Target Design and Low Complexity Signal Detection for Two-Dimensional Magnetic Recording

Chaitanya Kumar Matcha and Shayan Garani Srinivasa
Department of Electronic Systems Engineering,
Indian Institute of Science,
Bangalore, 560012, India.
E-mail: {mchaitanyakumar, shayan.gs}@dese.iisc.ernet.in

Abstract—**Partial response maximum likelihood (PRML) scheme is a well known technique to equalize the data read from 1D magnetic recording channels. The PRML scheme uses a linear equalizer followed by a maximum likelihood (ML) detector. This paper is novel in addressing the following aspects: a) We propose two different methods to design separable and non-separable 2D PR targets that help in signal detection. b) We propose an extension of 1D Viterbi detector for signal detection in 2D ISI channels. We use the detector to study the efficacy of PR targets designed for a particular choice of 2D ISI channel.**

Index Terms—**2D Partial response target design, 2D ISI, MMSE, Viterbi detector, 2D SOVA detector**

I. Introduction

Magnetic recording channels are characterized as slowly time varying due to the wear and tear, temperature variations and other factors. PRML scheme provides a solution to this by having a linear equalizer that adapts to varying channel conditions so that the signal detection is not impacted. This scheme further helps by reducing the extent of ISI to a predefined target response as seen by the detector.

There are several well known methods for PR target design [1], [2] for 1D ISI channels. In 1D ISI channels, the computational complexity of detectors grow exponentially with the number of taps, while the performance is a function of signal-to-noise ratio (SNR). Hence, the design approaches deal with maximizing SNR or minimizing the mean square error (MMSE) for a finite length target. Additional considerations that were studied in [1] include a) taps with integer coefficients to aid quantization, b) unit energy constraint and c) monic constraint. The integer coefficient constraint was shown to give poor performance compared to the other two constraints.

Design of 2D symbol detectors and soft detectors is non trivial and NP hard. Several 2D detection algorithms approximating ML criterion are proposed motived by the 1D Viterbi and BCJR algorithms [3], [4], [5]. Detection scheme using a multi-trach equalizer followed by 1D Viterbi detector for bit-patterned media is proposed in [3]. Separability of the channel response can help in significantly reducing the computational needs of the detectors as proposed in [4], at the cost of performance. Row and column detectors combined with a 2D equalizer was proposed in [5] that iteratively achieves MAP performance.

Generalized belief propogation (GBP) algorithm studied in [6], [7] provides a different approach for signal detection. The GBP algorithm uses message passing between regions instead of message passing between nodes as done in the well known belief propagation algorithm. The GBP algorithm is known to have very high computational complexity and its performance in relation to the ML criterion is not well understood.

In this paper, we look into the MMSE criterion to jointly design PR target and the equalizer for 2D ISI channels. We propose methods to design separable 2D PR targets under monic and unit energy constraints. These PR targets can be used along with the low complexity detectors specifically tuned for separable targets.

We propose a 2D symbol detection algorithm that extends the the 1D hard decision based Viterbi algorithm [8], [9]. We further extend this algorithm toward soft updates similar to the 1D soft output Viterbi algorithm (SOVA) [10]. We compare the performance and complexity of our approach to that in [5] through simulations. Similar to 1D SOVA in [10], the proposed algorithm provides soft outputs with just a forward pass and is suitable to be used along with forward error correction.

This paper is organized as follows. In Section II, we describe the 2D ISI channel model and the notations used in this paper. In Section III, we discuss techniques to design non-separable and separable 2D PR targets. In Section IV, we propose a low complexity ML based 2D signal detection algorithm, and analyze its performance and complexity. In Section V, we study the performance of various 2D PR targets designs through simulations. Section VI concludes the paper.

II. Channel Model

A 2D ISI channel can be modeled as a linear time-invariant (LTI) system with additive noise as

$$y(i,j) = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \hat{h}(m,n) a(i-m, j-n) + w(i,j), \qquad (1)$$

where $\hat{h}(m,n)$ is the 2D response of the channel, $w(i,j)$ are noise samples, $a(i,j)$ and $y(i,j)$ are the input symbols and received samples respectively. Defining $h(i,j) = \hat{h}(-i,-j) \forall i,j$,

We thank the Department of Science of Technology (DST), India for generously supporting this work under the grant SERB/F/3371/2013-14.

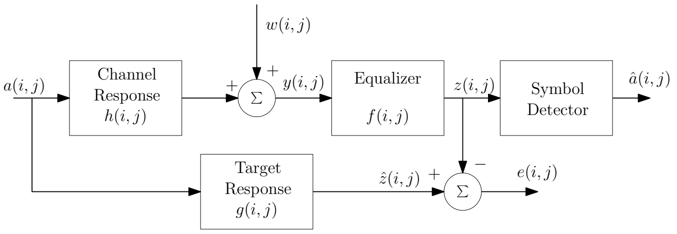


Figure 1. Partial response target design. The samples received from the channel are equalized to match the target response.

we can write (1) as

$$y(i,j) = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} h(m,n)a(i+m,j+n) + w(i,j). \quad (2)$$

We call this representation of $h(i,j)$ as the channel response "mask". $w(i,j)$ is the white noise sample at the position (i,j). The noise characteristics depend on the nature of the channels. For example, in magnetic recording channels studied in [11], [16], noise due to position jitter is significant. For the design techniques and algorithms discussed in this paper, we restrict our discussion to additive white Gaussian noise.

Figure 1 shows the channel model used for PR target design. The channel is modeled as linear ISI channel with additive white Gaussian noise. In the PRML scheme, the samples received from the channel are equalized using a PR equalizer achieving an overall response approximating the PR target response.

We will use following notation through out the paper:

Channel, target and equalizer responses are represented as 2D masks as in (2).

$a(i,j)$, $i,j = \cdots,-2,-1,0,1,2,\cdots$ defines the "plane" of input symbols. Typically, for practical constraints, the plane is of finite dimensions i.e., $i = 0,\cdots N_i$ and $j = 0,\cdots N_j$, for sufficiently large N_i and N_j.

$h(i,j)$ defines the ISI introduced by the channel as defined in (2). The set of coordinates (i,j) defines the shape of the mask. For example, $\{(i,j) \mid -1 \leq i \leq 1, -1 \leq, j \leq i\}$ defines a 3×3 rectangular mask. Let $\mathbf{H}$ be a 2D matrix whose elements are $h(i,j)$. Note that the rows and columns of $\mathbf{H}$ can be indexed with negative integers with $h(0,0)$ being the center tap.

$y(i,j)$ for $i,j = \cdots,-2,-1,0,1,2,\cdots$ defines the received samples. The received samples as a function of the input symbols and the channel response is given in (2).

The PR target is a finite size mask denoted by $g(i,j)$ where (i,j) takes values depending on the choice of PR target mask shape. Let $\mathbf{G}$ denote the 2D matrix whose elements are $g(i,j)$ with $g(0,0)$ being the center tap.

Similarly, the equalizer mask is denoted by $f(i,j)$, where (i,j) takes values depending on the choice of PR equalizer mask shape. Let $\mathbf{F}$ denote the 2D matrix whose elements are $f(i,j)$ with $f(0,0)$ being the center tap.

The output at the equalizer is

$$z(i,j) = \sum_{n_i,n_j=-\infty}^{\infty} f(n_i,n_j)\, y(i+n_i, j+n_j). \quad (3)$$

The desired output at equalizer as per the partial response is

$$\hat{z}(i,j) = \sum_{n_i,n_j=-\infty}^{\infty} g(n_i,n_j)\, a(i+n_i, j+n_j). \quad (4)$$

The error at position (i,j) is

$$e(i,j) = \hat{z}(i,j) - z(i,j). \quad (5)$$

In order to simplify the above equations and the facilitate the usage of matrix algebra, we introduce the vectorization operator $\mathrm{vec}(\cdot)$ that creates a column vector from the elements of a matrix by linearly indexing them row wise.

Let $\underline{v}_{\mathbf{X}}^{(i,j)}$ denote a vector for some matrix $\mathbf{X}$, where the elements $v(i,j)$ are ordered similar to the operation $\mathrm{vec}(\mathbf{X})$ considering $v(i,j)$ as the center element. We use this notation for $v(i,j)$ representing data samples or symbols at any stage of processing.

For example, if $g(i,j)$, with $i,j \in \{0,1\}$ is vectorized as

$$\mathrm{vec}(\mathbf{G}) = \begin{bmatrix} g(0,0) & g(0,1) & g(1,0) & g(1,1) \end{bmatrix}^T,$$

then $\underline{a}_{\mathbf{G}}^{(i,j)}$ is formed by similarly ordering $a(i,j)$ as

$$\underline{a}_{\mathbf{G}}^{(i,j)} = \begin{bmatrix} a(i,j) & a(i,j+1) & a(i+1,j) & a(i+1,j+1) \end{bmatrix}^T.$$

We also use following vector notation for the responses of the channel, equalizer and the PR target:

$$\underline{h} = \mathrm{vec}(\mathbf{H}), \qquad \underline{f} = \mathrm{vec}(\mathbf{F}), \qquad \underline{g} = \mathrm{vec}(\mathbf{G}).$$

While the matrix representation of a mask is appropriate for a rectangular shaped masks, the vector representation is suitable for masks of any shape and size.

With the vector representations, (2) can be written as

$$y(i,j) = \underline{h}^T \underline{a}_{\mathbf{F}}^{(i,j)} + w(i,j).$$

Similarly, the equations (3), (4) and (5) can be written as

$$\begin{array}{rcll} z(i,j) & = & \underline{f}^T \underline{y}_{\mathbf{F}}^{(i,j)}, & (6) \\ \hat{z}(i,j) & = & \underline{g}^T \underline{a}_{\mathbf{G}}^{(i,j)}, & (7) \\ e(i,j) & = & \underline{f}^T \underline{y}_{\mathbf{F}}^{(i,j)} - \underline{g}^T \underline{a}_{\mathbf{G}}^{(i,j)}. & (8) \end{array}$$

III. Two Dimensional Partial Response Target Design

The goal is to minimize the error signal $e(i,j)$ achieved by the equalizer as compared to the target chosen. We use the Minimum Mean Squared Error (MMSE) criterion to jointly design the partial response target and the equalizer.

A. Equalizer design

Assuming all values to be real numbers ($\mathbb{R}$), the mean squared error is

$$\begin{array}{rcl} \mathscr{E} & = & \mathbb{E}\left[|e(i,j)|^2\right]. \\ \mathscr{E} & = & \mathbb{E}\left[\left|\underline{f}^T \underline{y}_{\mathbf{F}}^{(i,j)} - \underline{g}^T \underline{a}_{\mathbf{G}}^{(i,j)}\right|^2\right]. \\ \mathscr{E} & = & \underline{f}^T \mathbb{E}\left[\underline{y}_{\mathbf{F}}^{(i,j)} \left(\underline{y}_{\mathbf{F}}^{(i,j)}\right)^T\right] \underline{f} + \underline{g}^T \mathbb{E}\left[\underline{a}_{\mathbf{G}}^{(i,j)} \left(\underline{a}_{\mathbf{G}}^{(i,j)}\right)^T\right] \underline{g} \\ & & -2\underline{f}^T \mathbb{E}\left[\underline{y}_{\mathbf{F}}^{(i,j)} \left(\underline{a}_{\mathbf{G}}^{(i,j)}\right)^T\right] \underline{g}. \qquad (9) \end{array}$$

Assuming that the channel is stationary, define

$$\begin{aligned} \mathbf{R}_{aa} &= \mathbb{E}\left[\underline{a}_{\mathbf{G}}^{(i,j)}\left(\underline{a}_{\mathbf{G}}^{(i,j)}\right)^{T}\right], \\ \mathbf{R}_{yy} &= \mathbb{E}\left[\underline{y}_{\mathbf{F}}^{(i,j)}\left(\underline{y}_{\mathbf{F}}^{(i,j)}\right)^{T}\right], \\ \mathbf{R}_{ya} &= \mathbb{E}\left[\underline{y}_{\mathbf{F}}^{(i,j)}\left(\underline{a}_{\mathbf{G}}^{(i,j)}\right)^{T}\right]. \end{aligned}$$

We can write (9) as

$$\mathscr{E} = \underline{f}^T \mathbf{R}_{yy} \underline{f} + \underline{g}^T \mathbf{R}_{aa} \underline{g} - 2\underline{f}^T \mathbf{R}_{ya} \underline{g}. \tag{10}$$

Note that $\mathbf{R}_{yy}$ and $\mathbf{R}_{aa}$ are auto-correlation matrices and hence are positive definite matrices.

Re-writing (10) as,

$$\begin{aligned} \mathscr{E} &= \left(\underline{f} - \mathbf{R}_{yy}^{-1}\mathbf{R}_{ya}\underline{g}\right)^T \mathbf{R}_{yy}\left(\underline{f} - \mathbf{R}_{yy}^{-1}\mathbf{R}_{ya}\underline{g}\right) \\ &= +\underline{g}^T \mathbf{R}_{aa}\underline{g} - \underline{g}^T \mathbf{R}_{ya}^T \mathbf{R}_{yy}^{-1}\mathbf{R}_{ya}\underline{g}. \\ \mathscr{E} &= \left(\underline{f} - \mathbf{R}_{yy}^{-1}\mathbf{R}_{ya}\underline{g}\right)^T \mathbf{R}_{yy}\left(\underline{f} - \mathbf{R}_{yy}^{-1}\mathbf{R}_{ya}\underline{g}\right) \\ &\quad +\underline{g}^T\left(\mathbf{R}_{ya} - \mathbf{R}_{ya}^T\mathbf{R}_{yy}^{-1}\mathbf{R}_{ya}\right)\underline{g}. \end{aligned} \tag{11}$$

$\mathbf{R}_{yy}$ is a positive definite matrix and hence the term $\left(\underline{f} - \mathbf{R}_{yy}^{-1}\mathbf{R}_{ya}\underline{g}\right)^T \mathbf{R}_{yy}\left(\underline{f} - \mathbf{R}_{yy}^{-1}\mathbf{R}_{ya}\underline{g}\right)$ can be minimized by setting

$$\underline{f} = \mathbf{R}_{yy}^{-1}\mathbf{R}_{ya}\underline{g}. \tag{12}$$

This gives a relation between the equalizer coefficients and the PR target and can be used with any choice of PR target.

With this, the MSE is

$$\mathscr{E}' = \underline{g}^T\left(\mathbf{R}_{aa} - \mathbf{R}_{ya}^T\mathbf{R}_{yy}^{-1}\mathbf{R}_{ya}\right)\underline{g}. \tag{13}$$

By definition, $\mathscr{E}' \geq 0$ holds true for any choice of $\underline{g}$ and hence $\left(\mathbf{R}_{aa} - \mathbf{R}_{ya}^T\mathbf{R}_{yy}^{-1}\mathbf{R}_{ya}\right)$ is a positive semi definite matrix.

B. Non-separable target

1) Unit energy constraint: With $\underline{g}^T\underline{g} = 1$ constraint, the error in (13) is minimized [12] when $\underline{g}$ is the eigenvector of $\left(\mathbf{R}_{aa} - \mathbf{R}_{ya}^T\mathbf{R}_{yy}^{-1}\mathbf{R}_{ya}\right)$ corresponding to the smallest eigenvalue $\lambda_{\min}$ of the matrix $\left(\mathbf{R}_{aa} - \mathbf{R}_{ya}^T\mathbf{R}_{yy}^{-1}\mathbf{R}_{ya}\right)$. The MMSE in this case is

$$\mathscr{E}_{\min}^{\mathrm{UE}} = \lambda_{\min}.$$

2) Monic constraint: We would like to set $g(0,0) = 1$. This is achieved by using a vector $\underline{u}$ that has one element as 1 and all others zeros such that $\underline{u}^T\underline{g} = g(0,0)$. Using the Lagrange multiplier λ, we have the cost function as

$$C = \underline{g}^T\left(\mathbf{R}_{aa} - \mathbf{R}_{ya}^T\mathbf{R}_{yy}^{-1}\mathbf{R}_{ya}\right)\underline{g} + 2\lambda\left(\underline{u}^T\underline{g} - 1\right).$$

To obtain the minima, we set the partial derivatives of C with respect to $\underline{g}^T$ and λ to zero. Using calculus of variations, we have

$$\begin{aligned} \frac{\partial C}{\partial \underline{g}^T} &= 2\left(\mathbf{R}_{aa} - \mathbf{R}_{ya}^T\mathbf{R}_{yy}^{-1}\mathbf{R}_{ya}\right)\underline{g} + 2\lambda\underline{u}^T = 0, \\ \frac{\partial C}{\partial \lambda} &= 2\left(\underline{u}^T\underline{g} - 1\right) = 0. \end{aligned}$$

Considering that $\left(\mathbf{R}_{aa} - \mathbf{R}_{ya}^T\mathbf{R}_{yy}^{-1}\mathbf{R}_{ya}\right)$ is a positive definite matrix, we can write the above equations as

$$\begin{aligned} \underline{g} &= -\lambda\left(\mathbf{R}_{aa} - \mathbf{R}_{ya}^T\mathbf{R}_{yy}^{-1}\mathbf{R}_{ya}\right)^{-1}\underline{u}, \\ \underline{u}^T\underline{g} &= 1. \end{aligned}$$

This gives the solution as

$$\underline{g} = \frac{\left(\mathbf{R}_{aa} - \mathbf{R}_{ya}^T\mathbf{R}_{yy}^{-1}\mathbf{R}_{ya}\right)^{-1}\underline{u}}{\underline{u}^T\left(\mathbf{R}_{aa} - \mathbf{R}_{ya}^T\mathbf{R}_{yy}^{-1}\mathbf{R}_{ya}\right)^{-1}\underline{u}}.$$

The error with this choice is

$$\mathscr{E}_{\min}^{\mathrm{MC}} = \frac{1}{\underline{u}^T\left(\mathbf{R}_{aa} - \mathbf{R}_{ya}^T\mathbf{R}_{yy}^{-1}\mathbf{R}_{ya}\right)^{-1}\underline{u}}.$$

C. Separable target

Considering square lattices for the recording medium, we want the target to be a rectangular mask such that

$$g(i,j) = g_c(i)\, g_r(j),$$

for $i = -N_i^{(g)}, \cdots, 0, 1, \cdots, N_i^{(g)}$, $j = -N_j^{(g)}, \cdots, 0, 1, \cdots, N_j^{(g)}$.

Let

$$\begin{aligned} \underline{g}_c &= \begin{bmatrix} g_c(-N_i^{(g)}) & \cdots & g_c(0) & \cdots & g_c(-N_i^{(g)}) \end{bmatrix}^T, \\ \underline{g}_r &= \begin{bmatrix} g_r(-N_i^{(g)}) & \cdots & g_r(0) & \cdots & g_r(-N_i^{(g)}) \end{bmatrix}. \end{aligned}$$

In this case, we can write

$$\underline{g} = \begin{bmatrix} g_c(-N_i^{(g)})\underline{g}_r & \cdots & g_c(0)\,\underline{g}_r & \cdots & g_c(-N_i^{(g)})\underline{g}_r \end{bmatrix}^T$$

$$\text{i.e., } \underline{g} = \underline{g}_c \otimes \underline{g}_r^T$$

where $\otimes$ is the Kronecker product.

In this case, (13) can be written as

$$\mathscr{E}' = \left(\underline{g}_c^T \otimes \underline{g}_r\right)\left(\mathbf{R}_{aa} - \mathbf{R}_{ya}^T\mathbf{R}_{yy}^{-1}\mathbf{R}_{ya}\right)\left(\underline{g}_c \otimes \underline{g}_r^T\right). \tag{14}$$

We prove a result in Lemma 1 that helps in analyzing expressions of the form given in (14).

Lemma 1. *Let $\underline{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_{N_x} \end{bmatrix}^T$ and $\underline{y} = \begin{bmatrix} y_1 & y_2 & \cdots & y_{N_y} \end{bmatrix}^T$ be column vectors of lengths N_x and N_y respectively. Let $\mathbf{A}$ be a square matrix of size $N_xN_y \times N_xN_y$. Let $\underline{z} = \underline{x} \otimes \underline{y}$, the Kronecker product of the two vectors. Then the quantity $\underline{z}^T\mathbf{A}\underline{z}$ can be uniquely represented as*

$$\underline{z}^T\mathbf{A}\underline{z} = \underline{x}^T\mathbf{B}_y\underline{x} = \underline{y}^T\mathbf{C}_x\underline{y}$$

for suitable matrices $\mathbf{B}_y$ and $\mathbf{C}_x$.

Proof: Writing $\mathbf{A}$ in block matrix representation as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} & \cdots & \mathbf{A}_{1,N_y} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} & \cdots & \mathbf{A}_{2,N_y} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{N_y,1} & \mathbf{A}_{N_y,2} & \cdots & \mathbf{A}_{N_y,N_y} \end{bmatrix},$$

Algorithm 1 Algorithm to obtain separable target under unit energy constraint

```
- Input Parameters: $\mathbf{C} = \left(\mathbf{R}_{aa} - \mathbf{R}_{ya}^T \mathbf{R}_{yy}^{-1} \mathbf{R}_{ya}\right)$, size of mask $N_r \times N_c$, learning threshold $\tau$
- Initialization: Set the first element of $\underline{g}_r^{(0)}$ and $\underline{g}_c^{(0)}$ to 1, and remaining elements to 0. $\mathscr{E}^{(0)} = 0$.
- Target Identification: Iterate following over $n$ with exit condition : $\left|\mathscr{E}^{(n+1)} - \mathscr{E}^{(n)}\right| \leq \tau$
    1) Identify the matrix $\mathbf{A}_{g_r^{(n)}}$ such that
           $\underline{u}^T \mathbf{C} \underline{u} = \left(\underline{g}_c^{(n)}\right)^T \mathbf{A}_{g_r^{(n)}} \left(\underline{g}_c^{(n)}\right),$
           $\underline{u} = \mathrm{vec}\left(\underline{g}_c^{(n)} \otimes \underline{g}_r^{(n)}\right).$
    2) Identify $\underline{a}_{\min}$, the eigenvector for the smallest eigen value of $\mathbf{A}_{g_r^{(n)}}$. Update $\underline{g}_c^{(n+1)} = \underline{a}_{\min}$.
    3) Identify the matrix $\mathbf{B}_{g_c^{(n+1)}}$ such that
           $\underline{v}^T \mathbf{C} \underline{v} = \left(\underline{g}_r^{(n)}\right)^T \mathbf{B}_{g_c^{(n+1)}} \left(\underline{g}_r^{(n)}\right),$
           $\underline{v} = \mathrm{vec}\left(\underline{g}_c^{(n+1)} \otimes \underline{g}_r^{(n)}\right).$
    4) Identify $\underline{b}_{\min}$, the eigenvector for the smallest eigen value of $\mathbf{B}_{g_c^{(n+1)}}$. Update $\underline{g}_r^{(n+1)} = \underline{b}_{\min}^T$.
    5) Update $\mathscr{E}^{(n+1)} = \underline{b}_{\min}$.
- Output: The separable mask is given by $\mathbf{G} = \underline{g}_c^* \otimes \underline{g}_r^*$. Minimum mean squared error is $\mathscr{E}^*$.
```

where $\mathbf{A}_{i,j}$ are all matrices of size $N_x \times N_x$. Similarly $\underline{z}$ can also be represented in block matrix representation as

$$\underline{z} = \begin{bmatrix} y_1\underline{x}, & y_2\underline{x}, & \cdots & y_{N_y}\underline{x} \end{bmatrix}^T.$$

Using block matrix multiplication, we can write

$$\begin{aligned} \underline{z}^T \mathbf{A} \underline{z} &= \sum_{i=1}^{N_y} \sum_{j=1}^{N_y} \left(y_i \underline{x}^T\right) \mathbf{A}_{i,j} \left(y_j \underline{x}\right), \\ \underline{z}^T \mathbf{A} \underline{z} &= \underline{x}^T \left(\sum_{i=1}^{N_y} \sum_{j=1}^{N_y} y_i \mathbf{A}_{i,j} y_j \right) \underline{x}. \end{aligned}$$

This gives the representation of $\underline{z}^T \mathbf{A} \underline{z} = \underline{x}^T \mathbf{B}_y \underline{x}$.

To show the representation as $\underline{y}^T \mathbf{C}_x \underline{y}$, we use the property of Kronecker product of vectors that $\underline{x} \otimes \underline{y} = \mathbf{P}\left(\underline{y} \otimes \underline{x}\right)$, for an appropriate permutation matrix $\mathbf{P}$. Hence,

$$\left(\underline{x} \otimes \underline{y}\right)^T \mathbf{A} \left(\underline{x} \otimes \underline{y}\right) = \left(\underline{y} \otimes \underline{x}\right)^T \mathbf{P}^T \mathbf{A} \mathbf{P} \left(\underline{y} \otimes \underline{x}\right).$$

By representing $\mathbf{P}^T \mathbf{A} \mathbf{P}$ as a block matrix, the above quantity can be represented as $\underline{y}^T \mathbf{C}_x \underline{y}$. ■

1) Unit energy constraint: Since $\underline{g}^T \underline{g} = \left(\underline{g}_c^T \underline{g}_c\right) \left(\underline{g}_r \underline{g}_r^T\right)$, we can further enforce constraints

$$\begin{aligned} \underline{g}_c^T \underline{g}_c &= 1, \\ \underline{g}_r \underline{g}_r^T &= 1. \end{aligned}$$

From Lemma 1, we can write

$$\mathscr{E}' = \underline{g}_c^T \mathbf{A}_{g_r} \underline{g}_c = \underline{g}_r \mathbf{B}_{g_c} \underline{g}_r^T,$$

for appropriate matrices $\mathbf{A}_{g_r}$ and $\mathbf{B}_{\mathbf{g_c}}$. Under the unit energy constraint, for a given $\underline{g}_r$, $\mathscr{E}'$ is minimized when $\underline{g}_c$ is eigenvector corresponding to smallest eigen value of $\mathbf{A}_{g_r}$. Similarly, for a given $\underline{g}_c$, $\mathscr{E}'$ is minimized when $\underline{g}_r^T$ is eigen vector corresponding to smallest eigen value of $\mathbf{B}_{g_c}$.

The optimal pair $\underline{g}_c$ and $\underline{g}_r^T$ can be obtained using alternating minimization technique as given in Algorithm 1. Lemma 2 below proves that this algorithm achieves the optimal solution under unit energy constraint.

Lemma 2. *The Algorithms 1 and 2 converge to achieve minimum mean squared error.*

Proof: The algorithms use alternating minimization technique. Consider a single iteration in the algorithm. Let $\mathscr{E}^{(n)}$ be the value of MSE after n^{th} iteration and let $\varepsilon^{(n)}$ be the value of MSE after computation of $g_c^{(n)}$. The update to $g_c^{(n)}$ is such that $\varepsilon^{(n)} \leq \mathscr{E}^{(n-1)}$. Within the n^{th} iteration, the update to $g_r^{(n)}$ is such that $\mathscr{E}^{(n)} \leq \varepsilon^{(n)}$. Therefore $\mathscr{E}^{(n)} \leq \mathscr{E}^{(n-1)}$ i.e., the MSE value decreases after every iteration. Since the mean squared error value is bounded below by 0, the algorithm converges to a finite value of MSE. The algorithms minimize a quadratic function with lower bound of 0 and hence has a unique local minimum. Therefore, the minimum achieved is the global minimum of the function. ■

2) Monic constraint: The constraint can be specified as $\underline{u}_r^T \underline{g}_r^T = 1$ and $\underline{u}_c^T \underline{g}_c = 1$.

From Lemma 1, we can write

$$\mathscr{E}' = \underline{g}_c^T \mathbf{A}_{g_r} \underline{g}_c = \underline{g}_r \mathbf{B}_{g_c} \underline{g}_r^T.$$

For a given $\underline{g}_r$,

$$\underline{g}_c = \frac{\mathbf{A}_{g_r}^{-1} \underline{u}_c}{\underline{u}_c^T \mathbf{A}_{g_r}^{-1} \underline{u}_c} \tag{15}$$

minimizes $\mathscr{E}'$. Similarly, for a given $\underline{g}_c$,

$$\underline{g}_r^T = \frac{\mathbf{B}_{g_c}^{-1} \underline{u}_r}{\underline{u}_r^T \mathbf{B}_{g_c}^{-1} \underline{u}_r} \tag{16}$$

minimizes $\mathscr{E}'$. The optimal pair $\underline{g}_c$ and $\underline{g}_r^T$ can be obtained using alternating minimization technique as given in Algorithm 2. Lemma 2 proves that this algorithm achieves the optimal solution under monic constraint.

IV. Low Complexity Symbol Detection for 2D ISI Channels

For 1D ISI channels, the ML decision is the "path" that maximizes the likelihood probability among all possible choices of the path. Hard decision Viterbi algorithm in this case is implemented by making local decisions for the individual bits by looking at finite length paths. These local decisions will provide an approximation of the ML path [9], [8].

In this section, we extend this idea to 2D ISI channels. Let us consider a finite sized page of bits sent through an

Algorithm 2 Algorithm to obtain separable target under monic constraint

```
• Input Parameters: $\mathbf{C} = \left(\mathbf{R}_{aa} - \mathbf{R}_{ya}^T \mathbf{R}_{yy}^{-1} \mathbf{R}_{ya}\right)$, size of mask $N_r \times N_c$, learning threshold $\tau$
• Initialization: Set first element of $\underline{g}_r^{(0)}$ and $\underline{g}_c^{(0)}$ to 1, and remaining elements to 0. $\mathscr{E}^{(0)} = 0$.
• Target Identification: Iterate following over $n$ with exit condition : $\left|\mathscr{E}^{(n+1)} - \mathscr{E}^{(n)}\right| \le \tau$
    1) Identify the matrix $\mathbf{A}_{g_r^{(n)}}$ such that
           $\underline{u}^T \mathbf{C} \underline{u} = \left(\underline{g}_c^{(n)}\right)^T \mathbf{A}_{g_r^{(n)}} \left(\underline{g}_c^{(n)}\right),$
           $\underline{u} = \mathrm{vec}\left(\underline{g}_c^{(n)} \otimes \underline{g}_r^{(n)}\right).$
    2) Update $\underline{g}_c^{(n+1)} = \mathbf{A}_{g_r^{(n)}}^{-1} \underline{u}_c / (\underline{u}_c^T \mathbf{A}_{g_r^{(n)}}^{-1} \underline{u}_c)$.
    3) Identify the matrix $\mathbf{B}_{g_c^{(n+1)}}$ such that
           $\underline{v}^T \mathbf{C} \underline{v} = \left(\underline{g}_r^{(n)}\right)^T \mathbf{B}_{g_c^{(n+1)}} \left(\underline{g}_r^{(n)}\right),$
           $\underline{v} = \mathrm{vec}\left(\underline{g}_c^{(n+1)} \otimes \underline{g}_r^{(n)}\right).$
    4) Update $\underline{g}_r^{(n+1)} = \left(\mathbf{B}_{g_c^{(n+1)}}^{-1} \underline{u}_r / (\underline{u}_r^T \mathbf{B}_{g_c^{(n+1)}}^{-1} \underline{u}_r)\right)^T.$
    5) Update $\mathscr{E}^{(n+1)} = 1/(\underline{u}_r^T \mathbf{B}_{g_c^{(n+1)}}^{-1} \underline{u}_r)$.
• Output: The separable mask is given by $\mathbf{G} = \underline{g}_c^* \otimes \underline{g}_r^*$. Minimum mean squared error is $\mathscr{E}^*$.
```

ISI channel. Among all possible choices of the page, the ML decision is the page that maximizes the likelihood probability. As an approximation to the ML decision, we can restrict the search to a local region to make decisions for the individual bits. We take this motivation to propose an algorithm towards the ML criterion. In 1D case, the symbols are detected from past to future. However, the detection order in 2D is not trivial. We choose the raster scan order for the proposed algorithm. The symbols are detected from left to right, top to bottom of the page.

A. ML detection for 2D ISI channels

Let us consider the case where the channel response is given by the mask $\mathbf{G}$ and the noise samples are white and Gaussian distributed with 0 mean and variance σ_w^2 as given by

$$\begin{aligned} y(i,j) &= \underline{g}^T \underline{a}_{\mathbf{G}}^{(i,j)} + w(i,j), \\ p(w(i,j)) &= \frac{1}{\sqrt{2\pi\sigma_w^2}} \exp\left(-\frac{1}{2\sigma_w^2}(w(i,j))^2\right). \end{aligned}$$

The likelihood probability is

$$Pr\left[\left\{y(n_i,n_j)\right\}_{n_i,n_j=-\infty}^{\infty} \mid \left\{a(n_i,n_j)\right\}_{n_i,n_j=-\infty}^{\infty}\right]. \quad (17)$$

To detect the symbol $a(i,j)$, we approximate the likelihood probability by computing the likelihood probability over a local region surrounding the position (i,j). Let $\mathbf{M}$ denote this local ML region mask. We approximate the likelihood probability in (17) to the likelihood probability of $\underline{y}_{\mathbf{M}}^{(i,j)}$ given by

$$\begin{aligned} & Pr\left[\underline{y}_{\mathbf{M}}^{(i,j)} \mid \left\{a(n_i,n_j)\right\}_{n_i,n_j=-\infty}^{\infty}\right] \\ &= \text{constant} \times \exp\left(-\left\|\underline{y}_{\mathbf{M}}^{(i,j)} - \underline{\hat{y}}_{\mathbf{M}}^{(i,j)}\right\|^2 / 2\sigma_w^2\right), \quad (18) \end{aligned}$$

where $y(i,j)$ are received samples and $\hat{y}(i,j)$ computed as

$$\hat{y}(i,j) = \underline{g}^T \underline{a}_{\mathbf{G}}^{(i,j)}. \quad (19)$$

Consider the case where the $\mathbf{G}$ and $\mathbf{M}$ are rectangular masks of sizes $(2N_i^{(g)}+1) \times (2N_j^{(g)}+1)$ and $(2N_i^{(m)}+1) \times (2N_j^{(m)}+1)$ respectively[1]. Let $N_i = N_i^{(g)} + N_i^{(m)}$ and $N_j = N_j^{(g)} + N_j^{(m)}$. Let the center (zero "delay") taps of the masks be at the geometric center of the individual masks. In this case, computation of the likelihood probability in (18) uses

$$\left\{a(k,l) \mid -N_i \le k-i \le N_i, -N_j \le l-j \le N_j\right\}.$$

While detecting the symbol $a(i,j)$, due to detection order chosen, the symbols $\{a(k,l) \mid k \le i \text{ or } k=i, l<j\}$ are already detected. Hence, the likelihood probability depends on the choice of

$$\begin{aligned} \{a(k,l) \quad | \quad & (0 \le k-i \le N_i, -N_j \le l-j \le N_j) \\ & \text{or } (k=i, 0 \le l-j \le N_j)\}. \quad (20) \end{aligned}$$

Let $P^{(i,j)}$ denote the symbols in the neighboring of (i,j) that are already decoded as given by

$$\begin{aligned} P^{(i,j)} = \{a(k,l) \quad | \quad & (-N_i \le k-i < 0, -N_j \le l-j \le N_j) \\ & \text{or } (k=i, -N_j \le l-j \le -1)\}. \quad (21) \end{aligned}$$

Similarly, let $S^{(i,j)}$ denote the symbols in the neighboring of (i,j) that will be decoded in future as given by

$$\begin{aligned} S^{(i,j)} = \{a(k,l) \quad | \quad & (0 < k-i \le N_i, -N_j \le l-j \le N_j) \\ & \text{or } (k=i, 1 \le l-j \le N_j)\}. \quad (22) \end{aligned}$$

The likelihood probability in (18) is a function of $P^{(i,j)}$, $a(i,j)$ and $S^{(i,j)}$ and can be computed over all choices of $a(i,j)$ and $S^{(i,j)}$. Among these choices, the one that maximizes the likelihood probability will be used to make decision on $a(i,j)$.

The likelihood probability in (18) is maximized by minimizing the ML metric given by

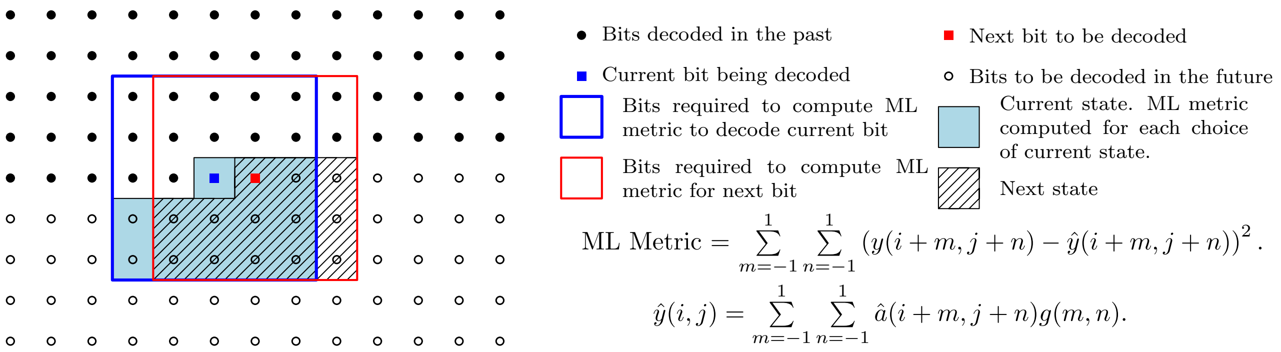
$$\Gamma\left(P^{(i,j)}, a(i,j), S^{(i,j)}\right) = \left\|\underline{y}_{\mathbf{M}}^{(i,j)} - \underline{\hat{y}}_{\mathbf{M}}^{(i,j)}\right\|^2. \quad (23)$$

Using (23), the decision $\hat{a}(i,j)$ is given by

$$\hat{a}(i,j) = \arg\min_{a(i,j)} \left\{\min_{S^{(i,j)}} \Gamma\left(P^{(i,j)}, a(i,j), S^{(i,j)}\right)\right\}. \quad (24)$$

Figure 2 shows the detection procedure for the case where $\mathbf{M}$ and $\mathbf{G}$ are of size 3 × 3. The detection algorithm for rectangular $\mathbf{G}$ and $\mathbf{M}$ is given in Algorithm 3.

[1] The super scripts in $N_i^{(g)}, N_j^{(g)}, N_i^{(m)}$ and $N_j^{(m)}$ are for associating the dimensions to the matrices $\mathbf{G}$ and $\mathbf{M}$. The subscripts i and j are to indicate the number of rows and number of columns respectively.


● Bits decoded in the past
■ Next bit to be decoded
■ Current bit being decoded
○ Bits to be decoded in the future
Bits required to compute ML metric to decode current bit
Current state. ML metric computed for each choice of current state.
Bits required to compute ML metric for next bit
Next state

$$\text{ML Metric} = \sum_{m=-1}^{1}\sum_{n=-1}^{1}\left(y(i+m,j+n)-\hat{y}(i+m,j+n)\right)^2.$$

$$\hat{y}(i,j) = \sum_{m=-1}^{1}\sum_{n=-1}^{1}\hat{a}(i+m,j+n)g(m,n).$$

Figure 2. Viterbi Detector extended to 2D symbol detection. The decoding order and criterion for computing path metric for a target response of size 3×3 is shown.

Algorithm 3 Viterbi Algorithm (Hard decisions) extended to 2D symbol detection.

```
• Inputs: A page of received samples $y(i,j)$ of size $N_i^{(y)} \times N_j^{(y)}$, target response mask $\mathbf{G}$ of size $\left(2N_i^{(g)}+1\right) \times \left(2N_j^{(g)}+1\right)$, local ML region $\mathbf{M}$ of size $\left(2N_i^{(m)}+1\right) \times \left(2N_j^{(m)}+1\right)$.
• Initialization: Set $\hat{a}(i,j) = -1$ for $(i,j)$ outside the boundary of $N_i^{(y)} \times N_j^{(y)}$ page.
• Signal Detection:
  For $i = 1$ to $N_i^{(y)}$
    – For $j = 1$ to $N_j^{(y)}$
        ∗ Compute the ML metric as
            $\Gamma\left(P^{(i,j)}, a(i,j), S^{(i,j)}\right) = \left\| \underline{y}_{\mathbf{M}}^{(i,j)} - \underline{\hat{y}}_{\mathbf{M}}^{(i,j)} \right\|^2,$
            $\hat{y}(i,j) = \underline{g}^T \underline{a}_{\mathbf{G}}^{(i,j)},$
          for all choices of $a(i,j)$ and $S^{(i,j)}$.
        ∗ Detected symbol is $\hat{a}(i,j)$ corresponding to smallest ML metric:
            $\hat{a}(i,j) = \arg\min_{a(i,j)} \left\{ \min_{S^{(i,j)}} \Gamma\left(P^{(i,j)}, a(i,j), S^{(i,j)}\right) \right\}.$
• Output: $N_i^{(y)} \times N_j^{(y)}$ page of detected symbols $\hat{a}(i,j)$.
```

B. Soft output extension for 2D detection

Analysis of 1D SOVA:

Let us look into the soft-output Viterbi algorithm proposed in [10] for 1D ISI. Following is the notation relevant to the 1D SOVA:

u_k: k^{th} bit.

δ: ML metrics are computed for paths of length δ. This corresponds to local ML region $\mathbf{M}$ defined in this paper.

l: The channel memory length. This corresponds to the size of $\mathbf{G}$.

s_k: State at time k defined as $(u_{k-l}, u_{k-l+1}, \cdots, u_{k-1})$

The paths are defined as a set of branches representing the transition from one state to the next.

Let us call the path given by the hard decisions of Viterbi algorithm as VA path. Even if the VA path diverges from the ML path at some instant, it will merge back into the ML path after sufficient time. SOVA looks at such a possible branching and merging of paths over a finite window of time into the past. Consider a pair of such paths P_1 and P_2 and the corresponding path metrics M_1 and M_2 with $M_1 < M_2$. The probability of choosing path P_1 over P_2 is considered to be

$$p_{s_k} = \frac{\exp\left(-\frac{M_1}{2\sigma_w^2}\right)}{\exp\left(-\frac{M_1}{2\sigma_w^2}\right) + \exp\left(-\frac{M_2}{2\sigma_w^2}\right)}. \tag{25}$$

This probability impacts all the bits wherever the paths P_1 and P_2 differ. If p_j is the reliability value of jth bit on P_1, the update to this reliability is done as

$$p_j \leftarrow p_j p_{s_k} + (1-p_j)\left(1-p_{s_k}\right). \tag{26}$$

The SOVA defines a specific way of identifying these differing paths and make the updates to the likelihood values of the bits:

1. For a state at s_k, two paths are chosen such they merge at s_k and have minimum path metric among all such paths. These two are competing paths. The reliability values are updated as in (26) for $(k-\delta^{\text{th}})$ bit to $(k-l)^{\text{th}}$ bit.

2. The procedure is repeated for all choices of s_k , then repeated for all choices of s_{k+1} and so on.

In SOVA, there are two intents of updating the likelihood values for all choices of s_k:

a) At time k, the VA path is decided only till $(k-\delta)^{\text{th}}$ bit. Making a particular choice of s_k for the updates is not desirable if the VA path does not ultimately touch the chosen state s_k.

b) Looking at all possible paths leads to MAP criterion.

Motivation from 1D SOVA for 2D detection:

For the 2D case, we borrow the ideas from SOVA and make soft decisions as follows:

1. In the 2D case, each choice of 5×5 bits as shown in Figure 2 gives a choice of a "local surface". This local surface is analogous to a path in 1D.

2. We make soft decisions after all the hard decisions are made. Hence, there is a particular choice of the "local surface" that we can make.

3. At position (i,j), M_1 is the ML metric corresponding to the "local surface" given by the hard decisions.

4. We define the competing surface as the one with minimum ML metric and whose bit value at (i,j) differs from the hard decision.

5. The reliability of the hard decision is computed in (25).

This choice of soft-output will only provide the reliability information for the hard already decisions made.

2D SOVA Equivalent Algorithm:

We extend the hard decision procedure in Algorithm 3 to give soft-outputs by looking at the alternate states whose ML metric comes close to the ML metric corresponding to the hard decisions.

At the position (i,j), let $M_1^{(i,j)}$ be the value of ML metric given by (23) corresponding to the the hard decisions given by Algorithm 3:

$$M_1^{(i,j)} = \Gamma\left(\hat{P}^{(i,j)}, \hat{a}(i,j), \hat{S}^{(i,j)}\right).$$

where $\hat{P}^{(i,j)}$ and $\hat{S}^{(i,j)}$ are the particular choices corresponding to the hard decisions.

We identify the alternate states which minimizes the ML metric, computed under the constraint that $a(i,j)$ differs from the hard decision. Let this be $M_2^{(i,j)}$:

$$M_2^{(i,j)} = \min_{a(i,j)\neq\hat{a}(i,j),S^{(i,j)}} \Gamma\left(\hat{P}^{(i,j)}, a(i,j), S^{(i,j)}\right). \quad (27)$$

Let

$$\Delta(i,j) = PM_2(i,j) - PM_1(i,j). \quad (28)$$

We define the reliability of the the decision $\hat{a}(i,j)$ as the probability $p_{(i,j)}$ of choosing the ML decision over the alternate states, given by

$$p_{(i,j)} = \frac{\exp\left(-\frac{PM_1(i,j)}{2\sigma_w^2}\right)}{\exp\left(-\frac{PM_1(i,j)}{2\sigma_w^2}\right) + \exp\left(-\frac{PM_2(i,j)}{2\sigma_w^2}\right)}. \quad (29)$$

Simplifying (29),

$$p_{(i,j)} = \frac{1}{1+\exp\left(-\frac{\Delta(i,j)}{2\sigma_w^2}\right)}. \quad (30)$$

The log-likelihood ratio corresponding to the decision $\hat{a}(i,j)$ is given by

$$LLR(i,j) = \hat{a}(i,j)\log\left(\frac{p_{(i,j)}}{1-p_{(i,j)}}\right) = \hat{a}(i,j)\frac{\Delta(i,j)}{2\sigma_w^2}. \quad (31)$$

$LLR(i,j)$ give the soft-decisions providing additional information on the reliability of the hard decisions from the Algorithm 3.

The SOVA equivalent for 2D is outlined in Algorithm 4.

Algorithm 4 Soft Output Viterbi Algorithm for 2D symbol detection

```
• Inputs: A page of received samples $y(i,j)$ of size $N_i^{(y)} \times N_j^{(y)}$, target response mask $\mathbf{G}$ of size $\left(2N_i^{(g)}+1\right) \times \left(2N_j^{(g)}+1\right)$, local ML region $\mathbf{M}$ of size $\left(2N_i^{(m)}+1\right) \times \left(2N_j^{(m)}+1\right)$ and the variance of AWGN $\sigma_w^2$.
• Initialization: Set $\hat{a}(i,j) = -1$ for $(i,j)$ outside the boundary of $N_i^{(y)} \times N_j^{(y)}$ page.
• Hard Decisions:
  For $i = 1$ to $N_i^{(y)}$
    – For $j = 1$ to $N_j^{(y)}$
        ∗ Compute the ML metric as
            $\Gamma\left(P^{(i,j)}, a(i,j), S^{(i,j)}\right) = \left\|\underline{y}_{\mathbf{M}}^{(i,j)} - \hat{\underline{y}}_{\mathbf{M}}^{(i,j)}\right\|^2,$
            $\hat{y}(i,j) = \underline{g}^T \underline{a}_{\mathbf{G}}^{(i,j)},$
          for all choices of $a(i,j)$ and $S^{(i,j)}$.
        ∗ Detected symbol is $\hat{a}(i,j)$ corresponding to smallest ML metric:
            $\hat{a}(i,j) = \arg\min_{a(i,j)} \left\{\min_{S^{(i,j)}} \Gamma\left(P^{(i,j)}, a(i,j), S^{(i,j)}\right)\right\}$
• We have a page of hard decisions given by $\hat{a}(i,j)$. Identify $\hat{S}^{(i,j)}$ and $\hat{P}^{(i,j)}$ corresponding to this detected page.
• Soft Decisions:
  For $i = 1$ to $N_i^{(y)}$
    – For $j = 1$ to $N_j^{(y)}$
        ∗ Compute the ML metric for the hard decisions :
            $M_1^{(i,j)} = \Gamma\left(\hat{P}^{(i,j)}, \hat{a}(i,j), \hat{S}^{(i,j)}\right)$
        ∗ Identify the minimum ML metric $M_2^{(i,j)}$:
            $M_2^{(i,j)} = \min_{a(i,j)\neq\hat{a}(i,j),S^{(i,j)}} \Gamma\left(\hat{P}^{(i,j)}, a(i,j), S^{(i,j)}\right)$
        ∗ The soft decisions are given by
            $LLR(i,j) = \hat{a}(i,j)\frac{1}{\sigma_w^2}\left(M_2^{(i,j)} - M_1^{(i,j)}\right).$
• Output: $N_i^{(y)} \times N_j^{(y)}$ page of LLRs $LLR(i,j)$.
```

C. Performance

The choice of the local ML region $\mathbf{M}$ defines the proximity of the algorithm's performance to the ML decisions. Since $\underline{y}_{\mathbf{G}}^{(i,j)}$ gives the samples that contain information about the symbol $a(i,j)$, choosing the mask shape $\mathbf{M}$ such that the shape of $\mathbf{G}$ is entirely contained in $\mathbf{M}$, helps in detection of $a(i,j)$. As the size of $\mathbf{M}$ becomes sufficiently large compared to the symbol page size, the algorithm behaves same as the ML criterion.

Figure 3 shows the performance of the algorithm for the ISI

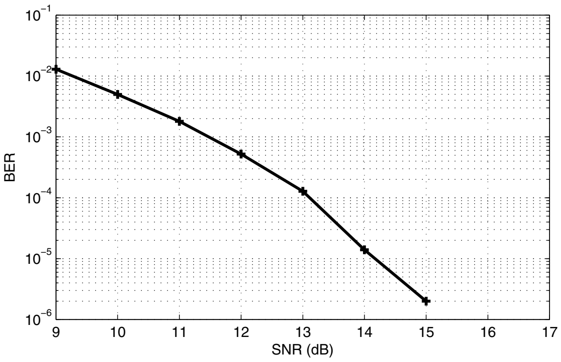


Figure 3. Uncoded performance of Viterbi Algorithm extended to 2D symbol detection for a 3 × 3 ISI mask.

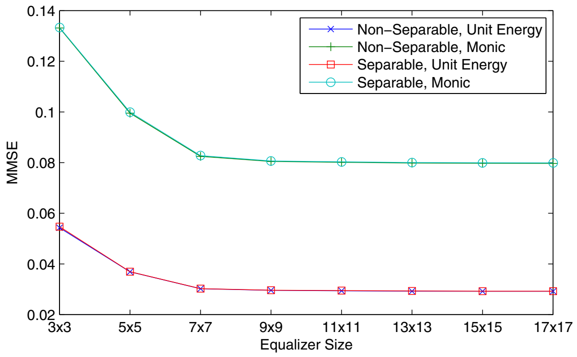


Figure 4. MMSE value achieved with equalizers of different sizes for 3 × 3 target for $\frac{1}{2\times 1.5}\exp\left(-\frac{1}{2\times 1.5}\left(x^2+y^2\right)\right)$ channel, SNR = 15 dB. As the channel response is separable, the MMSE values achieved for separable and non-separable targets is the same.

mask given by

$$\mathbf{H}=\begin{bmatrix}\frac{1}{16} & \frac{1}{4} & \frac{1}{16}\\ \frac{1}{4} & 1 & \frac{1}{4}\\ \frac{1}{16} & \frac{1}{4} & \frac{1}{16}\end{bmatrix}, \tag{32}$$

with the following definition of SNR

$$\mathrm{SNR(dB)}=10\log_{10}\left(\|\mathbf{H}\|^2/\sigma_w^2\right).$$

The algorithm is a one-pass procedure and the decision on a current pixel does not impact decisions of symbols on the page made in the past. Algorithms that either iterate to refine the decisions or provide feedback to the previous pixels can perform better as seen in [5]. We can also see that the performance is about 1 dB better than that of the single iteration of the row-column detector in [5]. This is due to choice of larger local ML region $\mathbf{M}$. The performance is within 1.5 dB of the joint 2D equalization and detection (JTED) algorithm in [5].

Since the proposed soft-output extension gives only the additional reliability information of the hard decisions made, the advantage can only be seen when the soft output values are used in conjunction with an error correction code.

D. Computational complexity

For a target $\mathbf{G}$ of size $\left(2N_i^{(g)}+1\right)\times\left(2N_j^{(g)}+1\right)$ and local ML region $\mathbf{M}$ of size $\left(2N_i^{(m)}+1\right)\times\left(2N_j^{(m)}+1\right)$, the number of ML metrics computed for each detected symbol is given by the number of choices of $a(i,j)$ and $S^{(i,j)}$. We have precomputed the values $\underline{\hat{y}}_k$ for all choices of $a(i,j)$ and $S^{(i,j)}$ to avoid duplication of computation. Hence, the number of additions and multiplications required to compute each ML metric is given by the size of $\mathbf{M}$.

For the additional steps in the soft-output extension, the ML metrics are computed for all choices of $S^{(i,j)}$ with a particular choice of $a(i,j)$. Hence, the additional complexity required will be approximately half the complexity of the hard decision algorithm. Tables I and II provide details of the computational complexity associated with two Algorithms 3 and 4. The numbers are also provided for a 3 × 3 target response $\mathbf{G}$ and the choice of 3 × 3 local ML region mask $\mathbf{M}$. For the channel response in (32), Algorithm 3 gives 1 dB gain compared to single iteration of the row-colum detector in [5] with the same computational complexity. The performance improvement is due to to the larger span of the signal used for computing the ML metrics.

V. Simulation and Results

A. Channel response

TDMR channels and corresponding read heads are still in the early stage of research. For this reason, we use the Gaussian pulse model of perpendicular recording channel and extend it to 2D. Following channel response is considered

$$h(x,y) \quad = \quad \frac{1}{2\pi\sigma^2}\exp\left(-\left(x^2+y^2\right)/2\sigma^2\right).$$

Note that $h(x,y)$ is separable. In the results to follow, the value of σ^2 and hence the width of the pulse is varied to compare the performance under various constraints.

B. Received signal generation

The input symbols $a(i,j)$ are chosen from the binary alphabet $\{-1,1\}$. The symbols are all random and independently generated with $Pr[a(i,j)=1]=0.5$ for all i,j.

At each instant, a page of size 64 × 64 bits is generated corresponding to 512 byte sectors traditionally used in recording media. The mask $h(x,y)$ is applied on it and white noise is added to it giving the output page of size 64 × 64. The symbols outside the boundary of the 64 × 64 page are all set to -1 to allow proper initialization of states near the boundary. The size of the boundary is same as the size of the local ML region mask $\mathbf{M}$.

C. Target and equalizer choice

We have fixed the size of target to 3 × 3. Based on our observations, we have noticed improvement in performance with increase in size of equalizer mask. We have fixed the size of equalizer mask to 15 × 15 as the improvement beyond that it is not very noticeable as shown in Figure 4.

Table I
COMPUTATIONAL REQUIREMENT OF HARD DECISION ALGORITHM 3 TO DETECT A PAGE OF SIZE $H\times W$ WITH ISI MASK OF SIZE $\left(2N_i^{(g)}+1\right)\times\left(2N_j^{(g)}+1\right)$ AND LOCAL ML REGION OF SIZE $\left(2N_i^{(m)}+1\right)\times\left(2N_j^{(m)}+1\right)$.

	Requirement per bit	3×3 masks, 64×64 page
Multiplications	$H\times W\left(2N_i^{(m)}+1\right)\left(2N_j^{(m)}+1\right)2^{(2N_j^{(g)}+2N_j^{(m)}+1)(N_i^{(g)}+N_j^{(m)})+(N_j^{(g)}+N_j^{(m)}+1)}$	3.02e8
Additions	$H\times W\left(2N_i^{(m)}+1\right)\left(2N_j^{(m)}+1\right)2^{(2N_j^{(g)}+2N_j^{(m)}+1)(N_i^{(g)}+N_j^{(m)})+(N_j^{(g)}+N_j^{(m)}+1)}$	3.02e8
Comparisons	$H\times W2^{(2N_i^{(g)}+2N_j^{(m)}+1)(N_i^{(g)}+N_j^{(m)})+(N_i^{(g)}+N_j^{(m)}+1)}$	3.02e8
Memory Units	$\left(2N_i^{(g)}+1\right)\left(2N_j^{(g)}+1\right)2^{(2N_j^{(g)}+2N_j^{(m)}+1)(N_i^{(g)}+N_j^{(m)})+(N_j^{(g)}+N_j^{(m)}+1)}$	7.38e4

Table II
COMPUTATIONAL REQUIREMENT OF SOFT DECISION ALGORITHM 4 TO DETECT A PAGE OF SIZE $H\times W$ WITH ISI MASK OF SIZE $\left(2N_i^{(g)}+1\right)\times\left(2N_j^{(g)}+1\right)$ AND LOCAL ML REGION OF SIZE $\left(2N_i^{(m)}+1\right)\times\left(2N_j^{(m)}+1\right)$.

	Requirement per bit	3×3 masks, 64×64 page
Multiplications	$H\times W\times\left(\frac{3}{2}\left(2N_i^{(m)}+1\right)\left(2N_j^{(m)}+1\right)2^{(2N_j^{(g)}+2N_j^{(m)}+1)(N_i^{(g)}+N_j^{(m)})+(N_j^{(g)}+N_j^{(m)}+1)}+1\right)$	4.53e8
Additions	$H\times W\times\left(\frac{3}{2}\left(2N_i^{(m)}+1\right)\left(2N_j^{(m)}+1\right)2^{(2N_j^{(g)}+2N_j^{(m)}+1)(N_i^{(g)}+N_j^{(m)})+(N_j^{(g)}+N_j^{(m)}+1)}+1\right)$	4.53e8
Comparisons	$H\times W\times 3\times 2^{(2N_i^{(g)}+2N_j^{(m)}+1)(N_i^{(g)}+N_j^{(m)})+(N_i^{(g)}+N_j^{(m)})}$	4.53e8
Memory Units	$\left(2N_i^{(g)}+1\right)\left(2N_j^{(g)}+1\right)2^{(2N_j^{(g)}+2N_j^{(m)}+1)(N_i^{(g)}+N_j^{(m)})+(N_j^{(g)}+N_j^{(m)}+1)}$	7.38e4

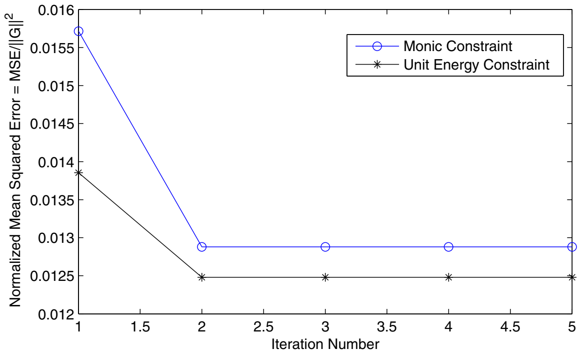


Figure 5. The convergence of the alternating minimization Algorithms 1 and 2 to obtain optimal separable targets for 3×3 target for $\exp\left(-\frac{1}{2}\left(x^2+y^2\right)\right)$ channel, SNR=20 dB. The mean squared error at each iteration is normalized to the energy of target.

For a separable target, the alternating minimization algorithms are observed to learn the target within 3 iterations as shown in Figure 5.

D. Results

Since the symbols are $\{+1,-1\}$, we define SNR as

$$\mathrm{SNR(dB)}=10\log_{10}\left(\|\mathbf{H}\|^2/\sigma_w^2\right).$$

where $\|\mathbf{H}\|$ is l_2 norm of the channel response and σ_w^2 is variance of white Gaussian noise introduced by the channel.

Figure 6 shows the BER vs. SNR curves for various choices of targets. For the Gaussian channel response, we see that the separable and non-separable targets show the same performance. However, as shown in the Figure 7, for a non-separable channel response given by $\frac{1}{0.5+x^2+y^2}$, a difference

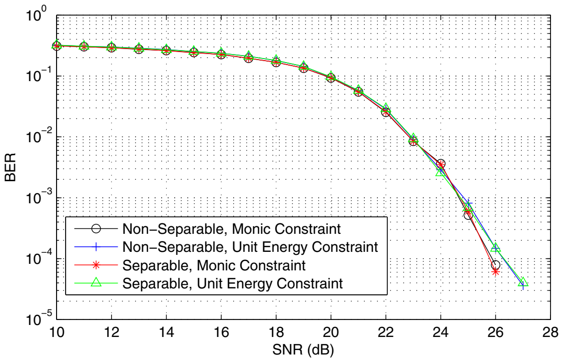


Figure 6. Performance of PRML system with 3×3 target for the channel response $\frac{1}{2}\exp\left(-\frac{x^2+y^2}{2}\right)$. The monic and unit energy targets give the same performance. Since the channel response is separable, same performance is observed for seperable and non-separable targets.

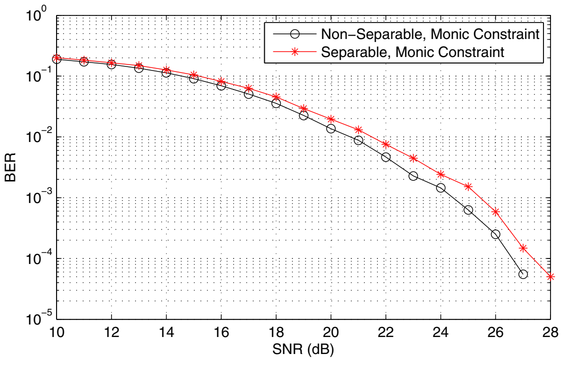


Figure 7. Performance of PRML system with 3×3 target for the channel response $\frac{1}{0.5+x^2+y^2}$. Since the channel response is non-separable, we observe difference in the performances of separable and non-separable targets. Non-seperable target performs better as expected.

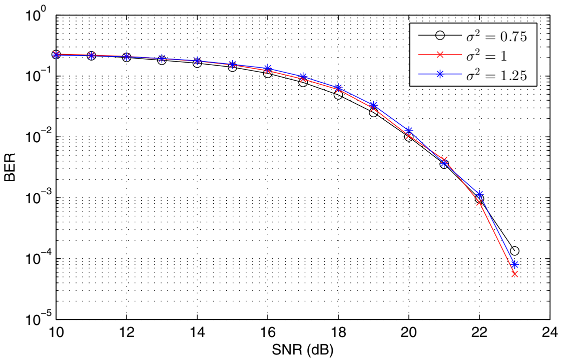


Figure 8. Performance of the detection algorithm with 3×3 ISI masks and AWGN. The ISI masks used here are non-separable 3×3 targets under monic constraint for a channel with the channel response $\frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$.

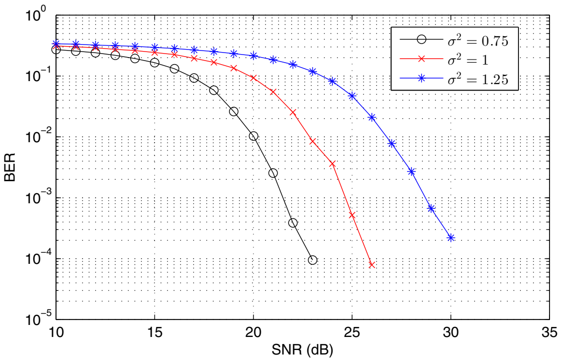


Figure 9. Performance of the PRML system for 3×3 monic target is compared by varying the width of the Gaussian pulse $\frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$. The performance of the system degrades as the extent of ISI increases.

of about 1 dB in performance between separable and non-separable targets is noticed.

Figure 8 shows the performance of our 2D detection algorithm when the target response is considered as the channel response. We have generated the targets by varying the width of the Gaussian pulse. Figure 9 shows the performance of the PRML system by varying the width of the Gaussian pulse. The BER observed in the PRML system will be dependent on the target as well as the channel capacity. From Figures 8 and 9, the degradation in performance of the PRML system with an increase in width of the Gaussian pulse can be attributed to the degradation in the channel capacity. This is indeed the case because the extent of ISI increases with the increase in the pulse width.

VI. Conclusion

We have extended 1D PR target design techniques to the 2D paradigm. With an eye to reduce the symbol detection complexity, we investigated into design of 2D separable targets that give a performance close to non-separable targets for a separable continuous time channel response. This allows us to deploy low complexity detectors specifically tuned for separable targets as needed.

The Viterbi algorithm from 1D is extended to 2D for low complexity signal detection by considering a 2D local span. The algorithm can be further extended with noise prediction to handle the colored noise from PRML system and the data dependent jitter noise present in magnetic recording channels.

References

[1] J. Moon and W. Zeng, "Equalization for Maximum Likelihood Detectors," *IEEE Transactions on Magnetics*, vol. 31, pp. 1083–1088, Mar. 1995.

[2] P. Kovintavewat, I. Ozgunes, E. Kurtas, J. Barry, and S. McLaughlin, "Generalized Partial-Response Targets for Perpendicular Recording With Jitter Noise," *IEEE Transactions on Magnetics*, vol. 38, pp. 2340–2342, Sept. 2002.

[3] S. Karakulak, P. Siegel, J. Wolf, and H. Bertram, "Joint-Track Equalization and Detection for Bit Patterned Media Recording," *IEEE Transactions on Magnetics*, vol. 46, pp. 3639–3647, Sept. 2010.

[4] Y. Wu, J. O'Sullivan, N. Singla, and R. Indeck, "Iterative Detection and Decoding for Separable Two-Dimensional Intersymbol Interference," *IEEE Transactions on Magnetics*, vol. 39, pp. 2115–2120, July 2003.

[5] Y. Chen and S. G. Srinivasa, "Joint Self-Iterating Equalization and Detection for Two-Dimensional Intersymbol-Interference," *IEEE Transactions on Communications*, vol. 61, pp. 3219–3230, Aug. 2013.

[6] S. Khatami and B. Vasić, "Detection for Two-Dimensional Magnetic Recording Systems," *Journal of Communications*, vol. 8, pp. 233–239, Apr. 2013.

[7] S. Khatami and B. Vasić, "Generalized Belief Propagation Detector for TDMR Microcell Model," *IEEE Transactions on Magnetics*, vol. 49, pp. 3699–3702, July 2013.

[8] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Transations on Information Theory*, vol. IT-13, pp. 260–269, Apr. 1967.

[9] G. D. Forney, "Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol-Interference," *IEEE Transations on Information Theory*, vol. IT-18, pp. 363–378, May 1972.

[10] J. Hagenauer and P. Hoeher, "A Viterbi Algorithm with Soft-Decision Outputs and its Applications," *Global Telecommunications Conference and Exhibition 'Communications Technology for the 1990s and Beyond'*, vol. 3, pp. 1680–1686, Nov. 1989.

[11] B. Vasić and M. E. Kurtas, *Coding and Signal Processing For Magnetic Recording Systems*. CRC Press LLC, 2004.

[12] G. Strang, *Linear Algebra and Applications*. 4 ed., 2006.

[13] J. Caroselli, S. Altekar, P. McEwen, and J. Wolf, "Improved Detection For Magnetic Recording Systems With Media Noise," *IEEE Transactions on Magnetics*, vol. 33, pp. 2779–2781, Sept. 1997.

[14] K. M. Chugg, "Performance of Optimal Digital Page Detection in a Two-Dimensional ISI/AWGN channel," *Proceedings - Asilomar Conference on Signals, Systems and Computing*, pp. 958–962, Nov. 1996.

[15] H. Yang and G. Mathew, "Joint Design of Optimum Partial Response Target and Equalizer for Recording Channels With Jitter Noise," *IEEE Transactions on Magnetics*, vol. 42, pp. 70–77, Jan. 2006.

[16] S. G. Srinivasa, Y. Chen, and S. Dahandeh, "A Communication-Theoretic Framework for 2-DMR Channel Modeling: Performance Evaluation of Coding and Signal Processing Methods," *IEEE Transactions on Magnetics*, vol. 50, pp. 6–12, Mar. 2014.