Linear Iterative Turbo-Equalization (LITE) for Dual Channels

Andrew Singer Jill Nelson Ralf Koetter
University of Illinois at Urbana-Champaign
Coordinated Science Laboratory
Urbana, IL 61801
{acsinger, jknelso1, koetter}@uiuc.edu

Abstract

We examine a point-to-point communications scenario in which two or more separate, but known, channels are available for data transmission. While sending the same data across multiple channels provides channel diversity, we introduce additional temporal diversity by permuting the order of the data prior to transmission over one or more of the channels. As a receiver, we introduce a low complexity iterative equalization algorithm, inspired by iterative decoders for turbo-codes, which we call linear iterative turbo-equalization (LITE). The LITE algorithm contains one minimum mean square error linear equalizer for each channel and passes soft-information between the different equalizers in the form of a prior over the transmitted data. The linear equalizers differ from conventional equalizers by incorporating this prior in the minimization. Through simulations, we compare the empirical performance of the LITE algorithm to that of conventional linear and decision feedback equalizers, as well as maximum likelihood decoding for the set of channels. Our simulations demonstrate that the LITE algorithm can achieve equalization performance comparable to maximum likelihood decoding with computational complexity comparable to that of linear equalization.

1. Introduction

An intersymbol-interference (ISI) channel is a common model for point-to-point communications, where the intersymbol interference is caused by multipath propagation induced by the dominant scatterers in the medium. While this channel is often time-varying for mobile communications, in many applications of interest the channel can be assumed known at the receiver, either through the use of pilot signals or suitable training data. As such, linear and decision feedback equalizers are often used to correct ISI and can provide adequate performance for reliable communication over such channels. In the context of a wireless communications infrastructure, such as that for cellular telephony or other personal communications systems, an interesting situation arises when a single user is given the opportunity to make simultaneous use of multiple channels. If the data rate of the user remains constant, then the presence of an additional channel provides a means for increasing the fidelity of the data which was previously sent over a single channel. While this problem falls within the general context of channel coding, in this paper, we explore a relatively simple transmission and equalization scheme to exploit the availability of additional channels. This method has low overhead, such that the overall computational complexity is comparable to that of linear equalization, yet performs nearly as well as maximum likelihood decoding.

2 Channel Model

The channel model considered includes two or more ISI channels that are known to the receiver. At the transmitter, the same data $b[n]$ is to be transmitted over each of the channels, thus creating a trivial rate $1/M$ code, where M is the number of channels available. For simplicity, in the remainder of this paper, we assume that $M = 2$, and note that extension to larger M is straightforward. In the two-channel case, prior to transmission, the data over one of the channels is segmented into blocks, and the data within each block is then interleaved to introduce additional temporal diversity. This transmission strategy was also used by Emmer and Franz in [4]. However, the decoding strategy described in [4] is based on the "turbo-equalization" method of Douillard et al. [3] and hence requires maximum a posteriori (MAP) decoding [2] of each of the channels, multiple times in succession. While the performance of this technique can approach the optimal in terms of probability of error, it comes at a price of computational complexity that is exponential in the length of the impulse response of the channel. In this paper, we consider a reduced complexity decoder for this transmission scheme, whose performance

0-7803-5700-0/99/$10.00©1999 IEEE

nearly attains that of "turbo-equalization," yet whose complexity is only polynomial in the channel impulse response.

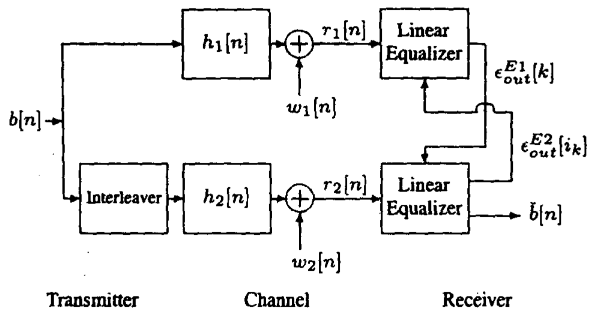

Figure 1. Equivalent baseband discrete-time channel model and LITE block diagram. The data $b[n]$ is segmented into blocks and then transmitted over each of two channels. Over one of the channels, the order of the data in the blocks has been interleaved. The two ISI channels $h_1[n]$ and $h_2[n]$ are assumed known to the receiver, and $w_1[n]$ and $w_2[n]$ are assumed to be independent AWGN sequences.

Specifically, our channel model is as shown in Figure 1. For simplicity, we assume that the data is binary phase-shift-key (BPSK)-encoded and that the overall effect of each of the transmitter-channel-receiver paths can be represented as a linear time-invariant discrete-time channel, $h_1[n]$ and $h_2[n]$. In Fig. 1, the data $b[n]$ is interleaved prior to transmission over $h_2[n]$. We also assume that additive white Gaussian noise $w_i[n]$ is present in each channel and that the noise in each of the channels is uncorrelated. The discrete-time baseband model of the two channel outputs for the transmitted data sequence $b[n]$ is then

$$r_1[n] = \sum_{k=-L_1}^{L_2} h_1[k]b[n-k] + w_1[n] \tag{1}$$

$$r_2[n] = \sum_{k=-E_1}^{E_2} h_2[k]\tilde{b}[n-k] + w_2[n], \tag{2}$$

where $i_n = \Pi(1, ..., N)$ is the data permutation index, $b[n]$ is the input symbol stream, and $\tilde{b}[n] = b[i_n]$ is the permuted symbol stream. It is well-known how to optimally estimate (in terms of minimizing the probability of error) the sequence of transmitted bits $b[n]$ based on only one of the received sequences $r_1[n]$ or $r_2[n]$. This is done using the so-called forward/backward algorithm [2]. However, the computational complexity of this approach is usually impractical for channels with significant ISI ($L_1 >> 1$ or $L_2 >> 1$), since the complexity of such an optimal equalization procedure grows exponentially in the length of the overall impulse response. Without considering the interleaver, the optimum receiver in terms of minimizing probability of symbol error, based on both $r_1[n]$ and $r_2[n]$, has computational complexity that is exponential in the sum of the lengths of the two channels. With the interleaver present, the effective channel length for the second channel becomes equal to the length of the interleaving block, making such a decoder completely impractical. In [4], Emmer and Franz propose an iterative equalization strategy based on the "turbo-equalization" method of Douillard et al. [3] which iteratively applies the forward/backward algorithm to each channel $h_1[n]$ and $h_2[n]$ in succession, until convergence is achieved. For channels with significant ISI, this approach also becomes prohibitively complex. As an alternative, we propose an iterative decoding algorithm which makes use of low complexity, minimum mean square error linear equalizers in place of the computationally heavy forward/backward equalizers used in "turbo-equalization."

3 Linear Iterative Turbo-style Equalization (LITE)

A central idea in iterative algorithms such as the "turbo-equalization" of [3], is that of message-passing or belief-propagation, where two or more devices take soft inputs, or beliefs about the symbol values, and produce soft outputs, or new beliefs for the values, and pass this information back and forth until some agreement is achieved. Since these devices both act on and produce soft information (rather than "hard" quantized symbol values), they are often called soft input/soft output (SISO) devices. A SISO equalizer takes as input a received sequence $r_i[n]$, $i = 1, 2$, and some additional information about the symbol sequence $b[n]$. In the context considered here, this additional soft information is a prior probability distribution over the bits, which can be interpreted as a set of values of the data and a level of belief in those values. The SISO equalizer then produces a soft estimate of the symbols which can be interpreted as a probability mass function indicating the likelihood of any symbol $b[n]$ being a particular value. Figure 1 depicts this interaction between the two SISO equalizers in the LITE setup.

For an overall channel impulse response $h_i[n]$, and received values $r_i[n]$, assume the priors $\epsilon_{in}[n]$ $n = 0 \ldots N$ to be given such that $0 \leq \epsilon_{in}[j] \leq 1$ for all j. A SISO equalizer can be defined mathematically as a device that performs an equalization function eq $: R^N \times [0,1]^N \rightarrow [0,1]^N$, $\text{eq}(y, \epsilon_{in}^E) \mapsto \epsilon_{out}^E$, where ϵ_{out}^E can be thought of as a vector of a posteriori probabilities.

The main idea behind the LITE algorithm, or more generally, belief propagation, is to let two or more SISO blocks exchange prior information about the data until they agree on a decision or until a maximal number of iterations is reached. Various termination criteria have been investigated in the literature in certain decoding contexts, such as turbo coding [1,5,6]. As soon as the termination criterion is satisfied, the outputs are suitably combined to construct the recovered data. In the case of the LITE algorithm, the output and the input of the SISO decoder are combined in order to form the form a soft combined equalization decision. If so desired, a hard decision on any symbol can be derived from this soft value.

In order to achieve good performance in such an iterative scheme, a permutation is included in the data path, which spreads out statistical dependencies between estimates. The situation depicted in Figure 1 contains an inherent feedback loop from one SISO equalizer back to itself. It is important that this feedback be mitigated (or delayed for as many iterations as possible), since the SISO equalizers are designed under the assumption that the observations are conditionally independent of the priors. This can be accomplished by enforcing a *well-behaved* property for the equalizers. We say that an equalizer is *well-behaved* if $\epsilon^E_{out}[j]$ is not a function of $\epsilon^E_{in}[j]$.

We now can formulate the LITE algorithm:

1. INPUT: A permutation Π, two vectors of received values $r_i, i = 1, 2$, and prior information about the symbols expressed as vector of probabilities ϵ^{E1}_{in} (ϵ^{E1}_{in} is usually a vector containing a value 0.5 in each position for BPSK data.)
2. Repeat the following steps until a termination criterion is reached.
   (a) $\epsilon^{E1}_{out} = \text{eq}(r_1, \epsilon^{E1}_{in})$
   (b) $\epsilon^{E2}_{in} = \Pi^{-1}(\epsilon^{E1}_{out})$
   (c) $\epsilon^{E2}_{out} = \text{eq}(r_2, \epsilon^{E2}_{in})$
   (d) $\epsilon^{E1}_{in} = \Pi(\epsilon^{E2}_{out})$
   (e) If the termination criterion is not satisfied Goto step 2a
3. OUTPUT hard decisions for symbol $\hat{b}_i$ based on $\epsilon^{E1}_{out}[i]$ and $\epsilon^{E2}_{out}[i]$

4 Soft-Input/Soft-Output Linear Equalization

A standard approach to reducing the effects of intersymbol interference and additive noise induced by the given channels is to use linear, or decision feedback equalizers tuned to those channels. A linear (affine) equalizer with coefficients $c[n, k]$, offset $g[n]$, and symbol estimate $\hat{b}[n]$ can be expressed in the form

$$\hat{b}[n] = \sum_{k=-N_1}^{N_2} c[n, k] r[n + k] + g[n],$$

where the equalizer coefficients are written as a function of n to enable the possibility of different coefficients used to estimate each symbol $\hat{b}[n]$, and the offset $g[n]$ provides a richer class of linear estimates which can account for a non-zero mean prior $\epsilon^E_{in}[n]$. For clarity, since the SISO equalizers used in each of the two channels follow the same development, we will drop the subscript $i = 1, 2$ from the signals $r_i[n], h_i[n]$ and $w_i[n]$ when it is unnecessary. The channel model for each channel can be written in matrix form as

$$\vec{r}[n] = H\vec{b}[n] + \vec{w}[n],$$

where the channel response matrix, H, is given by

$$\begin{bmatrix} h[L_2] & \dots & h[-L_1] & \dots & 0 \\ 0 & h[L_2] & \dots & h[-L_1] & \dots \\ & & & \vdots & \\ 0 & \dots & h[L_2] & \dots & h[-L_1] \end{bmatrix},$$

and the signal vectors are given by

$$\begin{aligned} \vec{r}[n] &= [r[n - N_1] \dots r[n] \dots r[n + N_2]]^T, \\ \vec{b}[n] &= [b[n - N_1 - L_2] \dots b[n] \dots b[n + N_2 + L_1]]^T, \\ \vec{w}[n] &= [w[n - N_1] \dots w[n] \dots w[n + N_2]]^T. \end{aligned}$$

The equalizer output can also be expressed in matrix form, simply as

$$\hat{b}[n] = \vec{c}[n]^T \vec{r}[n] + g[n],$$

where the equalizer coefficients $c[n, k]$ are written

$$\vec{c}[n] = [c[n, -N_1], \dots, c[n, N_2]].$$

With this channel model, the mean-square error of a symbol estimate $\hat{b}[n]$ is given by

$$E\{|b[n] - \hat{b}[n]|^2\}, \tag{3}$$

where the expectation is taken over the distribution of the symbols $b[n]$ and the noise $w[n]$. In the traditional approach to the design of complexity-constrained (finite N_1 and N_2) minimum mean-square error (MMSE) linear equalizers, it is assumed that the symbols $b[n]$ are equally likely to take on all possible symbol values, and that there is no additional information about their values available. The traditional equalizer is determined by finding the coefficient values $\vec{c}[n]$ and $g[n]$ which minimize the mean squared error (3), which, since the symbols are assumed unknown and equally likely for all time n, leads to a single set of coefficients, $\vec{c}$, and the offset is given by $g[n] = 0$. In the context of the LITE algorithm proposed here, the equalization algorithm will have available a set of priors over the symbols. For example, if the symbol alphabet is binary, then this would correspond to the availability of the sequence $\epsilon_{in}^E[n] = Prob\{b[n] = 1\}$. In the sequel, we assume that the channel response $h[n]$ is real-valued and that the symbol alphabet is $b[n] \in \{-1, 1\}$ for simplicity. Extension to complex baseband channels and higher-order symbol constellations is straightforward. In this case, an MMSE equalizer can be designed incorporating these priors into the optimization. Hence, the equalizer coefficients, $\vec{c}[n]$ and $g[n]$ can be determined by finding the minimum of the mean-squared error (3), where the expectation in (3) is over both the additive noise in the channel, and the given (time-varying) prior over the symbols. As a result, the equalizer coefficients will vary with time index, n. This leads to the following formulation,

$$\begin{aligned}\hat{b}[n] = E\{b[n]\} + & \\ & [E\{b[n]\vec{b}[n]\}H^T - E\{b[n]\}E\{\vec{b}[n]^T\}H^T] \times \\ & [HE\{\vec{b}[n]\vec{b}[n]^T\}H^T + E\{|w[n]|^2\}I - \\ & HE\{\vec{b}[n]\}E\{\vec{b}[n]^T\}H^T]^{-1}(\vec{r}[n] - HE\{\vec{b}[n]\}).\end{aligned} \tag{4}$$

Once the equalizer has produced MMSE linear estimates of the symbols $\hat{b}[n]$, these estimates must be mapped into priors ϵ_{out}^E. One method for mapping the outputs of the linear equalizers is to assume the output distribution $\hat{b}[n]$ is conditionally Gaussian, distributed about the symbol values. This leads to the following mapping

$$\text{Prob}\{b[n] = 1|\hat{b}[n]\} = \frac{1}{2}\left(1 + \tanh\left(\frac{\hat{b}[n]}{\sigma_{\hat{b}}^2}\right)\right),$$

where $\sigma_{\hat{b}}^2$ is the variance of the conditional output distribution given the symbol $\hat{b}[n] = \text{sign}(\hat{b}[n])$.

In order for this equalizer to be *well-behaved*, and thus avoid early limit-cycle behavior, the estimate $\hat{b}[n]$ cannot be a function of $\epsilon_{in}^E[n]$. Hence, the expectations in (4) must be taken over a distribution of the symbols which excludes $\epsilon_{in}^E[n]$ for the calculation of $\hat{b}[n]$. However, in calculating $\hat{b}[k], k \neq n$, $\epsilon_{in}^E[n]$ may be used. This leads to the following method for computing the output distribution given the observations, $r_i[n], i = 1, 2$, and the input distribution $\epsilon_{in}^{Ei}, i = 1, 2$.

1. Create buffers for the priors, the signal $r[n]$, the expectations $\vec{bb}[n] = E\{b[n]\vec{b}[n]\}$, the correlation matrix $B[n] = E\{\vec{b}[n]\vec{b}[n]^T\}$, and the means $\vec{mb}[n] = E\{\vec{b}[n]\}$

$$\begin{aligned}\vec{\epsilon}^{(n)} &\triangleq [\epsilon^{(n)}[-N_1 - L_2], \ldots, \epsilon^{(n)}[N_2 + L_1]]^T \\ \vec{r}^{(n)} &\triangleq [r^{(n)}[-N_1], \ldots, r^{(n)}[N_2]]^T \\ \vec{bb}^{(n)} &\triangleq [bb^{(n)}[-N_1 - L_2], \ldots, bb^{(n)}[N_2]]^T \\ &= [0, \ldots, 0, 1, 0, \ldots, 0]^T\end{aligned}$$

2. Initialize buffers for priors $\vec{\epsilon}^{(n)}$ and data $\vec{r}^{(n)}$, in terms of the signal $r[n]$ and the input ϵ_{in}^E.

$$\begin{aligned}\vec{r}^{(0)} &= [0, 0, \ldots, r[0], r[1], \ldots, r[N_2]]^T \\ \vec{\epsilon}^{(0)} &= [0, 0, \ldots, 0, \epsilon_{in}^E[0], \epsilon_{in}^E[1], \ldots, \epsilon_{in}^E[N_2 + L_1]]^T\end{aligned}$$

3. Loop over the data for $n = 0, \ldots, N$:

$$\begin{aligned}\epsilon^{(n)}[0] &= \frac{1}{2} \\ \vec{mb}^{(n)} &= 2\vec{\epsilon}^{(n)} - 1 \\ B &= \vec{mb}^{(n)}\vec{mb}^{(n)T} \\ \text{diag}(B) &= \text{diag}(1, 1, \ldots, 1) \\ \vec{c}[n] &= \left[H\left(B - \vec{mb}^{(n)}\vec{mb}^{(n)T}\right)H^T + \sigma_w^2 I\right]^{-1} H\vec{bb}^{(n)} \\ \hat{b}[n] &= \vec{mb}^{(n)} + \vec{c}^{(n)T}\left(\vec{r}^{(n)} - H\vec{mb}^{(n)}\right) \\ \vec{r}^{(n+1)} &= [r^{(n)}[-N_1 + 1], \ldots, r^{(n)}[N_2], 0] \\ \vec{\epsilon}^{(n+1)} &= [\epsilon^{(n)}[-N_1 - L_2 + 1], \ldots, \epsilon^{(n)}[N_2 + L_1], 0]\end{aligned}$$

if $n < N - N_2$

$$r^{(n+1)}[N_2] = r[n + 1 + N_2]$$

if $n < N - N_2 - L_1$

$$\epsilon^{(n+1)}[N_2 + L_1] = \epsilon_{in}^E[n + 1 + N_2 + L_1]$$

4. Estimate output variance:
$\sigma_{\hat{b}}^2 = (\text{var}(\hat{b}|\hat{b} > 0) + \text{var}(\hat{b}|\hat{b} < 0))/2$

5. Determine output priors:
$\epsilon_{out}^E = \frac{1}{2}\left(1 + \tanh\left(\frac{\hat{b}[n]}{\sigma_{\hat{b}}^2}\right)\right)$

5. Simulations and Conclusions

To enable meaningful comparison against standard equalization techniques, simulations of the LITE algorithm are shown for the scenario in which the two channels $h_1[n]$ and $h_2[n]$ are equivalent, with impulse response given by $h[n] = 0.407\delta[n + 1] + 0.815\delta[n] + 0.407\delta[n - 1]$. One thousand blocks of length of $N = 1,000$ symbols have been processed with an equalizer length of $L_1 + L_2 + 1 = 5$ taps. When the two channels are identical, and no interleaving is performed, the optimal joint constrained-complexity MMSE linear equalizer is equivalent to one equalizer operating on the average of the outputs of the two channels. The performance of this MMSE optimal equalizer (without interleaving) is compared to that of the LITE algorithm developed above. Perhaps a more appropriate comparison would be joint constrained-complexity MMSE equalization for the interleaved sequence. However, this would require inversion of a matrix of dimension n and thus has prohibitive computational complexity.

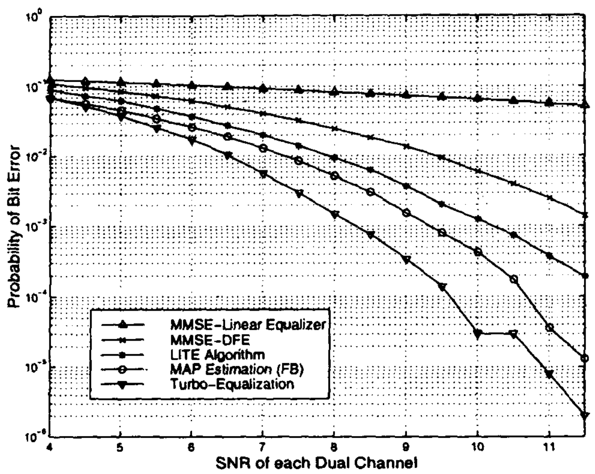

Figure 2. Performance of joint MMSE linear, MMSE DFE, and LITE algorithms along with maximum-likelihood decoding and turbo-equalization.

Since the LITE algorithm feeds back soft-decisions from each of the separate equalizers, we also compare its performance to that of an MMSE optimal decision-feedback equalizer for this dual channel scenario. Thus, we include the performance of an MMSE DFE for the channel given above, with 5 taps in the feedforward filter and 1 in the feedback filter.

These two MMSE equalizers are of comparable complexity to the LITE algorithm, and hence form a reasonable set of algorithms for comparison. Since the channel impulse response lengths are sufficiently short, and the signalling constellation is BPSK, we can also include in our comparison optimal maximum-likelihood symbol estimation for the identical two-channel model, again without interleaving. This is accomplished by once again averaging the outputs of the two channels, and then running the forward/backward algorithm on the resulting sequence. Finally, we also include the performance of the "turbo-equalization" algorithm of [3] as an estimate of the bound on the probability of error achievable for this two-channel case, when interleaving is included. If the channel impulse response lengths were to increase significantly, or the constellation were to increase in order significantly, then comparison against such maximum-likelihood techniques would rapidly become infeasible.

Figure 2 illustrates the performance of the joint MMSE linear, MMSE DFE, LITE, maximum likelihood (MAP), and turbo equalization algorithms for a range of signal to noise ratios. Since the channel considered for this simulation has a spectral null near the unit circle, the performance of the joint MMSE linear equalizer quickly levels off at a high probability of error. Although the joint DFE performs significantly better than the joint MMSE linear equalizer, it gives little improvement in the equalization of this channel for low SNR. The LITE algorithm, however, attains a much lower probability of error for all levels of SNR, and its gains increase with increasing SNR, from 1.5 dB to 2 dB. Further, this performance is obtained with a computational complexity similar to that of MMSE equalization yet is within 0.5 to 1 dB of maximum-likelihood decoding and within 2 dB of turbo-equalization, each of which have exponential complexity. These promising results indicate that LITE algorithm may provide a practical means for obtaining significant performance gains for a variety of communications applications.

References

[1] D. Agrawal. *Generalized Minimum-Distance and Iterative Decoding in Euclidean Space*. PhD thesis, University of Illinois, Urbana-Champaign, 1998.

[2] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Trans. on Inf. Theory*, pages 284–287, March 1974.

[3] C. Douillard, A. Picart, P. Didier, M. Jézéquel, C. Berrou, and A. Glavieux. Iterative correction of intersymbol interference: Turbo-equalization. *Euro. Trans. Telecommun.*, 6:507–511, September 1995.

[4] D. Emmer and V. Franz. Method and apparatus for data transmission. *International Patent EP912021A2 to Siemens AG*, April 1999.

[5] B. Frey and F. Kschischang. Early detection and trellis splicing: reduced-complexity iterative decoding. *IEEE Journal on Selected Areas in Communications*, 16(2):153–159, February 1998.

[6] J. Hagenauer. Iterative decoding of block and convolutional codes. In *Proceedings of the EIDMA Winter Meeting on Coding Theory, Information Theory and Cryptology*, page p. 32, Veldhoven, The Netherlands, 1994.