J Sign Process Syst (2008) 53:349–365
DOI 10.1007/s11265-008-0234-2

Adaptive Duplicated Filters and Interference Canceller for DS-CDMA Systems

François Nougarou · Daniel Massicotte · Messaoud Ahmed-Ouameur

Received: 3 April 2008 / Accepted: 23 April 2008 / Published online: 8 July 2008

Abstract A low complexity multiuser detection (MUD) technique, the Adaptive Duplicated Filters and Interference Canceller (ADIC) (patent pending), is proposed in the DS-CDMA context. Of particular interest is the use of adaptive filters block (AFB) dedicated to each user with its respective input signals independent from other users' contributions. The AFBs are mixed with interference canceller block in a cascade arrangement. As shown in this paper, this MUD can outperform the Decision Feedback Soft MultiStage Interference Canceller (DF-Soft-MPIC) MUD with complexity reduction by a factor of 4 to 8 for the data payload throughput from 64 kbps to 384 kbps, respectively. In addition to performance and algorithmic description of the proposed MUD method, a VLSI implementation strategy and hardware resources evaluation are investigated; permitting to estimate the maximum number of users in FPGA devices with respect to WCDMA constraints. The present work proposes a low complexity MUD wherein an interesting trade-off between performance and implementation complexity is described.

Keywords Multiuser detection · Interference cancellation · Low complexity · WCDMA · FPGA implementation · VLSI architecture

F. Nougarou · D. Massicotte (✉) · M. Ahmed-Ouameur
Laboratory of Signals and Systems Integration
(LSSI, www.uqtr.ca/Issi),
Department of Electrical and Computer Engineering,
Université du Québec à Trois-Rivières,
3351, Boulevard des Forges, C.P. 500,
Trois-Rivières, QC G9A 5H7, Canada
e-mail: daniel.massicotte@uqtr.ca

F. Nougarou
e-mail: francois.nougarou@uqtr.ca

M. Ahmed-Ouameur
e-mail: ahmed.ouameur@uqtr.ca

1 Introduction

DS-CDMA (direct sequence code-division multiple access) systems represent a bandwidth efficient solution in order to fulfill the third generation (3G) of mobile wireless communication requirements [1, 2, 4]. Unlike TDMA or FDMA, in CDMA systems, the entire bandwidth is shared among all active users used at the same time by the plurality of active users, by associating at each user unique orthogonal pseudo-noise codes.

Theoretically, in uplink (from users to base station) case, DS-CDMA scheme permits to detected each active user data at the base station (BS) receiver. But in practice, this multiple access technique stays interference-limited. Indeed, the transmission channel destroys the orthogonality between all active users [3, 4], and the receiver at BS can not correctly dissociate users' information. This phenomenon is referred as multiple access interferences (MAI). In addition, inter-symbols interferences (ISI) also appear due to multipath channel and increase proportionally with transmission data-rate.

It is well known that, in order to combat those uplink interferences, the base station receiver has to be efficiently designed [4]. However, conventional receiver, such as Rake (commonly used in second generation—2G), ignores MAI and considers them as additive white Gaussian noise (AWGN) while detecting the users of interest [3, 4]. Accordingly, Rake receiver suffers substantially from performance degradation as the number of users and/or data throughput increases. Therefore, many suggestions in the literature provide efficient solution to overcome MAI and ISI through multiuser detection (MUD) schemes [1, 3–15].

The interest in multiuser signal processing for CDMA stemmed from Verdú's seminal work in [1, 5], where he proposed and analyzed the optimal uplink multiuser device:

the maximum likelihood sequence detector (MLSD). Unfortunately, this optimal MUD is prohibitively complex for real-time implementation; its complexity increases exponentially with the number of users. Therefore, over the last two decades, research in this area has focused on several uplink sub-optimal MUD solutions [6–24] whose design objective is to reach a good tradeoff between performance and complexity: (1) to combat MAI and ISI in order to reliably detect information data of possibly a larger number of active users; while (2) maintaining a complexity permitting a real-time implementation for this maximum number of users. This implementation complexity issue explains the reason why the Rake receiver is still present in 3G base stations [4]. Indeed, even though many sub-optimal MUD methods have been proposed, most of these techniques show high computational complexity and can not be feasible in real-time on commercial baseband processors used in the BS. Nevertheless, some work has been conducted regarding real-time implementation aspect of MUDs [16, 24].

One possible classification of these sub-optimum MUD methods is to assign them to one of the two major classes with respect to whether or not channel estimates are required. Namely, direct methods require channel estimates in terms of channel's attenuations and delays to perform the detection process [6–9]. On the other hand, indirect techniques resort to some adaptive process to design the receiver using the possibly available training information [10–13]. To improve the performance-complexity tradeoff, MUD methods based on direct and indirect process are proposed [14] (patented).

Owing to their reduced computational complexity, the indirect (adaptive) technique [10–13] will take the major part of this paper. The computational complexity saving inherent in most adaptive methods, such as LMS or other techniques [13, 28, 29], stems from simple multiplication–addition operations wherein matrix manipulations (matrix–matrix, matrix–vector multiplications and/or matrix inversion etc.) are not usually involved. This approach supports VLSI implementation techniques such as pipelining (e.g. [30]) to maximize the number of users in a single device. Matrix manipulations cost become more expensive in time varying channels wherein the channel attenuations' variations dictate the frequency of the MUD's parameter (correlation matrices for MAI suppression in parallel interference cancellation techniques for instance) update [4].

In [4], J. G. Andrews addresses uplink MUD methods based on Interference Cancellation (IC) as the best solution for the DS-CDMA uplink problems. Given previously estimated symbols, this direct approach aims at regenerating the interferences and reconstructing the contributions of each user by subtracting in turn these interferences from the received signal; resulting in refined contributions from which new symbols estimates are deduced. This process is carried for a limited but sufficient number of iterations (stages). There are two main IC structures, namely: (1) The Multistage Parallel IC (MPIC) which makes the interferences cancellation of each user in parallel through several stages [7, 9]; (2) The Successive IC-SIC, which cancels the interferences successively from the least corrupted user to the most damaged one [6–8]. In both cases, a soft decision can be computed at the output of every stage so that a more accurate symbol detection is taken—Soft-MPIC. Depending on the transmission conditions, one technique may outperform the other [7]. It is worth mentioning at the current stage that interference cancellation can be made at the bit rate with the use of auto-correlation codes matrix, for better performance, or at chip rate, for a complexity efficiency. These IC MUD methods based on auto-correlation codes matrix shows excellent performances in combating MAI at the expense of a huge complexity. For acceptable complexity level most works do not use auto-correlation matrix [4] wherein works on their VLSI implementation appears in [17–22]. In comparison to the conventional Rake receiver, Hagerman and al. showed in [18] the uplink capacity improvement for WCDMA when using a MPIC: a 40% system-level increase has been found in typical urban environment for a voice transmission (12.2 kbps). Authors estimated that this MUD is 5 times more complex than Rake receiver, doubling the total receiver complexity. This increasing receiver complexity represents the principal constraint to deploy the MUD technology on commercial BS networks. Complexity reduction with respect to high performance represents the goal of proposed MUD solution.

This paper presents an Adaptive Duplicated Filters and Interference Canceller (ADIC) based on a mixed direct–indirect adaptive filtering approach for MUD design. This approach consists of creating synthesized training signals based on the known channel parameters given in turn by the channel estimator (e.g. Correlator [26, 27]). These synthesized training signals can be generated at the receiver to adapt the filter coefficients following indirect adaptation method. ADIC is a multistage detector wherein each stage comprises a bank of filters followed by an IC. Using an appropriate adaptation rule (LMS using adaptive step size, e.g. [29]) the filter bank's coefficients are obtained using a synthesized training signal. To maintain a pipeline implementation structure by preserving a low data dependency, a non feedback structure has been applied to ADIC MUD to obtain the expected performance. This pipeline structure propriety has been fully exploited in the presented VLSI implementation strategy (e.g. [30]).

The paper is organized as follows; in Section 2 a DS-CDMA signal model is presented. This model takes into account both traffic and pilot signal transmissions. Section 3

is devoted to describe in details the proposed MUD, ADIC. In Section 4, a VLSI implementation strategy of ADIC MUD is describes. Performance and complexity comparisons between ADIC and the Decision Feedback Soft MPIC (DF-Soft-MPIC) MUD and hardware resource evaluation to implement ADIC MUD method in FPGA are postponed to Section 5, while Section 6 draws some brief conclusions.

2 DS-CDMA Signal Model

Consider an uplink data transmission from K mobile units to a base station. To simplify the notation, we consider one receiving antenna. The kth user's transmitted baseband signal, vehiculing the information sequence $\{b_k[n]\}_{n=1}^{N}$ with k=1,2,...,K, can be written as

$$x_k(t) = \sum_{n=1}^{N} A_k b_k[n] d_k(t - nT; n), \tag{1}$$

where N is the number of symbols, A_k the signal gain and $b_k[n]$ the nth symbol of duration T. $b_k[n]$ is assumed to belong to S such that $S=\{\pm 1\}$, for BPSK signals. $d_k(t;n)$ is the spreading (scrambling) sequence for the nth symbol given by

$$d_k(t;n) = \sum_{\ell=1}^{N_c} c_{k,\ell}[n]\psi(t - \ell T_c), \tag{2}$$

here $N_c \triangleq T/T_c$ determines the spreading factor (processing gain) with T_c the chip period, $c_{k,\ell}[n]$ is the ℓth element of the spreading sequence for the nth symbol where $c_{k,\ell}[n] \in \{(\pm 1 \pm j)/\sqrt{2}\}$. $\psi(t)$ is a unit energy pulse shaping filter, possibly a raised cosine filter. We define $h_k(t; n)$ to be the kth user' multipath channel corresponding to the nth symbol as

$$h_k(t;n) = \sum_{p=1}^{P_k} h_{k,p}[n]\delta\left(t - \tau_{k,p}\right), \tag{3}$$

where P_k is the number of paths, $h_{k,p}[n]$ the pth path's complex amplitude (attenuation), $\tau_{k,p}$ the pth path's propagation delay and $\delta(t)$ the Dirac impulse function.

From (1) and (3) the total received signal for the nth symbol at the base station including the pilot-transmission signal, $\overline{r}_{k,p}\left(t - \tau_{k,p}; n\right)$, is

$$y(t;n) = \eta(t;n) + \sum_{k=1}^{K}\sum_{p=1}^{P_k} \overline{r}_{k,p}\left(t - \tau_{k,p}; n\right) + \overline{r}_{k,p}\left(t - \tau_{k,p}; n\right), \tag{4}$$

where $\eta(t; n)$ denotes the additive white Gaussian noise (AWGN) with zero mean and double side spectral density of $N_0/2$. The pilot-transmission signal undergoes the same multipath channel as the information traffic bearing signal $\overline{r}_{k,p}\left(t - \tau_{k,p}; n\right)$ such that

$$\overline{r}_{k,p}(t;n) = \sum_{p=1}^{P_k} \overline{r}_{k,p}\left(t - \tau_{k,p}; n\right) = A_k b_k[n]\Theta_k(t;n), \tag{5}$$

with $\Theta_k(t; n)$ is the effective code $h_k(t; n) \otimes d_k(t; n)$. $\otimes$ is the temporal convolution between the kth user's spreading waveform (2) at symbol instant n and the multipath channel (3).

3 ADIC Method

Each stage of the proposed multistage MUD, named ADIC (Adaptive Duplicated filters Interference Canceller; patent pending), see Fig. 1, consists of two distinct blocks: the adaptive filters block (AFB) and the interference canceller block (ICB). Of particular novelty is the use of adaptive filters (AFB) per user. Once adapted, the AFB is duplicated over the rest of stages reducing considerably the implementation complexity: all the stages of the same user share the same AFB, designed to combat both MAI and ISI. For an effective MAI and ISI cancellation, ICB is used in a cascade arrangement (see Fig. 1). This block regenerates all or part of the users' contributions using the AFB outputs. Once the interference cancellation is performed over the received signal, the interference-free signal per user is fed to the next AFB stage. As regards the AFB, two operational phases are considered, namely the adaptation phase wherein the filters' coefficients are adapted, and the detection phase involving ICB.

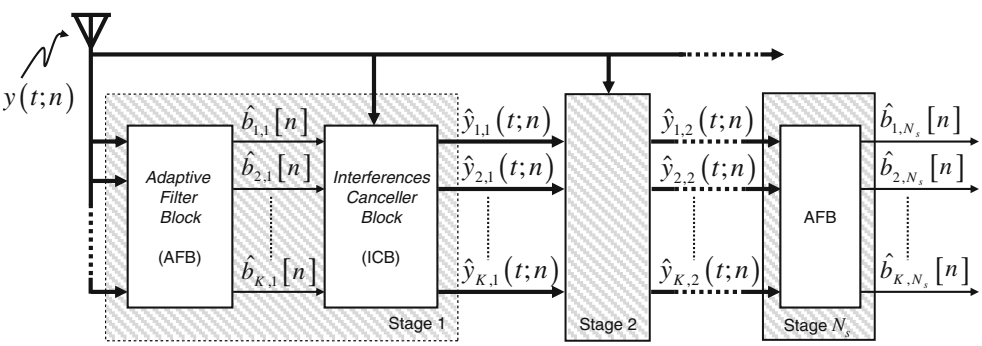

Figure 1 Detection phase of the proposed multistage MUD, for N_s stages.

3.1 Detection Phase

Being similar, we detail the detection phase flow of the signals within one stage only: the AFB and ICB blocks of stage s, where s=1,2,...,N_s (N_s the total number of stage). The AFB considers as inputs either the received signal (4), $\{y(t;n)\}_{n=1}^{N}$ if s=1, or $\{\hat{y}_{k,s-1}(t;n)\}_{n=1}^{N}$, the K estimated signals obtained from the previous stage s-1 if $s\neq1$. These inputs are sampled at a chip rate ($1/T_c$). The AFB's outputs are estimates of the traffic symbols ($1/T$) denoted by $\{\hat{b}_{k,s}[n]\}_{n=1}^{N}$. For a given user k, in order to describe the AFB operations, we adopt a vectorial representation of the AFB' estimated input $\{\hat{y}_{k,s-1}(t;n)\}_{n=1}^{N}$ from stage s-1, as

$$\hat{\mathbf{y}}_{k,s-1}[n]=\begin{bmatrix}\hat{y}_{k,s-1}(((n-1)N_w+1)T_c+\overline{\tau}_k;n)\\ \hat{y}_{k,s-1}(((n-1)N_w+2)T_c+\overline{\tau}_k;n)\\ \vdots\\ \hat{y}_{k,s-1}((nN_w)T_c+\overline{\tau}_k;n)\end{bmatrix}.$$

This vectorial representation corresponds to the simplest structure. For achieving such structure, one can consider one filter per path wherein each filter input is synchronized with an appropriate path delay, with N_w the vector dimension of the filter coefficients $\mathbf{w}_k$ as $\dim(\mathbf{w}_k)=N_{\mathbf{w}}\times1$ and $\overline{\tau}_k$ being a given path delay of the kth user's channel.

In the sth stage of the kth user, the corresponding raw AFB's output is

$$\widetilde{b}_{k,s}[n]=\mathbf{w}_k^{\mathrm{H}}\hat{\mathbf{y}}_{k,s-1}[n], \tag{6}$$

while the final output

$$\widehat{b}_{k,s}[n]=f\left(\widetilde{b}_{k,s}[n]\right), \tag{7}$$

wherein the kth user's filter coefficients in a vector form is $\mathbf{w}_k\triangleq[w_k(1),w_k(2),\ldots,w_k(N_{\mathbf{w}})]^{\mathrm{T}}$. $f(\cdot)$ (7) is a decision function, for example, the signum function in case of a hard decision function, or a tangent–hyperbolic or any other relevant function for a soft decision function (e.g. [6]). The final outputs depending to s corresponding to the estimate of the traffic information symbols are given by

$$\widehat{b}_{k,s}[n]=\begin{cases}f\left(\widetilde{b}_{k,s}[n]\right)=\tanh\left(\widetilde{b}_{k,s}[n]\right) & \text{if } s=1\\ f\left(\widetilde{b}_{k,s-1}[n],\widetilde{b}_{k,s}[n]\right)=\tanh\left(\left(\mathrm{sign}\left(\widetilde{b}_{k,s-1}[n]\right)+\mathrm{sign}\left(\widetilde{b}_{k,s}[n]\right)\right)/2\right) & \text{if } 1<s<N_s\\ f\left(\widetilde{b}_{k,s}[n]\right)=\mathrm{sign}\left(\widetilde{b}_{k,s}[n]\right) & \text{if } s=N_s\end{cases} \tag{8}$$

In the first stage, a tangent–hyperbolic function can be used. Such a function would softly limit the estimated information (binary) to within the pre-assumed safe dynamics. On the other hand at the last N_s stage, a hard decision is made. However, to delimit the flip-flop effect [6], for s=1,2,...,N_s, a decision function operates on the current and the previous filter outputs, namely, $\widetilde{b}_{k,s}[n]$ and $\widetilde{b}_{k,s-1}[n]$ from (8). In fact, the outcome from $\left(\mathrm{sign}\left(\widetilde{b}_{k,s-1}[n]\right)+\mathrm{sign}\left(\widetilde{b}_{k,s}[n]\right)\right)/2$ is −1, 1 or 0. If the outcome is + 1 or −1, this means that both the sth and (s-1)th stages agree that + 1 or −1 has been transmitted, respectively. On the other hand, a 0 outcome signals a flip-flop phenomena and the hard estimates are not involved in the ICB in the sth stage procedure (the related interference is not constructed nor eliminated which prevents an erroneous decision from propagating to the next (s+1)th stage. Applying the tangent–hyperbolic function on $\left(\mathrm{sign}\left(\widetilde{b}_{k,s-1}[n]\right)+\mathrm{sign}\left(\widetilde{b}_{k,s}[n]\right)\right)/2$ reduces to multiplying by 0.75, since $\tanh(\pm1)\backsimeq\pm0.75$. It is important to note that, if this decision function permits to improve interference cancellation performances at each stage, it also reduces the methods complexity by avoiding tangent–hyperbolic function calculations.

The data $\{\hat{b}_{k,s}[n]\}_{n=1}^{N}$ is the ICB inputs, Fig. 1. For a given stage s, the first role of the ICB is to construct the kth user's contribution $z_{k,s}(t;n)$ using

$$\begin{aligned}z_{k,s}(t;n)&=\hat{A}_k\,\hat{b}_{k,s}[n]\sum_{p=1}^{P_k}\hat{h}_{k,p}[n]d_k\left(t-nT-\hat{\tau}_{k,p};n\right)\\&=\hat{A}_k\,\hat{b}_{k,s}[n]\hat{\Theta}_k(t;n).\end{aligned} \tag{9}$$

This process is identical to (5). Unlike in (5), $\hat{A}_k$, $\hat{h}_{k,p}[n]$ and the delays $\hat{\tau}_{k,p}$, for k=1,2,...,K and p=1,...,P_k, are provided by a channel estimator, possibly a Correlator [27] or more performed method [26]. Accordingly, the total contribution from all the K users is given by the summation of all the users' contributions, such that:

$$Z_s(t;\,n)=\sum_{k=1}^{K}z_{k,s}(t;\,n) \tag{10}$$

Therefore, the kth user's interference can be deduced as:

$$\zeta_{k,s}(t;\,n)=Z_s(t;\,n)-z_{k,s}(t;\,n) \tag{11}$$

The next stage (s+1) input is built using the received signal (4) and the pre-estimated interference as

$$\hat{y}_{k,s}(t;n) = y(t;n) - \left(\sum_{k=1}^{K} z_{k,s}(t;n) - \hat{A}_k \hat{b}_{k,s}[n]\hat{\Theta}_k(t;n)\right), \tag{12}$$

where $\{\hat{y}_{k,s}(t;n)\}_{n=1}^{N}$ constitute the estimates of the received spread spectrum signals, essentially free from MAI and ISI.

3.2 Adaptation Phase

This phase consists of computing the filter coefficients $\mathbf{w}_k$ for k=1,2,...,K, on the basis of one filter per user. Upon convergence, $\mathbf{w}_k$ represents to some extent the inverse of effective codes, $\Theta_k(t; n)$ (5). Of interest is the fact that $\mathbf{w}_k$ is aimed to be much shorter than $\Theta_k(t; n)$ which saves considerable computation complexity. The adaptation phase is applied in the first stage only to compute the filter coefficients. These coefficients are duplicated on the next stage reducing considerably the adaptation complexity of the MUD.

Before describing the coefficient adaptation process, we need to construct a training data, Fig. 2. Indeed, for our method, existing commercial DS-CDMA systems (WCDMA and cdma2000) do not give access to pre-known or training data [2]—with the exception of pilot bits—in order to adjust the filter coefficients. It is important to note that, to assure the convergence, the filters need more than the already-available pilot bits to track channel variations as in fast fading context. Therefore, we may resort to synthesizing such training data along with a received signal using the estimated channel impulse response as follow [14] (patented):

1. randomly (or using a given distribution), we draws some training symbols $b_k^{\text{synth}}[n']$, per user k, from the same alphabet set as the original traffic symbols, S; n'= 1,2,...N^{synth}, N^{synth} being the training sequence length;
2. using pre-estimated channel parameters, $\hat{A}_k$, $\hat{h}_{k,p}[n]$ and $\hat{\tau}_{k,p}$, like in (12), we synthesize a received signal $y_k^{\text{synth}}(t;n')$ per user k as

$$\begin{aligned} y_k^{\text{synth}}(t;n') &= \bar{r}_k^{\text{synth}}(t;n') + \bar{r}_k^{\text{synth}}(t;n') + \eta^{\text{synth}}(t;n') \\ &= \hat{A}_k \hat{b}_k^{\text{synth}}[n']\hat{\Theta}_k(t;n') + \bar{r}_{k_{\text{pilot}}}^{\text{synth}}(t;n') + \eta^{\text{synth}}(t;n') \end{aligned} \tag{13}$$

In fact, the training data are synthesized using the channel model, we have replaced the real sample index n (Section 2) by n' to show synthetics sampling which have no constraint or dependence on the real time of transmitted data. As shown in (13), $y_k^{\text{synth}}(t;n')$ contains traffic, pilot and noise contributions. $b_k^{\text{synth}}[n']$ is of length N^{synth} symbols per user. With $y_k^{\text{synth}}(t;n')$ and $b_k^{\text{synth}}[n']$, the adaptation process can start.

In the short-code WCDMA context, the traffic spreading sequence—chip-by-chip multiplied scrambling and OVSF (Orthogonal Variable Spreading Factor) channelization codes—is 256 chips long [2]. In WCDMA, the spreading factors N_c or OVSF of 16, 8 and 4 correspond to the payload data throughput of 64, 144 and 384 kbps, respectively. Therefore we consider N_{nc}=256/N_c effective codes; this holds assuming that the channel is constant during one pilot symbol duration. Hence, for each user k,

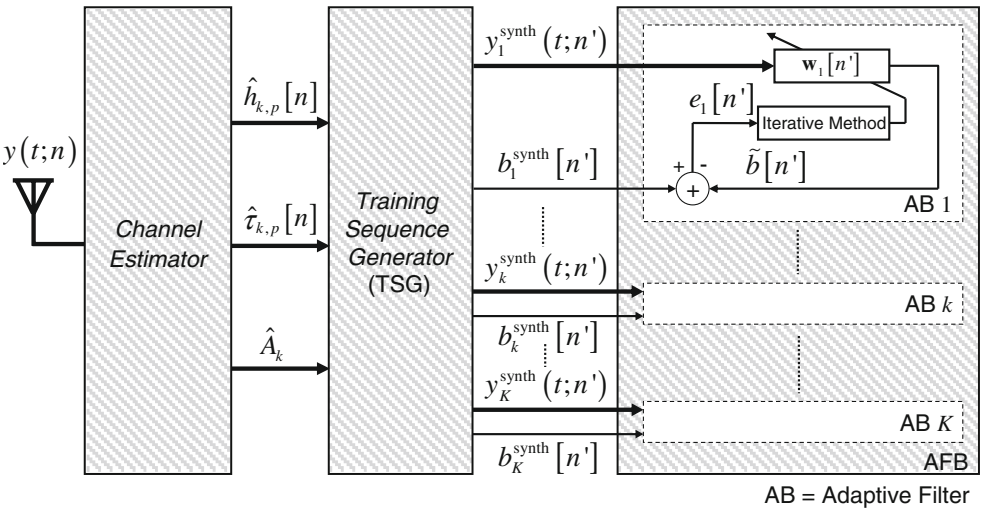

Figure 2 Adaptation phase of the proposed multistage MUD, for n'=1,2,…, N^{synth}, k=1,2,…,K.

$\mathbf{w}_k[n']$ consists of N_{nc} sub-filters, each aims to represent a short version of an inverse of the effective code. At first, we consider $N_{SF}=2N_c$ to be the length of each sub-filter, which yields a total filter length of $N_{\mathbf{w}}=N_{nc}N_{SF}$. So one can write

$$\mathbf{w}_k[n'] \triangleq \left[\mathbf{w}_{k,1}^{\mathrm{T}}[n'], \mathbf{w}_{k,2}^{\mathrm{T}}[n'], \ldots, \mathbf{w}_{k,v}^{\mathrm{T}}[n'], \ldots, \mathbf{w}_{k,N_{nc}}^{\mathrm{T}}[n']\right]^{\mathrm{T}}, \tag{14}$$

where $\mathbf{w}_{k,v}[n']$ is the sub-filter corresponding to the n'th training symbols, with $1 < v \triangleq \mathrm{mod}\left(n', N_{nc}\right) \leq N_{nc}$, mod(·) represents the modulo operator. The above specialization to short-code WCDMA signaling is extended to $y_k^{\mathrm{synth}}\left(t; n'\right)$, which is considered in a vector form as

$$\mathbf{y}_k^{\mathrm{synth}}\left[n'\right] = \begin{bmatrix} y_k^{\mathrm{synth}}\left(\left(\left(n'-1\right)N_{SF} - \frac{N_{SF}}{4} + 1\right)T_c + \overline{\tau}_k; n'\right) \\ y_k^{\mathrm{synth}}\left(\left(\left(n'-1\right)N_{SF} - \frac{N_{SF}}{4} + 2\right)T_c + \overline{\tau}_k; n'\right) \\ \vdots \\ y_k^{\mathrm{synth}}\left(\left(n'N_{SF} - \frac{N_{SF}}{4}\right)T_c + \overline{\tau}_k; n'\right) \end{bmatrix}. \tag{15}$$

It can be observed that the term $N_{SF}/4$ centers $\mathbf{y}_k^{\mathrm{synth}}\left[n'\right]$ with the n'th symbol of the kth user, in order to take into account of inter-symbols interferences (ISI). Of course, those representations of $\mathbf{w}_k[n']$ (14) and $\mathbf{y}_k^{\mathrm{synth}}\left[n'\right]$ (15) are more precisely used in ADIC method.

Coefficient adaptation can be implemented using many adaptive techniques (e.g. [28]). Set membership normalized LMS (SM-NLMS) possesses a good performance-complexity trade-off, at a convergence speed superior to the mother technique, NLMS [29]. The SM-NLMS algorithm has been considered in DS-CDMA context for multiuser detection [16], for channel estimation [17] and in order to estimate the interference power [13]. Of importance is the incorporation of a self adapting mechanism for the step-size, the adaptation, for a given user k, is described by

$$e_{k,v}\left[n'\right] = b_k^{\mathrm{synth}}\left[n'\right] - \mathbf{w}_{k,v}\left[n'\right]^{\mathrm{H}}\mathbf{y}_k^{\mathrm{synth}}\left[n'\right], \tag{16}$$

$$\mathbf{w}_{k,v}\left[n'+1\right] = \mathbf{w}_{k,v}\left[n'\right] + \mu_{k,v}\left[n'\right]\frac{e_{k,v}\left[n'\right]\mathbf{y}_k^{\mathrm{synth}}\left[n'\right]}{\mathbf{y}_k^{\mathrm{synth}}\left[n'\right]^{\mathrm{H}}\mathbf{y}_k^{\mathrm{synth}}\left[n'\right]}, \tag{17}$$

$$\mu_{k,v}\left[n'\right] = \begin{cases} 1-\lambda/\left|e_{k,v}\left[n'\right]\right|, & \text{if} \quad \left|e_{k,v}\left[n'\right]\right| > \lambda \\ 0, & \text{elsewhere} \end{cases}, \tag{18}$$

wherein $e_{k,v}[n']$ is the error for the n'th bit at the vth sub-filter output, $\mu_{k,v}[n']$ is dynamically conditional to a preset value of λ. Notice that (18) establishes two facts: (1) the term $1-\lambda/\left|e_{k,v}\left[n'\right]\right|$ is always less that 1 if $\left|e_{k,v}\left[n'\right]\right| > \lambda$ so that SM-NLMS is inherently stable; (2) otherwise $\mu_{k,v}[n']$ is set equal to 0 which alleviate some computational burden. Note that the SM-NLMS method complexity is lower than the NLMS algorithm considering the possibility of no coefficient is updated when $\mu_{k,v}[n']=0$. Indeed, as shown in (19) by replacing $\mu_{k,v}[n']$ in (17) by its expression in (18) when $\left|e_{k,v}\left[n'\right]\right| > \lambda$, the division operator in (18) disappears:

$$\begin{aligned} \mathbf{w}_{k,v}\left[n'+1\right] &= \mathbf{w}_{k,v}\left[n'\right] + \left(1 - \frac{\lambda}{\left|e_{k,v}\left[n'\right]\right|}\right)\frac{e_{k,v}\left[n'\right]\mathbf{y}_k^{\mathrm{synth}}\left[n'\right]}{\mathbf{y}_k^{\mathrm{synth}}\left[n'\right]^{\mathrm{H}}\mathbf{y}_k^{\mathrm{synth}}\left[n'\right]}, \\ &= \mathbf{w}_{k,v}\left[n'\right] + \left(e_{k,v}\left[n'\right] - \mathrm{sign}\left(e_{k,v}\left[n'\right]\right)\lambda\right)\frac{\mathbf{y}_k^{\mathrm{synth}}\left[n'\right]}{\mathbf{y}_k^{\mathrm{synth}}\left[n'\right]^{\mathrm{H}}\mathbf{y}_k^{\mathrm{synth}}\left[n'\right]}. \end{aligned} \tag{19}$$

Thus only one scalar division is used to update the vector of coefficients. As we can explain in next section, this division is not applied at each update to save hardware resources.

Finally, after convergence of the filter coefficients, $\mathbf{w}_k[N^{\mathrm{synth}}]$, at the synthetic sample or iterations N^{synth} are used by the detection phase as $\mathbf{w}_k=\mathbf{w}_k[N^{\mathrm{synth}}]$.

4 Implementation Description

In WCDMA, the received signal, $y(t)$, is composed of 4×10 ms-frames per block to compute the block error rate after decoding. Each 10 ms-frame has 38400 chips. These frames are divided in 15 slots of 38400/15=2,560 chips. Their duration is 10 s/15≈667 μs. Figure 3 shows the block diagram of ADIC MUD method. Three phases are used to estimate the transmitted data from K users, k=1,...,K:

- The channel estimation provides all channel amplitudes, $\hat{h}_k(t;n)$, and delays propagation, $\widehat{\tau}_k$, necessary to the ADIC adaptation phase. The channel estimation is not the object of this paper; a Correlator can be used instead but other methods can be used to boost the MUD performance (e.g. [26]);
- During ADIC's adaptation phase, (13)–(18), (using the channel information) the effective code, $\widehat{\Theta}_k(t;n)$, for all k, is constructed and the corresponding coefficient of each user, $\mathbf{w}_k$, is performed;
- Finally the detection phase, (6)–(12), suppresses interferences with its multistage arrangement and returns estimated symbols, $\left\{\widetilde{b}_{k,N_s}[n]\right\}_{n=1}^{N}$, N_s being the last stage.

These three phases are implemented to respect the timing constraint. We refer to [26] for implementation of channel estimation. As shown in Fig. 3, the latency is two slots; at the third slot all phases work concurrently until the end of the received data.

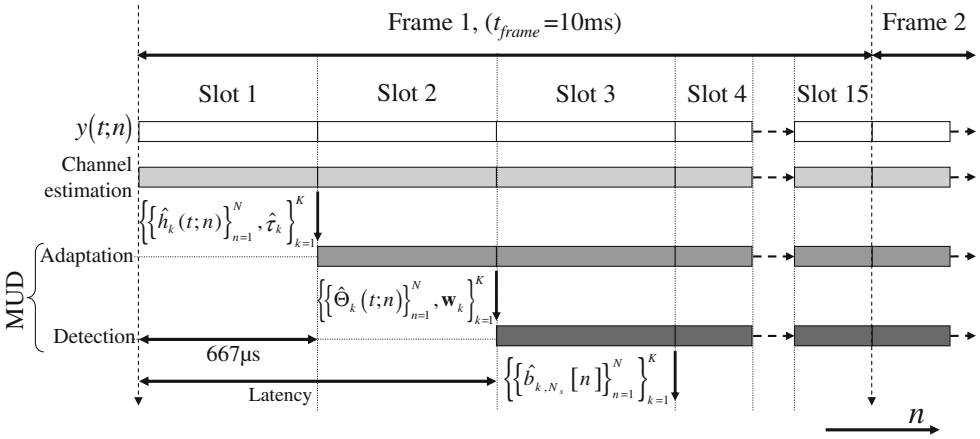

Figure 3 Timing diagram of ADIC MUD method.

4.1 Detection Phase

In this phase, we assume to have access to the filter coefficients, $\mathbf{w}_k=\mathbf{w}_k[N^{synth}]$, computed from the previous adaptation phase. Figure 4 describes the procedure for ADIC detection phase for the kth user following pipeline structure composed of three processing elements (PE). Each PE is shown in Fig. 5. We consider $N_s=3$ stages; as shown in performance analysis Fig. 10, three stages represent the best ADIC MUD performance-complexity trade-off. To reduce necessary memory size and localized the data communications in the respective PE, we divided N ($N=2560/N_c$—the number of data per frame) in Q sequences of length N_p, with $N_p=16$ and $Q=N/N_p$. We can write the data sequence $\left\{\widetilde{b}_{k,s}[n]\right\}_{n=1}^{N}$ as a vector $\widetilde{\mathbf{b}}_{k,s}$ by a concatenation of sub-vectors $\widetilde{\mathbf{b}}_{k,s,q}[i_q]$

$$\widetilde{\mathbf{b}}_{k,s}=\left[\widetilde{\mathbf{b}}_{k,s,1}^{T}[i_1],\ \widetilde{\mathbf{b}}_{k,s,2}^{T}[i_2],\ \ldots,\ \widetilde{\mathbf{b}}_{k,s,q}^{T}[i_q],\ \ldots,\ \widetilde{\mathbf{b}}_{k,s,Q}^{T}[i_Q]\ \right]^{T}, \tag{20}$$

where $\widetilde{b}_{k,s,q}[i_q]=[\widetilde{b}_{k,s,q}[i_{q,1}],\ \widetilde{b}_{k,s,q}[i_{q,2}],\ \ldots,\ \widetilde{b}_{k,s,q}[i_{q,n_p}],\ \ldots,\ \widetilde{b}_{k,s,q}[i_{q,N_p}]\ ]$ and $i_q=\left\{i_{q,n_p}\right\}_{n_p=1}^{N_p}$ with $i_{q,n_p}=(q-1)N_p+n_p$, for $q=1,2,\ldots,Q$ and $n_p=1,2,\ldots,N_p$.

In Fig. 4, the detection filter block (FB), for each stage, s, and each partition, q, of kth user, uses the same coefficients $\mathbf{w}_k$ and corresponds to

$$\widetilde{b}_{k,s,q}[i_{q,n_p}]=\mathbf{w}_k^{\mathrm{H}}\hat{\mathbf{y}}_{k,s-1,q}[i_{q,n_p}]. \tag{21}$$

Detection filter block consists in a PE presents in Fig. 5a.

The spreading block (SB) PE executes the equation (22). This PE is described in Fig. 5b where the block $\mathrm{B}_{\mathrm{add1}}$ consists of 5 parallel adders.

$$\begin{aligned} z_{k,s,q}(t;i_{q,n_p})&=\widehat{b}_{k,s,q}[i_{q,n_p}]\widehat{\Theta}_k(t;i_{q,n_p})=f\left(\widetilde{b}_{k,s,q}[i_{q,n_p}]\right)\widehat{\Theta}_k(t;i_{q,n_p}),\\ &=\widehat{b}_{k,s,q}[i_{q,n_p}]h_k(t;i_{q,n_p})\otimes d_k(t;i_{q,n_p}). \end{aligned} \tag{22}$$

A look up table (LUT) is employed to represent the tangent–hyperbolic function of the decision function $f(\cdot)$, equation (8). Furthermore, to compute $\widehat{\Theta}_k(t;i_{q,n_p})$, a multiplier free design can be used considering that $\{d_k(t;i_{q,n_p})\}$ as a sequence of ±1.

Expressions (23)–(25) represent operations of the interference canceller block, ICB. As we see in Fig. 5c, the

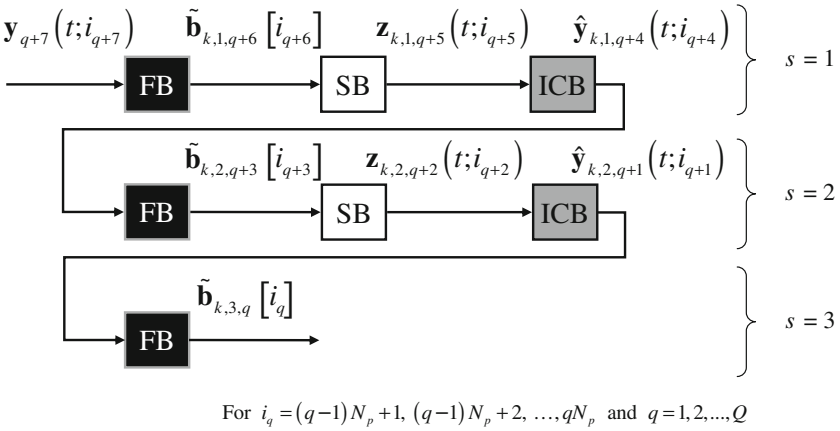

Figure 4 Procedure of ADIC detection phase for 3 stages ($N_s=3$) and a user k.

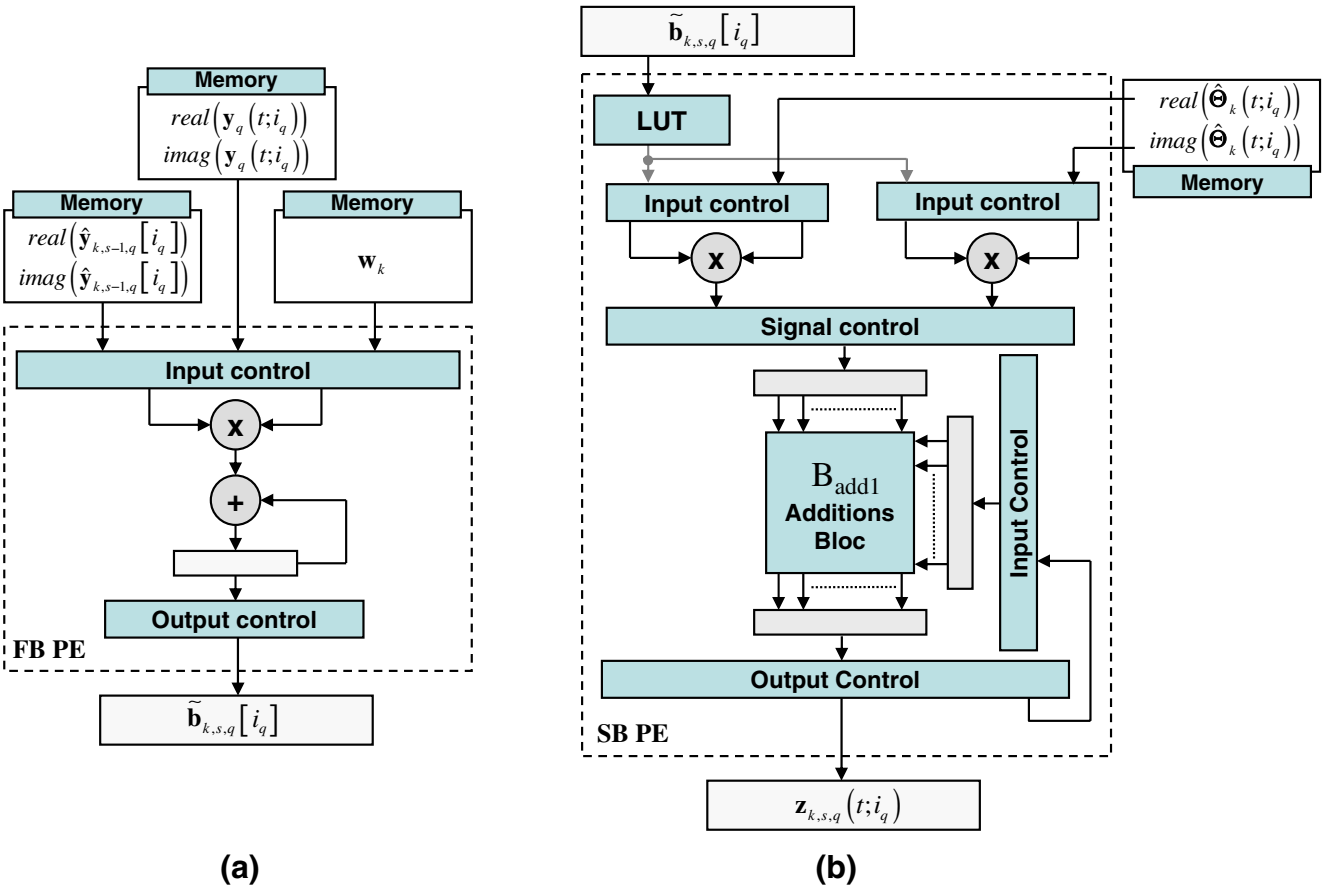

Figure 5 Hardware resources description for ADIC detection phase: **a** FB PE, **b** SB PE and **c** IC PE.

corresponding ICB PE is only realized with 5 parallel adders, B_{add2}.

$$Z_{s,q}\left(t;i_{q,n_p}\right)=\sum_{k=1}^{K} z_{k,s,q}\left(t;i_{q,n_p}\right), \tag{23}$$

$$\xi_{k,s,q}\left(t;i_{q,n_p}\right)=z_{k,s,q}\left(t;i_{q,n_p}\right)+Z_{s,q}\left(t;i_{q,n_p}\right), \tag{24}$$

$$\hat{y}_{k,s,q}\left(t;i_{q,n_p}\right)=y_q\left(t;i_{q,n_p}\right)+\xi_{k,s,q}\left(t;i_{q,n_p}\right). \tag{25}$$

Notice that on Figs. 4 to 7, vectors with indices i_q, and not i_{q,n_p}, are considered in order to represent groups of N_p data.

The 3D graph in Fig. 6, shows the data flow of the ADIC detection structure as a function of PE, stages (s) and for the kth user, we have: (1) FB PEs share the same coefficient; (2) SB PEs share the same effective code; and (3) for the same stage s, ICB PEs share the sum in (23) for all users. Note that we repeat the same detection multistage structure for all users, event if signals are different.

Detection phase timing diagram in Fig. 7, based on the graph data dependency in Fig. 6, describes for the user k how the Q partitions are propagated in the detection structure, applying a pipeline process. The detection clock cycle, T_{clk}^{d}, is the same for each PE and is imposed by the slowest PE which depends to the considering N_c: FB PE for N_c=16 and SB for N_c=8 and 4. The pipeline is full at q=3 and at s-2 to have an estimated data at each clock cycle. It results in the 16 first estimated symbols. Finally, the latency and the throughput are $7T_{clk}^{d}$ and T_{clk}^{d} respectively.

4.2 Adaptation Phase

The adaptation phase consists on three operations as depicted in Fig. 8: (1) the effective code computation, $\widehat{\Theta}_k(t;n)$, for all K users, (2) the synthetic signal construction using (13) and (3) the coefficients' update using an adaptive method based on SM-NLMS [29], to return $\mathbf{w}_k$ (16)–(18). Figure 8 presents the timing diagram for these operations. The coefficients' adaptation is divided in three other sub-operations each as a PE (Fig. 9): (1) the adaptation filter block (FB_{adapt}) to compute the equations (6)–(8), (2) the error and step-size block (ESB) to compute

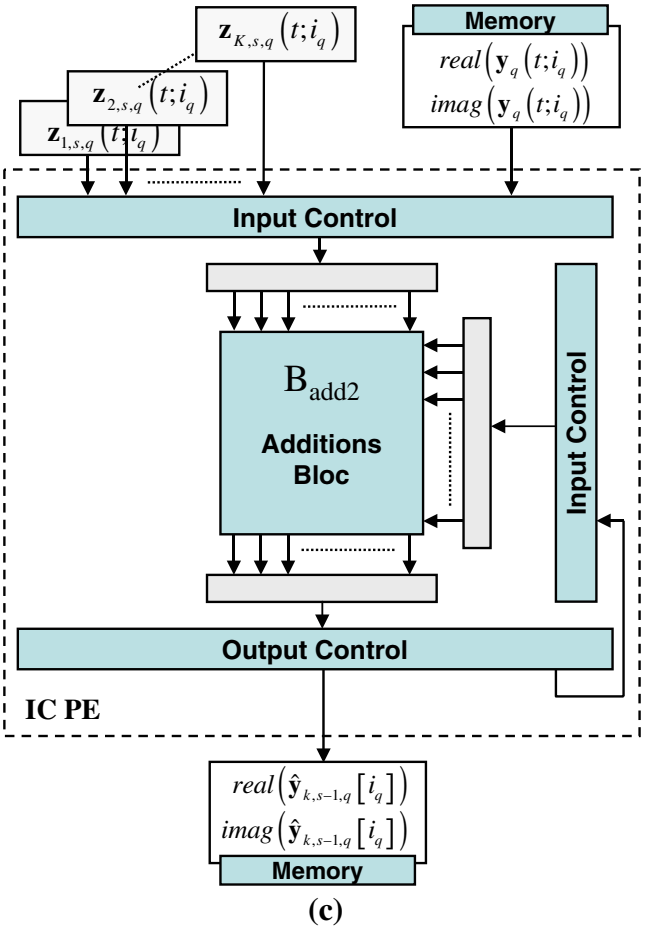

Figure 5 (continued).

(16) and (18), and (3) the update block (UB) for (17). Each of operators is executed by respective PE which have particular characteristics:

- FB_{adapt} PE, Fig. 9a, computes separately real and imaginary parts of considering data. In order to reduce hardware resources, this same PE is multiplexed to realize: (1) the effective code computation, and (2) bits estimation [right side of (16)] and (3) the synthetic signal norm calculation (19) of adaptive SM-NLMS treatment for one user.
- ESB PE, Fig. 9b, only computes the right side of the SM-NLMS method update expression (19) of the kth user. Considering implementation point of view, it is important to notice that the complex division operator present in this PE is not used at each n'th instant and can be multiplexed with other user. Thanks to $\mathbf{y}_k^{\text{synth}}[n']$ low dynamic magnitudes, instead of using the divisor N^{synth} times it can be used $(3N_c/256)N^{\text{synth}}$ times without performances loss for all user and all data rate.
- Figure 9c, the addition block presenting an arrangement of 3 parallel adders, B_{add3}, is the only one arithmetic operator of UB PE, witch permits to compute: (1) the synthesized received signal $\mathbf{y}_k^{\text{synth}}[n']$ from $b_k^{\text{synth}}[n']$ and $\widehat{\Theta}_k(t;n')$, and (2) new coefficients $\mathbf{w}_{k,v}[n'+1]$, using coefficients $\mathbf{w}_{k,v}[n']$, during SM-NLMS adaptation process (19). Once calculated, coefficients $\mathbf{w}_{k,v}[n'+1]$ replace the previous coefficients in the corresponding memory.

Applying a pipeline process, Fig. 8, the adaptation clock cycle, T_{clk}^a, is given by the slowest PE block. At the beginning, for $n'=1$, the FB is under operation. It results in a symbol estimate, $\mathbf{w}_{k,v}[n']^{\text{H}}\mathbf{y}_k^{\text{synth}}[n']$, [c.f. (16) for the kth user]. After T_{clk}^a, signals at $n'=2$ are available for BF and at $n'=1$ for ESB. This block gives $e_{k,v}[n']$ (16) and $\mu_{k,v}[n']$ (18). At the next T_{clk}^a, signals at $n'=3$ are available for BF, at $n'=2$ for ESB and at $n'=1$ for UB. This process permits to pipeline the architecture and continues until $n'=N^{synth}$. The latency and throughput are $2T_{clk}^a$ and T_{clk}^a respectively. There are K identical and independent adaptation process and structure, one per user. Noted that T_{clk}^a is independent of T_{clk}^d and each of them depend of the timing diagram shown in Fig. 3.

5 Simulation

5.1 Performance Results

Some experiments are conducted in a WCDMA environment. The simulation basic conditions are: pulse-shaping filter, $\psi(t)$, using Raised cosine with a roll off factor of 0.22; «Vehicular A» channel with $P_k=P=6$ paths; mobile speed of 3 km/h; carrier frequency of 2 GHz; one transmitting and one receiving antenna. Channel's amplitudes, $h_{k,p}$, are estimated by a Correlator [27] and channel's delays estimation $\tau_{k,p}$ are considered perfect. Note that the K pilot signals have been canceled at the receiver using a pilot cancellation process [14]. For the sake of reference and comparison, Rake and Decision Feedback Soft Multi-

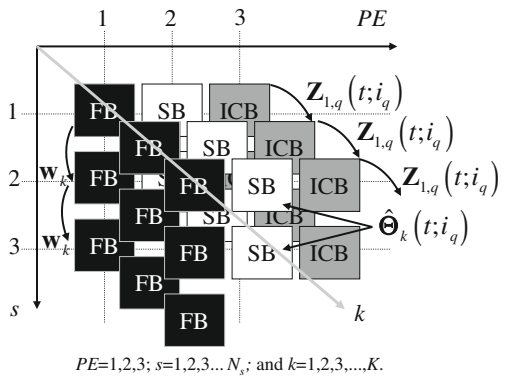

Figure 6 3-D data dependency graph of ADIC detection phase.

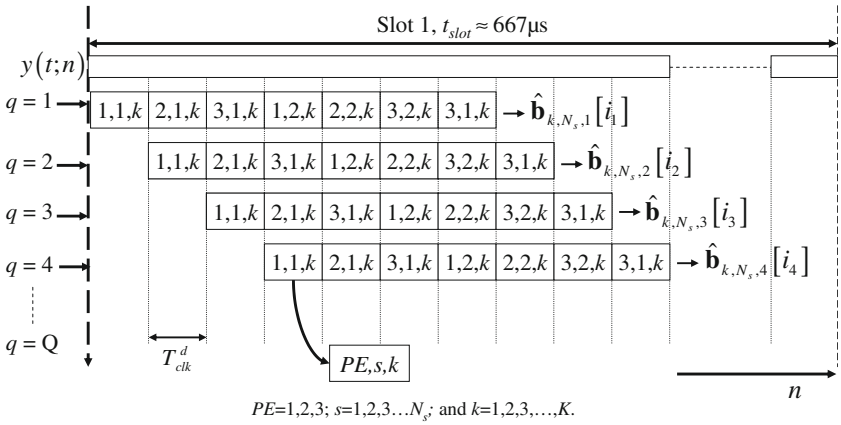

Figure 7 Timing diagram of ADIC detection phase for a user k.

stage Interference Canceller (DF-Soft-MPIC) [3, 4], using auto-correlation matrix with 5 stages, are included. Table 1 presents ADIC's parameters used for all simulations excepted if indicated in another way. Finally, in each simulation, 6000 data slots have been usually considered in order to generate satisfactory average raw bit error rate (BER) results. In our simulation, we consider that, for BER results under 5%, the decoder system following MUD is able to find the totality of transmitted data.

For N_c=16 (Fig. 10a) and N_c=8 (Fig. 10b), the fifth stage of ADIC gives BER equivalent or better than DF-Soft-MPIC. Performance-wise, ADIC MUD can be tailored to work with N_s=3 stages while maintaining good performance-complexity trade offs. It is worth mentioning that ADIC MUD provides the same results as the Rake at the first stage (N_s=1). It is an important interesting point knowing that this Rake output results can be used for other applications inside the BS such as the power control.

According to the mobiles speed, Fig. 11, MUD methods performances are degraded due to non-optimum performances of Correlator channel estimator. However ADIC is less sensitive than DF-Soft-MPIC. Indeed, more the speed increases, more the BER results of ADIC fourth stage is better than DF-Soft-MPIC fifth stage.

It is known that the commercial key component of MUD method consist to a low implementation complexity [4] to attain the desired performance. The adaptive approach proposed by ADIC make possible to fine tunes the performance-complexity tradeoff. The output sensitivity of ADIC and DF-Soft-MPIC methods with the periodicity to adapt the coefficients inside one time frame (1 frame=15 slots [2]) has been studied. When we changed the adapt time period from one slot to 15 slots, a lost of 0.35 dB and 0.45 dB have been observed for ADIC and DF-Soft-MPIC, respectively. For DF-Soft-MPIC, calculations of Rake and its matrix of auto-correlation represent the adaptation phase, and (13)–(18) for ADIC. ADIC is here, still, less sensitive, to obtain the same results as the DF-Soft-MPIC with 15 adaptations per frame, ADIC used only 10 adaptations per frame. In pedestrian and fast speed mobile unit contexts, we

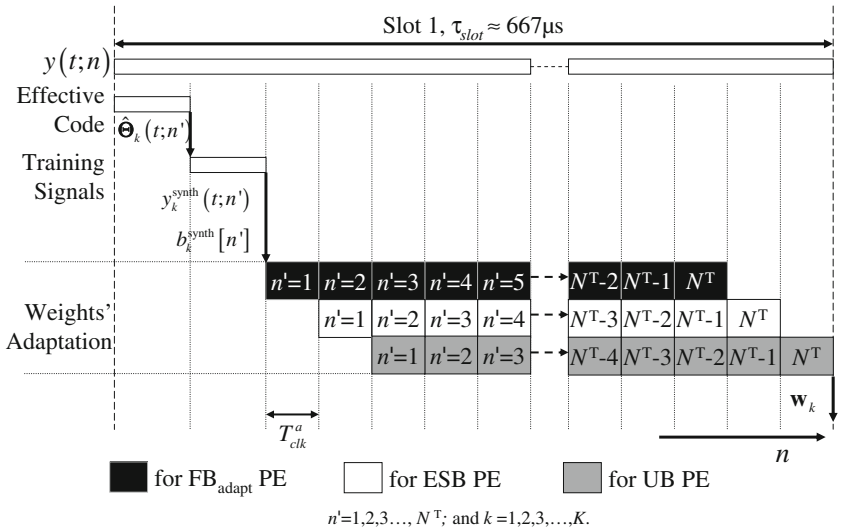

Figure 8 Timing diagram of ADIC adaptation phase N^T=N^{synth}.

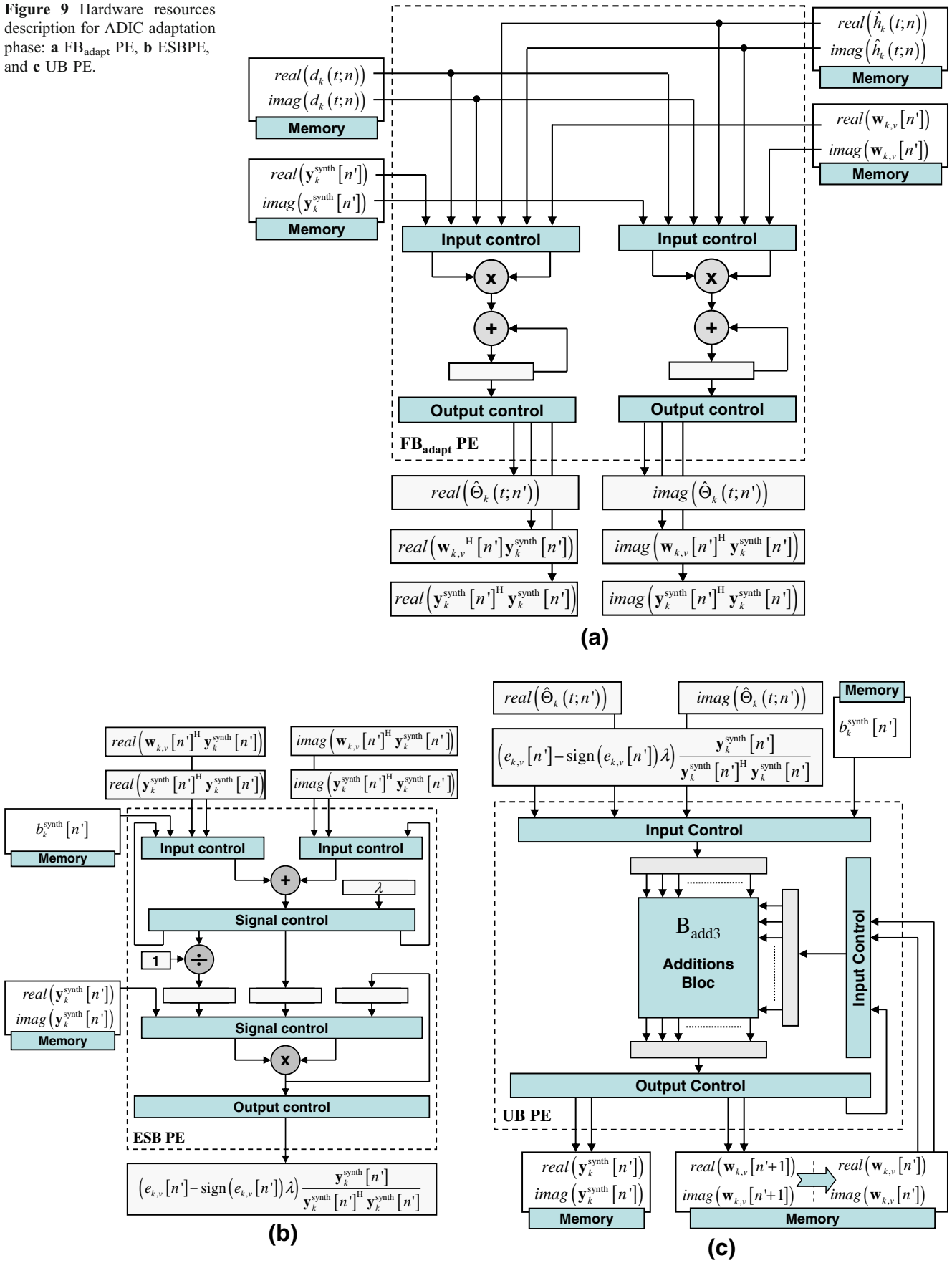

Figure 9 Hardware resources description for ADIC adaptation phase: **a** FB_{adapt} PE, **b** ESBPE, and **c** UB PE.

Table 1 ADIC's parameters for two different spreading codes.

OVSF	N_c=16 (64 Kb/s)	N_c=8 (144 Kb/s)
Symbols number in a slot (N)	$2560 \div N_c$=160	320
Sub-filters size (N_{SF})	$2N_c$=32	16
Iterations number per filter (N^{synth})	$3 \times N$=480	960
Iterations number per sub-filter	30	30
SM-NLMS parameter (λ)	0.005	0.005

can adapt the filter coefficients at each 15 slots and one slot, respectively.

5.2 Interest of the Proposed Adaptive Structure

To show the interest of ADIC adaptation phase structure and the use of SM-NLMS method has been studied to reduce the adaptation complexity and assure the convergence. For that, we introduced into simulations another adaptive MUD, AL-MMSE [10], based on a NLMS adaptation. To assure a convergence at pedestrian condition, AL-MMSE adaptation need a long training sequence of size N^{synth}=2400—5 times longer than ADIC one. AL-MMSE MUD adapts its K filters using the same received signal containing the K users contributions, contrary to ADIC which uses the user contribution corresponding to the filter considered (AB k in Fig. 2), in order to update each filter. Results, from two differently parameterized ADIC methods (λ=0.005 and 0.02), are presented in Fig. 12. As explained before, λ represents the error value from which the update will not be carried out. AL-MMSE method with its long training sequence performs less than the ADIC second stage for a complexity much higher (because of the necessary training sequence size) than ADIC. Moreover, the use of SM-NLMS adaptive method, dependently of the selected value λ, also allows important calculation savings. Indeed, compared to NLMS which uses all (100%) the training data for coefficient update, the SM-NLMS uses, with λ=0.005, 65% and, with λ=0.02, 55% of the update sequence (Eq. (17)), which is equivalent respectively to ≈20 and ≈17 iterations per sub-filter instead of the 30 iterations; an economy favorable for hardware implementation. These updated reductions are observed constant on all E_{b0} range of Fig. 10.

5.3 Complexity Analysis

In this section, we applied the approach used for a fair arithmetic complexity comparison, based on a complexity benchmark from a VLSI technology point of view such as FPGA and ASIC hardware implementation.

As a first step of the approach, it is necessary to compute the number of additions and multiplications. We consider the following parameters: N_c the spreading factor, N_h the maximum delay spread of the channel, P the number of path, N^{synth} the number of adaptive symbols in ADIC and m the MPIC parameter permitting to take into account ISI in its correlation matrix, $m = \lceil (N_c + N_h - 1)/N_c \rceil$, $\lceil \cdot \rceil$ being ceiling. Notice that these algorithms need a lot more additions than multiplication due to the presence of ±1 number in the algorithm execution. In our evaluation, we excluded the multiplication in presence of ±1 number. In order to make a fair arithmetic complexity comparison we use a unified framework for all these techniques by considering an elementary arithmetic unit used to realize

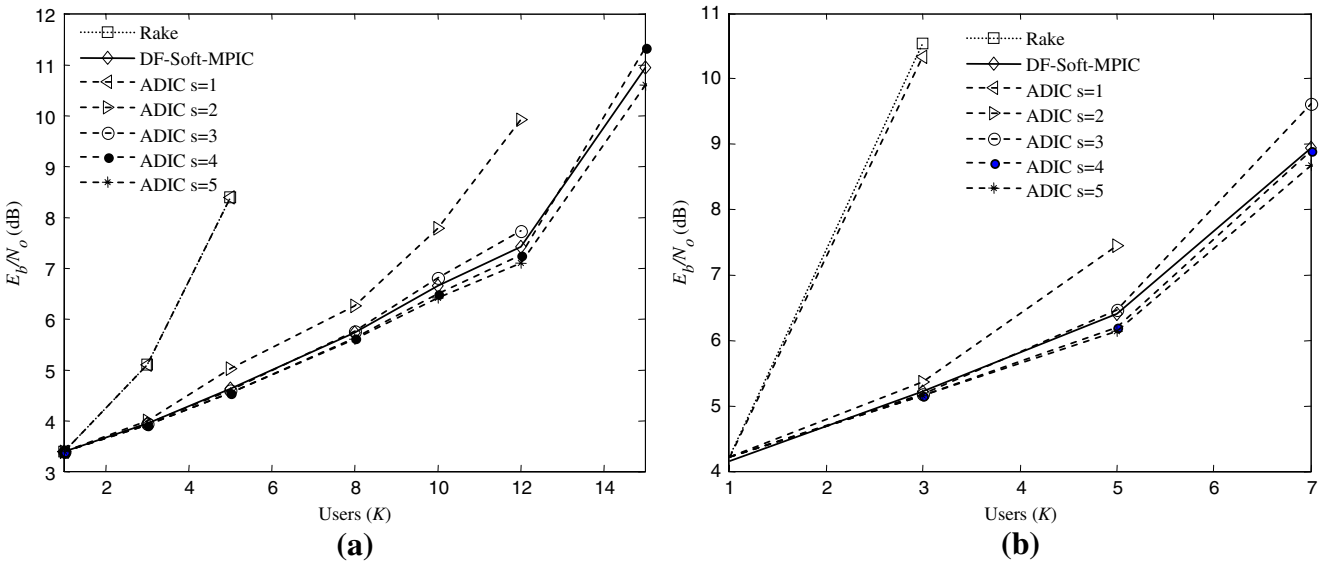

Figure 10 E_b/N_o (dB) versus users' number at 64 kbps **a** and 144 kbps **b**, at 3 kph, to obtain BER=5% with the Rake, DF-Soft-MPIC and ADIC.

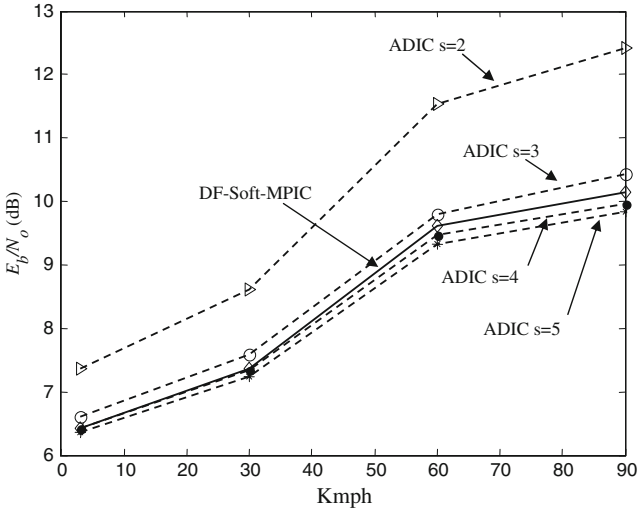

Figure 11 E_b/N_o (dB) versus mobiles speed for K=10 at 64 kbps to obtain BER=5% with DF-Soft-MPIC and ADIC.

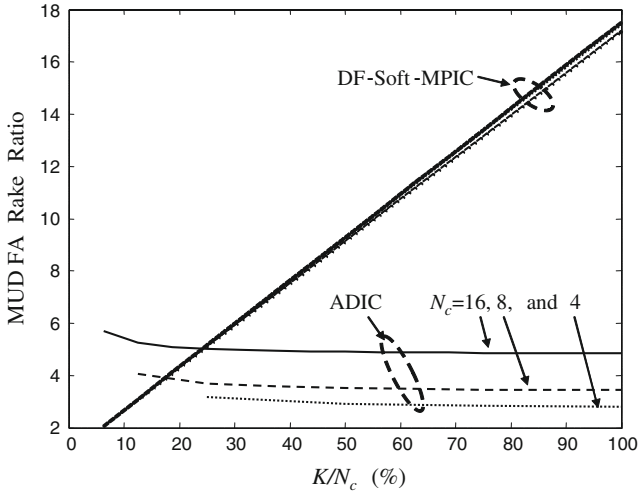

Figure 13 Required Number of FA for ADIC and DF-Soft-MPIC relative to the Rake receiver with 15 updates per frame, N_q=16 and N_s=3 bits for N_c={4,8,16}.

an adder and a multiplier, the number of full adder (FA). In a VLSI technology, multiplication and addition operations have the same binary structure with a bit word-length adjusted to assure the precision needed. We consider that an addition requires N_q FA and a multiplication N_q^2 FA, N_q being the number of bits needed to quantify each parameter of MUD studies.

At full-load BS receiver system, $K=N_c$, the required number of FA for ADIC and DF-Soft-MPIC relative to the Rake receiver for N_c=16, 8 and 4 is shown Fig. 13. For all methods, there are considered 15 update (adaptation phase) per frame (each slot), N_s=3 and N_q=16-bits. This result reveals that DF-Soft-MPIC is 34 times more complex than the conventional Rake receiver while ADIC is only 4.0 to 6.8 times more complex. For $K=N_c$, we can notice that ADIC presents a 4 to 8 complexity reduction compared to the DF-Soft-MPIC. ADIC presents a constant FA Rake ratio versus the number of simultaneously receive mobile users K.

5.4 Implementation Preliminary Results

In this section, we give some preliminary results about processing time and hardware resources estimation of the ADIC architecture described in previous section. Considering FPGA targeted technology integration, we drew each block architecture for detection (Fig. 5) and adaptation (Fig. 9) phases in term of additions, multiplications, multiplexers, registers, etc… We take into account N^{synth}=

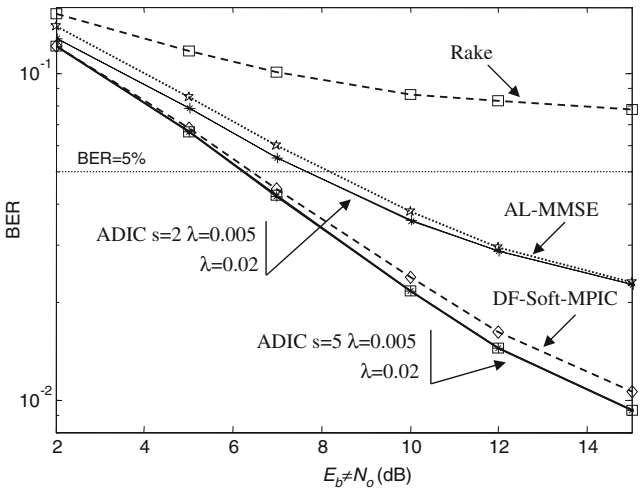

Figure 12 BER versus E_b/N_o with K=10 at 64 kbps and 3 kph, for ADIC with λ=0.02 and 0.005, DF-Soft-MPIC, the Rake and AL-MMSE receivers, simulated with 1500 data slots.

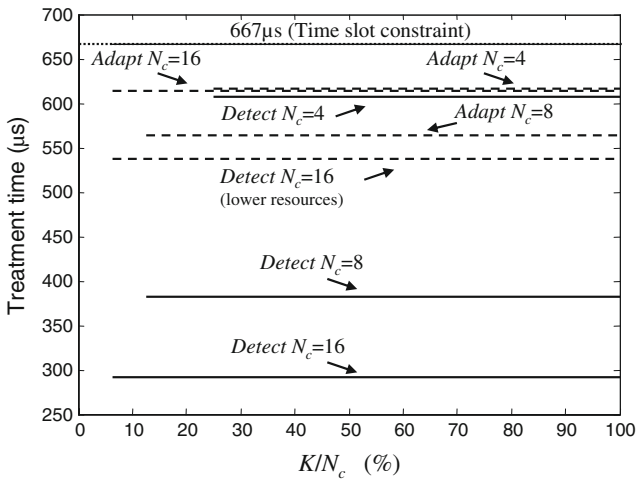

Figure 14 ADIC MUD treatment time for N_c={4,8,16}, for adaptation and detection phases and for a lower resources detection structure at N_c=16.

$3N$ for adaptation and N_s=3 for the detection. We evaluated ADIC in fixed-point bits and a word length of 16-bits is sufficient to keep the similar performances compare to floating-point with a lost of E_b/N_0 inferior to 0.1dB.

Assuming the pipeline implementation structure and that an addition and a multiplication operation can be respectively performed at a frequency of 200 MHz and 100 MHz [23], Fig. 14 presents processing time results for both adaptation and detection phases. Here, we consider for all N_c a full load receiver. From Fig. 14, one can draw the following remarks:

- For both ADIC phases and all N_c, processing times are lower than slot time, (reference time constrain);
- The processing time is independent of K because the resources grow with K (cf. Fig. 15);
- The adaptation process is lower than detection's one. Indeed, the two phases work in parallel, and share the same FPGA;
- The detection needs 3 FB, 2 SB and 2 ICB per user; it needs lots of FPGA embedded multipliers and slices. So we had to economize hardware to implant adaptation phase. This hardware economy reflects on adaptation processing time.

The arithmetic operations represent the most important hardware resources need to materialize the pipeline structure of ADIC considering the lowest complexity of control units that the previous proposed MUD [23]. These resources are shown in Fig. 15 presenting the total number of 16-bits adders and 16-bits multipliers, with respect to Fig. 14. For N_c=16, we need no more than 500 adders (7500 slices) in order to implement 16 users and 160 embedded multipliers are necessary. With these results and analysis of the required memory, and according to Virtex-II pro data sheet [25], ADIC MUD in full load integrated into a Virtex-II pro XC2VP40 family, which contains 19 392 slices and 192 embedded multipliers, accepts more users compared to the results proposed in [23].

From Fig. 15, we observed the hardware constrain imposed by the N_c=16 case and the low time consuming for detection phase. To decrease the hardware resources, a second detection structure has been proposed for 64 kbps with no impact for N_c=8 and 4. The modifications consist to take advantage of the PE regularities to time-multiplex the data computations: (1) use only one multiplier in the spreading block (SB) PE (Fig. 5b) and (2) decrease the number of parallel adders in B_{add1} and B_{add2} (Fig. 5c); 3 parallel adders instead of 5 in the both case. Of course the time processing of detection phase increases but those modifications permit, as shown in Fig. 15, to implement 16 users with 416 16-bits adders (6,240 slices) and 128 16-bits multipliers instead of 160 (20% reducing). In this case, a Virtex-II pro XC2VP30 family, which is constituted by 13,693 slices and 136 embedded multipliers, can be used to implement ADIC MUD for 64 kbps in full load.

5.5 Beyond the Arithmetic Complexity

Another important aspect to compare the implementation complexity is the algorithmic structure such as regularity, recursiveness, data flow, memory quantity and inherent parallelism—all qualities intrinsic to the non restrictive illustrative embodiments of the present invention.

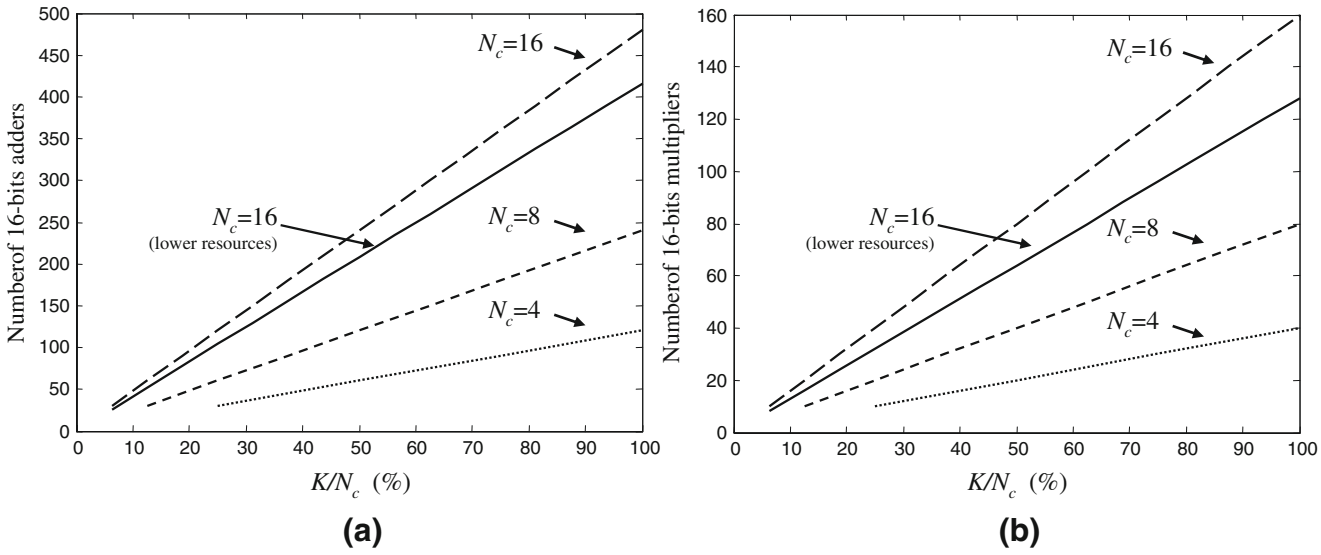

Figure 15 Total number of hardware arithmetic resources for ADIC: **a** adders and **b** multipliers for the proposed structure and a lower resources detection structure at N_c=16.

In this study, these aspects have not been included to compare MUD methods. However, an obvious consideration can be observed with the decision feedback structure of MUD. Indeed, even if the decision feedback structure might have relatively the same complexity level, the main drawback is the lack of parallelism that can be exploited, especially for the MPIC caused by data dependencies. In fact, a DF-Soft-MPIC at instant n and for user k needs to wait for all users so that the kth user proceeds to detect the current data before processing its own data. Such a structure looses its parallelism to apply pipeline or parallel techniques and to become serial operation limited for sequential DSP implementation. Hence, the DF-Soft-MPIC will always be limited by the DSP clock speed to respect the computational time imposed by the 3GPP time frame. Noted that, the present invention do not use decision feedback structure to exploit the parallel implementation techniques.

When $3 \leq N_s \leq 5$, it is worth mentioning that the ADIC can be optimized for a better performance-complexity trade-off. The performance represents the gains in dB saved to target a Bit Error Rate compared to the reference method and the complexity represents the implementation cost into VLSI technology such as DSP (Digital Signal Processor, FPGA—Field Programmable Gate Array, ASIC—Application Specific Integrated Circuit). Inherent to the illustrative embodiments of the present invention is a flexibility to tune the performance-complexity tradeoff based on the parameters such as N^{synth} and N_s. Compared to the most known technique, DF-Soft-MPIC, same performances in dB are obtained with less complexity in term of arithmetic implementations (see the results in the next section).

6 Conclusion

We proposed and investigated the performance and complexity of a new MUD based on adaptive filter block and interference canceller block in a cascade arrangement without the presence of decision feedback. It is known that the success key to commercially deploy the MUD in BS is to target a low complexity method offering the performances reach the soft multistage parallel interference canceller method (DF-Soft-MPIC). The adaptive duplicated filters and interference canceller (ADIC) proposed reaches this expectation. The AFB uses synthesized signals with the aid of the channel estimates to build a synthesized received signal per user. The latter is utilized as a training signal. The per-user-adaptation trend lowers the adaptation process complexity while the introduction of the ICB ensures, as the number of stages increases, interference free signals at the input of the next AFB. One short filter per user allows a considerable complexity reduction.

In addition, we have proposed a VLSI implementation strategy and hardware resources evaluation of ADIC MUD method. The presented implementation strategy takes into account the regularity of the algorithms, applying pipeline processes. Based on the arithmetic complexity, ADIC is 4 to 8 lower than the reference method, DF-Soft-MPIC. The evaluation of the hardware implementation and based on a pipeline strategy to take advantage of ADIC method, we noted that it is possible to implement in a Virtex-II pro XC2VP30 this MUD method at full-load base station receiver.

Future work will consist of exploiting ADIC method in to multi input multi output (MIMO) system and in orthogonal frequency-division multiple access (OFDMA) technologies always by respecting the performance-complexity trade-off.

Acknowledgment The authors would like to thank Axiocom Inc., the Natural Sciences and Engineering Research Council of Canada and the ReSMiQ for making the project possible.

References

1. Verdú, S. (1998). *Multiuse detection*. Cambridge: Cambridge University Press.
2. Prasad, R., & Ojanpera, T. (1998). An overview of CDMA evolution toward wideband CDMA. *IEEE Comm. Surveys*, *1*(1), 2–29.
3. Moshavi, S. (1996). Multi-user detection for DS-CDMA communications. *IEEE Communications Magazine*, *34*(10), 124–136.
4. Andrews, J. G. (2005). Interference cancellation for cellular systems: a contemporary overview. *IEEE Wireless Comm.*, pp. 19–29. April
5. Verdú, S. (1996). Minimum probability of error for asynchronous Gaussian multiple-access channels. *IEEE Transactions on Information Theory*, *IT-32*(1), 85–96 Jan.
6. Hui, A. L. C., & Letaief, K. B. (1998). Successive interference cancellation for multiuser asynchronous DS/CDMA detectors in multipath fading links. *IEEE Transactions on Communications*, *46*(3), 384–391.
7. Buehrer, M., Correal-Mendoza, N. S., & Woerner, B. D. (2000). A simulation comparison of multiuser receiver for cellular CDMA. *IEEE Transactions on Vehicular Technology*, *49*(4), 1065–1085 July.
8. Han, S. H., & Lee, J. H. (2002). Group-wise successive interference cancellation receiver with adaptive MMSE detection for dual-rate DS-CDMA system. *IEEE Global Telecommunications Conference*, *1*, 514–518 Nov.
9. Buehrer, M., & Woerner, B. D. (1996). Analysis of adaptive multistage interference cancellation for CDMA Using an improved gaussian approximation. *IEEE Transactions on Communications*, *44*(10), 1308–1321 Oct.
10. Miller, S. L. (1995). An adaptive direct-sequence code-division multiple-access receiver for multiuser interference rejection. *IEEE Transactions on Communications*, *43*(2), 1746–1755 Feb.

11. Miller, S. L. (1996). Training analysis of adaptive interference suppression for direct-sequence code-division multiple-access systems. *IEEE Transactions on Communications*, *44*(4), 488–495 April.
12. Massicotte, D., & Dahmane, A. O. (2004). Cascade filters receiver for DS-CDMA communication systems. WO04040789, 13 May.
13. de Lamare, R. C., & Diniz, P. S. R. (2006). Set-membership adaptive algorithms based on time-varying error bounds for DS-CDMA systems. *International Symposium on Circuits and Systems (ISCAS)*, *46*, 2372–2385.
14. Massicotte, D., & Dahmane, A. O. (2006). Mixed direct–indirect adaptation procedure applied to receiver filter. US patent application US2006233289.
15. Xue, G., Weng, J., Le-Ngoc, T., & Tahar, S. (1999). Adaptive multistage parallel interference cancellation for CDMA. *IEEE Journal on Selected Areas in Communications*, *17*(10), 1815–1827 Oct.
16. Nagaraj, S., Gollamudi, S., & Huang, Y. (1999). A fast converging adaptive multiuser detection scheme with reduced complexity for CDMA Systems. *Asilomar Conference*, *1*, 649–653 Oct.
17. Gollamudi, S., Nagaraj, S., & Huang, Y. (1997). Low complexity adaptive receiver for CDMA with Multipath fading. IEEE *Personal Wireless Communications,* 52–56, Dec.
18. Hagerman, B., Gunnarsson, F., Murai, H., Tadenuma, M., & Karlsson, J. (2005). WCDMA Uplink Parallel Interference Cancellation—System Simulation and Prototype Field Trials. *EURASIP Journal on Applied Signal Processing*, 1725–1735.
19. Sawahashi, M., et al. (2002). Expirements on pilot-assisted coherence multistage interference canceller for DSCDMA mobile radio. *IEEE Journal on Selected Areas in Communications*, *20*(2), 433–449.
20. Guo, Y., & Cavallaro, J. R. (2006). A low complexity and lower power SoC design architecture for adaptive MAI Suppression in CDMA systems. *Journal of VLSI Signal Processing*, *44*, 195–217.
21. Xu, G., et al. (2002). VLSI implementation of the multistage detector for next generation wideband CDMA receivers. *Journal of VLSI signal processing, Kluwer*, *30*, 21–33.
22. Correal, N., Buehrer, R., & Woerner, B. (1999). A DSP-based DS-CDMA multiuser receiver employing partial parallel interference cancellation. *IEEE Journal on Selected Areas in Communications*, *17*(4), 613–630 Apr.
23. Ho, Q.-T., Massicotte, D., & Dahmane, A. O. (2006). A low complexity adaptive multiuser detector and FPGA implementation for wireless DS-WCDMA communication systems. *EURASIP Journal on Applied Signal Processing—Special Issue on Designs Methods for DSP Systems*, paper ID52919, pp.1–12.
24. Rajagopal, S., Bhashyam, S., Cavallaro, J. R., & Aazhang, B. (2002). Real-time algorithms and architectures for multiuser channel estimation and detection in wireless base-station receivers. *IEEE Trans. on Wireless Comm.*, *1*(3), 468–479 July.
25. Xilinx, (2005). Virtex-II Pro and Virtex-II Pro X Platform FPGAs: Complete Data Sheet, v4.5, October
26. Ahmed-Ouameur, M., & Massicotte, D. (2007). Real-time DSP and FPGA implementation of wiener LMS based multipath channel estimation in 3G CDMA systems. *Journal of VLSI Signal Processing Systems*, *47*(3), 259–279 June.
27. Ling, F. (1993). Coherent detection with reference symbol based channel estimation for direct sequence CDMA uplink communication. *Proc. Vehicular Technology Conf.*, 400–403, May.
28. Haykin, S. (2002). *Adaptive filter theory*, 4th Edn., Prentice Hall.
29. Gollamudi, S., Nagaraj, S., Kapoor, S., & Huang, Y. F. (1998). Set-membership filtering and a set-membership normalized LMS algorithm with an adaptive step size. *IEEE Signal Processing Letters*, *5*(5), 111–114 May.
30. Kung, S. Y. (1987). *VLSI array processors*, Prentice Hall.

François Nougarou received a BS in electrical engineering from the École Supérieure de Technologie Électronique (ESTE) from the ESIEE group (France) in 1999, a Masters degree in electrical engineering and industrial electronics from the Université du Québec à Trois-Rivières (UQTR), Qc, Canada. He is currently pursuing his PhD in electrical engineering at the Université du Québec à Trois-Rivières where he joined the Laboratory of Signal and System Integration. His Ph.D. work consists of developments of multiuser detectors (MUD) for DS-WCDMA wireless communication systems of 3G/4G. His actual research interests include signal processing MUD conception and VLSI implementation, in order to reach the best trade off between performance and complexity in agreement with wireless communication demands.

Daniel Massicotte received the B.Sc.A. and M.Sc.A. degrees in electrical engineering and industrial electronics in 1987 and 1990 respectively from the Université du Québec à Trois-Rivières (UQTR), PQ, Canada. He obtained the Ph.D. degree in electrical engineering in 1995 at the École Polytechnique de Montréal, PQ, Canada. In 1994, he joined the Department of Electrical and Computer Engineering, Université du Québec à Trois-Rivières, where he is currently a Professor. He is currently Head of the Laboratory of Signal and Systems Integration and President Founder and Chief Technology Officer of Axiocom Inc. He received the Douglas R. Colton Medal for Research Excellence awarded by the Canadian Microelectronics Corporation, the PMC-Sierra High Speed Networking and Communication Award and the Second place at the Year 2000 Complex Multimedia/Telecom IP Design Contest from Europractice in 1997, 1999 and 2000 respectively. His research interests include VLSI implementation and digital signal processing for the communications and measurement problems such as nonlinear equalization, multiuser detection, channel estimation, and signal reconstruction.

Dr. Massicotte is also IEEE Senior member and member of the "Ordre des Ingénieurs du Québec", "Groupe de Recherche en Électronique Industrielle (GREI), and "Microsystems Strategic Alliance of Quebec" (ReSMiQ).

Messaoud Ahmed-Ouameur received the B.S. degree in electrical engineering from the institut national d'électronique et d'électricité (INELEC), Boumerdes (Algeria) in 1998, the M.S. and Ph.D. degrees with honours in electrical engineering from the Université du Québec à Trois-Rivières, Trois-Rivières (UQTR) QC, Canada, in 2002 and 2006 respectively. He received the master degree with honour in business administration (MBA) from graduate school of international studies at AJOU university, Souwan, South Korea in 2000. Since 2001, he worked for Axiocom inc. as director research and development.

His current research interests include wireless communications, spread spectrum systems, iterative (turbo) detection, channel estimation, smart antennas, Monte Carlo techniques for signal processing and real-time VLSI implementation issues.