ABSTRACT
We consider the equalization of two-dimensional intersymbol interference (2D ISI) channels. We propose a message passing algorithm based on Gaussian belief propagation (GaBP) to perform joint decoding and equalization. While in existing 2D equalizers the complexity is exponential in the length of the state-space, the complexity is only polynomial for the proposed algorithm. This enables to handle 2D channels with long memory with a tractable computational complexity.

Index Terms— Gaussian belief propagation, two-dimensional channels, intersymbol interference, turbo equalization.

1. INTRODUCTION
Coding and equalization for 2D ISI channels has attracted considerable interest during the last decade, due to recent developments in the area of storage systems [1].

Since maximum-likelihood detection is generally unfeasible on 2D ISI channels, several suboptimal approaches have been proposed in the literature. The minimum mean-square error (MMSE) equalizer and the decision feedback equalizer (DFE), have been adapted for 2D ISI channels. For channels with severe 2D ISI, more elaborated algorithms are needed, such as the decision feedback Viterbi algorithm (DF-VA) [2] and the iterative multistrip (IMS) algorithm [3]. The IMS algorithm being soft-input soft-output (SISO), joint equalization and decoding can be performed in a typical turbo equalization fashion [4].

In this paper, we consider the application of GaBP [5] to the equalization of 2D ISI channels. The resulting equalizer is essentially a generalization of the algorithm by Guo and Ping for 1D ISI channels [6]. Then, we show how to use the proposed algorithm for joint equalization and decoding. Finally, the performances of the proposed scheme are assessed through numerical simulations.

2. SYSTEM MODEL
Consider a 2D ISI channel whose input \( \{ b_{i,j}, 0 \leq i \leq I - 1, 0 \leq j \leq J - 1 \} \) is a matrix of i.i.d. (independent and identically distributed) binary data drawn from the alphabet \( \{-1, +1\} \). Let \( \{ h_{m,n}, 0 \leq m \leq M - 1, 0 \leq n \leq N - 1 \} \) be the 2D ISI coefficients, the channel output can be written as

\[
y_{i,j} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} h_{m,n} b_{i-m,j-n} + n_{i,j}, \quad 0 \leq i \leq I - 1, 0 \leq j \leq J - 1
\]  

(1)

where \( n_{i,j} \sim N(0, \sigma^2) \) is an additive white Gaussian noise (AWGN) term.

For the \( i \)-th line of observations \( \{ y_{i,k}, 0 \leq k \leq J - 1 \} \), we define the state \( x_k \) of size \( MN \times 1 \)

\[
x_k = \begin{bmatrix}
    b_{i-(M-1),k-(N-1)} \\
    \vdots \\
    b_{i,k} 
\end{bmatrix},
\]

(2)

which consists in raster scanning columnwise the data bits illustrated in Fig. 1. Let us define the vector of 2D ISI coefficients of size \( MN \times 1 \)

\[
h = \begin{bmatrix}
    h_{M-1,N-1} \\
    \vdots \\
    h_{0,0},N-1 \\
    \vdots \\
    h_{M-1,0} \\
    \vdots \\
    h_{0,0} 
\end{bmatrix},
\]

(3)

then we obtain the following state-space representation for the \( i \)-th line of observations

\[
\begin{cases}
    x_k = Fx_{k-1} + G_{ik} \\
    y_{i,k} = h^T x_k + n_{i,k}, \quad 0 \leq k \leq J - 1
\end{cases}
\]

(4)
where the new input vector at instant $k$ is defined as

$$i_k = \begin{bmatrix} b_{i-k-(M-1),k} \\ \vdots \\ b_{i,k} \end{bmatrix}.$$  

The state transition matrices are given by

$$F = \begin{bmatrix} 0_{N-1 \times 1} & I_{N-1} \\ 0 & 0_{1 \times N-1} \end{bmatrix} \otimes I_M$$

and

$$G = \begin{bmatrix} 0_{N-1 \times 1} \\ 1 \end{bmatrix} \otimes I_M,$$

where the symbol $\otimes$ denotes the Kronecker product.

The a posteriori probability mass function (pmf) of the input vectors and the 2D ISI states, $p(i_k, x_k, y_k | \{ i_n \}_{n=1}^k)$, can be easily factored. A portion of the corresponding factor graph [5] is depicted in Fig. 2, where the function node $f_k$ represents the factor $p(x_k | i_k, x_{k-1})$ and the function node $g_k$ represents the factor $p(y_k | i_k, x_k)$.

### 3. GABP-BASED EQUALIZATION

In this section, we derive a 2D equalizer based on GaBP by approximating all the messages exchanged by the sum-product algorithm [5] on the factor graph of Fig. 2 as Gaussian distributions. The proofs are analog to the derivation of the well-known Kalman filter [7] and are therefore omitted.

#### 3.1. Prior distributions

We assume that $p(x_0)$ is known and can be approximated as the Gaussian distribution $\mathcal{N}(x_0 | \bar{x}_0, P_0)$.

#### 3.2. Forward pass

Assuming that $a \text{ priori}$ log-likelihood ratios are available for the components of $i_k$, i.e.

$$La(b_{i-m,k}) = \ln \frac{p(b_{i-m,k} = -1)}{p(b_{i-m,k} = +1)}, \quad m = 0, \ldots, M - 1,$$

a Gaussian approximation of $p(i_k)$ of the form $\mathcal{N}(i_k | \bar{i}_k, Q_k)$ is sought. The standard solution consists in matching the parameters $\bar{i}_k$ and $Q_k$ to the mean and the covariance matrix of the binary input vector $i_k$, respectively [5]:

$$\bar{i}_k = \mathbb{E}[i_k] = \begin{bmatrix} m_a(b_{i-(M-1),k}) \\ \vdots \\ m_a(b_{i,k}) \end{bmatrix}$$

where

$$m_a(b_{i,m,k}) = -\tanh \left( \frac{La(b_{i,m,k})}{2} \right), \quad m = 0, \ldots, M - 1,$$

and

$$Q_k = \mathbb{E}[(i_k - \bar{i}_k)(i_k - \bar{i}_k)^T] = \text{diag} \left( \sigma_a(b_{i,k})^2, \ldots, \sigma_a(b_{i-(M-1),k})^2 \right),$$

where

$$\sigma_a(b_{i,m,k})^2 = 1 - m_a(b_{i-m,k})^2, \quad m = 0, \ldots, M - 1.$$
We can show that (7) admits the following closed form expression

\[ \mu_{f_k \rightarrow x_k}(x_k) \propto N(x_k : \hat{x}_k|k-1, P_{k|k-1}), \]  

where

\[ \begin{cases} 
\hat{x}_k|k-1 = F\hat{x}_{k-1}|k-1 + G\eta_k \\
\hat{P}_k|k-1 = FP_{k-1|k-1}F^T + G\hat{Q}_kG^T. 
\end{cases} \]  

Now, applying the sum-product rule to the variable node \( x_k \)

\[ \mu_{x_k \rightarrow f_k+1}(x_k) = \mu_{f_k \rightarrow x_k}(x_k) \mu_{y_k \rightarrow x_k}(x_k) \propto N(x_k : \hat{x}_k|k-1, P_{k|k-1})N(y_k, h^T x_k, \sigma^2). \]

We recognize the correction step of the Kalman filter, thus

\[ \mu_{x_k \rightarrow f_k+1}(x_k) \propto N(x_k : \hat{x}_k|k, P_{k|k}), \]  

where

\[ \begin{cases} 
K_k = P_{k|k-1}h(h^T P_{k|k-1}h + \sigma^2)^{-1} \\
\hat{x}_k|k = \hat{x}_k|k-1 + K_k(y_k - h^T \hat{x}_{k-1}|k-1) \\
P_{k|k} = P_{k|k-1} - K_k h^T P_{k|k-1}. 
\end{cases} \]  

3.3. Backward pass

Let us apply the sum-product rule to the function node \( f_{k+1} \),

\[ \mu_{f_{k+1} \rightarrow x_k}(x_k) = \int \int p(x_k+1|x_{k+1}) \mu_{x_k \rightarrow f_{k+1}}(x_k+1) dx_k+1 dx_{k+1}. \]

We easily recognize that (10) corresponds to the backwards prediction step in a two-filter Kalman smoother [7], therefore, \( \mu_{x_k \rightarrow f_{k+1}}(x_k+1) \propto p(y_{k+1}, z_k|x_k+1) \). It follows that \( \mu_{x_k \rightarrow f_{k+1}}(x_k+1) \) is a likelihood, which in general cannot be assimilated to a Gaussian probability density of \( x_k \), as required for GaBP. However, from Bayes rule we have

\[ p(x_{k+1}|y_{k+1}, f_{k+1}) \propto p(y_{k+1}|f_{k+1})p(f_{k+1}|y_{k+1})p(x_{k+1}|x_k+1). \]

Thus \( p(x_{k+1}) \mu_{x_k \rightarrow f_{k+1}}(x_k+1) \) can be parameterized by a Gaussian density of \( x_k \) such that

\[ p(x_{k+1}) \mu_{x_k \rightarrow f_{k+1}}(x_k+1) \propto N(x_k : \hat{x}_{k|k+1}, P_{k+1|k+1}). \]  

We must rearrange (10) so that (11) appears explicitly in the recursion. First, we express the forward dynamics \( p(x_{k+1}|x_k) \) as a function of the backward dynamics \( p(x_k|x_{k+1}) \) using Bayes rule

\[ p(x_{k+1}|x_k) = \frac{p(x_{k+1}|x_k+1)p(x_k+1)}{p(x_k)}. \]

We can show that the prior distribution of the 2D ISI states and the corresponding backward dynamics are Gaussian of the form

\[ \begin{aligned}
&\{ p(x_k) = N(x_k : \hat{x}_k, \Pi_k) \\
&p(x_k|x_{k+1}) = N(x_k : \hat{F}_{k+1}x_{k+1} + c_k + 1, \hat{Q}_{k+1})
\end{aligned} \]

(see proposition 7 in [8] for the demonstration and the expression of the parameters \( \hat{x}_k, \Pi_k, \hat{F}_{k+1}, c_k+1, \hat{Q}_{k+1} \)).

Then, injecting (12) into (10) and using (13) leads to

\[ \begin{aligned}
p(x_k) &\mu_{f_{k+1} \rightarrow x_k}(x_k) \\
&= \int \int p(x_k|x_{k+1}) \mu_{x_{k+1} \rightarrow f_{k+1}}(x_k+1) dx_{k+1} \\
&\propto N(x_k : \hat{x}_{k|k+1}, \Pi_{k|k+1}).
\end{aligned} \]

Again, we simplify this expression to

\[ \begin{aligned}
p(x_k) &\mu_{x_k \rightarrow f_k}(x_k) \propto N(x_k : \hat{x}_{k|k}, \Pi_{k|k}).
\end{aligned} \]

3.4. Computation of extrinsic log-likelihood ratios

The sum-product algorithm computes the \textit{a posteriori} marginal probability distribution of the 2D ISI state \( x_k \) as the product of all incoming messages to variable node \( x_k \) in Fig. 2.

\[ p(x_k|y_{1:k}) \propto \mu_{f_k \rightarrow x_k}(x_k) \times p(x_k) \mu_{f_{k+1} \rightarrow x_k}(x_k) \mu_{y_k \rightarrow x_k}(x_k) \]

which, according to (8), (15) and (13), can be rewritten as

\[ p(x_k|y_{1:k}) \propto N(x_k : \hat{x}_{k|k-1}, P_{k|k-1})N(x_k : \hat{x}_{k|k-1}, P_{k|k-1}). \]

After straightforward algebraic manipulations, we obtain the following simplification

\[ p(x_k|y_{1:k}) \propto N(x_k : \hat{x}_{k|k}, \Pi_{k|k}). \]
where

\[
\begin{align*}
\mathbf{P}_{k|1:I,J} &= \left[ \mathbf{P}^{-1}_{k|k-1} + \mathbf{P}^{-1}_{k|k,J} - \Pi_{k}^{-1} \right]^{-1} \\
\hat{\mathbf{x}}_{k|1:I,J} &= \mathbf{P}_{k|1:I,J}^{-1} \mathbf{P}_{k|k-1} \hat{\mathbf{x}}_{k|k-1} + \mathbf{P}_{k|k,J}^{-1} \hat{\mathbf{x}}_{k|J} - \Pi_{k}^{-1} \mathbf{x}_{k}.
\end{align*}
\]  

(17)

The last M coordinates of \( \mathbf{x}_{k} \) correspond to the data vector \( \mathbf{1}_k = [b_{i-(M-1)}, \ldots, b_{i}]^T \). Therefore, the \textit{a posteriori} probability distribution of \( b_{i-m,k} \), seen as a real random variable, has the form

\[
p(b_{i-m,k}|y_{1:i,J}) \propto N(b_{i-m,k} : m_p(b_{i-m,k}), \sigma_p(b_{i-m,k})^2),
\]

(18)

where the mean \( m_p(b_{i-m,k}) \) and the variance \( \sigma_p(b_{i-m,k})^2 \) are easily extracted from (17) by marginalization. Converting \( b_{i-m,k} \) back to a binary random variable in \( \{-1, +1\} \), the expression of the \textit{a posteriori} log-likelihood ratio is [5]

\[
L_p(b_{i-m,k}|y_{1:i,J}) = -\frac{2m_p(b_{i-m,k})}{\sigma_p(b_{i-m,k})^2}, \quad m = 0, \ldots, M - 1.
\]

However, in message passing algorithms, the messages of interest are extrinsic log-likelihood ratios obtained as [6]

\[
L_e(b_{i-m,k}|y_{1:i,J}) = -2w \left( \frac{m_p(b_{i-m,k})}{\sigma_p(b_{i-m,k})^2} - \frac{m_a(b_{i-m,k})}{\sigma_a(b_{i-m,k})} \right).
\]

(19)

A scaling factor \( 0 < w < 1 \) is introduced, because the proposed suboptimal GaBP equalizer has a tendency to overestimate the value of the output reliabilities.

**Remark 3.1** The computational complexity of the proposed GaBP equalizer is \( \mathcal{O}(MN^3) \) per data bit. Therefore, the complexity is only polynomial in the size of the 2D ISI state space, instead of exponential when using BCJR equalization [3].

### 4. SIMULATION RESULTS

We adopt a channel with \( I = J = 30, M = N = 5 \) and

\[
[h_{m,n}] = \begin{bmatrix}
0.0269 & -0.0478 & -0.1513 & -0.0478 & 0.0269 \\
-0.0478 & 0.0851 & 0.2693 & 0.0851 & -0.0478 \\
-0.1513 & 0.2693 & 0.8522 & 0.2693 & -0.1513 \\
-0.0478 & 0.0851 & 0.2693 & 0.0851 & -0.0478 \\
0.0269 & -0.0478 & -0.1513 & -0.0478 & 0.0269
\end{bmatrix}
\]

A rate-4/5 (3,15) regular LDPC code is used. In this case, IMS turbo equalization [3] would be a formidable task, since each BCJR equalizer would have to work on a trellis with 2^{20} states (i.e. more than one million discrete states). However, GaBP equalization is still manageable, although the inversion of 20 \times 20 matrices is required for each data symbol (see (17)).

Fig. 3 compares turbo equalization with simultaneous channel estimation based on 64 pilots (6 iterations), known-channel turbo equalization (6 iterations) and known-channel MMSE equalization followed by ten rounds of LDPC decoding.

### 5. REFERENCES


