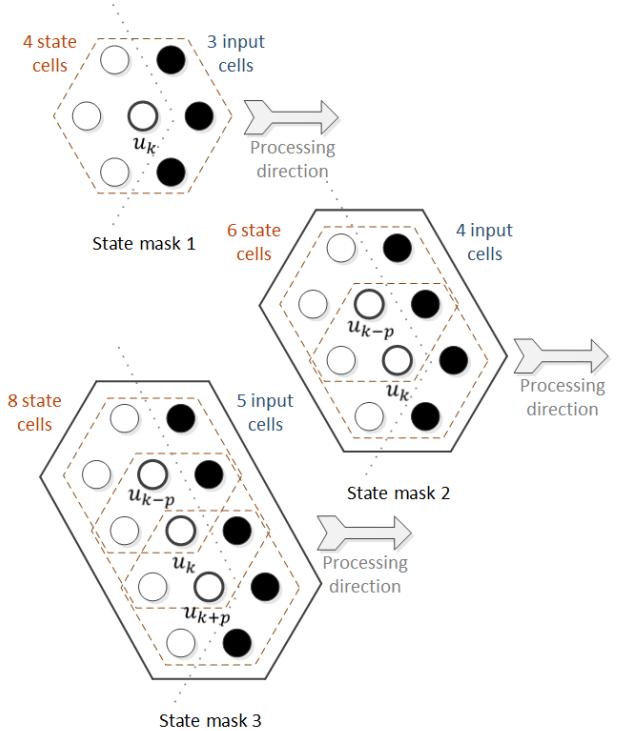


Fig. 4. State masks for the 2D BCJR algorithm

where m is the number of channel observations utilized in the calculation. Once the trellis is defined and the branch metric computation method is established, it is straightforward to execute the BCJR algorithm, although the results would not yield the quality of the optimal *a posteriori* probability estimator. To approach optimal performance, the size of the state mask must be large, but complexity increases quickly as the mask size increases. For example, suppose we wish to utilize seven channel observations (i.e., $m = 7$) within a state mask (like FM1 in Fig. 2), the state mask will contain 14 state cells and 5 input cells. In this case, the trellis diagram will have 16,384 states and 32 branches arriving at each state node, the complexity of which is clearly out of hand. In this paper, we attempt simulations corresponding to the three state masks shown in Fig. 4.

C. Multi-Directional Self-Iterating Soft Equalizer

The proposed equalizer consists of multiple constituent 1D linear equalizers which generate soft decisions and exchange them with one another in the form of extrinsic information. More specifically, each component equalizer is basically a linear equalizer running in one of three directions: horizontal, diagonal and reverse-diagonal. Each component equalizer generates and passes extrinsic information that becomes *a priori* symbol information for the next component equalizer. This process continues until decision quality no longer improves or enough overall iterations have taken place. See Figs. 5 and 6.

The component equalizer basically takes the form of the soft linear MMSE equalizer devised in [13]. Consider the

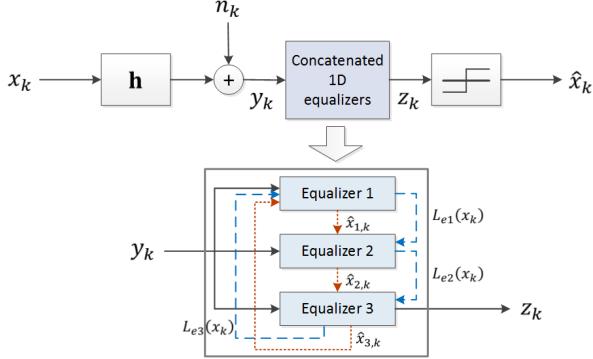


Fig. 5. Self-iterating equalizer based on multiple constituent 1D equalizers

equalizer along the horizontal direction as shown in Fig. 6. The dark gray islands correspond to the tap positions of the 1D linear equalizer. The gray ones above and below the dark gray islands can be viewed as the source of offtrack interference as far as the 1D equalizer running on the dark gray islands is concerned. The offtrack interference is first canceled via hard-decision feedback utilizing best up-to-date soft information in the collaborating component equalizers. Given the rotational symmetry of the hexagonal ISI pattern, the component equalizers running in other directions work in exactly the same way. The direction along which the dark gray islands are aligned represents the path for the component equalizer in each of three subfigures in Fig. 6.

Going back to Fig. 5, cancellation of offtrack signals using decisions from the previous component equalizer is implied in the inflow of the estimated hard decisions $\hat{x}_{j,k}$, where $j \in \{1, 2, 3\}$ points to a particular component equalizer. The passing of the extrinsic information $L_{e,j}$ to the next component equalizer is also shown. Note that the passing of both the extrinsic information and the hard decisions occur in blocks, after each component equalizer completely scans the entire 2D array of cells.

The component 1D equalizer design is based on the classical MMSE method that utilizes only the second order statistics of the underlying random sequences including the input symbol sequence. As suggested in [13], if the means and variances of the input symbols can be estimated based on extrinsic information coming from other processors (e.g., decoder or some other collaborating equalizer), then it is generally beneficial to make use of these estimated sequences of input symbol means and variances in computing the optimal equalizer tap weights. With this design principle in mind and after suppressing the use of the current symbol's *a priori* information (i.e., *intrinsic* information) in an effort to generate only the *extrinsic* information, the expressions for the equalizer output in (4) and the equalizer tap weights in (6) are modified to yield:

$$z_{e,j}(x_k) = \mathbf{g}_k^T (\mathbf{y}'_k - \mathbf{H}^T \bar{\mathbf{x}}_k + \bar{x}_k \mathbf{s}) \quad (9)$$

where \mathbf{y}'_k is the observation vector after offtrack interference cancellation, the over-bar denotes the statistical mean (or a

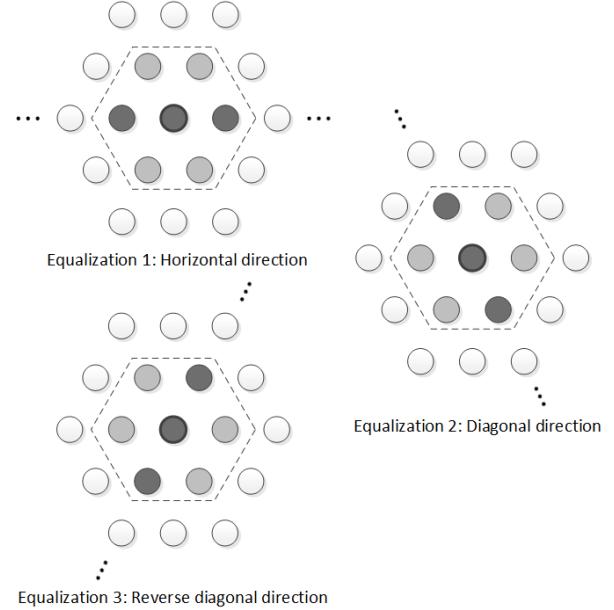


Fig. 6. Equalization directions for three component 1D equalizers

vector of means when placed over a vector) and $\mathbf{s} = \mathbf{H}^T \mathbf{e}$. The first subscript e for z emphasizes that the output contains only the extrinsic information. The equalizer taps are given by

$$\mathbf{g}_k = \{\sigma_n^2 \mathbf{I}_N + \mathbf{H}^T \mathbf{R}_{xx,k} \mathbf{H} + (1 - v_k) \mathbf{s} \mathbf{s}^T\}^{-1} \mathbf{s} \quad (10)$$

where σ_n^2 is the noise variance, \mathbf{I}_N is the identity matrix of size N with N denoting the number of taps used in each component linear equalizer and v_k is the variance for each input symbol.

The mean and the variance are calculated as

$$\bar{x}_k = \frac{e^{L_{e,j}(x_k)} - 1}{1 + e^{L_{e,j}(x_k)}}$$

and

$$v_k = 1 - |\bar{x}_k|^2,$$

respectively, where $L_{e,j}(x_k)$ is the received extrinsic information from the previous component equalizer in the form of log-likelihood ratio (LLR).

Given $z_{e,j}(x_k)$, each component equalizer generates new extrinsic LLR information by scaling it:

$$L_{e,j}(x_k) = \frac{2z_{e,j}(x_k)}{1 - \mathbf{s}^T \mathbf{g}_k} \quad (11)$$

where the denominator of the right side can be shown to be the variance of the equalizer error at the output of the equalizer [13].

For the first self-iteration, the *a priori* probability values for component equalizer 1 are all set to 1/2 (i.e., $L_{e,3}(x_k) = 0$), and $\hat{x}_{3,k}$ is determined by simple threshold detection of the channel observations. In order to pass hard-decision estimates for offtrack interference cancellation in the next component equalizer, the *a priori* LLR and the newly generated $L_{e,j}(x_k)$ are combined to construct $\hat{x}_{j,k}$.

Notice that the equalizer coefficients (10) are necessarily time-varying (TV). This is a consequence of the fact that the estimated mean and variance change for each input symbol. The TV filter becomes highly impractical as the tap coefficients need to be recalculated entirely at every processing symbol interval. Fortunately, a practical solution has also been provided by the authors of [13]. Namely, for each iteration only the average mean and average variance are taken over the entire data array and used in the computation of the filter taps. In this way, the tap weights are recalculated only when the corresponding component equalizer gets restarted after receiving the sequence of extrinsic LLRs from the previous component equalizer; during the equalizer's output generation in its turn, the tap weights are fixed and time-invariant. Let us call this equalizer a quasi-time-invariant (QTI) equalizer. Its tap weights can be shown to be

$$\mathbf{g}_{qti} = \left(\sigma_n^2 \mathbf{I}_N + \mathbf{H}^T \bar{\mathbf{R}}_{xx} \mathbf{H} \right)^{-1} \mathbf{s} \quad (12)$$

where $\bar{\mathbf{R}}_{xx}$ is the matrix replacing all variance terms appearing in $\mathbf{R}_{xx,k}$ with a time-averaged version $\bar{v} = 1/M \sum_{i=0}^{M-1} v_i$ (M is the data array size) except that v_k is forced to 1 (i.e., $\bar{\mathbf{R}}_{xx} = \text{diag} [\bar{v} \dots \bar{v} \ 1 \ \bar{v} \dots \bar{v}]$). We will show the simulation results using both the TV filters and the QTI filters. The performance losses associated with the QTI filters are only marginal.

IV. SIMULATION RESULTS AND DISCUSSION

Figs. 7 and 8 show the BER versus SNR curves for the three equalization methods described in the previous section for mild ISI of $c = 0.1$ as well as relatively severe ISI of $c = 0.3$. For useful reference points, we also present two curves: for zero ISI ($c = 0.0$) and for no equalization efforts whatsoever. Of the nine curves corresponding to equalization, three correspond to 2D linear MMSE equalizer with three different filter masks (FM1, FM2, FM3), three to 2D BCJR running on three different state masks (SM1, SM2, SM3) and the remaining three to the proposed MD-SISE with 1, 2 and 3 rounds of self-iterations. The label "No ISI" means the channel output is simply $y_k = x_k + n_k$, and its BER curve represents the fictitious matched filter bound (MFB). The BER for "No Equalization" is obtained from hard-slicing of y_k in (1) without any equalization efforts. For the proposed scheme, 1 round of self-iterations means that each of three component equalizers gets to process the entire array of the channel observations once. Also, for each component equalizer, only three taps are allowed. We observe that for the given ISI mask assumed in this work, little is gained by taking more taps. The SNR is defined as the total signal power captured within the ISI mask, which is normalized to 1 in this work, divided by the variance of the additive noise n_k .

A. Equalization for $c = 0.1$

For $c = 0.1$, performance gaps among three 2D linear equalizers are not obvious. In particular, the performance improvement for FM3 is negligible. 2D BCJR with SM1 or

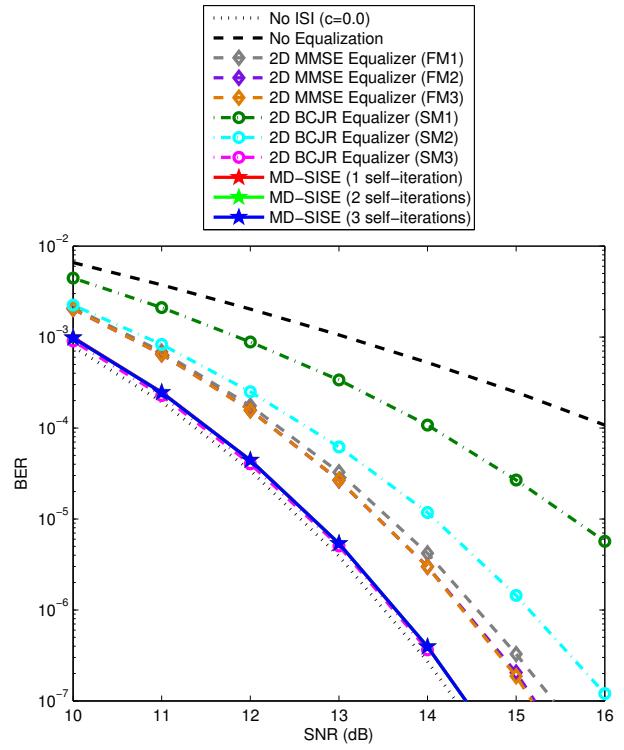


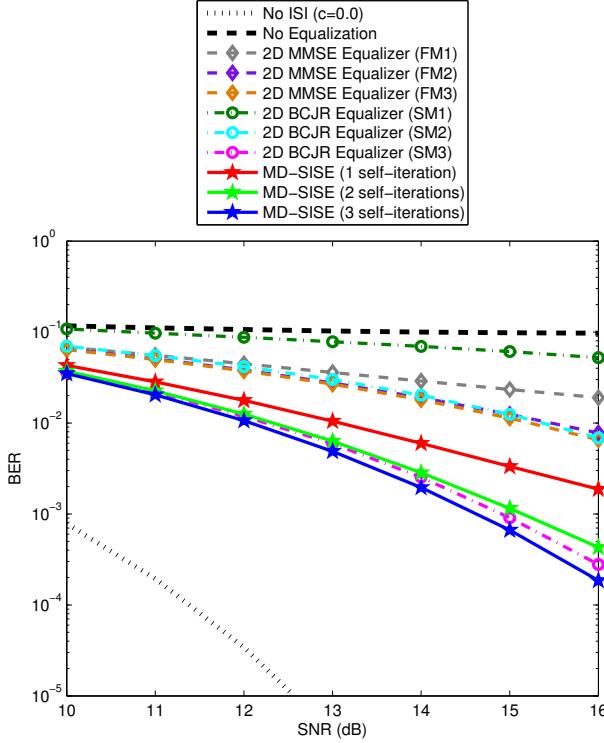
Fig. 7. Equalization for $c = 0.1$

SM2 is worse than any 2D linear MMSE method, whereas 2D BCJR with SM3 closely approaches the zero-ISI performance. Notice that MD-SISE with only 1 round of self-iteration gives nearly the zero-ISI performance for this mild ISI channel.

While the taps in MD-SISE in Fig. 7 are TV, the performance curves shown in Fig. 9 indicate that for this channel the use of QTI taps incurs no significant performance loss. Overall we conclude that for the $c = 0.1$ channel the proposed scheme with just 1 round of self-iteration performs better than all the 2D linear equalizers, which are more complex, as well as the 2D BCJR methods with state masks SM1 and SM2, which are considerably more complex. The proposed MD-SISE consisting of three component equalizers each with just three filter taps gives after only 1 round of self-iterations nearly the same performance as 2D BCJR with SM3, which runs on a 256-state trellis with 32 competing branches for each state node.

B. Equalization for $c = 0.3$

As ISI increases, the 2D linear MMSE equalizers show poor performance even with the largest filter mask, as seen in Fig. 8. As for 2D BCJR, using state mask SM2 it performs better than 2D linear MMSE with FM1, but requires the larger state mask SM3 to exceed the performance of 2D linear schemes with filter masks FM2 and FM3. With SM3, 2D BCJR also

Fig. 8. Equalization for $c = 0.3$

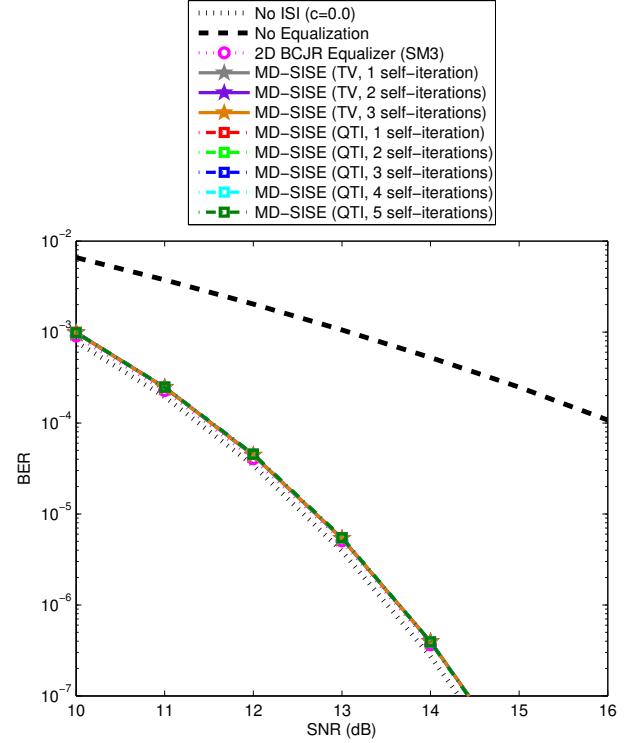
gives lower error rates than MD-SISE up to 2 rounds of self-iterations, but with 3 rounds of self-iterations MD-SISE again exceeds 2D BCJR with even the largest state mask. We note, however, that all schemes fall well short of the MFB without turbo equalization. Also, compared to the 2D linear equalizers, MD-SISE with just 1 round of self-iteration outperforms them all.

As seen in Fig. 10, for the $c = 0.3$ channel, the QTI taps show slight performance degradations, requiring more iterations to make up for the performance loss. Overall, however, it is clear that the MD-SISE schemes with practical QTI taps still offer considerably more favorable performance-complexity trade-off options.

Finally, Fig. 11 shows how the error rate curve gets lowered with an increasing number of iterations. The figure also shows how the performance improves as the process moves from one component equalizer to next. The curves show that 3 rounds provide nearly the full performance potential of the MD-SISE scheme for this particular channel.

V. CONCLUSION

A low-complexity equalization technique has been proposed for handling 2D ISI. The idea is based on multiple simple 1D linear equalizers working together via iterative exchange of soft information to eventually reach highly reliable decisions. For hexagonal 2D ISI channels and using component linear

Fig. 9. MD-SISE with QTI filter taps: $c = 0.1$

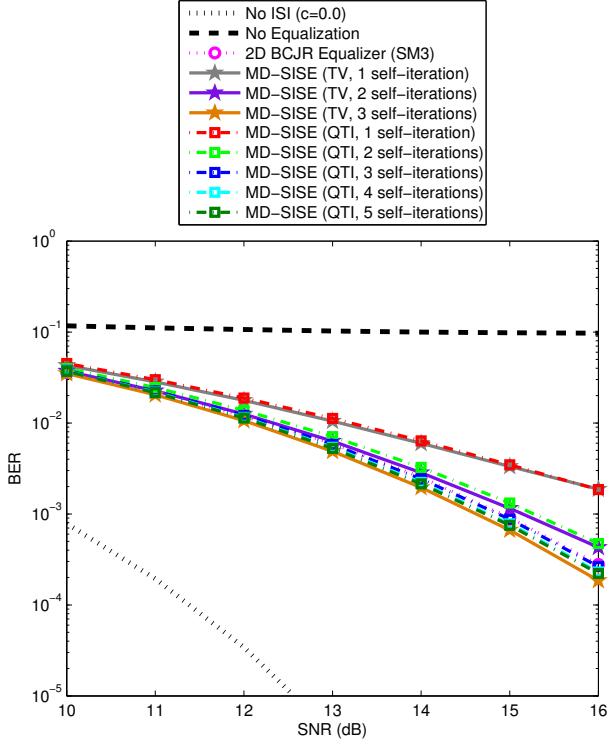
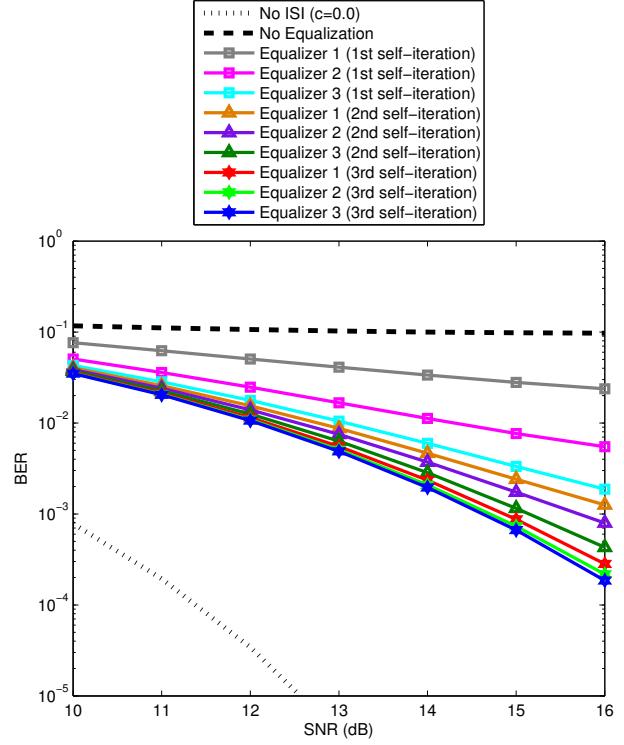
equalizers running in three different directions, we show that the proposed idea yields better performance compared to much more complex 2D linear equalizers based on 2D filter masks and even more complex 2D BCJR detectors operating on suboptimal finite-state-machine 2D ISI models. Interesting future work includes performance evaluation in turbo setting in conjunction with an outer error correcting code as well as investigating the potentials of other simple equalizers like the decision feedback equalizer as constituent equalizers.

ACKNOWLEDGMENT

This work has been supported in part by the NRF of Korea under grant no. 2012-0008719 and the ASTC.

REFERENCES

- [1] J. Moon, J. No, S. Lee, S. Kim, and J. Yang, "Statistical analysis of flash memory read data," in *Proc. IEEE GLOBECOM*, Houston, TX, Dec. 2011.
- [2] J. Moon, J. No, S. Lee, S. Kim, S. Choi, and Y. Song, "Statistical Characterization of Noise and Interference in NAND Flash Memory," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 8, pp. 2153-2164, Aug. 2013.
- [3] T. Rausch *et al.*, "HAMR Drive Performance and Integration Challenges," *IEEE Trans. Magnetics*, vol. 49, no. 2, pp. 730-733, Feb. 2013.
- [4] S. Nabavi *et al.*, "Two-Dimensional Pulse Response and Media Noise Modeling for Bit-Patterned Media," *IEEE Trans. Magnetics*, vol. 44, no. 11, pp. 3789-3792, Nov. 2008.

Fig. 10. MD-SISE with QTI filter taps: $c = 0.3$ Fig. 11. Performance improvement with iteration in MD-SISE with TV taps: $c = 0.3$

- [5] M.Yamashita *et al.*, “Read/Write Channel Modeling and Two-Dimensional Neural Network Equalization for Two-Dimensional Magnetic Recording,” *IEEE Trans. Magnetics*, vol. 47, no. 10, pp. 3558-3561, Oct. 2011.
- [6] Y. Wu, J. A. O’Sullivan, N. Singla, and R. S. Indeck, “Iterative detection and decoding for separable two-dimensional intersymbol interference,” *IEEE Trans. Magnetics*, vol. 39, no. 4, pp. 2115-2120, July 2003.
- [7] M. Marrow, and J. K. Wolf, “Iterative detection of 2-dimensional ISI channels,” in *Proc. IWT 2003*, Paris, France, Mar. 2003, pp. 131-134.
- [8] Y. Chen, T. Cheng, P. Njeim, B. Belzer, and K. Sivakumar, “Iterative soft decision feedback zig-zag equalizer for 2D intersymbol interference channels,” *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 2, pp. 167-180, Feb. 2010.
- [9] Y. Chen, B. Belzer, and K. Sivakumar, “Iterative row-column soft-decision feedback algorithm using joint extrinsic information for two-dimensional intersymbol interference,” in *Proc. 44th Conf. on Info. Sci. and Syst. (CISS 2010)*, Princeton, NJ, Mar. 2010.
- [10] S. Jeong, and J. Moon, “Self-Iterating Soft Equalizer,” to appear in *IEEE Trans. Comm.*
- [11] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Trans. Inform. Theory*, vol. 20, no. 2, pp. 284-287, Mar. 1974.
- [12] S. Haykin, *Communication Systems*, 3rd ed. New York, USA: Wiley, 1994.
- [13] M. Tuchler, R. Koetter, and A. C. Singer, “Turbo Equalization: Principles and New Results,” *IEEE Trans. Comm.*, vol. 50, no. 5, pp. 754-767, May 2002.