Iterative Joint Channel Estimation and Data Detection Using Superimposed Training: Algorithms and Performance Analysis

Xiaohong Meng, Jitendra K. Tugnait, Fellow, IEEE, and Shuangchi He

Abstract—Channel estimation for single-input multiple-output time-invariant channels is considered using superimposed training. A periodic (nonrandom) training sequence is arithmetically added (superimposed) at low power to the information sequence at the transmitter before modulation and transmission. We extend a recently proposed first-order statistics-based channel estimation approach (IEEE Commun. Lett., vol. 7, p. 413, 2003) to iterative joint channel estimation and data detection using a conditional maximum likelihood approach where the information sequence is exploited to enhance performance instead of being viewed as interference. An approximate performance analysis of the iterative channel estimation method is also presented under certain simplifying assumptions. Illustrative computer simulation examples are presented.

Index Terms—Communications channels, iterative channel estimation, maximum likelihood estimation, superimposed training.

I. INTRODUCTION

Consider a single-input multiple-output (SIMO) finite-impulse response linear channel with \( N \) outputs. Let \( \{ s(n) \} \) denote a scalar sequence that is inputted to the SIMO channel with discrete-time impulse response \( \{ h_l(t) \} \). The vector channel may be the result of multiple receive antennas and/or oversampling at the receiver. Then, the symbol rate channel output vector is given by

\[
x(n) := \sum_{l=0}^{L} h_l(n-l)
\]

where \( L \) is the channel order. The noisy measurements of \( x(n) \) are given by \( \{ v(n) \} \). In several approaches, this requires knowledge of the channel impulse response \( \{ v(n) \} \). In a training-based approach, \( s(n) = c(n) \) is a training sequence (known to the receiver) for \( n = 0, 1, \ldots, M_t - 1 \), and \( s(n) = n > M_t - 1 \) is the information sequence (unknown \( a \ priori \) to the receiver) \( \{ v(n) \} \). Therefore, given \( c(n) \) and corresponding noisy \( y(n) \), one estimates the channel via least-squares and related approaches. For time-varying channels, one has to send training signal frequently and periodically to keep up with the changing channel. An alternative is to estimate the channel based solely on noisy \( y(n) \), which exploits statistical and other properties of \( \{ s(n) \} \). This is the blind channel estimation approach. More recently, a superimposed training-based approach has been explored where one takes

\[
s(n) = b(n) + c(n)
\]

where \( \{ b(n) \} \) is the information sequence, and \( \{ c(n) \} \) is a training (pilot) sequence added (superimposed) at low power to the information sequence at the transmitter before modulation and transmission. There is no loss in information rate; on the other hand, some useful power is wasted in superimposed training, which could have otherwise been allocated to the information sequence \( \{ s(n) \} \). Superimposed training-based approaches have been discussed in [3]–[5], [7], and [12]–[14] for single-input single-output (SISO) and/or SIMO systems.

Periodic superimposed training for channel estimation via first-order statistics for SISO and/or SIMO systems have been discussed in [3], [7], [12], and [14]. In [2], performance bounds for training and superimposed training-based semi-blind SISO channel estimation for time-varying flat-fading channels have been discussed.

In the aforementioned superimposed training-based approaches, the contribution of the information sequence is viewed as interference. In [4] (also [5]), a novel block transmission method (as opposed to serial transmissions in [3], [7], [12], and [14]) has been proposed, where a data-dependent component is added to the superimposed training such that the interference due to data (information sequence) is greatly reduced in channel estimation at the receiver. This method requires “data blocking” for block transmissions and insertion of a cyclic prefix in each data block in order for the approach to work. In this paper, we restrict ourselves to serial transmissions and propose an iterative joint channel estimation and data detection approach where data are exploited to enhance performance instead of being viewed as interference. It should also be noted that for data detection, [4] and [5] also require an iterative approach.

Superimposed training is similar to code-multiplexed pilots used in code division multiple access [6], [9]. Also, the iterative channel estimation and data detection approach used in this paper is not novel.
(for instance, see [6], [9], and [10]), but its application to superimposed training-based systems is.

A. Objectives and Contributions

These are threefold.

1) We extend the first-order statistics-based channel estimation approach of [12] to iterative joint channel estimation and data detection approaches using linear minimum mean-square error (LMMSE) equalizers or Viterbi detectors. The first-order statistics-based approach views the information sequence as interference, whereas in the proposed approaches, it is exploited to enhance channel estimation and information sequence detection. The proposed approaches utilize existing techniques (e.g., [10]), but their application to superimposed training-based systems is novel. Furthermore, they provide alternatives to the approaches of [4] and [5]. One of our objectives is to further investigate certain aspects of superimposed training-based systems, as recently articulated in [3]–[5], [7], [12], and [13]. However, we do not have a specific standard or system in mind when discussing superimposed training.

2) An approximate (“semi-analytic”) performance analysis of the iterative channel estimation method is also presented under certain simplifying assumptions and simulated via simulations.

3) A simulation-based comparison with the approaches of [4] and [5] is provided both from performance (bit error rate, BER) and computational complexity (CPU time) points of view.

B. Notation

Superscripts $H$, $T$, and $\dagger$ denote complex conjugate transpose, transpose, and Moore–Penrose pseudo-inverse operations, respectively. $\delta(\tau)$ is the Kronecker delta, and $I_N$ is the $N \times N$ identity matrix. The symbol $\otimes$ denotes the Kronecker product. $\tr(A)$ is the trace of the matrix $A$.

II. FIRST-ORDER STATISTICS-BASED SOLUTION OF [12]

Here, we briefly review the approach of [12], introduce notation, and present our underlying system model assumptions. Assumptions (H1)–(H3) are as in [12]. Assume the following:

(H1) The information sequence $\{b(n)\}$ is zero-mean independent and identically distributed (i.i.d) drawn from a known finite alphabet with

$E[\{b(n)\}^2] = \sigma_b^2$.

(H2) The measurement noise $\{v(n)\}$ is nonzero-mean ($E[\{v(n)\}] = m$) white complex Gaussian independent of $\{b(n)\}$ with $E[\{v(n + \tau) - m\}v(n) - m]^H] = \sigma_v^2 I_N \delta(\tau)$. The mean vector $m$ is unknown. We will also use the notation $v(n) = \tilde{v}(n) + m$.

(H3) The superimposed training sequence $c(n) = c(n + mP)$ $\forall m, n$ is a nonrandom periodic sequence with period $P$. Let $\sigma_c^2 := (1/P) \sum_{n=0}^{P-1} |c(n)|^2$.

By (H3), we have $c_m := (1/P) \sum_{n=0}^{P-1} c(n)e^{-j\alpha_m n}$ and

$$c(n) = \sum_{m=0}^{P-1} c_m e^{j\alpha_m n} \forall n, \quad \alpha_m := 2\pi m/P.$$  \hspace{1cm} (4)

The coefficients $c_m$’s are known at the receiver since $\{c(n)\}$ is known. We have

$$E\{y(n)\} = \sum_{m=0}^{P-1} \sum_{l=0}^{L} c_m h(l)e^{-j\alpha_m l} e^{j\alpha_m n} + m. \quad (5)$$

The sequence $E\{y(n)\}$ is periodic with cycle frequencies $\alpha_m$, $0 \leq m \leq P - 1$. A mean-square (m.s.) consistent estimate $d_m$ of $d_m$ for $\alpha_m \neq 0$ follows as [12]

$$d_m = \frac{1}{T} \sum_{n=0}^{T-1} y(n)e^{-j\alpha_m n}. \quad (6)$$

As $T \rightarrow \infty$, $d_m \rightarrow d_m$ m.s. if $\alpha_m \neq 0$ and $d_0 \rightarrow 0 + m$ m.s. if $\alpha_m = 0$.

It is established in [12] that given $d_m$ for $1 \leq m \leq P - 1$, we can (uniquely) estimate $h(l)$’s if $P \geq L + 2$ and $c_m \neq 0 \forall m \neq 0$. Since $m$ is unknown, we will omit the term $m = 0$ for further discussion. Define

$$V := \begin{bmatrix} 1 & e^{-j\alpha_1} & \ldots & e^{-j\alpha_L} \\ 1 & e^{-j\alpha_2} & \ldots & e^{-j\alpha_{L+1}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-j\alpha_{P-1}} & \ldots & e^{-j\alpha_{P+L}} \end{bmatrix}_{(P-1) \times (L+1)} \quad (7)$$

$$g := [h^H(0) \quad h^H(1) \quad \ldots \quad h^H(L)]^H \quad (8)$$

$$D := [d_1^H \quad d_2^H \quad \ldots \quad d_{P-1}^H]^H \quad (9)$$

$$C := \text{diag}\{c_1, c_2, \ldots, c_{P-1}\} \otimes I_N. \quad \text{where} \quad \otimes \quad (10)$$

Omitting the term $m = 0$ and using the definition of $d_m$ from (5), it follows that

$$Cd = D. \quad (11)$$

It is shown in [12] that if $P - 1 \geq L + 1$ and $\alpha_m$’s are distinct, $\text{rank}(C) = N(L + 1)$; hence, we can determine $h(l)$’s uniquely. Define $D$ as in (9) with $d_m$’s replaced with $d_m$’s. Then we have the channel estimate

$$\hat{g} = C^{\dagger}D := (C^H C)^{-1}C^H D. \quad (12)$$

Lemma 1 [12]: If $P \geq L + 2$ and $c_m \neq 0 \forall m \neq 0$, then (11) has a unique solution. 

III. JOINT CHANNEL ESTIMATION AND DATA DETECTION

The first-order statistics-based approach in Section II views the information sequence as interference. Since the training and information sequences of a given user pass through an identical channel, this fact can be exploited to enhance channel estimation performance. This is the objective of this section, where we consider joint channel and information sequence estimation via an iterative conditional maximum likelihood (CML) (or deterministic ML, since the information sequence is modeled as unknown but nonrandom) approach.
Suppose that we have collected $T - L$ samples of the observation $Y = [y(T − 1), \ldots, y(T)]^T$. We then have the following linear model $V(n) := v(n) − m$:

$$
Y = T(s)
\begin{bmatrix}
    h(0) \\
    \vdots \\
    h(L)
\end{bmatrix}
+ \begin{bmatrix}
    \hat{v}(T-1) \\
    \vdots \\
    \hat{v}(L)
\end{bmatrix}
+ m
\begin{bmatrix}
    \vdots \\
    \vdots \\
    \vdots
\end{bmatrix}
\tag{13}
$$

where $V = \hat{V} + \mathcal{M}$ is a column vector consisting of samples of noise $\{v(n)\}$, $g$ is the vector of the channel parameters

$$
T(s) := \begin{bmatrix}
    s(T - 1)I_N & \cdots & s(T - L - 1)I_N \\
    \text{Block} & \text{Hankel} & \text{Matrix} \\
    s(L)I_N & \cdots & s(0)I_N
\end{bmatrix}
\tag{14}
$$

and a block Hankel matrix has identical block entries on its block antidiagonals. Consider (1), (2), and (13). Under the assumption of white Gaussian noise, consider the joint estimators

$$
\{\hat{g}, \hat{s}, \hat{m}\} = \arg \min_{g, s, m} \|Y - T(s)g - \mathcal{M}\|^2
\tag{15}
$$

where

$$
s := [s(T - 1), s(T - 2), \ldots, s(0)]^T
\tag{16}
$$

and $\hat{s}$ is the estimate of $s$. In the above, we have followed a deterministic ML (CML) approach assuming no statistical model for the input sequences $\{s(n)\}$. Under white Gaussian noise assumption, the CML estimators are obtained by the nonlinear least-squares optimization (15). Note that the observation vector $Y$ is linear in both the channel and the input parameters individually. In particular, we have

$$
Y = T(s)g + \hat{V} + \mathcal{M} = \mathcal{C}(g)s + \hat{V} + \mathcal{M}
\tag{17}
$$

where

$$
\mathcal{C}(g) = \begin{bmatrix}
    h(0) & \cdots & h(L) \\
    \vdots & \ddots & \vdots \\
    h(0) & \cdots & h(L)
\end{bmatrix}
\tag{18}
$$

is the $[N(T - L)] \times T$ so-called filtering matrix. We therefore have a separable nonlinear least-squares problem that can be solved sequentially. The finite alphabet properties of the information sequences can also be incorporated. These algorithms, which were first proposed by Seshadhri [10] for SISO systems, iterate between estimates of the channel and input sequences. At iteration $k$, with an initial guess of the channel $\hat{g}^{(k)}$ and the mean $\hat{m}^{(k)}$, the algorithm estimates the input sequence $s^{(k)}$ and the channel $\hat{g}^{(k+1)}$ and mean $\hat{m}^{(k+1)}$ for the next iteration by

$$
s^{(k)} = \arg \min_{s \in \mathcal{S}} \left\| Y - \mathcal{C}(\hat{g}^{(k)}) s - \mathcal{M}^{(k)} \right\|^2
\tag{19}
$$

$$
\hat{g}^{(k+1)} = \arg \min_{\hat{g}} \left\| Y - T(s^{(k)}) \hat{g} - \mathcal{M}^{(k)} \right\|^2
\tag{20}
$$

$$
\hat{m}^{(k+1)} = \arg \min_{\hat{m}} \left\| Y - T(s^{(k)}) \hat{g}^{(k+1)} - \mathcal{M}^{(k)} \right\|^2
\tag{21}
$$

where $\mathcal{S}$ is the (discrete) domain of $s$. The optimizations in (20) and (21) are linear least-square problems, whereas the optimization in (19) can be achieved by using the Viterbi algorithm (VA) [8]. Since the above iterative procedure involving (19)–(21) decreases the cost at every iteration, one achieves a local minimum of the nonlinear least-squares cost (local maximum of CML function).

We now summarize our CML approach.

1) a) Use (12) to estimate the channel using the first-order (cyclostationary) statistics of the observations. Denote the channel estimates by $\hat{g}^{(1)}$ and $\hat{h}(l)$. Estimate the mean as $\hat{m}^{(1)}$ given by

$$
\hat{m}^{(1)} := (1/T) \sum_{n=0}^{T-1} \left[ y(n) - \sum_{i=0}^{L} \hat{h}(i)c(n - i) \right].
\tag{22}
$$

b) Design a Viterbi sequence detector to estimate $\{s(n)\}$ as $\{\hat{s}(n)\}$ using the estimated channel $\hat{g}^{(1)}$, mean $\hat{m}^{(1)}$, and cost (19) with $k = 1$. (Note that knowledge of $\{c(n)\}$ is used in $s(n) = b(n) + c(n)$; therefore, we are in essence estimating $b(n)$ in the Viterbi detector.)

2) a) Substitute $\hat{s}(n)$ for $s(n)$ in (1) and use the corresponding formulation in (13) to estimate the channel $g$ and mean $m$ as

$$
\hat{g}^{(2)} = T^\dagger(\hat{s}) \left[ Y - \hat{M}^{(1)} \right].
\tag{23}
$$

$$
\hat{m}^{(2)} := (1/(T - L)) \sum_{n=L}^{T-1} \left[ y(n) - \sum_{i=0}^{L} \hat{h}^{(2)}(i)\hat{s}(n - i) \right].
\tag{24}
$$

b) Design a Viterbi sequence detector using the estimated channel $\hat{g}^{(2)}$, mean $\hat{m}^{(2)}$, and cost (19) with $k = 2$, as in Step 1b.

3) Step 2 provides one iteration of (19)–(21). Repeat until the incremental change in $\hat{g}$ over successive iterations is below a threshold.

Remark 1: Since a Viterbi detector can be expensive, one may replace it in steps 1b and 2b by an LMMSSE equalizer followed by hard decisions. Define $\tilde{y}(n) := y(n) - \sum_{i=0}^{L} \hat{h}(i)c(n - i) - \hat{m}^{(1)}$. Equalize the channel by applying the LMMSSE equalizer to $\{\tilde{y}(n)\}$ to estimate $\{b(n)\}$ as $\{\hat{b}(n)\}$ and then quantize $\{\hat{b}(n)\}$ into $\{\hat{b}(n)\}$ with the knowledge of the symbol alphabet (hard decisions).

IV. PERFORMANCE ANALYSIS

In this section, we carry out a performance analysis of the iterative channel estimation method [(20) or similar estimate at a later stage in Section III] under some simplifying assumptions. The simplifying assumptions are similar to that in [1], where they have been used in a different problem dealing with multiple-input multiple-output flat-fading systems with conventional training-based iterative channel estimation approaches.

In the following, $\hat{s}$ will denote the quantized estimate of $s$ (obtained as $\hat{b} + c$, where $\hat{b}$ and $c$ are defined in a manner similar to $s$ and $\hat{b}$ is the quantized estimate of $b$). By (17), we have

$$
Y = C(g)\hat{s} + \hat{V} + \mathcal{M} = C(g)(s - \hat{s})
= C(g)\hat{s} + \hat{V} + \mathcal{M} + C(g)(\hat{b} - b)
\tag{25}
$$

$$
= T(\hat{s})g + \mathcal{W}.
\tag{26}
$$

In (26), $T(\hat{s})g$ is the effective signal, and $\mathcal{W}$ is the effective noise. We may also write

$$
\hat{Y} := Y - \hat{M} = C(g)\hat{s} + \mathcal{W} = T(\hat{s})g + \mathcal{W}
\tag{27}
$$
where
\[ \vec{W} := \mathbf{W} - \hat{\mathbf{M}}. \] (28)

Further define
\[ b_v := \mathbf{b} - \hat{\mathbf{b}} \quad \text{and} \quad b_v(n) := b(n) - \hat{b}(n). \] (29)

Assume the following:

(H4) The information sequence \{b(n)\} is zero-mean i.i.d. binary with \[ E[(b(n))^2] = \sigma_b^2. \]

(H5) The sequence \{b(n)\} is the sequence of detected information bits with the BER \( p_e \). It is assumed to be zero-mean i.i.d. just as \{b(n)\}.

(H6) The error sequence \{b_v(n)\} is independent of the noise \{\vec{v}(n)\}.

(H7) The effective noise \{\mathbf{v}(n)\} [the sequence counterpart to the column vector \( \mathbf{W} \) in (25)] is independent of \{\hat{\mathbf{b}}(n)\}.

(H8) Components of the channel coefficient \( h(l) \)'s are assumed to be Gaussian random variables with zero mean and variance \( 1/(N(L + 1)) \). We also assume that \( h_i(l) \) and \( h_m(k) \) are statistically independent if \( k \neq l \) or \( i \neq m \), where \( h_i(l) \) denotes the \( i \)th component of \( h(l) \).

(H9) The unknown mean \( \mathbf{m} \) has been accurately estimated and accounted for so that \( \hat{\mathbf{Y}} \) and \( \mathbf{W} \) are taken to be zero mean with \( \mathbf{M} = \hat{\mathbf{M}} \).

Assumptions (H5)–(H7) are not (exactly) true but can be justified for the case of high signal-to-noise ratio (SNR) and low BER. These three assumptions allow the problem to become more tractable. Moreover, they are based on similar assumptions used in [1] in a different context. Assumption (H9) allows us to treat \( \hat{\mathbf{Y}} \) as zero mean in a later stage of our iterative algorithms in Section III. We will relax the binary assumption on \( b(n) \) later in Section IV-A.

By (H4), \( b(n) \) is either \( \sigma_b \) or \( -\sigma_b \) with probability 0.5 each. It then follows that (w.p. stands for with probability)
\[ \hat{b}(n) = \begin{cases} b(n), & \text{w.p. } 1 - p_e \\ -b(n), & \text{w.p. } p_e \end{cases} \] (30)

It follows from (29), (30), (H4), and (H5) that
\[ b_v(n) = \begin{cases} 0, & \text{w.p. } 1 - p_e \\ -2\hat{b}(n), & \text{w.p. } p_e \end{cases} \] (31)

leading to
\[ E\{b_v(n_1)b_v^*(n_2)\} = \begin{cases} 4p_e^2 E\{\hat{b}(n_1)\} E\{\hat{b}^*(n_2)\}, & \text{if } n_1 \neq n_2 \\ 4p_e E\{\hat{b}(n_1)\}^2, & \text{if } n_1 = n_2. \end{cases} \] (32)

Hence, it follows that
\[ E\{b_v b_v^H\} = 4p_e \sigma_b^2 I_T. \] (33)

From (27), we have the channel estimate
\[ \hat{\mathbf{g}} = T^\dagger(\hat{\mathbf{s}})Y = \mathbf{g} + T^\dagger(\hat{\mathbf{s}})\hat{\mathbf{W}}. \] (34)

Using (25)–(28), (33), (H6), (H7), and (H9), we have
\[ E\{\mathbf{WW}^H|\mathbf{g}, \hat{\mathbf{b}}\} = E\{\mathbf{WW}^H|\mathbf{g}\} = \sigma_b^2 I_{N(T-L)} + 4p_e \sigma_b^2 C(\mathbf{g}) C^H(\mathbf{g}). \] (35)

Furthermore, we have
\[ C(\mathbf{g}) C^H(\mathbf{g}) = \begin{bmatrix} \sum_{l=0}^{L} h(l)h^H(l) & \cdots & h(L)h^H(0) & 0 \\ \vdots & \ddots & \vdots & \vdots \\ h(0)h^H(L) & \cdots & h(L)h^H(0) & 0 \\ 0 & \cdots & 0 & \sum_{l=0}^{L} h(l)h^H(l) \end{bmatrix} \] (36)

Under (H8), it then follows that
\[ E\{C(\mathbf{g}) C^H(\mathbf{g})\} = I_{(T-L)} \otimes E\left\{ \sum_{l=0}^{L} h(l)h^H(l) \right\} = N^{-1} I_{N(T-L)}. \] (37)

Therefore, using (35)–(37), we have
\[ E\{\mathbf{WW}^H\} = \left( \sigma_b^2 + \frac{4p_e \sigma_b^2}{N} \right) I_{N(T-L)}. \] (38)

From (23) and (34)–(38), it follows that
\[ E\{\mathbf{AA}^H|\hat{\mathbf{b}}\} = \left( \sigma_b^2 + \frac{4p_e \sigma_b^2}{N} \right) \left[ T^H(\hat{\mathbf{s}})T(\hat{\mathbf{s}}) \right]^{-1}. \] (39)

Using (H5) and the strong law of large numbers, we have
\[ \lim_{T \to \infty} \frac{1}{T-L} T^H(\hat{\mathbf{s}})T(\hat{\mathbf{s}}) = \sigma_b^2 I_{N(L+1)} + \frac{1}{T-L} T^H(\mathbf{e})T(\mathbf{e}) \text{ w.p.1.} \] (40)

Using (34), (39), and (40), it follows that
\[ \lim_{T \to \infty} E\{(\mathbf{g} - \mathbf{g})(\mathbf{g} - \mathbf{g})^H\} = \frac{1}{T-L} \left( \sigma_b^2 + \frac{4p_e \sigma_b^2}{N} \right) \left[ \sigma_b^2 I_{N(L+1)} + \frac{1}{T-L} T^H(\mathbf{e})T(\mathbf{e}) \right]^{-1}. \] (41)

The variance of the channel estimate will be defined as
\[ \sigma_b^2 := E\{(\mathbf{g} - \mathbf{g})(\mathbf{g} - \mathbf{g})^H\} = \frac{1}{T-L} \left( \sigma_b^2 + \frac{4p_e \sigma_b^2}{N} \right) \times \text{tr}\left\{ \sigma_b^2 I_{N(L+1)} + \frac{1}{T-L} T^H(\mathbf{e})T(\mathbf{e}) \right\}^{-1}. \] (42)

A simplification of (42) is possible if we use (scaled) m-sequences (maximal length pseudorandom binary sequences) for \{c(n)\}, in which case we have \( (1/P) \sum_{n=1}^{P} c^*(n-l_1)c(n-l_2) = \sigma_b^2 \) for


\[ l_1 = l_2 \mod P, \text{ else } -\sigma_c^2/P. \] For \( P \) “large,” we may consider \( \{c(n)\} \) to be (periodically) white. Under this approximation, we have (for large \( T \), or \( T = L \) equal to an integer multiple of \( P \))

\[
\frac{1}{T-L}T^H(\mathbf{c})T(\mathbf{c}) \approx \sigma_h^2 I_{N(L+1)}. \tag{43}
\]

Then we have

\[
\sigma_h^2 \approx \frac{1}{T-L} \left( \sigma_o^2 + \frac{4p_e\sigma_b^2}{N} \right) \frac{N(L+1)}{\sigma_b^2 + \sigma_o^2}. \tag{44}
\]

A. M-Phase-Shift Keying (M-PSK) Information Sequences

Suppose that \( \{b(n)\} \), hence \( \{\hat{b}(n)\} \), is no longer binary; rather, it belongs to a nonbinary finite alphabet with equally likely symbols. Let \( p_e \) denote the symbol error rate (instead of the BER). The main problem now is how to characterize (30) and (31). We now make further simplifying assumptions and assume a nearest neighbor selection when making symbol detection errors. For a “typical point” in a given signal constellation, let there be \( m \) nearest neighbors with distance

\[ d := |b(n) - \hat{b}(n)|, \]

each equally likely to occur conditioned on the event that an error has occurred. We will assign zero conditional error probability to nonnearest neighbor points. For instance, for an M-PSK constellation, we have \( m = 2 \) and \( d = 2\sigma_b \sin(\pi/M) \) when \( E(|b(n)|^2) = \sigma_b^2 \). For a quaternary PSK (QPSK) constellation \((M = 4)\), this leads to \( d = \sqrt{2}\sigma_b \).

Under these assumptions, we have

\[
|b_e(n)| = \begin{cases} 0, & \text{w.p. } 1 - p_e, \\ d, & \text{w.p. } p_e \end{cases} \tag{45}
\]

leading to

\[
E\{b_e(n_1)b_e^*(n_2)\} = \begin{cases} 0, & \text{if } n_1 \neq n_2, \\ d^2p_e, & \text{if } n_1 = n_2. \end{cases} \tag{46}
\]

Hence, it follows that

\[
E\{b_e b_e^H\} = p_e d^2 I_T. \tag{47}
\]

Mimicking the previous developments, we can easily derive the counterpart to (42) as

\[
\begin{align*}
\sigma_h^2 &= \frac{1}{T-L} \left( \sigma_o^2 + \frac{p_e d^2}{N} \right) \\
&\times \text{tr} \left\{ \left[ \sigma_h^2 I_{N(L+1)} + \frac{1}{T-L} T^H(\mathbf{c})T(\mathbf{c}) \right]^{-1} \right\}. \tag{48}
\end{align*}
\]

The counterpart to (44) is given by

\[
\sigma_h^2 \approx \frac{1}{T-L} \left( \sigma_o^2 + \frac{4p_e\sigma_b^2}{N} \sin^2\left(\frac{\pi}{M}\right) \right) \frac{N(L+1)}{\sigma_b^2 + \sigma_o^2}. \tag{49}
\]

In particular, for \( M \)-PSK constellations, we have

\[
\sigma_h^2 \approx \frac{1}{T-L} \left( \sigma_o^2 + \frac{4p_e\sigma_b^2}{N} \right) \frac{N(L+1)}{\sigma_b^2 + \sigma_o^2}. \tag{50}
\]

Remark 2: Notice that the channel variance expression in (44) and (50) requires the knowledge of the probability of error \( p_e \), which is not available in general. In the simulation results presented in Section V, we obtain \( p_e \) from simulations; hence, our performance analysis should be more appropriately called a semi-analytic performance analysis.

V. Simulation Examples

A. Using LMMSE Equalizer in CML Algorithm

We consider a (time-invariant) random frequency-selective Rayleigh fading channel. We took \( N = 1 \) and \( L = 2 \) in (1) with \( h(l) \) mutually independent for all \( l \) zero-mean complex Gaussian with variance as in (H8) of Section IV. Additive noise was zero-mean complex white Gaussian. The SNR refers to the energy per bit over one-sided noise spectral density with both information and superimposed training sequence counting toward the bit energy. The information sequence as well as the superimposed training was binary. We took the superimposed training sequence period \( P = 7 \) in (H3); it is a scaled \( m \)-sequence (maximal length pseudorandom binary sequence) having a peak-to-average power ratio of 1 (the smallest possible). The average transmitted power in \( c(n) \) (scaled binary) is \( 0.2 \) or 0.67 of the power in \( h(n) \)—a small penalty in SNR, which leads to the training-to-information sequence power ratio (TIR) of 0.2 or 0.67, respectively. The LMMSE equalizer of length 11 bits and equalization delay of 5 bits was used throughout.

Given the channel estimate and the true channel at the \( \text{ith} \) Monte Carlo run as \( \hat{h}^{(i)}(l) \) and \( h^{(i)}(l) \), respectively, the normalized channel m.s. error (NCMSE) is defined as

\[
\text{NCMSE} := \frac{1}{M_r} \sum_{i=1}^{M_r} \frac{\sum_{l=0}^{N-1} \left| \hat{h}^{(i)}(l) - h^{(i)}(l) \right|^2}{\sum_{l=0}^{N-1} \left| h^{(i)}(l) \right|^2} \tag{51}
\]

where \( M_r \) is the number of Monte Carlo runs. We also implemented a conventional (time-multiplexed) training-based approach where the first 67 bits \((\pm 1) \) were reserved for training and the remaining bits were information bits: this leads to a training-to-information bits ratio of 0.2 (as in TIR = 0.2 for superimposed training). The simulation results averaged over 500 runs are shown in Figs. 1 and 2: Fig. 1 shows the BER, whereas Fig. 2 shows the NCMSE. We see that (Figs. 1 and 2) iterated enhancement is competitive with conventional training at lower SNRs (the “practical range”). Furthermore, we see that (Fig. 2) although the channel estimation errors may be much lower for superimposed training with iterated enhancement because of SNR penalty (power wasted in superimposed training), this channel accuracy advantage does not necessarily translate into a large BER advantage. Note also that a higher TIR translates to a lower channel estimation error, but because of the higher SNR penalty, one does not necessarily get a lower BER.

In Fig. 3, we investigate the effects of varying TIR on BER performance via simulations. Both time-multiplexed training and superimposed training-based approaches were considered. We show the results (averaged over 1000 runs) for the second iteration of the proposed approach and the conventional training-based approach for TIR = 0.05, 0.1, 0.2, and 0.3, and the corresponding conventional training lengths of 19, 36, 67, and 92 bits, respectively, such that the TIR/training-length pairs lead to the same training-to-information overhead. It is seen from Fig. 3 that superimposed training results improve with decreasing TIR (smaller SNR loss in training) until one reaches a “threshold” of 0.05 when initial (first step) channel estimates are not good enough to lead to a substantial improvement via iterative enhancements. On the other hand, conventional training works for all
B. Using VA in CML Algorithm

We now repeat the example in Section V-A but use the iterative CML approach with VA for signal detection. [The various parameters are as in the example in Section V-A.] The results corresponding to Figs. 1 and 2 are now shown in Figs. 4 and 5. The results based on VA are superior to that based on LMMSE equalizers. The comments made regarding Figs. 1 and 2 apply to Figs. 4 and 5 also.

C. Performance Analysis

We again consider the example in Section V-A except that the information sequence is now either binary PSK (BPSK) or QPSK, whereas superimposed training remains binary. Given the channel estimate and the true channel at the $i$th Monte Carlo run as $\hat{h}^{(i)}(l)$ and $h^{(i)}(l)$, respectively, the channel m.s. error (CMSE) is now defined as

$$CMSE := \frac{1}{M} \sum_{i=1}^{M} \sum_{l=0}^{2} \| \hat{h}^{(i)}(l) - h^{(i)}(l) \|^2$$

(52)
Fig. 5. As in Fig. 4 except that NCMSE as defined in (51) is shown.

Fig. 6. Comparison of channel MSE based on simulations and analytical results at various stages of iterative channel estimation and data detection when Viterbi detector is used. [Simulation channel MSE is given by (52).] BPSK signal. Training-to-information symbol power ratio TIR = 0.2. Record length = 400 bits. Results based on 500 Monte Carlo runs.

Fig. 7. Comparison of MSE based on simulations and analytical results at various stages of iterative channel estimation and data detection when Viterbi detector is used. [Simulation channel MSE is given by (52).] QPSK signal. Training-to-information symbol power ratio TIR = 0.2. Record length = 400 symbols. Results based on 500 Monte Carlo runs.

shown in Fig. 6 for BPSK signals and in Fig. 7 for QPSK signals at various stages of our iterative approaches. The analytical results shown are based on the theoretical value $\sigma^2_\hat{h}$ given by (44) or (50), where the value of $p_e$ was obtained from our simulation averages (under various SNRs). The analytical results for $p_e = 0$ correspond to the case where the entire information sequence plus the superimposed training acts as conventional training, thus providing a lower bound to the performance (including that of time-multiplexed conventional training). [Note that the first step results in Figs. 6 and 7 are based on the first-order statistics of the data and are taken from [13].] It is seen from Figs. 6 and 7 that by the second iteration of the proposed scheme (CML method exploiting the VA), the agreement between the theoretical and simulation results is quite good for both BPSK and QPSK information signals for SNRs of 15 dB or higher. One expects the assumptions such as (H5)–(H7) underlying the performance analysis in Section IV to be better satisfied in the later iterations when $p_e$ is likely to be “small.”

D. Comparisons With [4] and [5]

We again consider the example in Section V-A with BPSK information sequence and a record length of 399 bits. Similar to [5], we remove the mean value of the training sequence so that no power is wasted at zero frequency. Consider

$$\tilde{c}(n) := c(n) - \frac{1}{P} \sum_{k=0}^{P} c(k)$$

with $c(n)$ as an $m$-sequence with period $P = 7$. The superimposed training sequence for our approaches is taken to be a scaled version (to achieve desired TIR) of $\tilde{c}(n)$. For [5], we followed the recommendation given therein by picking a sequence satisfying Lemma 2 in [5] with (in the notation of [5]) $P = 3 (= L + 1)$, $Q = 133$, and $t = 1$. All superimposed training sequences were scaled to achieve
yields significantly lower BER compared to that in [4] and [5] but with other hand, the proposed iterative approach with the Viterbi detector with an order of magnitude lower computational requirement. On the LMMSE equalizer yields almost the same BER as in [4] and [5] but using LMMSE equalizers and Viterbi detector, respectively, in our proposed iterative algorithms; the approach in [5] is the same in the two figures. Table I shows the CPU time per run averaged over seven SNRs and 500 runs per SNR for the various cases [MATLAB 7.0.4 on a 2.8 GHz Pentium-based computer operating under Windows XP]. It is seen that the proposed iterative approach with LMMSE equalizer yields almost the same BER as in [4] and [5] but with an order of magnitude lower computational requirement. On the other hand, the proposed iterative approach with the Viterbi detector yields significantly lower BER compared to that in [4] and [5] but with three to five times higher computational requirements.

VI. CONCLUSION

The approach in [12] to SIMO channel estimation using superimposed training sequences (hidden pilots) and first-order statistics was extended to iterative joint channel estimation and data detection, where the information sequence is exploited to enhance performance instead of being viewed as interference. The results were illustrated via a simulation example involving frequency-selective quasi-static Rayleigh fading. In the considered uncoded case, the proposed methods are competitive with the conventional training method without incurring any information rate loss. Considering channel coding would ultimately determine whether the bandwidth gain of the superimposed training-based methods could compensate for their SNR performance loss. Furthermore, we did not consider time-varying channels where one would send time-multiplexed training frequently and periodically to keep up with the changing channel, thereby incurring a higher training overhead. In the quasi-static case considered in this paper, the data block length can be made quite large relative to the time-

![Fig. 9. BER. As in Fig. 8 except that the proposed iterative approach uses Viterbi detector.](Image)