Transactions Papers

Joint Synchronization and SNR Estimation for Turbo Codes in AWGN Channels

Jian Sun, *Member, IEEE,* and Matthew C. Valenti, *Member, IEEE*

***Abstract*—Turbo codes are sensitive to both (timing) synchronization errors and signal-to-noise ratio (SNR) mismatch. Since turbo codes are intended to work in environments with very low SNR, conventional synchronization methods often fail. This paper investigates blind symbol-timing synchronization and SNR estimation based on oversampled data frames. The technique is particularly suitable for low-rate turbo codes operating in additive white Gaussian noise at low SNR and modest data-transfer rates, as in deep space, satellite, fixed wireless, or wireline communications. In accordance with the turbo principle, intermediate decoding results are fed back to the estimator, thereby facilitating decision-directed estimation. The analytical and simulated results show that with three or more samples per symbol and raised cosine-rolloff pulse shaping, performance approaches that of systems with perfect timing and SNR knowledge at the receiver.**

***Index Terms*—Channel coding, channel estimation, iterative decoding, synchronization, turbo codes.**

I. Introduction

TURBO codes are capable of remarkable performance in very low signal-to-noise ratio (SNR) environments [1]. However, the full potential of turbo codes is only achieved if the channel state is known by the receiver. In addition to estimating the SNR of the channel [2], the receiver must synchronize with the bit epochs. Without a synchronization algorithm, the matched filter will (almost always) not be sampled at the proper instant. While timing synchronization is an issue for any digital transmission system, it is especially important for turbo codes, which operate at SNRs that are often too low for conventional synchronization techniques to work reliably.

Because of the challenge of capacity-approaching codes, synchronization has received renewed interest in the recent literature. Liu *et al.* [3] and Nayak *et al.* [4] discuss the performance of Mueller and Müller's synchronization method [5] with a phase-locked loop (PLL), and use it to track timing in systems that are protected by turbo or low-density parity-check (LDPC) codes. Nayak *et al.* [6] derived a lower bound for iterative timing recovery with a PLL-based structure, which has a gap of 7 dB to the Cramer–Rao bound (CRB) [7]. Wu *et al.* [8] discuss the effect of interpolation timing recovery (ITR) using the minimum mean-square error (MMSE) algorithm over a sliding window; the accuracy of the estimator increases with larger window sizes and higher SNR, but the complexity of this algorithm grows exponentially with the window size. In [9], Lu and Wilson propose a front-end synchronizer that uses a combination of an early–late gate and decision feedback (DF) using hard decisions on the code symbols (i.e. soft outputs from the turbo decoder are not exploited). In [10], Mielczarek and Svensson investigate the distribution of extrinsic information within a turbo decoder as a function of timing offset, and introduce a soft-bit combining method for synchronization that employs two separate turbo decoders and generates the likelihood of each data bit by combining the two decoder's soft outputs.

Another widely used technique is equalization [11]. An equalizer can compensate for the signal distortion caused not only by the channel and modulation, but also from the timing offset. However, the channel must be precisely estimated. Furthermore, symbol-rate equalizers cannot recover signal energy loss due to improper timing.

In this paper, we propose symbol timing-synchronization techniques for low-rate, capacity-approaching turbo codes operating over additive white Gaussian noise (AWGN) channels with random timing offset, bandlimited pulse shapes, and modest data rates. The proposed solutions are suitable for deep space, satellite, fixed wireless, or wireline communications. The goal of our study is to develop a synchronization algorithm with performance that is comparable to the one proposed in [10], but requires only a single turbo decoder. Our method jointly estimates SNR and timing of the received signal. In the first part of this paper, symbol synchronization using Gardner's method [12] and ITR is achieved separately from decoding, i.e., without requiring information from the decoder to be fed back to the estimator. This strategy is an extension of the SNR-estimation work of Summers and Wilson [2]. In the second part of this paper, soft estimates of the codeword are fed back from the decoder to the estimator and used to refine the joint timing/SNR estimate. It will be shown that this improves performance, although at the cost of additional complexity.

The remainder of this paper is organized as follows. Section II presents the system model. Section III discusses our approach to combined synchronization and SNR estimation. Section IV considers iterative decoding and timing estimation. Section V gives simulation results, and Section VI concludes the discussion. An Appendix is included that derives the mean and variance of the online statistic that forms the foundation of the proposed technique.

Paper approved by A. K. Khandani, the Editor for Coding and Information Theory of the IEEE Communications Society. Manuscript received November 18, 2003; revised July 23, 2004. This work was supported by the Office of Naval Research under Grant N00014-00-0655. This paper was presented in part at the IEEE International Conference on Communications, Anchorage, AK, May 2003.

J. Sun is with the Department of Electrical and Computer Engineering, University of Pittsburgh, Pittsburgh, PA 15261 USA (e-mail: jiansun@ieee.org).

M. C. Valenti is with the Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV 26506 USA (valenti@ieee.org).

Digital Object Identifier 10.1109/TCOMM.2005.851586

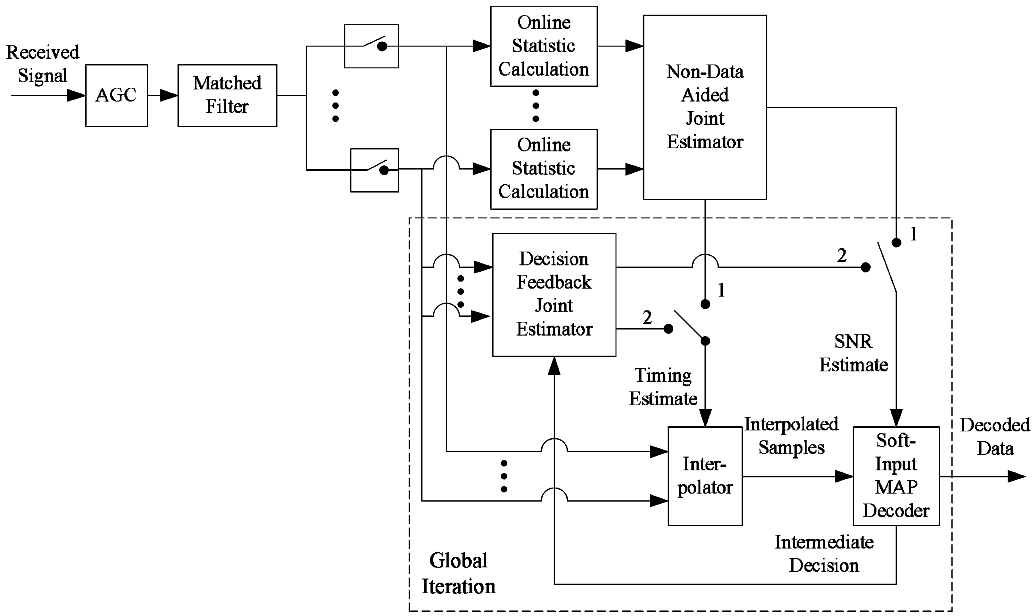

Fig. 1. Proposed receiver structure featuring joint timing/SNR estimation and iterative decoding/estimation.

II. System Model

The turbo-coded system considered in this paper is shown in Fig. 1. The system consists of a matched filter, whose output is sampled N times per symbol period, where N is assumed to be an integer with typical values between 2 and 4. For clarity, the diagram shows N samplers, each clocked at the symbol rate at sample instants that are shifted from one sampler to the next by T/N, where T is the symbol period. A practical *synchonous sampling* implementation would use a single sampler clocked at N times the symbol rate [12]. Each of the N samplers passes its samples of the entire codeword to an algorithm that calculates an online statistic for that sampler. All the samples are stored in memory according to their sampling order, forming N sample sequences. The online statistics are then passed to an estimator that jointly performs timing estimation and SNR estimation.

With the switches being placed in position 1, the timing estimate is used to control an interpolator which combines the samples from the matched filter to yield a sufficient statistic for each symbol. The turbo decoder uses the interpolated samples as input. The turbo decoder is implemented with the maximum *a posteriori* (MAP) algorithm [1], so it needs an SNR estimate. We assume that the receiver has perfect carrier and phase synchronization. Furthermore, we assume that perfect frame synchronization is achieved, so that the channel is quasi-static in the sense that it behaves as an AWGN channel for the duration of the frame, and that the timing offset is constant for the entire frame (although the channel SNR and timing offset may vary from frame to frame).

The part enclosed by dotted lines is the subsystem for decision-aided estimation. Due to the iterative nature of turbo processing, timing and SNR estimates can be refined using data-aided methods. In order to do this, intermediate decoding results are used as a reference to the transmitted code word. The turbo decoder runs a few decoding iterations (local iterations), and then feeds intermediate decisions back to a DF estimator, forming a global iteration. The switches flip to position 2. The new timing estimate is used to reinterpolate the samples, reconstructing a new input to the turbo decoder. The turbo decoder reuses the extrinsic information from previous local iterations, but uses the new reconstructed samples and SNR estimates for further decoding. The DF estimation requires knowledge of the energy transmitted per symbol $\mathcal{E}_s$ (if the code rate is R, then the energy per information bit $\mathcal{E}_b$ is $\mathcal{E}_s/R$), therefore, an automatic gain control (AGC) block is needed. We assume that perfect AGC is available, so that $\mathcal{E}_s$ is normalized to unity.

Assuming that bandlimited pulse shaping is used, the output of the matched filter is

$$r(t) = \sum_{k=-\infty}^{\infty} \sqrt{\mathcal{E}_s} a_k g(t - kT) + w(t) \tag{1}$$

where $\{a_k\}$ is the transmitted code sequence, which for binary phase-shift keying (BPSK) is $a_k \in \{-1, +1\}$, $g(t)$ is the pulse shape, and $w(t)$ is additive Gaussian noise. The variance of $w(t)$ is $\sigma^2 = N_0/2$, where N_0 is the one-sided power spectral density of the Gaussian noise prior to matched filtering (under the assumption that $g(0) = 1$). In the following discussion, we assume that $g(t)$ is a (root) raised cosine-rolloff (RC-rolloff) pulse shape with rolloff factor α [13], although we note that our results can be generalized to other pulse shapes.

The received signal $r(t)$ is sampled N times per symbol period, and the kth sample taken by the nth sampler is

$$r_n[k] = r\left(kT + \frac{n-1}{N}T - \frac{T}{2} + \tau\right), \quad 1 \le n \le N \tag{2}$$

where τ is the timing offset. The assumption of perfect frame synchronization implies that $-T/2N \le \tau \le T/2N$. With perfect timing (i.e., $\tau = 0$), only one sample per symbol is needed (i.e., $N = 1$), and the output of the matched filter has no intersymbol interference (ISI) if the pulse shape satisfies the Nyquist criteria [13]. Thus the SNR of the samples is exactly $\mathcal{E}_s/N_0$. However, when perfect timing is not available, the performance may degrade significantly. For RC-rolloff pulse shaping, the performance degradation is not only due to the loss in received signal power, but also due to the presence of rather severe ISI.

When the received signal is free of noise, the error caused by imperfect timing is $r(kT+\tau)-\sqrt{\mathcal{E}_s}a_k$. The timing offset results in interference from adjacent symbols and a loss in received signal energy. This interference can be characterized by the normalized mean squared error (MSE). When a Nyquist pulse shape is used and the timing offset $\tau \neq 0$, this MSE is [14]

$$M(\tau)=\sum_{k=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} m_{k-j}g(\tau-kT)g(\tau-jT) -2\sum_{k=-\infty}^{\infty} m_k g(\tau-kT)+m_0 \quad (3)$$

where m_k is the autocorrelation of the coded sequence $\{a_k\}$. Assuming that the a_k's are independent and zero mean, the autocorrelation of the coded sequence is $m_k=1$ if $k=0$, and zero elsewhere. Therefore, the normalized MSE is

$$M(\tau)=\sum_{k=-\infty}^{\infty} g^2(\tau-kT)-2g(\tau)+1. \quad (4)$$

Note that this is an even function of τ. For RC-rolloff pulses, $g(0)=1$, and therefore, $M(0)=0$.

Because the ISI is independent of the channel noise, it can be modeled as an additional Gaussian noise component [10]. More specifically, we define the *effective SNR* β as the SNR at a particular timing offset when all the effects are counted, including the additive noise, the ISI, and the loss of signal power. The effective SNR can be expressed as a function of both the channel SNR β_0 and the timing offset τ

$$\beta(\tau,\beta_0)=\frac{\frac{g(\tau)\mathcal{E}_s}{N_0}}{\frac{2M(\tau)\mathcal{E}_s}{N_0}+1} \quad (5)$$

where $\beta_0=\beta(0,\mathcal{E}_s/N_0)=\mathcal{E}_s/N_0$.

III. Frame-Based Synchronization

A. Online Statistics

Since multiple samples per symbol are taken, they can be used to first estimate the effective SNR and then solve (5) for τ and β_0. If a set of $N \geq 2$ effective SNRs and the timing differences between them are known, an equation array can be established using (5). Now with $N \geq 2$ equations and two unknowns, it is possible to jointly determine β_0 and τ. Interpolation can help to recover the loss in signal energy when there are multiple samples available.

Successful implementation of the joint estimation strategy requires fairly accurate estimates of the effective SNRs for each of the N sample positions averaged over the entire frame. To compute the effective SNR estimate, we use the approach proposed by Summers and Wilson [2] which computes online statistics using sample means of r_n^2 and $|r_n|$, i.e.,

$$\hat{s}_n=\frac{\frac{1}{K}\sum_{k=1}^{K} r_n^2[k]}{\left[\frac{1}{K}\sum_{k=1}^{K}|r_n[k]|\right]^2} \quad (6)$$

where K is the number of symbols in a frame. This particular online statistic is of interest because it is directly related to the SNR, yet does not require knowledge of the data (which cannot be achieved until after the first pass of decoding). As K approaches infinity, the sample means become expected values [2]

$$s_n=\frac{E\left[r_n^2\right]}{\{E\left[|r_n|\right]\}^2}=\frac{1+2\beta_n}{\left[\sqrt{\frac{2}{\pi}}e^{-\beta_n}+\sqrt{2\beta_n}\mathrm{erf}(\sqrt{\beta_n})\right]^2}=f(\beta_n) \quad (7)$$

where $\beta_n=\beta(((n-1)/N)T-(T/2)+\tau,\beta_0)$, $1\leq n\leq N$. Since β_n is a function of β_0 and τ, s_n is also related to β_0 and τ.

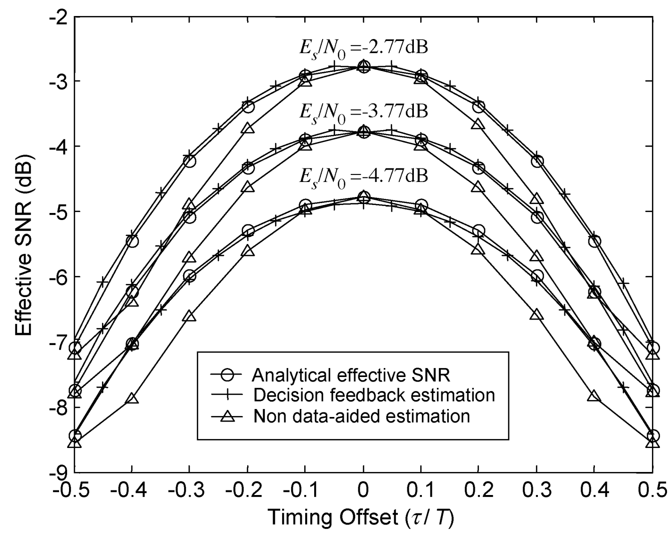

Fig. 2. Effective SNR in the presence of improper timing, both using the NDA estimation and DF methods (100 trials with a frame size 4590). RC-rolloff pulse shaping is used with rolloff factor $\alpha=0.5$.

Fig. 2 shows the results of a simulation that compares the estimated and analytical effective SNRs. Analytical effective SNRs are calculated from (5). Nondata-aided (NDA) estimates of effective SNRs are obtained by calculating $\{s_n\}$ and inverting (7). In Fig. 2, the NDA-estimated effective SNR fits the analytical results well for $|\tau|\leq 0.1T$. The NDA estimation produces pessimistic results of effective SNRs when $0.1T<|\tau|<0.5T$, because, as discussed shortly, the online statistic generates a biased estimator of the effective SNR.

The received sequence $r_n[k], 1\leq k\leq K$ is a discrete random process, therefore online statistics calculated by (6) are random variables. Before the synchronizer can be designed and evaluated, the online statistics must first be characterized. Detailed derivations are shown in the Appendix. The expected value of $\hat{s}_n$ is

$$\mathbf{E}[\hat{s}_n]=(1+2\beta_n)\left[\frac{1}{\sqrt{\frac{2}{\pi}}\exp(-\beta_n)+\sqrt{2\beta_n}\mathrm{erf}(\sqrt{\beta_n})}\right]^2 +\frac{\left[g\left(\frac{n-1}{N}T-\frac{T}{2}+\tau\right)\mathcal{E}_s\right]^2}{K}\cdot\left(1+\frac{1}{2\beta_n}\right) \times\left(1+\frac{1}{2\beta_n}-\left[\sqrt{\frac{2}{\pi}}\exp(-\beta_n)+\sqrt{2\beta_n}\mathrm{erf}(\sqrt{\beta_n})\right]^2\right). \quad (8)$$

The first term in (8) is exactly the same as the right-hand side of (7). However, the second term in (8) is greater than zero, thus, the online statistics are inevitably biased. Since the online statistic s_n is a decreasing function of β_n, the estimated effective SNR is always less than the real value. If either K or β_n is large, then the second term is negligible and $\hat{s}_n$ is approximately an unbiased estimator. The variance of the online statistic is

$$\mathbf{V}[\hat{s}_n]=\left[\frac{2\sigma^4}{K}+\frac{4\sigma^2}{K}+(\sigma^2+\mathcal{E}_s)^2\right](4\sigma_v^2+2\sigma_v^4)+\left(\frac{2\sigma^4}{K}+\frac{4\sigma^2}{K}\right) \times\left\{\left[\frac{1}{\sigma\sqrt{\frac{2}{\pi}}\exp\left(-\frac{\mathcal{E}_s}{2\sigma^2}\right)+\sqrt{\mathcal{E}_s}\mathrm{erf}\left(\sqrt{\frac{\mathcal{E}_s}{2\sigma^2}}\right)}\right]^2+\sigma_v^2\right\} \tag{9}$$

where σ_v^2 is the variance of $(1/K)\sum_{k=1}^{K}|r_n[k]|$. $\mathbf{V}[\hat{s}_n]$ decreases when K grows. Intuitively, this is reasonable because an estimate becomes more reliable with a larger observation window.

B. MMSE Algorithm

The estimation problem is to find the timing offset τ and SNR β_0 of the channel, based on multiple sequences of samples obtained from the matched filter. Online statistics are calculated using these sample sequences. The relationship between $\{s_n\}$ and the pair of parameters τ and β_0 is

$$s_n = s\left(-\frac{T}{2}+\frac{n-1}{N}T+\tau,\beta_0\right) \tag{10}$$

where $s(\tau,\beta_0) = f[\beta(\tau,\beta_0)]$. Intuitively, the optimal MMSE solution to this problem would result in a curve-fitting algorithm. The idea is to find the effective SNR curve with a certain pair of τ and β_0 that best fits the existing series of $\{\hat{s}_n\}$. An example is shown in Fig. 3 for $N = 4$ samples per symbol, when $\tau = 0.13T$ and $\beta_0 = -3.77$ dB. The error is defined as the Euclidean distance of the candidate curve to the values in $\{\hat{s}_n\}$. The goal is to pick $\hat{\tau}$ and $\hat{\beta}_0$ that minimizes the following error function:

$$e(\hat{\tau},\hat{\beta}_0) = \sum_{n=1}^{N}\left\{s\left(-\frac{T}{2}+\frac{n-1}{N}T+\hat{\tau},\hat{\beta}_0\right)-\hat{s}_n\right\}^2. \tag{11}$$

The corresponding optimization equations are

$$\frac{\partial e(\hat{\tau},\hat{\beta}_0)}{\partial\hat{\beta}_0}=0$$
$$\frac{\partial e(\hat{\tau},\hat{\beta}_0)}{\partial\hat{\tau}}=0. \tag{12}$$

Direct solution of (12) is difficult. Alternatives are to solve the problem numerically, or to make assumptions that simplify it. The numerical approach requires a large library of entries to be stored in lookup tables, and is sensitive to noise. Hence, we seek an approach to simplify the problem by applying a linearized approximation to the relationship of effective SNR and the pair of parameters (τ,β_0), as shown in Fig. 3. Since the online statistics $\{\hat{s}_n\}$ are directly available without needing to compute each $\hat{\beta}_n$, and uniquely map to the effective SNR as in (10), the algorithm can be directly applied to $\hat{s}_n$.

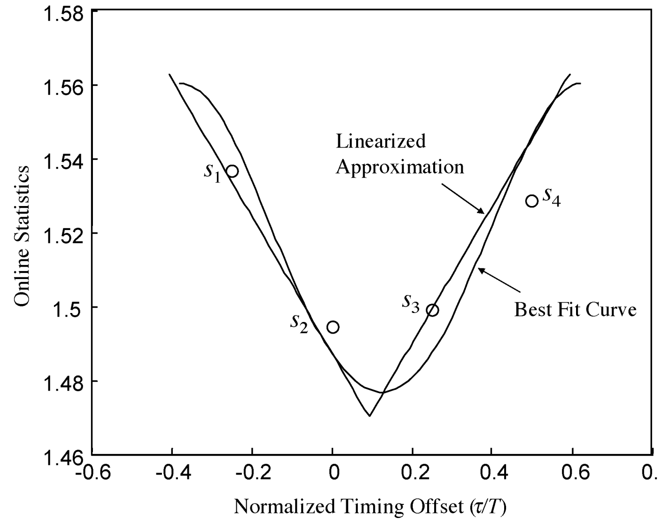

Fig. 3. Example series of online statistics $\{\hat{s}_n\}$, the corresponding best-fit curve, and a linearized approximation.

C. Reduced-Complexity Estimation of β_0 and τ

Because the exact MMSE solution is complex and sensitive to noise, we have developed the following reduced-complexity method for estimating β_0 and τ. An initial estimate of β_0 is found by selecting the minimum $\{\hat{s}_n\}$ and then inverting (7)

$$\hat{\beta}_0 = f^{-1}\left(\min_{1\le n\le N}\hat{s}_n\right). \tag{13}$$

For the curve-fitting method, this estimate is used to select the initial candidate curve. In the linearized approximation, $\hat{\beta}_0$ is used to determine the slope, i.e., the linearized approximation is

$$s(\hat{\tau},\hat{\beta}_0) = C(\hat{\beta}_0)|\hat{\tau}| + \hat{s}_0 \tag{14}$$

where $C(\hat{\beta}_0)$ is the slope and $\hat{s}_0 = f(\hat{\beta}_0)$. Values of $C(\hat{\beta}_0)$ are found by evaluating the relationship between online statistics and τ. For a specific β_0

$$C(\beta_0) = \frac{s(0.5T,\beta_0)-s_0}{0.5T}. \tag{15}$$

Simulations are used to obtain empirical values of $C(\beta_0)$. More specifically, for each value of β_0 ranging from -4.8 dB to -1.8 dB in 0.1 dB increments, 1000 realizations of the online statistic $s(0.5T,\beta_0)$ are simulated and the average is used to determine the slope $C(\beta_0)$, which is stored in a lookup table.

The optimization function for $\hat{\tau}$ is

$$\frac{\partial e}{\partial\hat{\tau}} = \sum_{n=1}^{N}2C(\hat{\beta}_0)\mathrm{sign}\left(n-1-\frac{N}{2}+\hat{\tau}\right) \times\left[C(\hat{\beta}_0)\left|-\frac{T}{2}+\frac{n-1}{N}T+\hat{\tau}\right|+\hat{s}_0-\hat{s}_n\right]. \tag{16}$$

$\hat{\tau}$ is found by setting (16) equal to zero. The slope $C(\hat{\beta}_0)\neq 0$ is, hence

$$\hat{\tau} = \frac{T}{2N}-\frac{1}{C(\hat{\beta}_0)N}\sum_{n=1}^{N}\left[\mathrm{sign}\left(-\frac{T}{2}+\frac{n-1}{N}T\right)(\hat{s}_0-s_n)\right]. \tag{17}$$

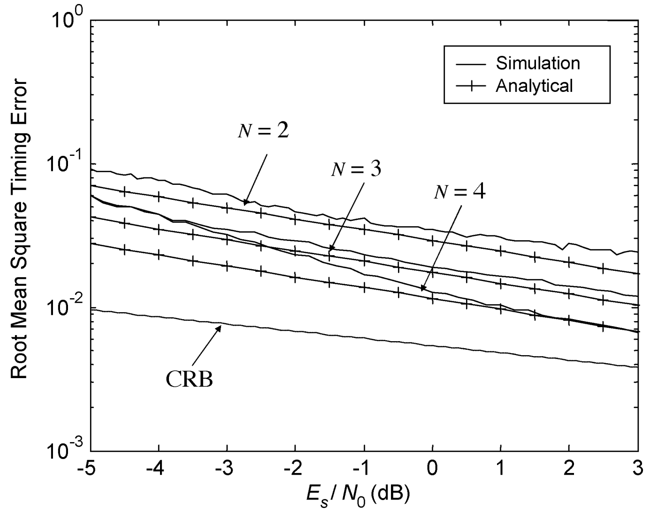

Fig. 4. RMS timing error when $K = 4590$, with $N = 2, 3, 4$, uncoded data sequence with BPSK modulation and rolloff factor $\alpha = 0.5$.

Under the linearized approximation, the true value of τ should satisfy

$$\tau = \frac{T}{2N} - \frac{1}{C(\beta_0)N}\sum_{n=1}^{N}\left[\mathrm{sign}\left(-\frac{T}{2}+\frac{n-1}{N}T\right)(\hat{s}_0-\bar{s}_n)\right] \tag{18}$$

where $\bar{s}_n = \mathbf{E}[\hat{s}_n]$. The mean square error (MSE) of the timing estimate is found to be

$$\begin{aligned}\mathbf{E}\left[(\tau-\hat{\tau})^2\right] = &\frac{1}{\left[C(\hat{\beta}_0)N\right]^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\rho_{ij}\\ &\times \mathrm{sign}\left[\left(i-1-\frac{N}{2}\right)\left(j-1-\frac{N}{2}\right)\right]\sqrt{\mathrm{V}[\hat{s}_i]\mathrm{V}[\hat{s}_j]}\end{aligned} \tag{19}$$

where ρ_{ij} is the correlation coefficient of s_i and s_j, $1 \leq i, j \leq N$. This coefficient is symmetric, $\rho_{ij} = \rho_{ji}$, and is periodic due to the fact that the online statistics count all the samples in a frame. When $\alpha = 0.5$, $\rho_{i,i+1} = 0.1657$, 0.3229, and 0.4792 when $N = 2$, 3, and 4, respectively.

The root-mean square (RMS) timing error with two, three, and four samples per symbol using the proposed joint estimation algorithm is shown in Fig. 4. The analytical curves are found by plotting (19). The simulated and analytical results agree, especially as the SNR and number of samples per symbol gets large.

D. Interpolation

Once the estimate of τ is available, the interpolator combines information from candidate samples to construct what the samples would have been had the waveform been sampled at one sample per symbol at exactly the proper instant. This is equivalent to reconstructing the waveform and sampling it with proper timing. These samples are input to the turbo decoder. The simplest interpolator is a linear interpolator, which generates a weighted summation of selected samples. In the proposed strategy, just the two samples that are closest to the estimated timing offset are selected. The other samples, if available, are not used because they are subject to higher ISI, and consequently, are not reliable. Assuming that the two samples come from the nth and $(n+1)$th samplers, then the sampling times associated with these samples are $\hat{\tau}_n = -(T/2) + ((n-1)/N)T + \hat{\tau}$ and $\hat{\tau}_{n+1} = \hat{\tau}_n + T/N$, respectively. The result of the linear interpolation is the following statistic for the kth symbol:

$$r[k] = w_n r_n[k] + w_{n+1} r_{n+1}[k] \tag{20}$$

where the weights $w_n = (|\hat{\tau}_{n+1}|/(|\hat{\tau}_n| + |\hat{\tau}_{n+1}|))$ and $w_{n+1} = (|\hat{\tau}_n|/(|\hat{\tau}_{n+1}| + |\hat{\tau}_{n+1}|))$.

This simple interpolation method results in a signal energy loss. However, this loss is negligible when $N > 2$. We have also tested higher order polynomial interpolators [15]. The performance of more complex interpolaters is usually worse than with using simple linear interpolation, because these interpolaters are too sensitive to noise, so they are not suitable for the low-SNR environment where turbo codes are implemented.

E. Final Estimation of β_0

As shown in Fig. 3, the linearized approximation reaches its minimum value at

$$\hat{s}_0 = \frac{1}{N}\sum_{n=1}^{N}\left[\hat{s}_n + C(\hat{\beta}_0)\left|-\frac{T}{2}+\frac{n-1}{N}T+\hat{\tau}\right|\right]. \tag{21}$$

When $\hat{\tau}$ is available, a final estimate of β_0 is found

$$\hat{\beta}_0' = f^{-1}\left(\frac{1}{N}\sum_{n=1}^{N}\left[\hat{s}_n + C(\hat{\beta}_0)\left|-\frac{T}{2}+\frac{n-1}{N}T+\hat{\tau}\right|\right]\right). \tag{22}$$

This estimate is used in the turbo decoder to generate the channel reliability information. Turbo codes are not sensitive to moderate errors in the estimation of the SNR [2]. Our simulation also shows that the performance loss due to estimating β_0 is negligible.

IV. Iterative Synchronization

A. Data-Aided (DA) SNR Estimation

When the data is known (or can be accurately estimated), the variance of the additive noise can be estimated as proposed by Reed and Asenstorfer in [16]. The additive noise includes the effects of both the channel noise and the ISI. Let σ_n^2 be the additive noise on the nth sample sequence. Assuming that the transmitted data is available at the receiver, the variance of the additive noise is a function of (τ, β_0), and can be expressed as

$$\begin{aligned}v(\tau,\beta_0) &= \mathbf{E}\left[\left\{r(kT+\tau)-\sqrt{\mathcal{E}_s}a_k\right\}^2\right]\\ &= \mathcal{E}_s\left\{[g(\tau)-1]^2 + M(\tau) + \frac{1}{2\beta_0}\right\}.\end{aligned} \tag{23}$$

$v(\tau, \beta_0)$ is an even function of τ and $v(0, \beta_0) = \sigma^2$. The effective SNR is

$$\beta(\tau,\beta_0) = \frac{g(\tau)\mathcal{E}_s}{2v(\tau,\beta_0)}. \tag{24}$$

The ideal AGC block in the system model normalizes $\mathcal{E}_s$. Therefore, $\mathcal{E}_s = 1$ in the following. The validity of (23) is verified by simulation, as shown in Fig. 2. This simulation uses the estimated variance $\hat{\sigma}_n^2$ derived below and the actual data for DF. When implemented in a turbo decoder, the feedback simply uses a hard decision of the log-likelihood ratio (LLR) $\hat{a}_k = \text{sign}(y[k])$, where $y[k]$ is the LLR of the kth information bit. We also considered soft-DF by using $\hat{a}_k = \tanh(y[k]/2)$, but found essentially no performance difference when compared with hard-DF.

For limited-length sequences, an estimate of σ_n^2 can be found by replacing the expected value with the sample mean

$$\hat{\sigma}_n^2 = \frac{1}{K}\sum_{k=1}^{K}\left[(r_n[k] - a_k)^2\right]. \tag{25}$$

The point estimator defined in (25) is an unbiased estimator for $v(((n-1)/N)T + (T/2) + \tau, \beta_0)$. Its variance is

$$\mathbf{V}\left[\hat{\sigma}_n^2\right] = \frac{2}{K}\left[v\left(\frac{n-1}{N}T + \frac{T}{2} + \tau, \beta_0\right)\right]^2. \tag{26}$$

Similar to NDA estimation, we establish the following error function:

$$e_{\text{DA}}(\tau, \beta_0) = \sum_{n=1}^{N}\left\{v\left(-\frac{T}{2} + \frac{n-1}{N}T + \hat{\tau}, \hat{\beta}_0\right) - \hat{\sigma}_n^2\right\}^2 \tag{27}$$

which we wish to minimize with respect to $\hat{\tau}$ and $\hat{\beta}_0$.

B. *Estimation of β_0 and τ*

Since $v(\tau, \beta_0) \geq \sigma^2$, we select the estimate of σ^2 as $\hat{\sigma}^2 = \min\limits_{1 \leq n \leq N} \hat{\sigma}_n^2$. Thus, the corresponding estimate of the channel SNR is

$$\hat{\beta}_{0,\text{DA}} = \frac{1}{2\hat{\sigma}^2}. \tag{28}$$

The linearized approximation of (23) is

$$v(\hat{\tau}, \hat{\beta}_0) = C_{\text{DA}}(\hat{\beta}_0)|\hat{\tau}| + \hat{\sigma}^2. \tag{29}$$

As with the feedforward case, empirical values for $C_{\text{DA}}(\beta_0)$ are found by evaluating (29) through simulation and then stored in a lookup table.

Similar to the process of (16) and (17), we find the following DA estimate of τ:

$$\hat{\tau}_{\text{DA}} = \frac{T}{2N} - \frac{1}{C_{\text{DA}}(\hat{\beta}_{0,\text{DA}})N} \times \sum_{n=1}^{N}\left[\text{sign}\left(-\frac{T}{2} + \frac{n-1}{N}T\right)\left(\hat{\sigma}^2 - \hat{\sigma}_n^2\right)\right]. \tag{30}$$

The MSE of the timing estimate is

$$\mathbf{E}\left[(\tau - \hat{\tau}_{\text{DA}})^2\right] = \frac{1}{\left[C_{\text{DA}}(\hat{\beta}_0)N\right]^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\rho'_{ij} \times \text{sign}\left[\left(i - 1 - \frac{N}{2}\right)\left(j - 1 - \frac{N}{2}\right)\right]\sqrt{\text{V}\left[\hat{\sigma}_i^2\right]\text{V}\left[\hat{\sigma}_j^2\right]}. \tag{31}$$

When $\alpha = 0.5$, the correlation values $\rho'_{i,i+1} = 0.537$, 0.779, and 0.874 when $N = 2$, 3, and 4, respectively. These values were calculated using the sample mean and sample variances of the simulation results. The noise is correlated because the matched filter colors the additive noise. Therefore, $w(t)$ has an autocorrelation function equal to $g(t)$. The RMS timing-estimation error is shown in Fig. 5. The analytical curves are found by plotting (31). It is found that with knowledge of the transmitted data, the estimation error is close to the CRB, but the performance is sensitive to the accuracy of linearized approximation. When $N = 2$, or when the channel SNR is high, the linearized approximation turns inaccurate.

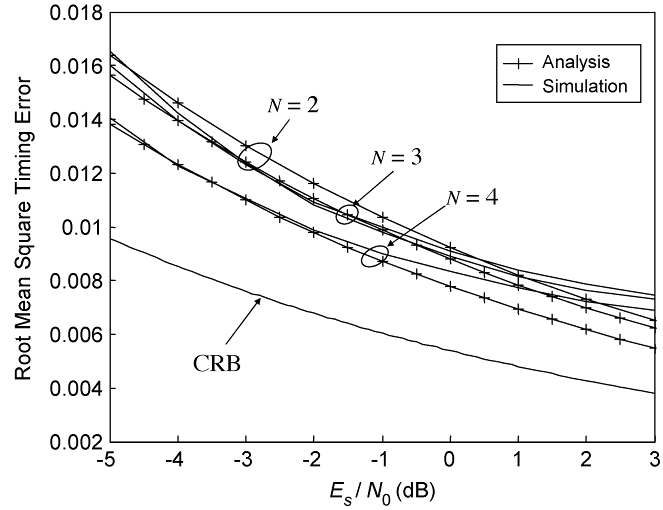

Fig. 5. Root MSE of the DF timing estimate for uncoded BPSK modulation with rolloff factor $\alpha = 0.5$, frame size $K = 4590$, and $N = \{2, 3, 4\}$ samples per symbol.

V. Simulation Study

A simulation campaign was completed in order to illustrate the effectiveness of the proposed estimation techniques. In the simulations, BPSK modulation was used over an AWGN channel. The timing error was quasi-static in the sense that the offset was constant throughout the frame, but varied independently from frame to frame according to a uniform distribution. The timing offset was estimated using the linearized approximation, and a lookup table with 31 candidate slope values was used, with $\mathcal{E}_s/N_0$ ranging from -4.8 dB to -1.8 dB in 0.1 dB increments. This range corresponds to $\mathcal{E}_b/N_0$ between 0–3 dB when the coding rate is 1/3. This range was selected because this is where the bit-error rate (BER) performance of typical turbo codes changes most rapidly, i.e., the range contains the so-called "waterfall." Turbo decoding is performed using the log-MAP algorithm [1].

We define two types of iteration, *local* and *global*. A local iteration is merely an iteration within the turbo decoder. On the other hand, a global iteration is an iteration between the turbo decoder and the estimator. The number of times the SNR and timing offset are estimated is equal to the number of global iterations. In all simulations, the total number of local iterations is set to 10. Systems that do not use DF only execute one global iteration, while systems with DF execute two global iterations (with five local iterations per global iteration). The balance between the number of global and local iterations is important. On the one hand, decision-directed estimation works better when

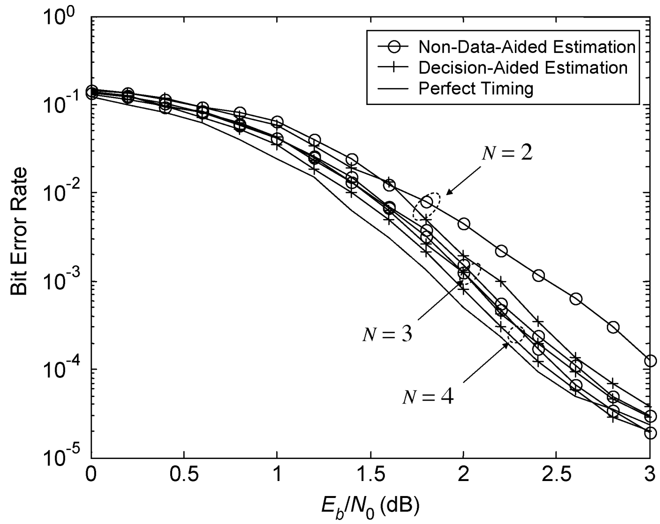

Fig. 6. Performance of turbo code 1 using the proposed joint timing/SNR algorithms and $N = \{2, 3, 4\}$ samples per symbol. The interleaver size is 256, overall code rate is 1/2, RSC generators (37, 33), and decoding uses 10 total iterations of log-MAP algorithm.

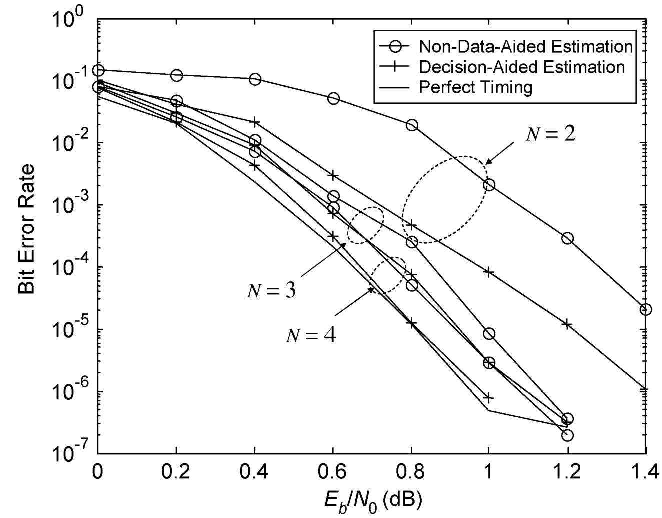

Fig. 7. Performance of turbo code 2 using the proposed joint timing/SNR algorithms and $N = \{2, 3, 4\}$ samples per symbol. Code is as specified in the cdma2000 standard with overall code rate 1/3 and interleaver size 1530. Decoding uses 10 total iterations of log-MAP algorithm.

the intermediate decision is accurate, and thus, it is important to not feed back information prematurely. On the other hand, it is important not to wait too long before feeding back information to the estimator, or else the benefits of the DF estimator will not be realized.

As in [10], the first turbo code tested uses a 256-bit random interleaver. The constituent recursive systematic convolutional (RSC) code uses octal generators (37, 33), and the parity bits are alternatively punctured to obtain a coding rate of 1/2. The two constituent codes are terminated independently. The BER performance is shown in Fig. 6 for $N = 2$, 3, and 4 samples per symbol, and both the NDA and the decision-directed estimation techniques. The performance with perfect timing and SNR estimates is also shown for comparison purposes. With only two samples per symbol and NDA estimation, the BER performance is within 0.8 dB of perfect timing at a BER of 10^{-4}. If a second global iteration is used (DF estimation), then this loss is reduced to only about 0.4 dB. With four samples per symbol and NDA estimation, performance is within 0.2 dB of perfect timing at a BER of 10^{-4}, and with DF estimation, the loss is less than 0.1 dB.

The second code that we consider is one of the turbo codes defined in the cdma2000 standard [17]. In particular, an interleaver size of 1530 and overall code rate of 1/3 is selected, which will make this code significantly stronger than the first code. This code uses RSC constituent codes with octal generator (15,13) and interleaver as defined in the standard. The BER performance is shown in Fig. 7, again for $N = 2$, 3, and 4 samples per symbol and both estimation techniques. The BER losses in this case are very similar to the losses observed with the first code, indicating that the estimation technique is robust enough to work with stronger codes, and at the corresponding low SNRs. In particular, with $N = 2$ and at a BER of 10^{-4}, the loss relative to perfect timing with the NDA estimator is 0.8 dB, and with the decision-directed estimator is about 0.4 dB. With $N = 4$, these two losses are 0.2 and 0.1 dB, respectively. In the above systems, it is shown that the BER performance improves as N increases if

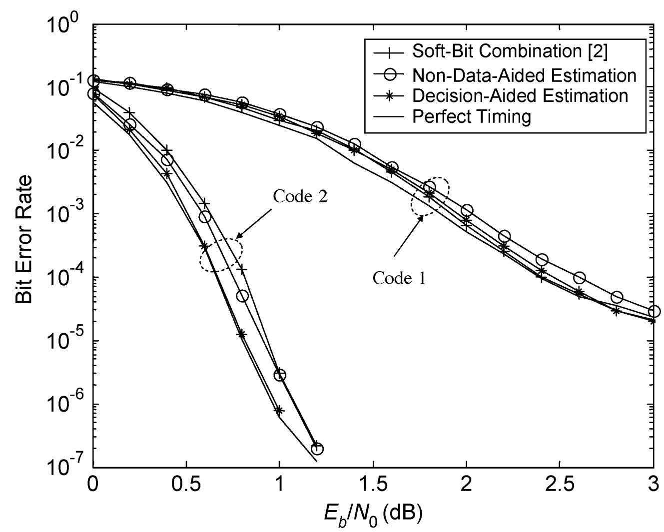

Fig. 8. Comparison of Mielczarek's soft-combining method and joint estimation of SNR and timing-offset method with four samples per symbol. Code 1: interleaver size $= 256$, coding rate $= 1/2$, constituent RSC code is (37, 33). Code 2: interleaver size $= 1530$, coding rate $= 1/3$, constituent RSC code is (15, 13). Random interleaver is used, and 10 iterations.

perfect timing is not available. Data-directed estimation helps to close the gap, thereby approaching the BER performance with perfect timing.

In Fig. 8, we compare the performance of both codes using our proposed estimation techniques with the previously proposed soft-combining method of [10] with four samples per symbol. While soft-bit combining works better with the weaker code, the proposed joint estimation techniques outperform it when using the stronger code. This is due to the fact that the estimation error of the proposed frame-based estimator is inversely proportional to the window (frame) size. Recall, however, that the method in [10] requires two turbo decoders running in parallel, while our method only requires a single turbo decoder. The overall system complexity of the proposed approach is, therefore, significantly less than the one proposed in [10].

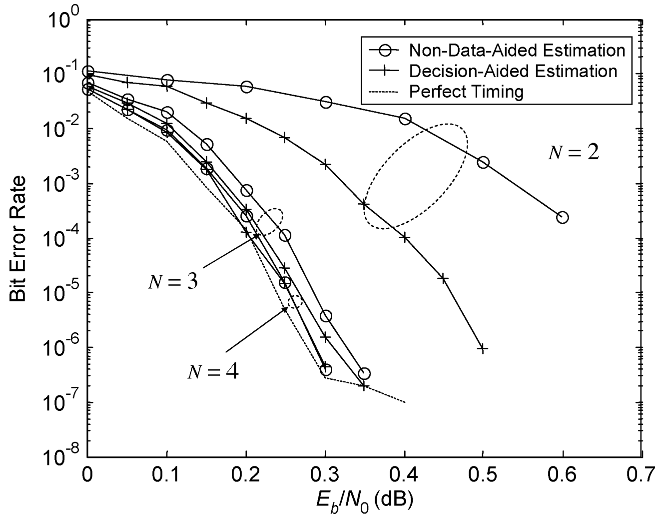

Fig. 9. Performance of turbo code 3 using the proposed joint timing/SNR algorithms and $N = \{2, 3, 4\}$ samples per symbol. Code is as specified in the cdma2000 standard with overall code rate 1/3 and interleaver size 20730. Decoding uses 10 total iterations of log-MAP algorithm.

We also considered a third code, shown in Fig. 9, which is much stronger than the first two. This code is also defined in cdma2000 standard [17]. The interleaver size is 20730. The code can get a BER of 10^{-5} at a SNR of 0.23 dB with perfect timing. With only two samples per symbol and NDA estimation, the coding gain loss is within 0.8 dB of the BER performance of perfect timing at a BER of 10^{-4}. When $N = 2$ and two global iterations are invoked, the coding gain loss is about 0.3 dB at a BER of 10^{-5}. With four samples per symbol and NDA estimation, the coding gain loss is within 0.05 dB of the BER of perfect timing at a BER of 10^{-5}. When two global iterations are called, the coding gain does not improve much because of the low SNR.

In all of the above simulations, it is found that the performance of $N = 3$ with two global iterations is comparable to that of $N = 4$ with only one global iteration. The performance gain is achieved at the cost of one call of decision-aided joint estimation and reinterpolation. This means we can design a system with a lower sampling rate and additional system complexity to achieve a similar performance to higher sampling rate and lower complexity.

VI. Conclusions

Imperfect timing causes a loss in effective SNR, which results in a severe BER performance degradation for timing shifts greater than about 10% of the symbol period. This performance loss can be recovered by a proper estimation algorithm. However, the situation is complicated by the fact that the channel SNR over which turbo codes operate is both very small and not known to the receiver. Also, practical systems are likely to use Nyquist pulse shaping, which introduces ISI in the presence of timing errors. Our approach to synchronization and SNR estimation involves sampling the signal multiple times per symbol period, and computing an online statistic for each of the sample instances over the entire frame. These online statistics are then used to simultaneously estimate the channel SNR and timing offset. A simple linear interpolation algorithm is then used to reconstruct the matched-filter samples at the estimated timing instants. Tentative decisions from the decoder can be fed back and used to refine the timing and SNR estimates.

The proposed algorithm recovers much of the loss due to poor synchronization, and does so with negligible added complexity and latency (compared with that of the turbo decoding algorithm itself). With feedback from the decoder to the estimator, the simulated coding gain loss is negligible, i.e., about 0.1 dB with four samples per symbol when the interleaver size $= 1530$ at a BER of 10^{-5}. The DF technique provides a method to improve coding gain by implementing iterative timing/SNR estimation with a complexity much smaller than the turbo decoder. When the sampling rate is critical, this technique recovers the coding gain comparable to a system with a higher sampling rate, but does not use DF. Finally, it is noted that the proposed technique is suitable for more than just turbo codes. Indeed, any system can use the NDA technique. Furthermore, since only hard decisions were fed back from the decoder to the estimator, the proposed iterative synchronization strategy is suitable for any error-control code, not just those that use soft-output decoders.

Appendix

The online statistics are computed with (6). At high SNR, $|r[k]|$ is approximately Gaussian. For Gaussian random variables, first-order statistics are independent from second-order statistics [18], and so we may assume that the numerator and denominator in (6) are independent. Then, we have the expected value

$$\mathbf{E}[s] = \mathbf{E}\left[\frac{1}{K}\sum_{k=1}^{K} r[k]^2\right]\mathbf{E}\left[\frac{1}{\left[\frac{1}{K}\sum_{k=1}^{K}|r[k]|\right]^2}\right]. \quad \text{(A.1)}$$

$(1/K)\sum_{k=1}^{K} r^2[k]$ is a noncentral chi-square distribution with K degrees of freedom. The first- and second-order statistics of $(1/K)\sum_{k=1}^{K} r[k]^2$ are

$$\mathbf{E}\left[\frac{1}{K}\sum_{k=1}^{K} r^2[k]\right] = \sigma^2 + \mathcal{E}_s \quad \text{(A.2)}$$

$$\mathbf{V}\left[\frac{1}{K}\sum_{k=1}^{K} r^2[k]\right] = \frac{2}{K}\sigma^4 + \frac{4}{K}\sigma^2. \quad \text{(A.3)}$$

From the central limit theorem, $(1/K)\sum_{k=1}^{K}|r[k]|$ has (approximately) a normal distribution, where

$$\begin{aligned} m_v &= \mathbf{E}\left[\frac{1}{K}\sum_{k=1}^{K}|r[k]|\right] \\ &= \sigma\sqrt{\frac{2}{\pi}}\exp\left(-\frac{\mathcal{E}_s}{2\sigma^2}\right) + \sqrt{\mathcal{E}_s}\operatorname{erf}\left(\sqrt{\frac{\mathcal{E}_s}{2\sigma^2}}\right) \quad \text{(A.4)} \\ \sigma_v^2 &= \mathbf{V}\left[\frac{1}{K}\sum_{k=1}^{K}|r[k]|\right] \\ &= \frac{\mathcal{E}_s + \sigma^2}{K} - \frac{1}{K}\left[\sigma\sqrt{\frac{2}{\pi}}\exp\left(-\frac{\mathcal{E}_s}{2\sigma^2}\right)\right. \\ &\qquad \left. + \sqrt{\mathcal{E}_s}\operatorname{erf}\left(\sqrt{\frac{\mathcal{E}_s}{2\sigma^2}}\right)\right]^2. \quad \text{(A.5)} \end{aligned}$$

$Y = ((1/K)\sum_{k=1}^{K}|r[k]|)^2$ has a noncentral chi-square distribution. If K is sufficiently large, then $\sigma_v^2 \ll 1$, and consequently, $\mathbf{E}[1/Y]$ and $\mathbf{V}[1/Y]$ are approximately $((1/m_v^2)+\sigma_v^2)$ and $(2\sigma_v^4 + 4\sigma_v^2)$, respectively. Hence, we have the expected value of online statistic

$$\begin{aligned}\mathbf{E}[s] &= (\mathcal{E}_s+\sigma^2)\\ &\quad\times\left(\left[\frac{1}{\sigma\sqrt{\frac{2}{\pi}}\exp\left(-\frac{\mathcal{E}_s}{2\sigma^2}\right)+\sqrt{\mathcal{E}_s}\mathrm{erf}\left(\sqrt{\frac{\mathcal{E}_s}{2\sigma^2}}\right)}\right]^2\right)\\ &\quad+(\mathcal{E}_s+\sigma^2)\left(\frac{\mathcal{E}_s+\sigma^2}{K}-\frac{1}{K}\left[\sigma\sqrt{\frac{2}{\pi}}\exp\left(-\frac{\mathcal{E}_s}{2\sigma^2}\right)\right.\right.\\ &\qquad\left.\left.+\sqrt{\mathcal{E}_s}\mathrm{erf}\left(\sqrt{\frac{\mathcal{E}_s}{2\sigma^2}}\right)\right]^2\right)\\ &=(1+2\beta_0)\left[\frac{1}{\sqrt{\frac{2}{\pi}}\exp(-\beta_0)+\sqrt{2\beta_0}\mathrm{erf}(\sqrt{\beta_0})}\right]^2\\ &\quad+\frac{1}{K}\left(1+\frac{1}{2\beta_0}\right)\left(1+\frac{1}{2\beta_0}-\left[\sqrt{\frac{2}{\pi}}\exp(-\beta_0)\right.\right.\\ &\qquad\left.\left.+\sqrt{2\beta_0}\mathrm{erf}(\sqrt{\beta_0})\right]^2\right)\end{aligned}\tag{A.6}$$

where $\beta_0 = \mathcal{E}_s/N_0$ is the SNR. The term in the last line of (A.6) represents a bias, which is unavoidable due to the division of two random variables.

The variance of the statistic is

$$\begin{aligned}\mathbf{V}[s] &= \left[\frac{2\sigma^4}{K}+\frac{4\sigma^2}{K}+(\sigma^2+\mathcal{E}_s)^2\right]\left(4\sigma_v^2+2\sigma_v^4\right)+\left(\frac{2\sigma^4}{K}+\frac{4\sigma^2}{K}\right)\\ &\quad\times\left\{\left[\frac{1}{\sigma\sqrt{\frac{2}{\pi}}\exp\left(-\frac{\mathcal{E}_s}{2\sigma^2}\right)+\sqrt{\mathcal{E}_s}\mathrm{erf}\left(\sqrt{\frac{\mathcal{E}_s}{2\sigma^2}}\right)}\right]^2+\sigma_v^2\right\}.\end{aligned}\tag{A.7}$$

References

[1] C. Berrou, A. Glavieux, and P. Thitimasjshima, "Near-Shannon-limit error-correcting coding and decoding: Turbo-codes," in *Proc. IEEE Int. Conf. Commun.*, Geneva, Switzerland, May 1993, pp. 1064–1070.

[2] T. A. Summers and S. G. Wilson, "SNR mismatch and online estimation in turbo decoding," *IEEE Trans. Commun.*, vol. 46, no. 4, pp. 421–423, Apr. 1998.

[3] J. Liu, H. Song, and B. V. K. V. Kumar, "Symbol timing recovery for low-SNR partial response recording channels," in *Proc. IEEE GLOBECOM*, 2002, pp. 1129–1136.

[4] A. R. Nayak, J. R. Barry, and S. W. McLaughlin, "Joint timing recovery and turbo equalization for coded partial response channels," *IEEE Trans. Magn.*, vol. 38, no. 9, pp. 2295–2297, Sep. 2002.

[5] K. H. Mueller and M. Müller, "Timing recovery in digital synchronous data receivers," *IEEE Trans. Commun.*, vol. COM-24, no. 5, pp. 516–531, May 1976.

[6] A. R. Nayak, J. R. Barry, and S. W. McLaughlin, "Lower bounds for the performance of iterative timing recovery at low SNR," in *Proc. 15th Int. Symp. Math. Theory Netw. Syst.*, South Bend, IN, Aug. 2002, pp. WM2–WM5.

[7] C. N. Georghiades and M. Moeneclaey, "Sequence estimation and synchronization from nonsynchronized samples," *IEEE Trans. Inf. Theory*, vol. 37, no. 11, pp. 1649–1657, Nov. 1991.

[8] Z.-N. Wu, J. M. Cioffi, and K. D. Fisher, "A MMSE interpolated timing recovery scheme for the magnetic recording channel," in *Proc. IEEE Int. Conf. Commun.*, vol. 3, Montreal, QC, Canada, Jun. 1997, pp. 1625–1629.

[9] L. Lu and S. G. Wilson, "Synchronization of turbo coded modulation systems at low SNR," in *Proc. IEEE Int. Conf. Commun.*, 1998, pp. 428–432.

[10] B. Mielczarek and A. Svensson, "Timing error recovery in turbo coded systems on AWGN channels," *IEEE Trans. Commun.*, vol. 50, no. 10, pp. 1584–1592, Oct. 2002.

[11] D. Raphaeli and Y. Zarai, "Combined turbo equalization and turbo decoding," in *Proc. IEEE GLOBECOM*, vol. 2, Nov. 1997, pp. 639–643.

[12] F. M. Gardner, "A BPSK/QPSK timing-error detector for sampled receivers," *IEEE Trans. Commun.*, vol. COM-34, no. 5, pp. 423–429, May 1986.

[13] J. G. Proakis, *Digital Communications*, 4th ed. New York: McGraw-Hill, 2000.

[14] L. E. Franks, "Further results on Nyquist's problem in pulse transmission," *IEEE Trans. Commun. Technol.*, vol. COM-16, no. 4, pp. 337–340, Apr. 1968.

[15] L. Erup, F. M. Gardner, and R. A. Harris, "Interpolation in digital modems—Part II: Implementation and performance," *IEEE Trans. Commun.*, vol. 41, no. 6, pp. 998–1008, Jun. 1993.

[16] M. Reed and J. Asenstorfer, "A novel variance estimator for turbo-code decoding," in *Proc. Int. Conf. Telecommun.*, Melbourne, Australia, Apr. 1997, pp. 173–178.

[17] Third Generation Partnership Project 2. (2002, May). Physical layer standard for cdma2000 spread spectrum systems, Rel. C. [Online]. Available: http://www.3gpp2.org, Rep. 3GPP2 C.S0002-C

[18] R. V. Hogg and A. T. Craig, *Introduction to Mathematical Statistics*, 5th ed. Englewood Cliffs, NJ: Prentice-Hall, 1995.

Jian Sun (S'03–M'05) received the B.S. and M.S. degrees in electrical engineering from Shanghai Jiao Tong University, Shanghai, China, in 1997 and 2000, respectively, and the Ph.D. degree in 2004 from West Virginia University, Morgantown.

Since November 2004, he has been a Research Associate with the University of Pittsburgh, Pittsburgh, PA, on wireless sensor networks. His area of interests includes error-correction coding, channel estimation, and synchronization.

Matthew C. Valenti (M'99) received the B.S.E.E. degree from Virginia Polytechnic Institute and State University (Virginia Tech), Blacksburg, in 1992, the M.S.E.E. degree from The Johns Hopkins University, Baltimore, MD, in 1995, and the Ph.D. degree in electrical engineering from Virginia Tech in 1999.

He is currently an Associate Professor with the Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown. His research interests are in the areas of communication theory, error-correction coding, applied information theory, and wireless multiple-access networks.

Dr. Valenti was a Bradley Fellow at Virginia Tech during his doctoral work. He serves as an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and has been on the technical program committee for several international conferences.