A GENERALIZED BCJR ALGORITHM AND ITS USE IN TURBO SYNCHRONIZATION

Jacob H. Gunther, David Keller, Todd K. Moon

Department of Electrical & Computer Engineering • Utah State University

ABSTRACT

A generalization of the BCJR algorithm is derived to compute joint posterior probabilities of arbitrary sets of symbols given received data. A special case of the new algorithm that computes pairwise joint probabilities is used to evaluate the second order statistics needed for EM estimation of carrier phase and symbol timing offsets in a digitally modulated waveform. This EM estimator is integrated into a turbo synchronization loop in which soft information is exchanged between the generalized BCJR algorithm (the synchronizer) and an LDPC decoder.

1. INTRODUCTION

This paper considers the problem of synchronizing a digital communication receiver to a carrier signal that is modulated by a root-Nyquist pulse train in which the symbols are protected by LDPC coding [1]. At the receiver, the sampled matched filter outputs contain an unknown phase rotation and intersymbol interference (ISI) due to sample timing offset. The phase and time offsets are unknown and must be estimated. In a turbo receiver, errors in phase and timing estimates degrade the accuracy of the information exchanged during turbo iterations which increases the number of iterations needed to achieve decoding success and may prevent successful decoding altogether.

In this paper, carrier phase and symbol timing are estimated anew in each turbo iteration. Expectation maximization [2] (EM) is used as the basis for estimation. The EM equations describing the optimum phase and sample times depends on the first and second order posterior conditional statistics of the symbols (posterior conditional mean and covariance) which require single variable marginal and pairwise joint posterior probabilities for their evaluation. Single variable posterior probabilities are computed by both standard LDPC decoder and BCJR algorithm. However, neither LDPC or BCJR provides higher order joint probabilities. Pairwise joint probabilities may be computed using a generalized BCJR algorithm which is derived in this paper. The generalized BCJR computes the joint posterior probability of arbitrary sets of symbols given received data. Pairwise joint probabilities are obtained as a special case. Simulation results comparing generalized-BCJR-based turbo synchronization to the ideal case in which phase and timing are known precisely are given in Section 4. The paper begins by establishing the data model (Section 2) and sets up the carrier phase and symbol timing estimation problem using expectation maximization.

Phase and timing estimation based on EM was originally performed in [3] but was not done in combination with decoding. Zhang and Burr [4] used extrinsic information produced during turbo decoding in a standard maximum likelihood based carrier phase estimator. Hard symbol decisions from the output of the turbo decoder were fed back to the phase estimator. Nuriyev and Anastasopoulos [5] used an optimal strategy to allocate power to a single pilot symbol inserted at the beginning of block of LDPC encoded bits for iterative carrier phase estimation. Carrier phase estimation was combined with turbo codes in [6]. Joint timing recovery and turbo equalization was investigated in [7, 8].

A general framework for turbo synchronization based on the EM algorithm was presented by Noels, *et al.* [9]. However, [9] used a simplifying assumption that made it possible to estimate unknown parameters using only the single variable marginal posterior probabilities computed by turbo decoders. Lottici and Luise [10] embedded carrier phase recovery into the iterations of a turbo decoder with the assumption of zero ISI to simplify the EM equations as in [9] so that only single variable posteriors were involved.

The following notation is used in the remainder of the paper. Boldface upper and lower case letters are used to denote matrices and vectors, respectively. The notation $[\mathbf{A}]_{(i,j)}$ and $[\mathbf{a}]_i$ is used to refer to specific elements of matrices and vectors. The dimensions of matrices and vectors will be given when defined. The notation $\mathbf{A} \odot \mathbf{B}$ is the element-by-element product of the two matrices $\mathbf{A}$ and $\mathbf{B}$. The Kronecker delta function δ_k will be used: $\delta_k = 1$ if $k = 0$ and $\delta_k = 0$ if $k \neq 0$. The notation $\mathbf{n} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$ indicates that $\mathbf{n}$ is a vector with jointly Gaussian distributed elements with mean $\boldsymbol{\mu}$ and covariance $\mathbf{C}$.

2. DATA MODEL

Consider a bandpass BPSK transmitted signal,

$$s(t) = \sum_{n=0}^{N-1} s_n g(t - nT)\sqrt{2}\cos(2\pi f_c t), \tag{1}$$

where $s_n \in \{+\sqrt{\mathcal{E}}, -\sqrt{\mathcal{E}}\}$ are BPSK symbols, $g(t)$ is a causal, unit-energy, square-root Nyquist pulse of duration t_0, T is the symbol period, and f_c is the carrier frequency. In this paper, the symbols s_n are derived from bits appearing at the output of an LDPC encoder via the map: $1 \to +\sqrt{\mathcal{E}}, 0 \to -\sqrt{\mathcal{E}}$. The code word length is N bits. Therefore, (1) represents an isolated transmission of a single LDPC codeword.

Propagation delay in the channel leads to a time delay τ_0 in the received pulse train and a carrier phase offset φ_0 as follows,

$$r(t) = \sum_{n=0}^{N-1} s_n g(t - nT - \tau_0)\cos(2\pi f_c t + \varphi_0) + n(t).$$

Time and carrier phase offsets in this paper are defined relative to the transmitter. Additive white Gaussian noise with (two-sided) power spectral density $N_0/2$ has also been added.

After I/Q demodulation using the a phase of φ_1, matched filtering and sampling at time $t = t_0 + kT + \tau_1$, the pair of samples taken during the k^{th} symbol period may be written in vector form as,

$$\mathbf{y}_k = \begin{bmatrix}\cos\varphi \\ \sin\varphi\end{bmatrix} \sum_{n=0}^{N-1} s_n R_{gg}([k-n]T - \tau) + \mathbf{n}_k, \qquad (2)$$

$$\mathbf{n}_k \sim \mathcal{N}\left(\mathbf{0}, \frac{N_0}{2}\mathbf{I}_{2\times 2}\right).$$

where $\varphi = \varphi_0 - \varphi_1$ is the phase error, $\tau = \tau_0 - \tau_1$ is the sample timing offset, and

$$R_{gg}(t) = \int_{-\infty}^{t_0} g(\nu + t) g(\nu) d\nu$$

is the pulse autocorrelation function. Since $g(t)$ is a square-root Nyquist pulse, $R_{gg}(t)$ is a Nyquist pulse. Therefore, $R_{gg}([k-n]T - \tau) = \delta_{k-n}$ if and only if $\tau = 0$. In this paper, we assume that the timing error is less than a half symbol period, $|\tau| < \frac{T}{2}$.

The support of $R_{gg}(\tau)$ is $0 \le t \le 2t_0$. Let P be the number of whole symbol periods of duration T in $2t_0$ seconds and let α be the fractional remainder, $P = \lfloor \frac{2t_0}{T} \rfloor$ and $\alpha = \frac{2t_0}{T} - P$. Let $\beta = \frac{\tau}{T}$. Then $|\beta| < \frac{1}{2}$. Now, since $R_{gg}(\tau) = 0$ for $\tau < 0$ and $\tau > 2t_0$, the only values of n for which $R_{gg}([k-n]T - \tau)$ is nonzero are in the range $k - \beta - [P + \alpha] \le n \le k - \beta$. Define Q as follows,

$$Q = \lceil P + \alpha + 1 \rceil = \begin{cases} P+1 & \alpha = 0 \\ P+2 & 0 < \alpha < 1 \end{cases}.$$

Now (2) may be written as,

$$\mathbf{y}_k = \mathbf{c}(\varphi)\mathbf{s}_k^T \mathbf{p}(\tau) + \mathbf{n}_k, \qquad k = 0, \cdots, N+Q-2$$

with the definition of the vectors

$$\mathbf{c}(\varphi) = \begin{bmatrix}\cos\varphi & \sin\varphi\end{bmatrix}^T$$

$$\mathbf{s}_k = \begin{bmatrix}s_k & s_{k-1} & \cdots & s_{k-Q+1}\end{bmatrix}^T$$

$$\mathbf{p}(\tau) = \begin{bmatrix}R_{gg}(0\cdot T - \tau) & \cdots & R_{gg}([Q-1]\cdot T - \tau)\end{bmatrix}^T.$$

In the preceeding development, the timing error and phase offset were implicitly assumed to be constant over the whole data block. However, the development can be generalized so that both of these parameters are time varying. To account for this generalization, subscripts shall be added to the parameters to indicate the time at which they are applicable. Maximum likelihood estimates for the symbol timing error, τ_n, and the carrier phase offset, φ_n, at time n based on the single sample $\mathbf{y}_n$ may be obtained by minimizing $-\log p(\mathbf{y}_n|\tau_n, \varphi_n)$ where $p(\mathbf{y}_n|\tau_n, \varphi_n)$ is the probability density function of $\mathbf{y}_n$ parameterized by the unknowns τ_n and φ_n. However, due to the presence of unknown symbols, only the distribution $p(\mathbf{y}_n|\mathbf{s}_n, \tau_n, \varphi_n)$ is available. When applied to the problem of synchronization, the EM principle begins with initial estimates $\tau_n^{[0]}$ and $\varphi_n^{[0]}$ and iteratively minimizes,

$$\begin{pmatrix}\tau_n^{[i+1]} \\ \varphi_n^{[i+1]}\end{pmatrix} = \arg\min_{\tau,\varphi} \left\{ \begin{array}{c} \|\mathbf{y}_n\|^2 - 2\mathbf{c}^T(\varphi)\mathbf{B}_n^{[i]}\mathbf{p}(\tau) \\ +\mathbf{p}^T(\tau)\mathbf{R}_n^{[i]}\mathbf{p}(\tau) \end{array} \right\} \qquad (3)$$

where $\tau_n^{[i]}$ and $\varphi_n^{[i]}$ are estimates of τ_n and φ_n obtained on the i^{th} iteration, and where the matrices $\mathbf{R}_n^{[i]} = E(\mathbf{s}_n\mathbf{s}_n^T|\mathbf{y}_n, \tau_n^{[i]}, \varphi_n^{[i]})$ and $\mathbf{B}_n^{[i]} = \mathbf{y}_n E(\mathbf{s}_n^T|\mathbf{y}_n, \tau_n^{[i]}, \varphi_n^{[i]})$ are the posterior conditional autocorrelation of the symbols and cross-correlation between symbols and data. Given τ, the minimum of (3) with respect to φ is easily found to be,

$$\varphi_n^{[i+1]} = \arctan\frac{[\mathbf{B}_n^{[i]}\mathbf{p}(\tau)]_2}{[\mathbf{B}_n^{[i]}\mathbf{p}(\tau)]_1}. \qquad (4)$$

In evaluating (4), τ is replaced by $\tau_n^{[i]}$. Note that the solution for $\varphi_n^{[i]}$ only depends on the first order statistics (posterior conditional mean) of the symbols contained in $\mathbf{B}_n^{[i]}$.

A closed form solution for $\tau_n^{[i+1]}$ is difficult because τ appears in the argument of the pulse. The minimization with respect to τ in (3) may be carried out by gradient descent. The gradient of (3) is easily calculated to be

$$g(\tau, \varphi) = 2\left[\mathbf{c}^T(\varphi)\mathbf{B}_n^{[i]} - \mathbf{p}^T(\tau)\mathbf{R}_n^{[i]}\right]\frac{\partial \mathbf{p}(\tau)}{\partial \tau}. \qquad (5)$$

3. COMPUTING JOINT A POSTERIORI PROBABILITIES WITH A GENERALIZED BCJR ALGORITHM

The BCJR algorithm [11] is commonly used to compute single variable posterior probabilities. A generalization of the BCJR algorithm showing how to compute joint posterior probabilities of arbitrary sets of symbols is presented which reproduces the standard BCJR algorithm as a special case. Formulas for computing the joint posterior probability of a pair of symbols will be derived first. From this, formulas for joint posterior probabilities of arbitrary sets of symbols is easily inferred.

Suppose a sequence of M-ary symbols $\{s_n\}_{n=0}^{N-1}$ is passed through an ISI channel with L memory elements and impulse response $h_0, \cdots, h_L$. The channel may be in one of $Q = M^L$ states $(s_{k-1}, \cdots, s_{k-L})$. Let $\xi_k \in \{0, \cdots, Q-1\}$ be the state index at time k. Let $Y = \{y_0, \cdots, y_{N+L-1}\}$ be the set of measurements (sampled matched filter outputs) at the receiver where $y_k = \sum_{\ell=0}^{L} h_\ell s_{k-\ell} + n_k = \mathbf{s}_k^T\mathbf{h} + n_k$. Given the distribution of the noise and the channel impulse response, likelihoods such as $p(y_k|\mathbf{s}_k)$ may be computed. We desire to evaluate the posterior probability $P(s_k = a, s_\ell = b|Y)$ where, without loss of generality, $\ell > k$.

Let $Q_{a,k}$ be the set of all state transitions at time k caused by the input symbol $s_k = a$. For example, $(p, q) \in Q_{a,k}$ if $s_k = a$ caused a transition from state $\xi_k = p$ to state $\xi_{k+1} = q$. With these definitions, the question about the joint probability of symbols s_k, s_ℓ may be transformed into a question about the joint probabilities of states as follows,

$$P\left(\begin{array}{c} s_k = a, \\ s_\ell = b \end{array}\middle| Y\right) = \sum_{\substack{(p,q)\in Q_{a,k} \\ (r,s)\in Q_{b,\ell}}} P\left(\begin{array}{c} \xi_k = p, \\ \xi_{k+1} = q, \\ \xi_\ell = r, \\ \xi_{\ell+1} = s \end{array}\middle| Y\right).$$

Following reasoning similar to that used in standard BCJR developments [11], split the data into the five sets, $Y = (Y_{n<k}, y_k, Y_{k<n<\ell}, y_\ell, Y_{n>\ell})$, where $Y_{n<k} = \{y_1, \cdots, y_{k-1}\}$, $Y_{k<n<\ell} = \{y_{k+1}, \cdots, y_{\ell-1}\}$ and $Y_{n>\ell} = \{y_{\ell+1}, \cdots, y_N\}$. By careful use of conditioning, the joint probability of the states may be factored as,

$$\begin{aligned}
&P(\xi_k = p, \xi_{k+1} = q, \xi_\ell = r, \xi_{\ell+1} = s|Y)P(Y) \\
&= P(\xi_k = p, \xi_{k+1} = q, \xi_\ell = r, \xi_{\ell+1} = s, \\
&\qquad Y_{n<k}, y_k, Y_{k<n<\ell}, Y_\ell, Y_{n>\ell}) \\
&= P(Y_{n>\ell}|\xi_{\ell+1} = s, \xi_\ell = r, \xi_{k+1} = q, \xi_k = p, \\
&\qquad y_\ell, Y_{k<n<\ell}, y_k, Y_{n<k}) \\
&\times P(y_\ell, \xi_{\ell+1} = s|\xi_\ell = r, \xi_{k+1} = q, \xi_k = p, Y_{k<n<\ell}, y_k, Y_{n<k}) \\
&\times P(Y_{k<n<\ell}, \xi_\ell = r|\xi_{k+1} = q, \xi_k = p, y_k, Y_{n<k}) \\
&\times P(y_k, \xi_{k+1} = q|\xi_k = p, Y_{n<k}) \\
&\times P(Y_{n<k}, \xi_k = p).
\end{aligned}$$

The conditional probabilities may be simplified by invoking the following two facts:

1. y_ℓ is conditionally independent of y_k for all $k < \ell$ given the state ξ_ℓ, and
2. the state $\xi_{\ell+1}$ is conditionally independent of ξ_k for all $k < \ell$ given ξ_ℓ.

Application of these independence properties leads to the following simplifications,

$$P\begin{pmatrix} Y_{n>\ell}|\xi_{\ell+1}=s, \xi_\ell=r, \xi_{k+1}=q, \\ \xi_k=p, y_\ell, Y_{k<n<\ell}, y_k, Y_{n<k}\end{pmatrix} = P(Y_{n>\ell}|\xi_{\ell+1}=s)$$

$$P\begin{pmatrix} y_\ell, \xi_{\ell+1}=s|\xi_\ell=r, \xi_{k+1}=q, \\ \xi_k=p, Y_{k<n<\ell}, y_k, Y_{n<k}\end{pmatrix} = P\begin{pmatrix} y_\ell, \xi_{\ell+1}=s \\ |\xi_\ell=r\end{pmatrix}$$

$$P\begin{pmatrix} Y_{k<n<\ell}, \xi_\ell=r|\xi_{k+1}=q, \\ \xi_k=p, y_k, Y_{n<k})\end{pmatrix} = P\begin{pmatrix} Y_{k<n<\ell}, \xi_\ell=r \\ |\xi_{k+1}=q\end{pmatrix}$$

$$P(y_k, \xi_{k+1}=q|\xi_k=p, Y_{n<k}) = P(y_k, \xi_{k+1}=q|\xi_k=p)$$

Inserting these simplified probabilities along with the definitions of α, β, and γ from the standard BCJR algorithm,

$$\alpha_k(p) = P(Y_{n<k}, \xi_k=p),$$
$$\beta_{\ell+1}(s) = P(Y_{n>\ell}|\xi_{\ell+1}=s),$$
$$\gamma_k(p,q) = P(y_k, \xi_{k+1}=q|\xi_k=p, Y_{n<k}),$$

leads to the following formula for the pairwise joint probability,

$$P(\xi_k=p, \xi_{k+1}=q, \xi_\ell=r, \xi_{\ell+1}=s|Y)P(Y) = \qquad (6)$$
$$\alpha_k(p)\gamma_k(p,q)P(Y_{k<n<\ell}, \xi_\ell=r|\xi_{k+1}=q)\gamma_\ell(r,s)\beta_{\ell+1}(s).$$

The only new quantity appearing in (6) is $P(Y_{k<n<\ell}, \xi_\ell = r| \xi_{k+1}=q)$ which is the probability of observing $y_{k+1}, \cdots, y_{\ell-1}$ and ending in state r at time ℓ given that channel started in state q at time $k+1$. This probability can be factored into sums and products of likelihoods and transition probabilities. To see this, consider the following special cases.

- Suppose $\ell = k+1$. Then, $Y_{k<n<\ell} = \{\}$ and the probability in question is simply $P(\xi_{k+1}=r|\xi_{k+1}=q) = \delta_{r-q} = 1$ if $r=q$ and 0 otherwise.

- If $\ell = k+2$, then $Y_{k<n<\ell} = \{y_{k+1}\}$ and $P(y_{k+1}, \xi_{k+2} = r|\xi_{k+1}=q) = \gamma_{k+1}(q,r)$ by definition.

- More interesting situations arise when $\ell > k+2$. Let $\ell = k+3$. Then $Y_{k<n<\ell} = \{y_{k+1}, y_{k+2}\}$. Introduce a new state variable ξ_{k+2} and marginalize it out as follows,
$$\begin{aligned} &P(y_{k+2}, y_{k+1}, \xi_{k+3}=r|\xi_{k+1}=q) \\ &= \sum_{u\in Q} P(y_{k+2}, y_{k+1}, \xi_{k+3}=r, \xi_{k+2}=u|\xi_{k+1}=q) \\ &= \sum_{u\in Q} \begin{matrix} P(y_{k+2}, \xi_{k+3}=r|\xi_{k+2}=u)\times \\ P(y_{k+1}, \xi_{k+2}=u|\xi_{k+1}=q)\end{matrix} \\ &= \sum_{u\in Q} \gamma_{k+2}(u,r)\gamma_{k+1}(q,u). \end{aligned}$$

- Following the same procedure when $\ell = k+4$ leads to
$$P(y_{k+2}, y_{k+1}, \xi_{k+3}=r|\xi_{k+1}=q)$$
$$= \sum_{u\in Q}\sum_{v\in Q} \gamma_{k+3}(u,r)\gamma_{k+2}(v,u)\gamma_{k+1}(q,v).$$

- In general, for any $\ell > k$,
$$P(Y_{k<n<\ell}, \xi_\ell=r|\xi_{k+1}=q)$$
$$= \sum_{u_1}\sum_{u_2}\cdots\sum_{u_{\ell-k-2}} \left\{\begin{matrix} \gamma_{\ell-1}(u_{\ell-k-2}, r)\times \\ \vdots \\ \gamma_{k+2}(u_1,u_2)\times \\ \gamma_{k+1}(q,u_1)\end{matrix}\right\}, \qquad (7)$$
in which there are $\ell-k-2$ sums involving products of $\ell-k-1$ $\gamma_i(u,v)$ terms.

Thus we see that computing arbitrary pairwise joint posterior probabilities uses quantities already computed in the standard forward-backward recursions.

The sum-product form of (7) is reminiscent of repeated matrix multiplication. In [12], writing down the BCJR equations in matrix form led to a very compact description of the algorithm. Here, matrix notation also simplifies the formulas for the joint probabilities of arbitrary sets of symbols.

Below, the BCJR algorithm is expressed in both sum-product form and in the more compact matrix-vector form. Unless specified otherwise, all summations are over all Q states. To rewrite sums over transitions $(p,q) \in Q_{a,k}$, define the zero-one matrix $\mathbf{A}_{a,k}$ with elements defined by $[\mathbf{A}_{a,k}]_{(p,q)} = 1$ if $(p,q) \in Q_{a,k}$ and is zero otherwise. Also define the matrix $\mathbf{\Gamma}_k$ and vectors $\boldsymbol{\alpha}_k(p)$ and $\boldsymbol{\beta}_k(p)$ by $[\mathbf{\Gamma}_k]_{(p,q)} = \gamma_k(p,q), [\boldsymbol{\alpha}_k]_p = \alpha_k(p)$, and $[\boldsymbol{\beta}_k]_p = \beta_k(p)$.

Forward-Backward Equations:

$$\alpha_{k+1}(q) = \sum_p \alpha_k(p)\gamma_k(p,q) \qquad \boldsymbol{\alpha}_{k+1}^T = \boldsymbol{\alpha}_k^T\mathbf{\Gamma}_k$$

$$\beta_k(p) = \sum_q \gamma_k(p,q)\beta_{k+1}(q) \qquad \boldsymbol{\beta}_k = \mathbf{\Gamma}_k\boldsymbol{\beta}_{k+1}$$

Single Variable Marginal A Posteriori Probability:

$$P(s_k=a|Y) = \frac{\sum_{(p,q)\in Q_{a,k}} \alpha_k(p)\gamma_k(p,q)\beta_{k+1}(q)}{\sum_p\sum_q \alpha_k(p)\gamma_k(p,q)\beta_{k+1}(q)}$$
$$= \frac{\boldsymbol{\alpha}_k^T\left(\mathbf{\Gamma}_k \odot \mathbf{A}_{a,k}\right)\boldsymbol{\beta}_{k+1}}{\boldsymbol{\alpha}_k^T\mathbf{\Gamma}_k\boldsymbol{\beta}_{k+1}}$$

The probability calculation in (7) may be written in matrix notation as,

$$P(Y_{k<n<\ell}, \xi_\ell=r|\xi_{k+1}=q)$$
$$= \left[\prod_{i=k+1}^{\ell-1}\mathbf{\Gamma}_i\right]_{(q,r)} = \begin{cases} [\mathbf{I}]_{(q,r)} = \delta_{r-q} & \ell=k+1, \\ [\mathbf{\Gamma}_{k+1}]_{(q,r)} = \gamma_{k+1}(q,r) & \ell=k+2, \\ [\mathbf{\Gamma}_{k+1}\mathbf{\Gamma}_{k+2}]_{(q,r)} & \ell=k+3, \\ [\mathbf{\Gamma}_{k+1}\cdots\mathbf{\Gamma}_{\ell-1}]_{(q,r)} & \ell\ge k+4\end{cases}$$

Using this result, the pairwise joint probability in (6) becomes,

Pairwise Joint A Posteriori Probability:

$$P(s_k=a, s_\ell=b|Y)$$
$$= \frac{\boldsymbol{\alpha}_k^T\left(\mathbf{\Gamma}_k\odot\mathbf{A}_{a,k}\right)\left[\prod_{i=k+1}^{\ell-1}\mathbf{\Gamma}_i\right]\left(\mathbf{\Gamma}_\ell\odot\mathbf{A}_{b,\ell}\right)\boldsymbol{\beta}_{\ell+1}}{\boldsymbol{\alpha}_k^T\mathbf{\Gamma}_k\boldsymbol{\beta}_{k+1}}.$$

This formula is easily extended to higher order joint probabilities. Define the event $\mathcal{I} = \{s_{k_1}=a_1, \cdots, s_{k_J}=a_J\}$ where without loss of generality $k_1 < k_2 < \cdots < k_J$. The joint posteriori probability of $\mathcal{I}$ given Y is, *General Joint A Posteriori Probability:*

$$
\begin{aligned}
P(\mathcal{I}|Y) &= \boldsymbol{\alpha}_{k_1}^T \left(\Gamma_{k_1} \odot \mathbf{A}_{a_1,k_1}\right) \\
&\times \left[\prod_{i=k_1+1}^{k_2-1} \boldsymbol{\Gamma}_i \right] \left(\Gamma_{k_2} \odot \mathbf{A}_{a_2,k_2}\right) \\
&\times \left[\prod_{i=k_2+1}^{k_3-1} \boldsymbol{\Gamma}_i \right] \left(\Gamma_{k_3} \odot \mathbf{A}_{a_3,k_3}\right) \\
&\vdots \\
&\times \left[\prod_{i=k_{J-1}+1}^{k_J-1} \boldsymbol{\Gamma}_i \right] \left(\Gamma_{k_J} \odot \mathbf{A}_{a_J,k_J}\right) \boldsymbol{\beta}_{k_J+1} / \boldsymbol{\alpha}_k^T \boldsymbol{\Gamma}_k \boldsymbol{\beta}_{k+1}.
\end{aligned}
$$

4. SIMULATIONS

Five hundred independent trials were performed at several different SNRs. In each trial, random carrier phase, symbol timing offset, and noise sequence were generated. The observations and uniform priors were input to the EM-based turbo synchronizer. Our experiments showed that good initial guesses for τ and φ helped the EM-based turbo synchronizer to converge quickly. A technique for arriving at good initial guesses was developed and used in the simulations. Space limitation does not permit elaboration on this technique here. In our simulations, turbo iterations stop when either the LDPC decoder successfully decodes or a maximum of 10 turbo iterations is reached. Figure 1 shows RMS errors of phase and timing estimates generated by EM versus turbo iteration number. The estimates converge rapidly to less than one hundredth of a radian (in phase) and less than one hundredth of a symbol period (in time). Figure 2 illustrates the bit error rate (BER) performance of the EM-based turbo synchronizer compared a turbo synchronizer in which the phase and timing offset are known exactly. The EM-based turbo synchronizer performs as well after 10 iterations as a parameter informed turbo synchronizer. Also shown is the BER performance of a parameter informed turbo synchronizer when there is no phase or timing error. It appears that there is an average penalty of about 1 dB associated with synchronizing to an unknown phase and timing offset. A length 20,000, rate $\frac{1}{2}$ LDPC code was used in these simulations.

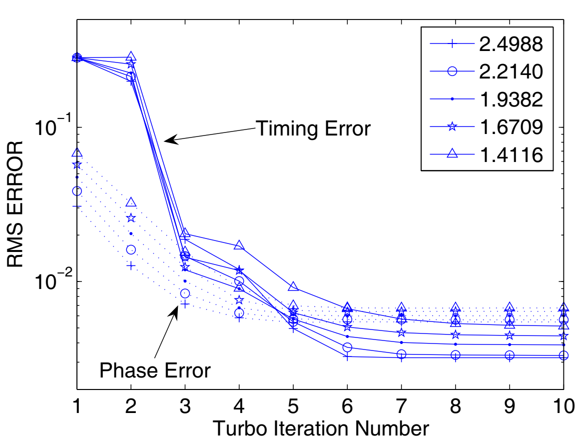

Fig. 1. Phase and timing timing estimation error.

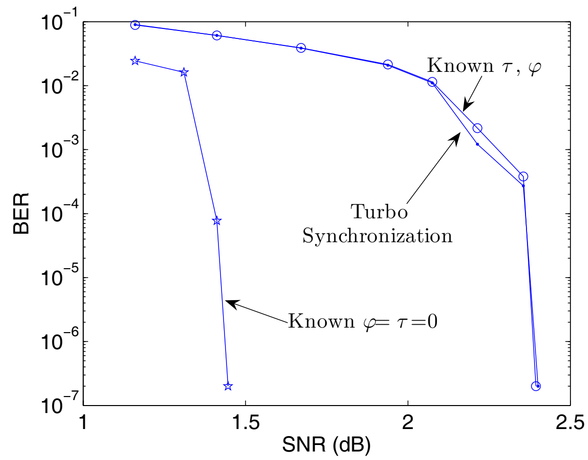

Fig. 2. BER after ten turbo iterations.

5. REFERENCES

[1] R. Gallager, "Low-density parity-check codes," *IRE Trans. Information Theory*, vol. IT-8, pp. 21–28, Jan 1962.

[2] A. Dempster, N. Laird, and D. Rubin, "Maximum-likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc., ser. B*, vol. 39, pp. 1–38, Jan. 1977.

[3] C. N. Georghiades and D. L. Snyder, "The expectation-maximization algorithm for symbol unsynchronized sequence detection," *IEEE Trans. Commun.*, vol. 39, pp. 54–61, Jan. 1991.

[4] L. Zhang and A. G. Burr, "A new method of carrier phase recovery for BPSK system using Turbo-codes over AWGN channel," in *Proc. 12th IEEE Internat. Symp. on Personal, Indoor and Mobile Radio Com*, vol. 1, pp. A179–A183, 2001.

[5] R. Nuriyev and A. Anastasopoulos, "Analysis of joint iterative decoding and phase estimation for the AWGN channel using density evolution," in *Proc. IEEE Inter. Symp. Info. Thry.*, p. 168, 2002.

[6] J. Hamkins and D. Divsalar, "Coupled receiver-decoders for low rate turbo codes," in *Proc. IEEE Int. Symp. Information Theory*, (Yokohama, Japan), p. 381, June 2003.

[7] J. R. Barry, A. Kavcic, S. W. McLaughlin, A. Nayak, and W. Zeng, "Iterative timing recovery," *IEEE Signal Processing Mag.*, pp. 89–102, 2004.

[8] A. R. Nayak, J. R. Barry, and S. W. McLaughlin, "Joint timing recovery and turbo equalization for coded partial response channels," *IEEE Trans. Magnetics*, vol. 38, pp. 2295–2297, Sept 2002.

[9] N. Noels, C. Herzet, A. Dejonghe, V. Lottici, H. Steendam, M. Moeneclaey, M. Luise, and L. Vandendorpe, "Turbo synchronization: an EM algorithm interpretation," in *Proc. IEEE Int. Conf. Commun.*, vol. 4, pp. 11–15, May 2003.

[10] V. Lottici and M. Luise, "Embedding carrier phase recovery into iterative decoding of turbo-coded linear modulations," *IEEE Trans. Commun.*, vol. 52, pp. 661–669, Apr. 2004.

[11] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. IT*, vol. 20, pp. 284–287, Mar. 1974.

[12] R. Koetter, A. C. Singer, and M. Tüchler, "Turbo equalization," *IEEE Signal Processing Magazine*, vol. 21, pp. 67–80, Jan. 2004.