Correction of Extrinsic Information for Iterative Decoding in A Serially Concatenated Multiuser DS-CDMA System

Pei Xiao (Member, IEEE)

School of Electrical, Electronic and Computer Engineering
Univ. of Newcastle, Newcastle Upon Tyne, NE1 7RU, United Kingdom

E-mail: pei.xiao@ncl.ac.uk

Erik G. Ström (Senior Member, IEEE)

Communication Systems Group, Dept. of Signals and Systems
Chalmers Univ. of Technology, SE-412 96 Göteborg, Sweden

E-mail: erik.strom@s2.chalmers.se

Abstract—The system under study is a coded asynchronous DS-CDMA system with orthogonal modulation in time-varying Rayleigh fading multipath channels. Information bits are convolutionally encoded, block interleaved, and mapped to M -ary orthogonal Walsh codes, where the last step is essentially a process of block coding. This paper aims at tackling the problem of joint iterative decoding of this serially concatenated inner block code and outer convolutional code and estimating frequency-selective fading channels in multiuser environments. The (logarithm) maximum a posteriori probability, (Log)-MAP criterion is used to derive the iterative decoding schemes. In our system, the soft output from inner block decoder is used as a priori information for the outer decoder. The soft output from outer convolutional decoder is used for two purposes. First, it may be fed back to the inner decoder as extrinsic information for the systematic bits of the Walsh codeword. Secondly, it is utilized for channel estimation and multiuser detection (MUD). We also show that the inner decoding can be accomplished without extrinsic information, and in some cases, e.g., when the system is heavily loaded, yields better performance than the decoding with unprocessed extrinsic information. This implies the need for correcting the extrinsic information obtained from outer decoder. Different schemes are examined and compared numerically, and it is shown that iterative decoding with properly corrected extrinsic information or with non-extrinsic/extrinsic adaptation enables the system to operate reliably in the presence of severe multiuser interference, especially when the inner decoding is assisted by decision directed channel estimation and interference cancellation techniques.

Index terms: DS-CDMA, iterative decoding, extrinsic information, channel estimation, multiuser detection.

I. INTRODUCTION

Turbo codes represent an important advancement in the area of power efficient communications. The practical importance of turbo codes stems from the fact that they enable reliable communications at signal-to-noise ratios close to the channel capacity with simple component codes, yet admit high performance iterative soft decoding algorithms with complexity not significantly higher than that of the decoder for single constituent code.

In a conventional communication receiver, only bits, or hard-decisions are passed between the subsystems. Information is lost and becomes unavailable to the subsequent stages whenever hard-decisions are made. Also, preceding stages can not benefit from the information derived by the following stages. The interface between each subsystem can be greatly improved by applying “turbo processing principle,” which was first employed for decoding parallel concatenated convolutional codes, known as Turbo codes. With turbo processing, each subsystem is implemented with a Soft-Input, Soft-Output (SISO) algorithm, such as MAP or Log-MAP. Soft decision values, typically in the form of log-likelihood ratios (LLRs), are passed down the chain and refined by the subsequent stages. The soft output of the final stage is then fed back to the first stage and a second iteration of the processing is initiated. Several iterations of turbo processing can be executed to improve performance.

The turbo principle is a general strategy of iterative feed-back decoding or detection [1], and can be used in a more general way than just for the decoding of parallel concatenated convolutional codes. It has been successfully applied to many detection/decoding problems such as serial concatenation, equalization, coded modulation, multiuser detection, joint source and channel decoding and others [2].

In this paper, we study iterative decoding of a serially cascaded asynchronous CDMA system which involves convolutional coding and orthogonal modulation. The orthogonal modulation is accomplished with Walsh (Hadamard) code which is suitable for combining the advantages of spreading and coding to achieve improved performance for spread spectrum (CDMA) systems. Convolutional codes are employed to further improve the performance and power efficiency of the system. It is believed that CDMA systems exhibit their full potential, when combined with forward error correction coding (FEC) [3].

The problem of iterative decoding for serially concatenated codes (consisting of inner code and outer codes) has been addressed e.g., in [4] for serially concatenated convolutional codes and in [5], [6] for serially concatenated block code

and convolutional code. Analogous to the decoding of turbo codes, the inner decoder extracts the soft information from the outer decoder to update and improve its soft decision on code bits. The inner decoder also provide the outer decoder with soft unquantized decisions to improve performance. The process of passing soft information between two SISO stages proceeds, and after a few iterations, the information data are decoded with a hard decision at the output of the outer decoder. In [6], [7], MAP demodulator and SOVA (soft-output VA) decoder were applied to a similar system using M-ary modulation and FEC. A performance gain of about 0.6dB at a bit error rate (BER) of 10^{-3} was noticed for single user system in AWGN channel when compared against the conventional non-SISO demodulator and decoder. It was indicated in [7] that interleaver design has significant impact on the system performance. However, some important issues e.g., channel estimation and MAI mitigation were not addressed in the above references.

In addition to FEC, multiuser detection (MUD) is another effective tool to increase the capacity of interference limited CDMA systems. Several iterative MUD schemes were proposed, e.g., in [8], [9] for uncoded M-ary orthogonal systems. In order to fully explore the potential of multiuser detectors, we need to acquire accurate measurements of the fading channel to do coherent detection or interference cancellation. It is shown in the above papers that the use of iterative multiuser detection (interference cancellation) with decision-directed channel estimation provides substantial capacity gains compared to the conventional receiver.

The problem of joint multiuser detection and decoding was treated, e.g., in [10]–[12]. Soft interference cancellation, linear MMSE filtering, or trellis based Log-MAP multiuser detection, etc. were proposed in those papers to reduce the deteriorative effect of interference before single user decoding was done. However, the algorithms developed in the above papers are confined to uncascaded systems with single convolutional code, and the issue of joint detection/decoding and channel estimation is not investigated, except in [11] where a soft input MMSE channel estimation algorithm was proposed.

The contribution of this paper is the treatment of joint multiuser detection, decoding and channel estimation by utilizing turbo processing principle for the systems in question. The iterative decoding is assisted by decision directed channel estimation and interference cancellation to effectively combat interference. Some correction and adaptation algorithms are proposed to better utilize the extrinsic information in bad channels (severe multiuser interference environment). The remainder of the paper is organized as follows. In Section II, we introduce the system model. The algorithms for inner and outer decoding, interference mitigation and channel estimation are discussed in Section III and IV. Their performance is numerically evaluated and compared in Section V. Extrinsic correction algorithms are proposed in Section VI. Conclusions are drawn in Section VII.

II. SYSTEM MODEL

The system model is only briefly described in this section. For a more detailed description, readers are referred to [8].

TABLE I
MAPPING BETWEEN INPUT BITS, SYMBOL INDICES, AND WALSH
CODEWORDS

Code bits	Index	Walsh codeword
$u_k^{[0]} u_k^{[1]} u_k^{[2]}$	$m = i_k(j)$	$\mathbf{w}_m$
+1 +1 +1	0	$\mathbf{w}_0 : +1 +1 +1 +1 +1 +1 +1 +1$
+1 +1 -1	1	$\mathbf{w}_1 : +1 +1 +1 +1 -1 -1 -1 -1$
+1 -1 +1	2	$\mathbf{w}_2 : +1 +1 -1 -1 +1 +1 -1 -1$
+1 -1 -1	3	$\mathbf{w}_3 : +1 +1 -1 -1 -1 -1 +1 +1$
-1 +1 +1	4	$\mathbf{w}_4 : +1 -1 +1 -1 +1 -1 +1 -1$
-1 +1 -1	5	$\mathbf{w}_5 : +1 -1 +1 -1 -1 +1 -1 +1$
-1 -1 +1	6	$\mathbf{w}_6 : +1 -1 -1 +1 +1 -1 -1 +1$
-1 -1 -1	7	$\mathbf{w}_7 : +1 -1 -1 +1 -1 +1 +1 -1$

The block diagram of the transmitter is shown in the upper part of Fig. 1. The k^{th} user's l^{th} information bit is denoted as $b_k[l] \in \{+1, -1\}$ ($k = 1, \dots, K$, $l = 1, \dots, L_b$, and L_b is the block length). The information bits are convolutionally encoded into code bits $\{u_k[l^n]\} \in \{+1, -1\}$, where $u_k[l^n]$ denotes the n^{th} code bit due to $b_k[l]$.

Code bits are subsequently interleaved and each block of $\log_2 M$ coded and interleaved bits $\{u_k[l^n]\} \in \{+1, -1\}$ is mapped into $\mathbf{w}_{i_k(j)} \in \{\mathbf{w}_0, \dots, \mathbf{w}_{M-1}\}$, which is one of the M Walsh codewords. The subscript $i_k(j) \in \{0, 1, \dots, M-1\}$ denotes the k^{th} user's j^{th} Walsh symbol index. The indices of the $\log_2 M$ systematic bits of each Walsh codeword $\mathbf{w}_{i_k(j)}$ is given by $i = M/2^{s+1}$ for $s = 0, 1, \dots, \log_2(M) - 1$. In case $M = 8$, the mapping rule is given in Table I. The columns corresponding to the three systematic bits, $\mathbf{w}_{i_k(j)}^1, \mathbf{w}_{i_k(j)}^2, \mathbf{w}_{i_k(j)}^4$, where $\mathbf{w}_{i_k(j)}^p$ denotes the p^{th} bit of the codeword, are highlighted in the table.

The interleaver and deinterleaver are denoted as Π and Π^{-1} , respectively, in Fig. 1 and the following figures. The purpose of interleaving is to separate adjacent code bits in time so that, ideally, each code bit will experience independent fading.

The Walsh codeword $\mathbf{w}_{i_k(j)} \in \{+1, -1\}^M$, is then repetition encoded into

$$\mathbf{s}_k(j) = \text{rep}(\mathbf{w}_{i_k(j)}, N/\log_2 M) \in \{+1, -1\}^N \quad (1)$$

where $\text{rep}(\cdot, \cdot)$ denotes the repetition encoding operation, where its first argument is the input bits and the second one is the repetition factor. Therefore, each bit of the Walsh codeword is spread (repetition coded) into $N_c = N/M$ chips, and each Walsh symbol is represented by N chips and denoted as $\mathbf{s}_k(j)$. The sequence $\mathbf{s}_k(j)$ is then scrambled (randomized) by a scrambling code unique to each user to form the transmitted chip sequence $\mathbf{a}_k(j) = \mathbf{C}_k(j)\mathbf{s}_k(j) \in \{+1, -1\}^N$, where $\mathbf{C}_k(j) \in \{-1, 0, +1\}^{N \times N}$ is a diagonal matrix whose diagonal elements correspond to the scrambling code for the k^{th} user's j^{th} symbol. In this paper, we focus on the use of long codes, e.g., the scrambling code differs from symbol to symbol. The purpose of repetition coding and scrambling is to spread the Walsh bits to N_c chips so that users can be separated. It is desirable to have low cross-correlations between different users' scrambling codes in order to reduce multiple access interference.

Fig. 1. Block diagram of the transmitter, channel, and receiver front end.

The scrambled sequence $\mathbf{a}_k(j)$ is pulse amplitude modulated using a unit-energy chip waveform $\psi(t)$ to form the baseband signal. For simplicity, we assume that $\psi(t)$ is a rectangular pulse with support $t \in [0, T_c)$ (The chip duration is denoted by T_c , its relation with symbol duration T is $T = NT_c$); 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 noise power spectral density $N_0/2$ and with L_k resolvable paths, having time-varying complex channel gains $h_{k,1}(t), h_{k,2}(t), \dots, h_{k,L_k}(t)$ and delays $\tau_{k,1}, \tau_{k,2}, \dots, \tau_{k,L_k}$ (see the lower part of Fig. 1). The received signal is the sum of K users' signals plus additive white complex Gaussian noise $n(t)$. After frequency down-conversion and chip matched filtering (CMF), the received signal corresponding to the k^{th} user's j^{th} transmitted Walsh sequence $\mathbf{s}_k(j)$ can be written in vector form as

$$\begin{aligned} \mathbf{r}(k, j) &= \mathbf{A}(k, j)\mathbf{h}(j) + \mathbf{n}(k, j) \\ &= \mathbf{X}_k(j)\mathbf{h}_k(j) + \text{ISI}(k, j) + \text{MAI}(k, j) + \mathbf{n}(k, j) \in \mathbb{C}^{N_k} \end{aligned} \quad (2)$$

where the columns of the matrix $\mathbf{A}(k, j)$ are the delayed version of transmitted chip sequences $\mathbf{a}_k(j)$ for $k = 1, 2, \dots, K$, one column per path. The length of the processing window N_k , is larger than the symbol interval N to account for the asynchronous and multipath nature of the channel. The columns are weighted together by $\mathbf{h}(j)$, whose elements are the path gains of all users' paths. The received vector $\mathbf{r}(k, j)$ can be written as the sum of four terms: the signal of interest $\mathbf{X}_k(j)\mathbf{h}_k(j)$, the intersymbol interference (ISI), the multiple access interference (MAI), and the noise represented by $\mathbf{n}(k, j)$, which is a vector of complex noise samples with zero mean and variance N_0 . The columns of the matrix $\mathbf{X}_k(j)$ are essentially the shifted versions of the chips due to the k^{th} user's j^{th} symbol, one column per path (the shift is determined by the path delay). The vector $\mathbf{h}_k(j) = [h_{k,1}(jT) \ h_{k,2}(jT) \ \dots \ h_{k,l}(jT) \ \dots \ h_{k,L_k}(jT)]^T$ corresponds to the channel gains of the k^{th} user's paths, it is part of $\mathbf{h}(j)$.

III. ITERATIVE DECODING ALGORITHMS

The iterative receiver structure for decoding the data transmitted by user k is illustrated in Fig. 2. It consists of two stages: a SISO inner block decoder, followed by a SISO outer convolutional decoder. The two stages are separated by the deinterleaver Π^{-1} and the interleaver Π . The k^{th} user's outer convolutional decoder takes $\lambda(u_k[l^n]; I)$, the extrinsic values of the code bits, as input. It delivers as output an update of the LLRs of the code bits $\lambda(u_k[l^n]; O)$, as well as the LLRs of the information bits $\lambda(b_k[l]; O)$, based on the code constraints. The latter are used for making hard decisions on transmitted information bits at the final iteration; while the former are used for two purposes: deriving extrinsic information $\lambda(u'_k[l^n]; I)$ for inner decoding and deriving estimate of transmitted Walsh sequence $\hat{\mathbf{s}}_k(j)$ for channel estimation and multiuser detection. In multiuser case, the extrinsic information $\lambda(u'_k[l^n]; I)$ should be the properly normalized or corrected version before feeding back to the inner decoder. This point will be elaborated later on in Section VI. The inner decoder accepts a priori information $\lambda(u'_k[l^n]; I)$ and channel values and delivers soft output value $\lambda(u'_k[l^n]; O)$. Decoding is based on alternately decoding the two component codes and passing the updated extrinsic information, which is part of the soft output of the SISO decoder, to the next decoding stage. The process is repeated several times and ended by making a hard decision on the LLR values of the information bits in the last iteration. Several iteration stopping criteria can be envisaged; however, we will adopt the simply rule to stop after a pre-determined number of iterations. We use the notation $\lambda(\cdot; I)$ and $\lambda(\cdot; O)$ at the input and output ports of SISO. They refer to the unconstrained LLRs when the second argument is I , and modified LLRs according to the code constraints when it is O . The second argument I or O is sometimes omitted to simplify notation whenever no ambiguity arises. Other soft values are denoted by $L(\cdot)$. They are usually soft input and output of non-SISO devices.

To avoid statistical dependencies between the soft values of several iteration steps, it is necessary to feed back only the extrinsic value $\lambda(u_k[l^n]; I) = \Pi^{-1}\{\lambda(u'_k[l^n]; O) - \lambda(u'_k[l^n]; I)\}$

Fig. 2. Block diagram of iterative decoding with extrinsic normalization/correction.

to the outer decoder and $\lambda(u'_k[l]; I) = \Pi\{\lambda(u_k[l]; O) - \lambda(u_k[l]; I)\}$ to the inner decoder as shown in Fig. 2. These two decoder modules are discussed in detail next.

A. SISO Outer Convolutional Decoder

Based on the soft input $\lambda(u_k[l]; I)$ and the trellis structure of the convolutional code, the k^{th} user's SISO channel decoder computes a posteriori LLR of each information bit $\lambda(b_k[l]; O)$ and each code bit $\lambda(u_k[l]; O)$ as

$$\lambda(b_k[l]; O) = \ln \frac{P[b_k[l] = +1 | \lambda(u_k[l]; I)]}{P[b_k[l] = -1 | \lambda(u_k[l]; I)]} \quad (3)$$

$$\lambda(u_k[l]; O) = \ln \frac{P[u_k[l] = +1 | \lambda(u_k[l]; I)]}{P[u_k[l] = -1 | \lambda(u_k[l]; I)]} \quad (4)$$

where $\lambda(b_k[l]; O)$ is used to make decision on the transmitted information bit at the final iteration, while $\lambda(u_k[l]; O)$ is used for channel estimation and interference cancellation in the demodulator at the next iteration.

Several SISO algorithms can be used to compute the channel decoder outputs (3) and (4). For estimating the states or outputs of a Markov process, the symbol-by-symbol MAP algorithm is optimal. It differs from the Viterbi algorithm (VA) in the optimality criterion. The VA minimizes the frame or packet error probability, and the MAP algorithm minimizes symbol error probability [13]. The MAP algorithm searches for the most probably transmitted symbol, given the received vector. It, however, poses numerical representation problems, and requires a large number of additions and multiplications. Max-Log-MAP solves the numerical problem and reduces the computational complexity, but are suboptimal especially at low SNR region. A further simplification yields the soft-output Viterbi algorithm (SOVA), it has simpler structure but inferior performance compared to Max-Log-MAP. By complementing the $\max(\cdot)$ operation with a correction function, Log-MAP algorithm avoids the approximations in the Max-Log-MAP and is equivalent to (true) symbol-by-symbol MAP, but without its major disadvantages. Therefore, we consider the use of Log-MAP for the purpose of this study. For a complete treatment on different SISO algorithms, their similarities, differences and performance comparisons, readers are recommended to consult [14].

B. SISO Inner Block Decoder

The LLR of a transmitted +1 and -1 for every coded and interleaved bit $u'_k[l]$ from each user $k = 1, 2, \dots, K$ is given according to [2], [6] by

$$\begin{aligned} \lambda(u'_k[l]; O) &= \ln \frac{P[u'_k[l] = +1 | \mathbf{y}]}{P[u'_k[l] = -1 | \mathbf{y}]} = \ln \frac{\sum_{m: u'_k[l] = +1} P(\mathbf{w}_m | \mathbf{y})}{\sum_{m: u'_k[l] = -1} P(\mathbf{w}_m | \mathbf{y})} \\ &= \ln \frac{\sum_{m: u'_k[l] = +1} \exp\left(\frac{1}{2} \sum_{i=1}^N L(w_i; y_i) w_i\right)}{\sum_{m: u'_k[l] = -1} \exp\left(\frac{1}{2} \sum_{i=1}^N L(w_i; y_i) w_i\right)} \\ &= \ln \frac{\sum_{m: u'_k[l] = +1} \exp\left(\frac{1}{2} \mathbf{L}^T \mathbf{w}_m\right)}{\sum_{m: u'_k[l] = -1} \exp\left(\frac{1}{2} \mathbf{L}^T \mathbf{w}_m\right)} \end{aligned} \quad (5)$$

where we use the notation $m : u'_k[l]_k = \pm 1$ to denote the set of Walsh codes $\{\mathbf{w}_m\}$ that correspond to the code bit $u'_k[l] = \pm 1$, and assume $u'_k[l]$ is one of the $\log_2 M$ systematic bits of the inner Walsh codeword. The i^{th} bit of the Walsh codeword $\mathbf{w}_m$ is denoted as $w_i \in \{+1, -1\}$. The vector $\mathbf{y}$ is of length M , and is due to the k^{th} user's j^{th} transmitted Walsh symbol, and is obtained by despreading and RAKE combining of the received vector $\mathbf{r}(k, j)$ or its interference canceled version $\mathbf{r}'(k, j)$. The vector $\mathbf{y}$ changes from one processing window to the next. The process of despreading and multipath combining will be elaborated shortly in the next subsection when different approaches of inner decoding are discussed. In equation (5), we denote T as vector transpose operation. Each element of the vector $\mathbf{L} = [L(w_1; y_1), L(w_2; y_2), \dots, L(w_M; y_M)]^T$ is defined as

$$L(w_i; y_i) = \begin{cases} L_c y_i + \lambda(u'_k[l]; I), & \text{for } i = \frac{M}{2^s + 1}, s = 0, 1, \dots, \log_2 M - 1; \\ L_c y_i, & \text{otherwise.} \end{cases}$$

which is the channel value y_i multiplied with channel reliability L_c supplemented with a priori information $\lambda(u'_k[l]; I)$ for the $\log_2 M$ systematic bits of each codeword $\mathbf{w}_m$, and L_c is defined such that

$$L_c y_i = \ln \frac{p(y_i | w_i = +1)}{p(y_i | w_i = -1)}$$

Typically, one term will dominate each sum in (5), which suggests the “dual-maxima” approximation [6], [15]

$$\lambda(u'_k[l]; O) \approx \frac{1}{2} \max_{m: u'_k[l] = +1} \mathbf{L}^T \mathbf{w}_m - \frac{1}{2} \max_{m: u'_k[l] = -1} \mathbf{L}^T \mathbf{w}_m \quad (6)$$

The vectors $\mathbf{y}$ and $\mathbf{L}$ should be formed and L_c computed according to the chosen strategy for the inner decoding, which can be a traditional single user approach or a MUD-aided approach as discussed next.

1) *Conventional single user approach*: The conventional inner decoding scheme is illustrated in Fig. 3. For simplicity of notation we will suppress the index k and/or j from $\mathbf{s}_k(j)$, $\mathbf{C}_k(j)$, $\mathbf{r}(k, j)$, $\mathbf{A}(k, j)$, $\mathbf{n}(k, j)$, $\mathbf{X}_k(j)$ and $\mathbf{h}_k(j)$, etc., whenever no ambiguity arises.

Let $\mathbf{r}_{k,l}$, ($l = 1, 2, \dots, L_k$) denote the delay aligned version of the received vector due to the transmission of the j^{th} symbol from the k^{th} user's l^{th} path. The vector $\tilde{\mathbf{r}}_{k,l} \in \mathbb{C}^N = [\tilde{r}_{k,l}[1] \ \tilde{r}_{k,l}[2] \ \dots \ \tilde{r}_{k,l}[N]]^T$ and $\tilde{\mathbf{n}}_{k,l} \in \mathbb{C}^N$ are $\mathbf{r}_{k,l}$ and the original noise vector $\mathbf{n}$ scrambled with the scrambling sequence $\mathbf{C}_k$. The symbol $\mathbb{C}$ denotes the complex field. To simplify the development of the receiver algorithm, we assume different users' scrambling sequences are orthogonal to each other (their cross-correlations are approximately zero) and their autocorrelations approximate delta function. Under this (optimistic) assumption, $\mathbf{r}_{k,l}$ will after descrambling and despreading only contain the contribution from k^{th} user's l^{th} path plus additive noise.

Let us assume unit chip energy and define

$$\tilde{\mathbf{r}}_{k,l}^d \in \mathbb{C}^M = [\tilde{r}_{k,l}^d[1] \ \tilde{r}_{k,l}^d[2] \ \dots \ \tilde{r}_{k,l}^d[M]]^T$$

The i^{th} element $\tilde{r}_{k,l}^d[i]$ is the output of the l^{th} path's despreaders corresponding to w_i , it is formed simply by

$$\begin{aligned} \tilde{r}_{k,l}^d[i] &= \sum_{n=1}^{N_c} \tilde{r}_{k,l}[(i-1)N_c + n] \\ &= h_{k,l}N_c w_i + \sum_{n=1}^{N_c} \tilde{n}_{k,l}[(i-1)N_c + n] \end{aligned} \quad (7)$$

where $N_c = N/M$ is the number of chips for each bit w_i . Let us define $\mathbf{y} = [y_1 \ y_2 \ \dots \ y_M]^T$ the output of Maximum Ratio Combiner (MRC) with element y_i computed as

$$\begin{aligned} y_i &= \text{Re} \left\{ \sum_{l=1}^{L_k} \hat{h}_{k,l}^* \tilde{r}_{k,l}^d[i] \right\} \\ &= \text{Re} \left\{ \sum_{l=1}^{L_k} \hat{h}_{k,l}^* \sum_{n=1}^{N_c} \tilde{r}_{k,l}[(i-1)N_c + n] \right\} \end{aligned} \quad (8)$$

where $\hat{h}_{k,l}$ is an estimate of the channel gain $h_{k,l}$. The superscript operator $(\cdot)^*$ is the conjugate transpose operation when applied to matrices, and simply the conjugate when applied to scalars. Substituting (7) into (8) and assume perfect

channel estimation, i.e., $\hat{h}_{k,l} = h_{k,l}$, we derive

$$\begin{aligned} y_i &= \text{Re} \left\{ \sum_{l=1}^{L_k} h_{k,l}^* \left(h_{k,l} N_c w_i + \sum_{n=1}^{N_c} \tilde{n}_{k,l}[(i-1)N_c + n] \right) \right\} \\ &= N_c w_i \sum_{l=1}^{L_k} |h_{k,l}|^2 + \text{Re} \left\{ \sum_{l=1}^{L_k} h_{k,l}^* \sum_{n=1}^{N_c} \tilde{n}_{k,l}[(i-1)N_c + n] \right\} \\ &= N_c w_i P_k + n_{yi} \end{aligned}$$

where $P_k = \sum_{l=1}^{L_k} P_{k,l} = \sum_{l=1}^{L_k} |h_{k,l}|^2$ is the received power from all paths of user k .

Since the descrambling operation does not change the noise statistic, the noise sample is complex Gaussian $\tilde{n}_{k,l}[i] \sim \mathcal{CN}(0, N_0)$, and $\sum_{n=1}^{N_c} \tilde{n}_{k,l}[(i-1)N_c + n] \sim \mathcal{CN}(0, N_c N_0)$. Thereby,

$$n_{yi} = \text{Re} \left\{ \sum_{l=1}^{L_k} h_{k,l}^* \sum_{n=1}^{N_c} \tilde{n}_{k,l}[(i-1)N_c + n] \right\} \sim \mathcal{N}(0, N'_0/2)$$

where $N'_0 = N_c P_k N_0$. Recall that $y_i = N_c P_k w_i + n_{yi}$, thus

$$\begin{aligned} p(y_i | w_i = \pm 1) &= \frac{1}{\sqrt{\pi N'_0}} \exp \left[\frac{-(y_i \mp N_c P_k)^2}{N'_0} \right] \\ \ln \frac{p(y_i | w_i = +1)}{p(y_i | w_i = -1)} &= \frac{-(y_i - N_c P_k)^2 + (y_i + N_c P_k)^2}{N'_0} \\ &= \frac{4N_c P_k y_i}{N_c P_k N_0} = \frac{4}{N_0} y_i \end{aligned} \quad (9)$$

From (9), we obtain the channel reliability value $L_c = 4/N_0$. In reality, the assumptions of code orthogonality and perfect channel estimation are not valid. Hence, the algorithm derived based on these assumptions is therefore quite suboptimal. Especially, the presence of MAI and ISI will deteriorate the system performance. One way to work around this problem would be to increase the value of N_0 to capture the MAI and ISI. A more effective measure to alleviate their effect is the use of MUD techniques, which will be discussed next.

2) *MUD approach*: The aperiodic nature of the long codes employed in this work usually precludes the use of linear multiuser detection schemes, like the MMSE detector and decorrelator, due to their high computational complexity. Therefore, only the nonlinear parallel interference cancellation (PIC) scheme is considered here. The inner decoding scheme combined with interference cancellation is depicted in Fig. 4. The PIC with decision feedback will be discussed next.

Interference cancellation is performed by estimating the transmitted signals in parallel for all the users, and then subtracting the estimated signals of the interfering users from the received signal $\mathbf{r}$ to form a new signal vector $\mathbf{r}_{k,l}^{\text{PIC}}$ for demodulation of the signal transmitted from the l^{th} path of user k . Mathematically, it is expressed as

$$\mathbf{r}_{k,l}^{\text{PIC}} = \mathbf{r} - \hat{\mathbf{A}}\mathbf{h} + \hat{\mathbf{a}}_k \hat{h}_{k,l}$$

where $\mathbf{r} \in \mathbb{C}^{N_k}$ denote the received signal vector due to the transmission of the j^{th} symbol from the k^{th} user's l^{th} path, it contains N_k chips (usually $N_k > N$ due to multipath delay spread). The vector $\mathbf{r}_{k,l}^{\text{PIC}} \in \mathbb{C}^{N_k}$ is its interference cancelled version after subtracting the contributions from all the other users and the same user's other paths using hard decision

Fig. 3. Internal structure of the conventional SISO inner decoder.

Fig. 4. Internal structure of MUD-aided SISO inner decoder.

feedback. The vector $\hat{\mathbf{A}}\hat{\mathbf{h}}$ represents the estimated contribution from all the users calculated by using the data matrix $\hat{\mathbf{A}}$ and channel vector $\hat{\mathbf{h}}$ estimated at the previous iteration. The vector $\hat{\mathbf{a}}_k\hat{h}_{k,l}$ is the estimated contribution from the l^{th} path of user k .

The derivation of L_c is the same as in Section III-B.1 except that $\mathbf{r}_{k,l}$ is replaced by $\mathbf{r}_{k,l}^{\text{PIC}}$, the delay compensated and MAI and ISI cancelled version of the received vector due to the transmission of the j^{th} symbol from the k^{th} user's l^{th} path. The MAI and ISI are estimated by making tentative hard decisions on the output from the outer decoder, i.e., $\hat{u}_k[l]^n = \text{sgn}\{\Pi(\lambda(u_k[l]^n; O))\}$ (see Fig. 2) for all k . Then we go through block encoding and spreading to produce an estimate of the transmitted chip sequence $\hat{\mathbf{s}}_k$, which is used for

both interference cancellation and channel estimation. Channel estimation will be treated in Section IV. For detailed description on interference cancellation in orthogonal signalling systems, refer to [8].

In the ideal situation, the MAI from other users and ISI from the same user's other paths are cancelled. Going through the same procedure as shown in (7)–(9), we come up with the same channel reliability value $L_c = 4/N_0$. However, the mechanisms for deriving $\mathbf{r}_{k,l}$ and $\mathbf{r}_{k,l}^{\text{PIC}}$ are different (single user and MUD approach respectively) which result in different $\mathbf{y}$ and $\mathbf{L}$ vectors used in equations (5)–(6) for computing LLR values.

It should be noted that the inner decoding can be accomplished without extrinsic information. The switch in Fig. 2 is

then turned off, and $L(w_i; y_i) = L_c y$ for all i in equation (5) and (6). The performance can still be improved at each iteration without extrinsic information because we get better estimate of the channel $\hat{\mathbf{h}}_k$ and transmitted sequence $\hat{\mathbf{s}}_k$ (better cancellation) as the iteration proceeds.

IV. ESTIMATION OF FADING PROCESSES

We need estimates of the complex channel gains to do maximum ratio combining as discussed in section III-B.1 and III-B.2. Recall that the received signal vector is formed as $\mathbf{r} = \mathbf{A}\mathbf{h} + \mathbf{n}$. The task of a channel estimator is to estimate the fading vector $\mathbf{h}$ given the received observation $\mathbf{r}$ and the transmitted data. Depending on the form of the data that can be retrieved, channel estimation can be either decision directed or pilot aided. The former uses decision feedback loops and utilizes the decisions on the transmitted signals $\hat{\mathbf{A}}$ to extract the channel coefficients. The second approach makes the use of pilot symbols or channels ($\mathbf{A}$ is known in this case). The use of pilots simplifies channel estimation with the penalty of wasting channel resources. In this paper, we focus on the first approach and estimate time-varying multipath Rayleigh fading channels in absence of pilot symbols. However, we assume that all path delays are known.

The principle is that the accuracy of channel estimation depends how accurate the receiver demodulates the data. In iterative decoding, data are usually better detected at the next iteration stage so that the channel can be better estimated, which in turn leads to the improvement of decoding performance in the upcoming stage.

The approach is to start by making a rather crude estimate of data using noncoherent matched filtering technique [8], then estimate the channel using the initially detected data. After that, iterative decoding described above can be applied to detect the data more accurately, and the channel estimate is refined by using the detected data, and so on.

The maximum likelihood (ML) estimator is described in this section to estimate frequency selective multipath channel gains for the orthogonal signalling systems. It is decision-directed method (the estimation procedure at the n^{th} iteration uses the data estimates from the $(n-1)^{\text{th}}$ stage) and can be inserted into the iteration loops in the previous section.

From (2), we recall that $\mathbf{r}(k, j) = \mathbf{A}(k, j) + \mathbf{h}(k, j) + \mathbf{n}(k, j)$, where $\mathbf{n}(k, j) \sim \mathcal{CN}(0, N_0)$. Assuming that $\mathbf{A}(k, j)$ is known and of full column rank and treating $\mathbf{h}(k, j)$ as unknown and deterministic, the maximum likelihood (ML) estimate of $\mathbf{h}(k, j)$ is $\hat{\mathbf{h}}(k, j) = \hat{\mathbf{A}}^\dagger(k, j)\mathbf{r}(k, j)$, where $\hat{\mathbf{A}}^\dagger(k, j) = (\hat{\mathbf{A}}^*(k, j)\hat{\mathbf{A}}(k, j))^{-1}\hat{\mathbf{A}}^*(k, j)$ is the left pseudoinverse of $\hat{\mathbf{A}}(k, j)$ [8].

If $\hat{\mathbf{A}}(k, j)$ is an estimate of the data matrix, then we can still form an estimate of $\mathbf{h}(k, j)$ as $\hat{\mathbf{h}}(k, j) = \hat{\mathbf{A}}^\dagger(k, j)\mathbf{r}(k, j)$. In the case of correct decisions $[\hat{\mathbf{A}}(k, j) = \mathbf{A}(k, j)]$, then $\hat{\mathbf{h}}(k, j) = \mathbf{A}^\dagger(k, j)\mathbf{A}(k, j)\mathbf{h}(k, j) + \mathbf{A}^\dagger(k, j)\mathbf{n}(k, j) = \mathbf{h}(k, j) + \mathbf{A}^\dagger(k, j)\mathbf{n}(k, j)$ which is an unbiased estimate of $\mathbf{h}(k, j)$. To be completely correct, this estimate is truly ML only if the data estimates are ML decisions. However, we will still refer to the algorithm as ML even for other data detectors.

This procedure will suffer from a so-called dimensionality problem. When the total number of paths of all the users is

greater than the number of chips in the vector $\mathbf{r}(k, j)$, i.e., $L_{\text{tot}} \geq N_k$, the matrix $\hat{\mathbf{A}}(k, j)$ will not have full column rank and the above mentioned procedure will become useless (e.g., $\hat{\mathbf{A}}^*(k, j)\hat{\mathbf{A}}(k, j)$ is not invertible). The problem can be resolved by stacking several $\mathbf{r}(k, j)$ vectors on top of each other and assuming the channel remains static during several symbol intervals. In particular, suppose $\mathbf{h}(k, j) \approx \mathbf{h}(k, j+1)$, we can then write

$$\begin{aligned} \begin{bmatrix} \mathbf{r}(k, j) \\ \mathbf{r}(k, j+1) \end{bmatrix} &= \begin{bmatrix} \mathbf{A}(k, j) & \mathbf{0} \\ \mathbf{0} & \mathbf{A}(k, j+1) \end{bmatrix} \begin{bmatrix} \mathbf{h}(k, j) \\ \mathbf{h}(k, j+1) \end{bmatrix} \\ &\quad + \begin{bmatrix} \mathbf{n}(k, j) \\ \mathbf{n}(k, j+1) \end{bmatrix} \\ &\approx \begin{bmatrix} \mathbf{A}(k, j) \\ \mathbf{A}(k, j+1) \end{bmatrix} \mathbf{h}(k, j) + \begin{bmatrix} \mathbf{n}(k, j) \\ \mathbf{n}(k, j+1) \end{bmatrix} \end{aligned}$$

The channel estimation algorithm using hard decision of the matrix $\hat{\mathbf{A}}$ can be reformulated as

$$\hat{\mathbf{h}}(k, j) = \begin{bmatrix} \hat{\mathbf{A}}(k, j) \\ \hat{\mathbf{A}}(k, j+1) \end{bmatrix}^\dagger \begin{bmatrix} \mathbf{r}(k, j) \\ \mathbf{r}(k, j+1) \end{bmatrix} \quad (10)$$

which will produce usable estimates as long as $2N_k > L_{\text{tot}}$. Obviously, this scheme can be extended further by stacking several $\mathbf{r}(k, j)$ vectors on top of each other, like

$$\hat{\mathbf{h}}(k, j) = \begin{bmatrix} \hat{\mathbf{A}}(k, j) \\ \vdots \\ \hat{\mathbf{A}}(k, j+D) \end{bmatrix}^\dagger \begin{bmatrix} \mathbf{r}(k, j) \\ \vdots \\ \mathbf{r}(k, j+D) \end{bmatrix} \quad (11)$$

Stacking also has the effect of noise averaging and tends to reduce the error of the channel estimation [16]. However, the quality of the estimates may be reduced, especially for fast fading channels.

As shown in Fig. 5, the initial ML channel estimates are noisy compared to the original channel. We know that the channel gains are correlated in time, and we should therefore be able to improve the estimates by smoothing. A simple smoothing procedure is to feed $\hat{\mathbf{h}}(j)$ through an FIR filter with impulse response $g(n)$, which yields the smoothed channel gain vector $\bar{\mathbf{h}}(j)$ as

$$\bar{\mathbf{h}}(j) = \sum_{k=j-N_s}^{j+N_s} \hat{\mathbf{h}}(k)g(j-k) \quad (12)$$

The impulse response of the smoothing filter is also plotted in Figure 5. Smoothing operation significantly improves the estimation performance. The channel estimate after smoothing operation (see the lower right corner of Fig. 5) is very close the original channel.

V. PARAMETERS SETTING AND INITIAL RESULTS

Different approaches are evaluated numerically with computer simulations. In the simulations, we employ a rate 1/3 Maximum Free Distance (MFD) convolutional code [17] with constraint length 5 and generator polynomials (25, 33, 37) in octal form for all the users. Block interleaving is applied to the convolutionally encoded bits to decorrelate the fading effect. Each block of $\log_2 8 = 3$ interleaved bits from each user is then converted into one of $M = 8$ Walsh codes spread to a

Fig. 5. ML channel estimates with and without smoothing.

total length of $N = 64$ chips. The number of chips per inner code bit is $N_c = N/M = 8$. If the orthogonal modulation is viewed as part of spreading, the effective spreading factor of the system is $N/\log_2 M = 64/3$ chips per convolutionally coded bit (64 chips per information bit).

Channels are independent Rayleigh fading channels with the classical “bath tub” power spectrum [18]. The channel gain $h_{k,l}(t)$ is a complex circular Gaussian process with autocorrelation function $E[h_{k,l}^*(t)h_{k,l}(t+\tau)] = \bar{P}_{k,l}J_0(2\pi f_D\tau)$ where f_D is the maximum Doppler frequency, $J_0(x)$ is the zeroth order Bessel function of the first kind. The Doppler shifts on each of the multipath components are due to the relative motion between the base station and mobile units. Here, the normalized Doppler frequency is assumed to be $f_D T = 0.01$. Channel estimation is conducted with the ML algorithm presented in Section IV. The choice of the FIR filter length for channel smoothing depends on the nature of fading, e.g., the normalized Doppler frequency. For $f_D T = 0.01$, it was shown in [16] that a smoothing filter derived from Hamming window of length 19 (normalized such that $\sum_{k=-N_s}^{N_s} g(k) = 1$) is appropriate, and is thus used in our simulations here.

Perfect slow power control is assumed in the sense that $\bar{P}_k = \sum_{l=1}^{L_k} \bar{P}_{k,l} = \sum_{l=1}^{L_k} E[|h_{k,l}|^2]$, the average received power, is equal for all users. The number of multipath channels L_k is set to be 3, ($L_k = L = 3$) for all k . The long scrambling codes C_k are randomly assigned. The noise variance N_0 and C_k as well as delays $\tau_{k,1}, \tau_{k,2}, \dots, \tau_{k,L_k}$ are assumed to be known to the receiver. The block size is set to 1540 Walsh symbols, which corresponds to $1540 \times 3 = 4620$ code bits. The interleaver size is set to be 66×70 for all the conducted experiments. The simulation results are averaged over random distributions of fading, noise, delay, and scrambling codes with a minimum of 10 blocks of data transmitted and at least 100 errors generated.

Fig. 6 shows the results of iterative decoding for single user system with conventional approach (no interference cancella-

Fig. 6. Performance of iterative decoding for a single user system.

tion is needed in this case). The gain by applying extrinsic information to the inner decoding is 1.3 dB at BER of 10^{-5} and 0.8 dB at BER of 10^{-3} when compared against non-extrinsic feedback case, which is more than the 0.6 dB gain reported in [6] for AWGN channel. This is due to the multipath diversity gain obtained by maximum ratio combining. If the approximation in (6) is used for inner decoding and the Log-MAP is replaced by Max-Log-MAP for the outer decoding, the performance loss is noticeable in low SNR region, and gradually becomes smaller as SNR increases. To study the behavior of each algorithm, the number of iterations is usually set to 7 (except in Fig. 14), since it is observed that almost all the algorithms would converge after 5 or 6 iterations.

Different schemes discussed above are compared in Fig. 7. As expected, PIC aided iterative decoding outperforms the conventional single user approach in a multiuser environment. The results also show that the reliability of the extrinsic information goes down as the number of user increases. It is attributed to the fact that these algorithms assume perfect channel estimation and perfect cancellation, which clearly is not the case in a system with high level of interference. When the number of user goes beyond 14, the performance of PIC aided decoding without extrinsic information surpasses the one with extrinsic information. A similar trend is also observed with conventional approach, not as drastic though (the gain by applying extrinsic information gradually diminishes as the number of user increases). In the next section, we propose some schemes to improve the reliability of the extrinsic information, and make it useful in heavily loaded systems.

All the results presented in this paper are obtained based on $L = 3$ -path model. Note that if $L > 3$, the system performance would improve compared to the presented results due to the fact that diversity gain increases as the number of paths increases, especially when the MUD-aided decoding is used and the interference is effectively removed. The opposite conclusion holds when $L < 3$. For a quantitative analysis of the impact of L on the system performance, readers are

Fig. 7. Comparison of different iterative decoding schemes in multiuser environments.

referred to [16].

In Fig. 8, we examine the effect of the stacking factor D in equation (11) on the channel estimation results. As indicated by the Cramer-Rao Lower Bound (CRLB) derived in [16], it seems that the larger D value, the smaller estimator error (we use estimation variance as performance measurement) we will get. That would be the case if the channel is static. However, for the time-varying fading channel, the channel changes beyond the coherence time. Therefore, D value has to be chosen accordingly. From the plot, one can see that for $f_D T = 0.01$, $D = 4$ appears to be the optimum value before smoothing, and $D = 2$ or $D = 3$ appears to be the optimum value after smoothing. The time-varying nature of the fading channel prohibits the use of a larger stacking. Also, the dependency between stacking and smoothing as shown by the simulation results has to be taken into account in the selection of the stacking factor D to achieve the best channel estimation and decoding performance. For the simulations conducted in this paper, D is set to 3.

It was also shown in [16] that when the ML estimator is coupled with interference cancellation technique, the decision directed ML channel estimator yields close performance to the pilot aided approach assuming exact knowledge of the transmitted data after the system convergence is reached.

VI. CORRECTION/ADAPTATION OF EXTRINSIC INFORMATION

It is stated in [19] that for bad channels the reliability information of soft decoder output is too optimistic. The output can be considered as being multiplied by a factor, that depends on the current BER. To become closer to the true LLR, the output has to be normalized. Although the authors drew the conclusion for the soft-output-Viterbi-decoder (SOVA) in bad channels (low SNR), we discovered similar problem also with MAP/Log-MAP decoder in severe interference environment. As indicated in Fig. 7, when the level of interference increases,

Fig. 8. The impact of the stacking factor on the performance of ML channel estimation.

the extrinsic information becomes unreliable and leads to worse performance than decoding without extrinsic feedback. The reason is that we assume perfect interference cancellation in the MUD-aided decoding, therefore, in the derivation of channel reliability value $L_c = 4/N_0$, N_0 only contains the noise variance. However, in a heavily loaded system, perfect interference cancellation can hardly be achieved, and N_0 should contain noise variance plus interference cancellation residual. The latter is determined by the channel statistics (the number of paths per user, the average received power per path), number of users, processing gain of the CDMA system as well as the probability of the erroneous cancellation which is related to error probability (BER) of previous decoding stage. However, BER performance is very difficult to analyze in an iterative decoding system. The inaccurate N_0 value due to the ignorance of interference cancellation residual leads to the reduced reliability of the extrinsic information. A similar justification can be made to the conventional decoding where different users' scrambling sequences are assumed to be orthogonal to each other. That implies the need for normalization or correction of the extrinsic values.

Fig. 9 shows the histogram of the output (λ) of the SISO outer decoder at different iterations. Apparently, λ can be approximated as Gaussian distributed variable with mean value m_λ (or $-m_\lambda$) and variance σ_λ^2 . The pdf of λ conditioned on the bit $u = \pm 1$ being transmitted can be expressed as

$$p(\lambda|u = \pm 1) = \frac{1}{\sqrt{2\pi}\sigma_\lambda} \exp \left[-\frac{1}{2\sigma_\lambda^2} (\lambda \mp m_\lambda)^2 \right]$$

The conditional LLR, given the observation of the decoder output is calculated as [19]

$$\begin{aligned} \lambda(u) &= \ln \frac{P(\lambda|u = +1)}{P(\lambda|u = -1)} = \ln \left[e^{-\frac{1}{2\sigma_\lambda^2} ((\lambda - m_\lambda)^2 - (\lambda + m_\lambda)^2)} \right] \\ &= \frac{2m_\lambda}{\sigma_\lambda^2} \lambda \end{aligned}$$

Fig. 9. Histograms of the extrinsic value (12-user system, $E_b/N_0 = 7\text{dB}$) at different iteration stages.

which means the output λ has to be multiplied with the factor $c = 2m_\lambda/\sigma_\lambda^2$ to obtain the real LLR. Since the value of c depends on the current BER of the decoder output, which can vary from block to block, c has to be calculated for each block individually. From our experiments, we also notice that slightly better results can be achieved when modifying the normalization factor as

$$c = \begin{cases} 2m_\lambda/\sigma_\lambda^2, & \text{if } 2m_\lambda/\sigma_\lambda^2 < 1; \\ 1, & \text{otherwise.} \end{cases}$$

The performance of this extrinsic normalization scheme is shown by the dashed curve in Fig. 10. It works rather well for a moderate number of users (up to 20 users). However, it gradually becomes ineffective as the system becomes more heavily loaded, which necessitates the use of other correction method. Obviously, the optimum correction factor c_{opt} should lie between 0 (meaning no extrinsic information) and 1 (meaning extrinsic without correction). Extrinsic information can not be exploited if c value is too small ($c \approx 0$) and is not properly corrected if c value is too big ($c \approx 1$). Considering the fact that the uncorrected extrinsic information becomes less effective when the system becomes more heavily loaded, we conjecture that c_{opt} should be in the vicinity of $1/K$, the reciprocal of the total number of user K to combat the detrimental effect of the interference. The optimum factor $c_{\text{opt}} = 1.3/K$ was found from the search discussed below. The solid line in Fig. 10 shows this correction method yields better performance than the extrinsic normalization scheme introduced above in severe MAI situations. This clearly indicates the necessity to do extrinsic correction in order to improve the performance of iterative decoding in multiuser environments.

We compared different correction factors $c = 1/K, 1.3/K, 1.5/K$ in Fig. 11. In PIC case, decoding with $c = 1.3/K$ and $c = 1.5/K$ give almost identical result, $c = 1/K$ is slightly worse. All of them perform better than decoding without extrinsic feedback, the gain

is $0.4 \sim 1.1\text{dB}$ in 15 user case. It proves that extrinsic information really helps improve the decoding performance if properly manipulated. Other values of c in cases of $c < 1/K$ and $c > 1.5/K$ are also tested, they yield worse performance than $c = 1.3/K$ or $c = 1.5/K$, and not shown in Fig. 11. While with conventional approach, the gain by introducing extrinsic correction is not noticeable: $c = 1.3/K$ and $c = 1$ (meaning no extrinsic correction) yield almost the same result. It is worth noticing the significant gain achieved by incorporating PIC into inner decoding compared to the conventional scheme, the difference can be as large as 2.4 dB. However, the price to pay for the performance improvement is the added complexity due to the interference cancellation process.

The results presented in Fig. 6 indicate that the extrinsic information (without correction) helps improve the system performance in single user channel (in absence of MAI), which suggests another work around method to mitigate the deteriorative effect of the interference and to exploit extrinsic information more efficiently. That is to use some adaptive (switching) scheme. The basic idea is to do decoding without extrinsic feedback for a few iterations, the channel becomes cleaner (MAI and ISI are more effectively suppressed) and closer to single user channel as the iteration goes on. Then we turn on the extrinsic feedback and let it run for a few more iterations. The results of this adaptive decoding scheme are shown Fig. 12 and Fig. 13. Seven iterations (4 stages without extrinsic feedback and 3 stages with unprocessed extrinsic feedback) are carried out in the test. It can be observed from Fig. 12 that adaptive decoding always performs better than decoding with unprocessed extrinsic information, it, however, converges to decoding without extrinsic feedback when the system becomes more heavily loaded (even performs slightly worse when the number of users goes beyond 19). But as indicated in Fig. 13 the gain achieved by the adaptive scheme tends to become bigger as SNR increases. The switching should be carried out adaptively according to various conditions, like the system load (the number of users K), path model (the number of paths L) and maximum Doppler frequency ($f_D T$), etc. For the configuration presented in Fig. 13 ($K = 15, L = 3, f_D T = 0.01$), the 4-stage non-extrinsic and 3-stage extrinsic switching was shown to be efficient. However, as the number of K and/or L increases, more iterations of decoding without extrinsic feedback are needed in order for the MAI and ISI to be more effectively cancelled [9], [16]. As $f_D T$ increases, the system performance will degrade. However, its influence on the convergence speed of interference cancellation process is not as prominent as other parameters like K and L [16].

The initial LLRs are statistically independent in the iterative decoding process, however, since the decoders use indirectly the same information, the improvement through the iterative process becomes marginal, as the LLRs become more and more correlated. The convergence property of the iterative decoding algorithms is examined in Fig. 14. One can observe from the figure that iterative decoding without extrinsic information converges faster than the one with extrinsic information. Clearly, when exploited properly, extrinsic information improves the system performance, especially when SNR and

Fig. 10. Performance of extrinsic correction/normalization schemes in iterative decoding.

Fig. 12. Performance of adaptive iterative decoding as function of K .

Fig. 11. Performance of different iterative decoding schemes as function of SNR.

Fig. 13. Performance of adaptive iterative decoding as function of SNR.

the number of iterations increases. In both cases, 6 or 7 stages would suffice for maximum achievable performance.

VII. CONCLUSIONS

In this paper, we presented an integrated approach to iterative multiuser detection, decoding and channel estimation for convolutionally coded and orthogonally modulated asynchronous CDMA systems in multipath Rayleigh fading channels. In addition to be used as extrinsic information for the inner decoder, tentative decisions can be made on the output of the outer decoder for MAI and ISI cancellation to improve the performance of the inner soft decoder. Decision directed channel estimation was also proposed for multipath RAKE combining before decoding is done. Inner decoding can be done with or without extrinsic information. In the latter case, the performance improvement at each iteration is due

to improved interference cancellation and channel estimation with decision feedback. The soft extrinsic information was found to have reduced reliability in bad channels (when the system is heavily loaded). Some extrinsic correction and non-extrinsic/extrinsic adaptation schemes were proposed to reduce the detrimental effect of the interference. The numerical results show that the inner decoding with corrected extrinsic feedback or with non-extrinsic/extrinsic adaptation outperforms the one without extrinsic feedback and that inner decoding with MAI and ISI cancellation is much superior to the conventional single user decoding.

REFERENCES

- [1] M. Valenti. "Turbo codes and iterative processing". *IEEE New Zealand Wireless Commun. Symp.*, Nov. 1998, proceedings pages unnumbered.
- [2] J. Hagenauer. "The Turbo principle: tutorial introduction and state of the art". *Proc. International Symposium on Turbo Codes*, pp. 1-11, Sept. 1997.

Fig. 14. Convergence property of the iterative decoding schemes (21-user case).

- [3] J. Hagenauer. "Forward error correcting for CDMA systems". *Proc. ISSSTA'96*, pp. 566-669, Sept. 1996.
- [4] S. Benedetto, D. Divsalar, G. Montorsi, F. Pollara. "Serial concatenation of interleaved codes: performance analysis, design, and iterative decoding". *IEEE Transactions on Information Theory*, vol. 44, no. 3, pp. 909-926, May 1998.
- [5] J. Hagenauer, E. Offer, L. Papke. "Iterative decoding of binary block and convolutional codes". *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 429-445, March 1996.
- [6] R. Herzog, J. Hagenauer, A. Schmidbauer. "Soft-in/soft-out Hadamard despreader for iterative decoding in the IS-95(A) system". *Proc. Vehicular Technology Conference*, vol 2, pp. 1219-1222, May 1997.
- [7] L. Boulet, H. Leib. "Decoding of the IS-95 uplink code". *IEEE Communication Letters*, vol. 5, no. 5, pp. 206-208, May 2001.
- [8] E. Ström, S. Miller. "Iterative demodulation and channel estimation of orthogonal signalling formats in asynchronous DS-CDMA systems". *IEICE Transactions on Electronics*, vol. E85-C, no. 3, pp. 442-451, March, 2002.
- [9] P. Xiao, E. Ström. "Multiuser detection and channel estimation algorithms for M-ary DS-CDMA systems in multipath Rayleigh fading channels". *Proc. PIMRC'2003*, vol. 2, pp. 1829-1834, Sept. 2003.
- [10] X. Wang, H. Poor. "Iterative (Turbo) soft interference cancellation and decoding for coded CDMA". *Transactions on Communications*, vol. 47, no. 7, pp. 1046-1061, July 1999.
- [11] H. Gamal, E. Geraniotis. "Iterative multiuser detection for coded CDMA signals in AWGN and fading channels". *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 1, pp. 30-41, Jan. 2000.
- [12] M. Valenti, B. Woerner. "Iterative multiuser detection for convolutionally coded asynchronous DS-CDMA". *Proc. IEEE Int. Symp. on Personal, Indoor, and Mobile Radio Comm. (PIMRC)*, pp. 213-217, Sept. 1998.
- [13] L. Bahl, J. Cocke, F. Jelinek, J. Raviv. "Optimal decoding of linear codes for minimizing symbol error rate". *IEEE Transactions on Information Theory*, vol. IT-20, pp. 284-287, 1974.
- [14] P. Robertson, P. Hoeher, E. Villebrum. "Optimal and sub-optimal maximum a posteriori algorithms suitable for Turbo decoding". *European Transactions on Telecommunications*, vol. 8, no. 2, pp. 119-125, March-April, 1997.
- [15] A. Viterbi. "An intuitive justification and a simplified implementation of the MAP decoder for convolutional codes". *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 260-264, Feb. 1998.
- [16] P. Xiao. *Iterative Detection, Decoding and Channel Parameter Estimation for Orthogonally Modulated DS-CDMA Systems*. PhD thesis, Department of Signals and Systems, Chalmers University of Technology, Gothenburg, Sweden. Jan. 2004. available at <http://www.s2.chalmers.se/~pxiao/Phd.Thesis.Xiao.pdf>
- [17] J. Proakis. *Digital Communications*, 4th edition, McGraw-Hill, 2000.
- [18] W. C. Jakes, Jr. *Microwave Mobile Communications*. John Wiley & Sons, New York, New York, 1974.

- [19] L. Papke, P. Robertson. "Improved decoding with the SOVA in a parallel concatenated (Turbo-code) scheme". *Proc. International Conference on Communication*, pp. 102-106, 1996.