Abstract—In this paper, we present different linear and nonlinear iterative data detection schemes for the asynchronous direct-sequence code-division multiple access (DS-CDMA) systems employing orthogonal signalling formats and long scrambling codes. Compared to the conventional receiver and other noncoherent multiuser detectors, coherent multiuser detection schemes achieve much better performance provided that the channels are accurately estimated. To this end, we propose several channel estimation algorithms to estimate multipath Rayleigh fading channels. Different data detection and channel estimation schemes are compared in terms of the BER performance. Based on the numerical results, some recommendations are made on how to choose multiuser detectors and channel estimation algorithms in practical CDMA systems.

I. INTRODUCTION

The system under study is an asynchronous DS-CDMA system with orthogonal modulation. The transmitted chip sequence from a particular user is the concatenation of one of $M$ possible Walsh sequences (representing the transmitted symbol) and a long scrambling code. Multiuser detection for $M$-ary orthogonal modulation in DS-CDMA systems has been the subject of study in several papers. For instance, parallel and successive interference cancellation were presented in [1], [2]. The interference is estimated and subtracted from the received signal before detection is done. Iterative schemes for demodulating $M$-ary orthogonal signalling formats in DS-CDMA systems were proposed in [3], [4], using nonlinear MMSE and PIC, respectively. Time-varying Rayleigh fading channel is assumed in those papers, necessitating channel estimation for effective interference cancellation. However, relatively little work has been done in exploring linear receivers for the system in question. In this paper, we present some linear schemes suitable for demodulating DS-CDMA signals with orthogonal modulation and compare their performance with nonlinear receiver, e.g., PIC. We start with an approximate maximum likelihood linear receiver, then derive some other algorithms based on interference suppression rather than interference cancellation. With interference suppression we mean that the receiver tries to filter out the estimated interference from the received signal. This is in contrast to cancellation techniques when the estimated interference is subtracted from the received signal.

The justification for using suppression instead of cancellation is that an erroneously estimated symbol will lead to a doubling of the interference when using cancellation. However, when using suppression, an erroneously estimated symbol will cause the suppression of a non-existing signal. This will lead to some suppression of the desired signal, but the overall penalty may be less than in the cancellation case.

It is well-known fact that coherent detection gives better performance than noncoherent detection when the channels are accurately estimated. Furthermore, to do interference cancellation we need knowledge of the complex channel gains. In this paper, we propose several methods for estimating the channels which, together with coherent detection, is shown to yield significant performance gains in time-varying multipath channels.

The remainder of the paper is organized as follows. In Section II, the system model is presented. Data detection and channel estimation algorithms are introduced in Section III and IV. Different algorithms are compared in Section V and conclusions are drawn in Section VI.

II. SYSTEM MODEL

Fig. 1 shows the signal path for the $k$th user. The $k$th user’s $j$th symbol is denoted by $s_k(j) \in \{1,2,\ldots,M\}$, and the $M$ orthogonal signal alternatives are defined by $M$ Walsh sequences $\{w_1(n), \ldots, w_M(n)\}$ with support $n \in \{1,2,\ldots,N\}$, where $N \geq M$. The Walsh chips are randomized by a scrambling code $c_k(n) \in \{-1,+1\}$. Hence, every symbol is spread by $N$ chips and each bit by $N/\log_2 M$ chips.

The transmitted chip sequence from the $k$th user, $a_k(n)$, is the product of the scrambling code $c_k(n)$ and the Walsh chips, i.e.,

$$a_k(n) = c_k(n) \sum_j w_{kj}(n-jN).$$

The baseband signal $s_k(t)$ is formed by pulse amplitude modulating $a_k(n)$ with the unit-energy chip waveform $\psi(t)$, i.e.,

$$s_k(t) = \sum_n a_k(n) \psi(t-nT_c),$$

where $T_c$ is the chip duration and $T = NT_c$ is the symbol duration. For simplicity, we assume that $\psi(t)$ is a rectangular pulse with support $t \in [0, T_c]$; however, the proposed methods can be extended for other waveforms, e.g., square-root raised cosine pulses.

The baseband signal is multiplied with a carrier and transmitted over a Rayleigh fading channel with $L_k$ resolvable paths with time-varying complex channel gains $h_{k1}(t), h_{k2}(t), \ldots, h_{kL_k}(t)$ and delays $\tau_{k1}, \tau_{k2}, \ldots, \tau_{kL_k}$. We assume, without loss of generality, that $\tau_{k1} < \tau_{k2} < \ldots < \tau_{kL_k}$. The received signal is the sum of all users’ contributions plus additive white Gaussian noise with power spectral density $N_0/2$. The passband signal, $r_B(t)$, is formed according to Fig. 1, and the complex envelope $r(t)$ of the received signal can be written as

$$r(t) = n(t) + \sum_{k=1}^{K} \sum_{i=1}^{L_k} h_{ki}(t-\tau_{ki}) s_k(t-\tau_{ki})$$

The passband signal, $r_B(t)$, can be written in terms of the complex envelope as $r_B(t) = \sqrt{2} \Re \left\{ r(t) e^{j\omega_0 t} \right\}$, where $\omega_0$ is the carrier frequency.
The columns of the product of the channel gain and a shifted version of the approach and shifted version of chip sequence transmitted by the user's path signals and the thermal noise. The vector $\mathbf{h}(j)$ consists of contributions from all users' signals. The channel vector $\mathbf{h}(j) \in \mathbb{C}^{n \times 1}$ is defined as

$$\mathbf{h}(j) = [h_{1}(j) \ h_{2}(j) \ \cdots \ h_{K}(j)]^{\top},$$

where $h_{k}(j)$ are the contributions from each user's path and is the the unity vector $\mathbf{1}$. Each column of the matrix $\mathbf{H}(t)$ contains intersymbol interference (ISI). The contribution only from the $k^{th}$ user's $j^{th}$ symbol can be written as $\mathbf{X}_{k}(j)$, containing $S_{k}(j)$ and the contribution excluding ISI is $\mathbf{X}_{k}(j)$.

As seen from Fig. 2, $y(t_{c})$ consists of contributions from all users' path signals and the thermal noise. The vector $\mathbf{h}(j)$ is a vector of the noise samples $v(t_{c})$. Each column of the matrix $\mathbf{H}(t)$ represents the contribution from each path and is the product of the channel gain and a shifted version of the appropriate user's chip sequence (the shift is due to the path delay). The columns of $\mathbf{H}(t)$ are weighted together by $\mathbf{h}(j)$, whose elements are the path gains of all users' paths. From Fig. 2, we see that $\mathbf{r}(k,j)$ can be written as the sum of four terms: the signal of interest, the intersymbol interference (ISI), the multiple access interference (MAI), and the noise. The signal of interest is the part of $y(t_{c})$ that is due to the $k^{th}$ user's $j^{th}$ symbol. In Fig. 2, the signal of interest for user 1 is marked with bold lines. The columns of the matrix $\mathbf{X}_{k}(j)$ are essentially the shifted versions of the dips due to the $k^{th}$ user's $j^{th}$ symbol, one column per path. The columns of $\mathbf{X}_{k}(j)$ are weighted together by the vector $\mathbf{h}(j)$, whose elements are the path gains of the $k^{th}$ user's paths.

The matrix $\mathbf{A}(k,j) \in \mathbb{R}^{n \times 4t}$, that is due to the $k^{th}$ user's $j^{th}$ symbol. The contribution to $\mathbf{r}(k,j)$ due to the $k^{th}$ user, including ISI, is $\mathbf{S}_{k}(j)1_{L_{k}}$, and the contribution excluding ISI is $\mathbf{X}_{k}(j)\mathbf{h}(j)$.

### III. Multiuser Detection Algorithms

The task of the receiver is to detect the symbols from all users, i.e., detect $i_{k}(j)$ for $k = 1, 2, \ldots, K$ given the observation $\mathbf{r}(k,j)$. The decision on the $k^{th}$ user's $j^{th}$ symbol, $\hat{i}_{k}(j)$ is found as

$$\hat{i}_{k}(j) = \arg \max_{m \in \{1, 2, \ldots, M\}} z_{k}(m),$$

where $z_{k}(m)$ is the soft decision.

The conventional detection technique is to form the soft decision by correlating the received signal with the $M$ possible transmitted waveforms. In the absence of an estimate of the fading
processes, we do this in a path-by-path noncoherent manner, and the soft decision is

\[ z_k(m) = \frac{\sum_{j=1}^{2^m} |x_{k,l,m}(j)r(k,j)|^2}{\|x_{k,l,m}(j)\|^2} \]  

(5)

where \( x_{k,l,m}(j) \) is the shifted chip sequence from the \( k^{th} \) user based on the hypothesis that the \( m^{th} \) Walsh symbol \( m = 1, 2, \ldots, M \) is transmitted. The shift is determined by the path delay. The vector \( x_{k,l,m}(j) \) is the \( j^{th} \) column corresponding to the \( k^{th} \) user’s \( l^{th} \) path in the matrix \( X_{k,l,m}(j) \), which is defined as

\[ X_{k,l,m}(j) = [x_{k,1,m}(j) \ldots x_{k,l,m}(j) \ldots x_{k,L,m}(j)] \]

Although optimum in the single user AWGN channel, the conventional receiver has poor performance in multiuser environment. In the following, we will describe how the soft decisions are formed for different multiuser detectors.

### A. Approximate Maximum Likelihood Multiuser Detector

In our case, the noise is complex Gaussian. Given the received observation \( r(k,j) \), the log-likelihood function of the received vector conditioned on a realization of the fading channels can be expressed as

\[ \text{constant} \cdot \frac{1}{N_0} ||r(k,j) - A(k,j)h(j)||^2 \]

From the above equation, one can see that maximization of this log-likelihood function is equivalent to minimization of the function

\[ ||r(j) - A(k,j)h(j)||^2 \]  

(6)

For simplicity of notation we will suppress explicit dependence on \( k \) and \( j \) of the various quantities whenever no ambiguity arises. In particular, we will use \( r, n, A, X_{k,m}, \) and \( h_k \) to denote \( r(k,j), n(k,j), A(k,j), X_{k,m}(k,j), \) and \( h(j) \), respectively.

It is shown in [4] that the ML estimate of the fading channel vector \( h \) is \( \hat{h} = A^+r \), where \( A^+ = (A^HA)^{-1}A^\dagger \) denotes the left psuedoinverse of \( A \). Substituting \( \hat{h} \) into (6), we derive the ML estimation of the data matrix \( A \) as the minimizer of

\[ ||r - Ah||^2_{h=A^\dagger r} = ||r - AA^\dagger r||^2 = ||(I - AA^\dagger)r||^2 = ||P_{A^\dagger}r||^2 \]

\[ = \text{trace}||P_{A^\dagger}r||^2 = \text{trace}\{P_{A^\dagger}^*P_{A^\dagger}r\} = \text{trace}\{r^*P_{A^\dagger}P_{A^\dagger}r\} \]

\[ = \text{trace}\{r^*P_{A^\dagger}r\} = \text{trace}\{P_{A^\dagger}P_{A^\dagger}r\} \]

where \( P_{A^\dagger} = I_N - AA^\dagger \) is the null space of \( A \), the matrix \( \hat{A} \) is the sample covariance matrix defined as \( \hat{A} = rr^* \). The ML detector can then be expressed as

\[ i_k^{\text{ML}} = \arg \min_{\hat{A}} \text{trace}\{P_{A^\dagger}P_{A^\dagger}r\} \]

(7)

This algorithm needs the estimate of the whole matrix \( A \) at a time. Under the assumption \( N_k < 2N \) for all \( k \), the matrix \( A \) is a function of \( i_k(j); i_l(n), l = 1, \ldots, k-1, k+1, \ldots, K; n = j-1, j, j+1 \). Therefore, the computational complexity of the ML detector defined by (7) grows exponentially with the number of users \( K \). A sub-optimum solution that reduces the complexity from exponential to polynomial is the multistage ML detector. We use the conventional detector for the first stage to get initial estimates of the transmitted data. Then the ML algorithm switches to the decision directed mode. To detect the data transmitted by the \( k^{th} \) user at the \( n^{th} \) stage, we replace \( A \) in equation (6) with \( \hat{A} \) defined as

\[ \hat{A} = [\hat{A}_1^{(n-1)} \ldots \hat{A}_k^{(n-1)} \ldots \hat{A}_K^{(n-1)}] \]

(8)

where the matrices \( \hat{A}_1^{(n-1)}(j), \hat{A}_2^{(n-1)}(j), \ldots, \hat{A}_K^{(n-1)}(j) \) are estimated MAI at the \( k^{th} \) user’s \( j^{th} \) symbol interval. They are obtained by substituting \( \hat{h}_i^{(n-1)} \), \( i = 1, \ldots, k-1, k+1, \ldots, K \) into equations (1), (3), and we assume the propagation delays are known.

Let us denote \( \hat{A} \) as the MAI matrix formed as \( \hat{A} = [\hat{A}_1 \ldots \hat{A}_{k-1} \hat{A}_{k+1} \ldots \hat{A}_K] \). The matrix \( \hat{A} \) is similar to \( \hat{h} \) except that the columns due to the \( k^{th} \) user are deleted. Similarly, we delete the estimate of the \( k^{th} \) user’s channel vector \( \hat{h}_k \) from \( \hat{h} \) and form the channel vector \( \hat{h}_k \) corresponding to the MAI:

\[ \hat{h} = [\hat{h}_1 \hat{h}_2 \ldots \hat{h}_{k-1} \hat{h}_{k+1} \ldots \hat{h}_K] \]

The expression of the approximate ML algorithm — equations (6) and (8) can
where $U$ knowing or estimating the fading. That leads to the noncoherent iteration stage.

The standard receiver [as defined by (5)]. We then iterate for $n \in \{4\}$. We can conclude that the PIC is an approximate ML algorithm.

Consider the matrix $U \in \mathbb{R}^{k \times (k \times l - k)}$ defined as

$$
U = [A_1 \cdots A_{k-1} A_{k+1} \cdots A_K] \quad (9)
$$

Hence, $U$ is formed from $A$ by deleting the columns that are due to the $k^{th}$ user. We can suppress the interference by projecting $r$ on the null space of $U$ which can be computed as $P^U_1 = I - UU^T$, where $U^T = (U^*U)^{-1}U^*$ denotes the left pseudoinverse of $U$. This implies that $P^U_1 A_i = 0$ for all $i \neq k$, and thus

$$
P^U_1 r = P^U_1 [Ah + n] = \sum_{i=1}^k P^U_1 A_i h_i + P^U_1 n = P^U_1 A_i h_i + P^U_1 n
$$

Note that we can compute estimates $\hat{A}_k$ of $A_k$ by substituting $\hat{\eta}_k$ into (1) and (3). We can do interference suppression without knowing or estimating the fading. That leads to the noncoherent version of IIS receiver. Let $\hat{\eta}_k^{(n)}$ denote the data estimates after the $n^{th}$ iteration, and $\hat{\eta}_k^{(1)}$ be the data estimates obtained from the standard receiver [as defined by (5)]. We then iterate for $n = 2, 3, \ldots, N_{\text{iter}}$

$$
z_k^{(n)}(m) = \sum_{i=1}^{k} \left( |x_k,i,m |^2 P_{(n-1)}^U r - \frac{1}{2} \left\| P_{(n-1)}^U x_k,i,m \right\|^2 \right) \quad (10)
$$

where $\hat{U}^{(n-1)}$ is estimated interference matrix at the $(n - 1)^{th}$ iteration stage.

If an estimate of the fading vector $h$ is available, the MAI matrix $U$ can be formed in different ways as follows

$$
U = [s_1 \cdots s_{k-1} s_k] \in \mathbb{C}^{N_k \times (k-1)} \quad (11)
$$

$$
U = [s_1 \cdots s_{k-1} s_k] \in \mathbb{C}^{N_k \times (k_{\max} - k)} \quad (12)
$$

$$
U = [s_1 \cdots s_{k-1} (s_k - X_i h_i) s_{k+1} \cdots s_k] \in \mathbb{C}^{N_k \times K} \quad (13)
$$

where $s_k = S_i I_k$. is the contribution from the $i^{th}$ user. A coherent version of the IIS receiver can now be formulated as

$$
z_k^{(n)}(m) = \text{Re} \left\{ \hat{h}_i^* X_{k,m}^* P_{(n-1)}^U r \right\} - \frac{1}{2} \left\| P_{(n-1)}^U X_k,m \hat{h}_i \right\|^2 \quad (14)
$$

Among the different ways of constructing suppression filter, we found out from experiments that the one constructed from the null space of (12) gives the best performance, and is therefore employed in our simulations.

C. Whitened Matched Filter (WMF)

As we know, the conventional matched filter achieves the best performance in the AWGN single user channel or in strict orthogonal synchronous channel. It’s not a good choice for multiuser detection in which interference must be taken into account in addition to the white Gaussian noise. Interference combined with Gaussian noise does not have a Gaussian distribution. A way to work around this problem is to whiten the combined interference and noise, which can be achieved by preprocessing the received vector $r$ with the matrix $R^{-1/2}$. The whitened vector $W = R^{-1/2} r$ has a “white” correlation matrix $E[WW^T] = R^{-1/2}RR^{-1/2} = I$. The matrix $R^{-1/2}$ is obtained from $R$ by Cholesky factorization [5], and $R$ is the correlation matrix of $r$ and can be estimated as $R = \hat{A}h^*h + N_0 I_N$.

This idea leads to another kind of linear interference suppression technique, namely, the whitened matched filter (WMF). A coherent WMF can be formed by correlating the whitened received vector with each candidates vector $\{R^{-1/2} X_k,m \hat{h}_i\}$

$$
z_k^{(n)}(m) = \text{Re} \left\{ \left( \hat{\hat{R}}^{(n-1)} X_k,m \hat{h}_i \right)^* \hat{R}^{(n-1)} r \right\} - \frac{1}{2} \left\| \hat{R}^{(n-1)} X_k,m \hat{h}_i \right\|^2
$$

$$
= \text{Re} \left\{ \hat{h}_i^* X_{k,m}^* \hat{R}^{(n-1)} r \right\} - \frac{1}{2} \hat{h}_i X_{k,m}^{*} \hat{R}^{(n-1)} X_{k,m} \hat{h}_i \quad (15)
$$

IV. ESTIMATION OF FADING PROCESSES

From above we know that the coherent data detection requires estimate of the fading processes, i.e., an estimate of $h$. Several channel estimation algorithms are presented in this section. All of them are decision-directed and can be inserted into the coherent iteration loops in the previous section. The estimation procedure at the $n^{th}$ iteration uses the data estimates from the previous stage, i.e., $\hat{\eta}_k^{(n-1)}$.

A. Maximum Likelihood Channel Estimation

Given estimates of the transmitted data, we can estimate $h$ [4] as $\hat{h} = \hat{A}^* r$. We obtain $\hat{A}$ as before by substituting $\hat{\eta}_k^{(n-1)}$ into (1) and (3). Recall that $r = Ah + n$, in the case of correct decisions ($A = \hat{A}$), then $\hat{h} = \hat{A}^* Ah + A^* n = h + A^* n$ which is an unbiased estimate of $h$.

This procedure will suffer from a so-called dimensionality problem. When $L_{\alpha} \geq N_{i}$, the matrix $\hat{A}$ will not have full column rank and the above mentioned procedure will become useless. The problem can be resolved by stacking several $r(k,j)$ vectors on top of each other and assume channel remains static during several symbol intervals [4]. In particular, suppose $h(k,j) = \alpha^2$

3The matrix $R$ is symmetric positive definite as long as $N > M$ and $N_0 > 0$. Hence, there exist a symmetric positive definite matrix $Q$ such that $QQ^T = R$, and we will use the notation $Q = R^{1/2}$, $Q^{-1} = R^{-1/2}$, $Q^* = R^{*1/2}$, and $(Q^{-1})^* = R^{-1/2}$.
\( h(k, j) \), we can then write
\[
\begin{bmatrix}
  r(k, j) \\
  r(k, j + 1)
\end{bmatrix} = \begin{bmatrix}
  A(k, j) & 0 \\
  0 & A(k, j + 1)
\end{bmatrix} \begin{bmatrix}
  h(k, j) \\
  h(k, j + 1)
\end{bmatrix} + \begin{bmatrix}
  n(k, j) \\
  n(k, j + 1)
\end{bmatrix}
\]
\[
\approx \begin{bmatrix}
  A(k, j) \\
  A(k, j + 1)
\end{bmatrix} h(k, j) + \begin{bmatrix}
  n(k, j) \\
  n(k, j + 1)
\end{bmatrix}
\]
Hence, we can estimate \( h(k, j) \) as
\[
\hat{h}(k, j) = \left[ \begin{bmatrix}
  A(k, j) \\
  A(k, j + 1)
\end{bmatrix} \right]^\dagger \begin{bmatrix}
  r(k, j) \\
  r(k, j + 1)
\end{bmatrix}
\] (16)
which will produce usable estimates as long as \( 2N_r > L_{tot} \). Obviously, this scheme can be extended further by stacking several \( r(k, j) \) vectors on top of each other; however, the quality of the estimates may be reduced, especially for relatively fast fading channels.

The estimation results can be further enhanced by applying smoothing operation on the original channel estimates [4].

B. Linear MMSE Channel Estimation

The LMMSE algorithm for channel estimation computes a matrix \( W \), which is chosen to minimize the mean square error \( \mathbb{E}[\| h - W r \|^2] \). The optimum matrix of \( W \) under LMMSE criterion can be computed as
\[
W_{\text{mmse}} = \arg \min_W \mathbb{E}[\| h - W r \|^2] = R^{-1} \Phi
\]
\[
R = \mathbb{E}[rr^\ast] = A \mathbb{E}[hh^\ast] A^\ast + N_0 I = APA^\ast + N_0 I
\]
\[
\Phi = \mathbb{E}[hh^\ast] = \mathbb{E}[(Ah + n)h^\ast] = A \mathbb{E}[hh^\ast] = AP
\]
\[
P = \text{diag}(P_{1,1}, P_{1,2}, \cdots, P_{K,1}, \cdots, P_{K,K})
\]
where \( P_{k,l} \) is the received power from the \( k^{th} \) user’s \( l^{th} \) path. Combining the above equations, the linear MMSE estimate of \( h \) can be formulated as
\[
\hat{h} = W_{\text{mmse}}^\ast r = \Phi^\ast R^{-1} r = \hat{P} \hat{A}^\ast (\hat{A} \hat{P} \hat{A}^\ast + N_0 I)^{-1} r
\] (17)
From (17), we see that the signal and noise power level \( P_{k,l} \) and \( N_0 \) must be known or estimated to carry out the LMMSE channel estimation.

V. NUMERICAL RESULTS

The simulated system is a \( K = 12 \)-user system. Each user transmits one of \( M = 8 \) Walsh codes spread to a total length of \( N = 64 \) chips. The effective spreading of the system is \( N / \log_2 M = 64 / 3 \) chips per bit. Different users are separated by different scrambling codes \( c_k(n) \) which are random, and differ from symbol to symbol (long-code system).

For simplicity, the simulated system is assumed to be chip-synchronous, i.e., all path delays are assumed to be multiples of \( T_c \). However, the system is asynchronous on the symbol level. The normalized Doppler frequency is assumed to be \( f_D T = 0.01 \). Perfect slow power control is assumed in the sense that \( P_k = \sum_{l=1}^{L_k} P_{k,l} \), the average received power, is equal for all users. The system was simulated for single-path channels, \( L_k = L = 1 \) and 3-path channels, \( L_k = L = 3 \) for all \( k \). In the case when \( L = 3 \),
the spacing between the three paths of each user is set to $2T_c$.

The simulation results are averaged over random distributions of fading, noise, delay, and scrambling code through numerous Monte-Carlo runs.

The performance of different multiuser detectors is compared in Fig. 3. The results are obtained after three iteration stages, initialized with conventional noncoherent first stage. The noncoherent IIS doesn’t show much improvement compared to the conventional receiver, while other coherent schemes with the ML channel estimation and smoothing achieve considerable performance gain. We can also see that PIC performs better than IIS and WMF. Considering the fact that when long spreading (scrambling) codes are used, which is the case in most of the practical systems, the linear filters (e.g., IIS, WMF) update at a symbol rate, the inverse of $U$ in (14), $R$ in (15) has to be recalculated every time, which significantly increases the computational complexity. On the contrary, PIC does not deal with any matrix inversion, thus considerably reduces the complexity compared to linear detectors. That makes PIC a more attractive detection algorithm in long-code CDMA systems.

The multistage approximate ML detector presented in Section III-A is not simulated. It, however, should have the same performance as PIC, since we have proved in Section III that these two schemes finally converge, although they detect data in different approaches (AML detection is linear scheme, while PIC is nonlinear). The advantage of using ML detection is the avoidance of estimating fading process, which is needed by PIC. While the performance of PIC can be improved considerably when combined with channel smoothing, which is not possible for the ML detector.

The upper plots of Fig. 4 shows the original fading channel and the results of channel estimation with different schemes. The original estimates are noisy due to the noisy observations. The quality of the estimated channel is greatly improved after applying the smoothing operation using an FIR filter with impulse response $g(n)$ which is a (non-causal) Hamming window of length 19, normalized such that $\sum g(n) = 1$. The impulse response is plotted at the bottom left corner of Fig. 4. In case of LMMSE estimator, we assume the signal and noise power levels are known.

Different channel estimators combined with the smoothing filter are compared in Fig. 5 in terms of BER performance when applied to 3-stage PIC and IIS. One can observe that the ML and LMMSE algorithms yield identical results. The PIC and IIS are also simulated with perfect channel estimates, i.e., when $h_k = \hat{h}_k$ (which we call genie-aided PIC/IIS) to see how close the performance of proposed channel estimators is to the ideal one. As expected, the genie-aided IIS always outperforms the IIS with channel estimates. To our surprise, the genie-aided PIC performs worse than the PIC with channel estimates in high SNR region. The reason might be that data detection error has greater detrimental effects to the genie-aided PIC due to larger erroneous cancellation (doubling the interference). While in low SNR region, noise is dominant, the channel is closer to single user channel. In this case, maximum ratio combining with perfect knowledge of the channel will naturally lead to better performance than that with imperfect channel estimates.

Fig. 6 shows the performance of multistage PIC with ML channel estimator and smoothing filter for single-path (flat fading) and 3-path (frequency selective fading) respectively. The iterations are initialized with a conventional noncoherent stage. One can observe that the gap between PIC and conventional receiver becomes larger as SNR increases, which means we need to maintain a reasonable level of signal to noise ratio in order to achieve considerable capacity gain by applying multiuser detectors. Multistage PIC in 3-path case achieves lower error probabilities compared to single-path case due to diversity gain. However, in both cases, their performance gets saturated after 3 or 4 iterations.

VI. CONCLUSIONS

In this paper, we compared the performance of different linear, nonlinear multiuser detectors and channel estimators. The conclusion is that nonlinear interference cancellation scheme like 3 or 4-stage PIC is better choice in practical systems with long spreading codes, in consideration of both performance and computational complexity. We proved that PIC is an approximate ML approach to data detection. In case of perfect cancellation, it is optimum multiuser detector in the sense of maximum likelihood data detection. The knowledge of the channel is essential to increase the system capacity. Coherent detection gives better performance than noncoherent detection when the channel gains are accurately estimated.

REFERENCES


