Soft Information in Interference Cancellation Based Multiuser Detection

Jerome M. Shapiro, *Member, IEEE,* and James Dunyak, *Member, IEEE,*

***Abstract*— This paper considers soft information in interference cancellation based multiuser detection algorithms. For the case of binary phase-shift keying (BPSK) modulation, a relationship between the soft information as provided by the log likelihood ratio (LLR) and the minimum mean-square error (MMSE) bit estimates is derived, whereby the bit estimate is given by the hyperbolic tangent of one-half the LLR. Although this result is known for Gaussian channels, we show that this result holds true regardless of the underlying channel distribution. Similar relationships are derived for quadrature phase-shift keying (QPSK) and also for more general constellations.**

Results for multiuser detection (MUD) are then derived. Starting with an analysis of the convergence dynamics, what follows is an analysis of four canonical iterative MUD algorithms that fit within the framework: parallel vs. serial interference cancellation and symbol-level vs. chip-level updates. The results help explain why chip-level algorithms lead to faster convergence for a given amount of computation than do symbol-level algorithms, even though the chip-level algorithms never explicitly calculate the matched filter that is known to be a sufficient statistic.

Finally, an efficient computational structure for a MUD processing element for the chip-level parallel interference cancellation (CPIC) is derived as a one-chip ahead prediction.

I. Introduction

THIS paper develops a framework for analyzing interference cancellation (IC) based multiuser detection (MUD) algorithms. The framework derives soft information in the form of the log-likelihood ratio (LLR) for the transmitted bits. It also develops a general expression for the minimum mean-square error (MMSE) estimator for the bits that can be used for interference cancellation. The results are motivated by and are a generalization of a class of chip-level MUD algorithms developed at the MITRE Corporation over the past few years for asynchronous code-division multiple-access (CDMA) systems with long codes [5]- [7], [9]- [11], [13].

Because multiple access interference (MAI) is the dominant degradation in CDMA systems, multiuser detection has been an active area of research (see [17] and the references therein). Indeed, the optimum maximum likelihood (ML) algorithms and the suboptimal decorrelating and minimum mean-square error (MMSE) algorithms described in Verdu's book apply primarily to the short-code CDMA where the same spreading code is used for each symbol and thus the matrices used in these algorithm are constant. For long code CDMA, however, the matrices would change for every bit as pointed out by Divsalar *et al* [4].

Worldwide cellular CDMA systems use long periodically repeating codes for the purpose of frequency reuse so that different base stations that have different offsets can share the same spectrum. For these systems, interference cancellation based MUD algorithms have been the only implementable approaches. Nevertheless, MUD has not seen widespread application because of the perception that these algorithms are still too computationally complex. We hope that the results in this paper change that perception.

IC algorithms can be classified into serial interference cancellation (SIC) and parallel interference cancellation (PIC).

SIC algorithms, in which for each stage a user's signal is estimated and cancelled before going on to the next user, have been described in [18] and [14]. The primary disadvantage of serial cancellation has been the long delay required because each successive user must wait until the previous user's bit is estimated.

Early PIC algorithms have been described in [15], [16], and [21]. In all of these algorithms an attempt is made to cancel all of the interference from each user in parallel. In [4], Divsalar *et al.* developed partial PIC in which the hard bit estimates are replaced by MMSE soft bit estimates. Derived from a Gaussian assumption, these soft estimates also have the relationship that the MMSE bit estimates are the hyperbolic tangent of half the likelihood ratio.

In all of the above mentioned SIC and PIC algorithms, the matched filter computed over a symbol is the starting point, and all of these algorithms operate at the symbol level.

Beginning in the mid 1990's, motivated by a Kalman filtering viewpoint of the multiuser detection problem, a series of improved chip-level interference cancellation algorithms were developed at MITRE [5]- [7], [9]- [11], [13]. The Mixed Gaussian Demodulator [6] is essentially a chip-level partial PIC algorithm that improves upon [4] by performing more frequent chip-level updates. As we shall show, it is also potentially much simpler computationally, both because of its natural fine-grain concurrency, and because it converges in fewer iterations.

What follows are both reflections, insights, and improvements upon [6] that we hope provide a framework for viewing interference cancellation algorithms with soft information.

Initially, we develop a framework for an observation at a single unit of time at a scalar receiver. Then we combine the results for more general receiver observations in CDMA. Then, we illustrate the principles by contrasting the specific algorithmic processing behind four canonical interference cancel-

Manuscript received July 8, 2005

The authors are with the MITRE Corporation, 202 Burlington Road, Bedford, MA 01730.

lation algorithms: 1) symbol-level parallel interference cancellation (SPIC), 2) symbol-level serial interference cancellation (SSIC), 3) chip-level parallel interference cancellation (CPIC) and 4) chip-level serial interference cancellation (CSIC). Finally, we show how the chip-level parallel interference cancellation algorithm has a simple computational structure when viewed as a one-step-ahead prediction algorithm.

Our primary contributions are 1) the derivation of fundamental relationships that exist, regardless of the underlying channel distribution, between the log-likelihood function and the MMSE symbol estimate, including the fact that the MMSE bit estimate for binary phase-shift keying (BPSK) is equal to the hyperbolic tangent of one-half the LLR, 2) our analysis of the four canonical algorithms in terms of soft information, 3) a chip-level serial interference cancellation algorithm that has not appeared in the literature before, and 4) the architectural interpretation of the chip-level interference cancellation algorithms as one-chip-ahead predictors.

II. Basic Framework of Interference Cancellation

The transmitted signal model for user k in a CDMA system is

$$x_k(t) = A_k h_k(t) b_k(t), \tag{1}$$

where A_k is the complex channel amplitude, $h_k(t)$ is the complex spreading waveform, and $b_k(t)$ is the BPSK mapping of a bit that maps a "0" bit to +1 and a "1" bit to -1.

The receiver sees the waveforms of many users added together plus noise.

$$y(t) = \sum_k A_k h_k(t) b_k(t) + w(t). \tag{2}$$

where $w(t)$ is zero-mean white gaussian noise with variance σ_w^2.

For a particular user l, it is convenient to write (2) as

$$\begin{aligned} y(t) &= A_l h_l(t) b_l(t) + \sum_{k \neq l} A_k h_k(t) b_k(t) + w(t) && (3) \\ &= A_l h_l(t) b_l(t) + I_l(t) + w(t) && (4) \end{aligned}$$

where the term $I_l(t)$ is the MAI seen by user l.

A multiuser detector computes an estimate of the bits $\hat{b}_k(t)$ and cancels the interference. However, unlike a conventional coherent receiver, which needs only an accurate phase estimate, in order to cancel the interference the receiver must also obtain a good estimate of the magnitude of A. Define the estimate of the complex amplitude $\hat{A}$ so that $A = \hat{A} + \tilde{A}$, where $\tilde{A}$ is the complex estimation error. In that case, we can rewrite (4) as

$$y(t) = \hat{A}_l h_l(t) b_l(t) + \tilde{A}_l h_l(t) b_l(t) + I_l(t) + w(t), \tag{5}$$

where we have added an additional term for the channel estimation error.

The interference cancelled signal for user l is

$$\begin{aligned} z_l(t) &= y(t) - \sum_{k \neq l} \hat{A}_k h_k(t) \hat{b}_k(t) && (6) \\ &= \hat{A}_l h_l(t) b_l(t) + \sum_k \tilde{A}_k h_k(t) b_k(t) + \\ &\quad \sum_{k \neq l} \hat{A}_k h_k(t) (b_k(t) - \hat{b}_k(t)) + w(t). && (7) \end{aligned}$$

It is convenient to define the *innovation* as the portion of the input signal remaining after all of the interference has been removed; it is given by

$$\begin{aligned} i(t) &= y(t) - \sum_k \hat{A}_k h_k(t) \hat{b}_k(t) && (8) \\ &= \sum_k \tilde{A}_k h_k(t) b_k(t) + \\ &\quad \sum_k \hat{A}_k h_k(t) (b_k(t) - \hat{b}_k(t)) + w(t). && (9) \end{aligned}$$

We can see that the innovation consists of three terms: the channel estimation error, the uncancelled interference, and the noise.

The interference cancelled signal can now be expressed as the sum of its prediction plus the innovation:

$$z_l(t) = \hat{A}_l h_l(t) \hat{b}_l(t) + i(t). \tag{10}$$

Presumably, if we've done a reasonable job of channel estimation and interference cancellation, the innovation variance will be lower than the variance of the uncancelled interference plus noise, and we will be able to demodulate $b_l(t)$; that is we will estimate the bit by applying a conventional demodulator to $z_l(t)$, with lower probability of error than we would get if we applied a conventional demodulator to $y(t)$. As can be seen from (9), however, MUD can only provide so much benefit before the performance limitation is the quality of the channel estimation. In other words, even if the bit estimates are correct, the variance of the channel estimation errors could dominate the remaining MAI.

III. Minimum Mean-Square Error Bit Estimation in Terms of the Log Likelihood Ratio

In this section we derive the MMSE bit estimate in terms of the LLR for BPSK. The results also clearly extend to quadrature phase-shift keying (QPSK) since in the absence of phase error in the channel estimate, QPSK decomposes into two independent BPSK cases. Similar relationships are also derived for more general constellations.

Viewing $b_l(t)$ as a random variable that is ± 1 with equal probability, the minimum mean-square (MMSE) estimator of $b_l(t)$ given a set of observables Y, is the conditional expectation $E[b_l(t)|Y]$. We can manipulate the conditional density using Bayes rule to obtain

$$p[b_l(t)|Y] = \frac{p[b_l(t)Y]}{p[Y]} \tag{11}$$
$$= \frac{p[Y|b_l(t)]p[b_l(t)]}{p[Y]} \tag{12}$$
$$= \frac{p[Y|b_l(t)]p[b_l(t)]}{\frac{1}{2}(p[Y|b_l(t)=1]+p[Y|b_l(t)=-1])}. \tag{13}$$

Therefore

$$E[b_l(t)|Y] = \frac{p[Y|b_l(t)=1]p[b_l(t)=1](1)+p[Y|b_l(t)=-1]p[b_l(t)=-1](-1)}{\frac{1}{2}(p[Y|b_l(t)=1]+p[Y|b_l(t)=-1])}$$
$$= \frac{(p[Y|b_l(t)=1]-p[Y|b_l(t)=-1])}{(p[Y|b_l(t)=1]+p[Y|b_l(t)=-1])}, \tag{14}$$

since $p[b_l(t)=1]=p[b_l(t)=-1]=\frac{1}{2}$.

It will be useful to manipulate (14) by dividing all terms by $p[Y|b_l(t)=-1]$

$$E[b_l(t)|Y] = \frac{\frac{p[Y|b_l(t)=1]}{p[Y|b_l(t)=-1]}-1}{\frac{p[Y|b_l(t)=1]}{p[Y|b_l(t)=-1]}+1}. \tag{15}$$

Defining the log likelihood ratio (LLR) as

$$L_l(t) = \log\frac{p[b_l(t)=1|Y]}{p[b_l(t)=-1|Y]} \tag{16}$$
$$= \log\frac{p[Y,b_l(t)=1]}{p[Y,b_l(t)=-1]} \tag{17}$$
$$= \log\frac{p[Y|b_l(t)=1]}{p[Y|b_l(t)=-1]}. \tag{18}$$

It can now readily be seen that we can rewrite (15) as

$$E[b_l(t)|Y] = \frac{e^{L_l(t)}-1}{e^{L_l(t)}+1} \tag{19}$$
$$= \tanh\frac{L_l(t)}{2}. \tag{20}$$

The mean-square is given by

$$E[b_l^2(t)|Y] = \frac{p[Y|b_l(t)=1]p[b_l(t)=1](1)^2+p[Y|b_l(t)=-1]p[b_l(t)=-1](-1)^2}{\frac{1}{2}(p[Y|b_l(t)=1]+p[Y|b_l(t)=-1])}$$
$$= 1. \tag{21}$$

Therefore the conditional variance is given by

$$\sigma^2_{b_l(t)|Y} = 1-\tanh^2\frac{L_l(t)}{2}. \tag{22}$$

Equation (18) tells how the LLR can be computed as the ratio of the conditional probability density of the observation given the two hypotheses evaluated for the observation. Equation (20) says that whatever the state of our knowledge as determined by the LLR, the optimal MMSE bit estimate is given by the hyperbolic tangent of half the LLR. The LLR is the soft information and provides both a hard-decision estimate of the bit in the sign and a reliability measure in the magnitude. It should be pointed out that this is a generalization of results found in [4] and [6], except that no Gaussian assumption is necessary.

It should also be pointed out that if we use this estimate for the bit estimate in interference cancellation, the actual amount of interference cancelled will vary directly with the reliability of the bit estimate. A reliability close to zero will lead to a bit estimate that is also close to zero, and thus little interference will be cancelled. Similarly, if the reliability is large, the magnitude of the bit estimate will be close to unity; the high reliability provides high confidence that the sign is correct and thus $\hat{b}_l(t)$ will likely be very close to $b_l(t)$. Therefore most of the interference is very likely to be cancelled.

The results extend to other general constellations. Let c_j be the set of constellation points. The MMSE estimate $\hat{c}_l(t|Y)$ is given by the conditional expectation

$$E[c_l(t)|Y] = \sum_j c_j p[c_l(t)=c_j|Y] \tag{23}$$
$$= \frac{\sum_j c_j p[Y|c_l(t)=c_j]p[c_l(t)=c_j]}{p[Y]} \tag{24}$$
$$= \frac{\sum_j c_j p[Y|c_l(t)=c_j]p[c_l(t)=c_j]}{\sum_j p[Y|c_l(t)=c_j]p[c_l(t)=c_j]} \tag{25}$$
$$= \frac{\sum_j c_j e^{L_{lj}(t|Y)}}{\sum_j e^{L_{lj}(t|Y)}}, \tag{26}$$

where the log-likelihoods are given by

$$L_{lj}(t|Y) = \log p[Y|c_l(t)=c_j]. \tag{27}$$

The result shows a similar relationship between the MMSE symbol estimate and the log-likelihood functions for each of the constellation points. For the remainder of the paper, we will discuss the BPSK case, but the analysis is easily extended to the more general modulation.

An interference cancellation algorithm is a series of computations designed to cancel the interference, thus reducing the variance in the innovations and thus increasing the reliability of the soft information. It can be seen that improved reliability of the soft information leads to improved bit estimates and *vice versa*. This analysis motivates many of the iterative cancellation algorithms whereby revised bit estimates are alternated with revised LLR calculations. It also suggests that in cases where the bits are the coded bits of an error-correcting code, an iteration step that updates the LLRs by imposing the code's constraints using a soft-input soft-output (SISO) algorithm [2], [12], followed by a revision of the bit estimates as the hyperbolic tangent of half the resulting LLRs can be placed right inside the iteration loop. Indeed "turbo" MUD has been studied in the context of more conventional MUD algorithms in [19], [20] and its references.

IV. Application to Multiuser Detection for CDMA

The pair of relationships in (18) and (20) can lead to a number of different algorithms to perform MUD, depending on the order in which LLRs and bit estimates are computed. Assuming that the input signal is sampled at the chipping rate $t = nT_c$, then we can work in discrete time at the sampling lattice of the data. In terms of notation, square brackets indicate discrete time i.e., $y(nT) \doteq y[n]$. We will assume that we know the spreading waveform at all times for all users $h_k(t)$ so that we can operate on this lattice. In practice, since users may be asynchronous, in order to work on the sampling lattice of the data, the spreading sequence must be interpolated. Linearly interpolating the spreading sequence, which is presumed to be a binary impulse train, can easily be implemented with lookup tables. We also make the simplification, that each sampling instance is associated with only one bit at a symbol boundary. While the linear interpolation that took place at the transmitter to form a continuous waveform from the spreading sequence necessitates that this assumption is violated near the symbol boundary, it greatly simplifies the analysis.

We define the chip demodulator $D()$to be the processing that takes as its input $z_l[n]$ and produces a *chip* LLR $\lambda_l[n]$ that does not make use of the fact that the symbol persists for more than one chip

$$\lambda_l[n] = D(z_l[n]). \tag{28}$$

For example, in the case where the innovations $i[n]$ can be modelled as Gaussian with variance σ^2, the demodulator function would be

$$\lambda_l[n] = D(z_l[n]) = \frac{\hat{A}_l^{\dagger} h_l^{\dagger}[n] z_l[n]}{\sigma^2}, \tag{29}$$

where the symbol † denotes complex conjugation. Assuming that user l has N_l chips per symbol, the *symbol* LLR, $L_l[m]$, where $n \in \Omega_l[m]$, where $\Omega_l[m]$ is set of N_l chips in the same symbol, is defined as

$$L_l[m] \doteq \sum_{n \in \Omega_l[m]} \lambda_l[n]. \tag{30}$$

Consider the original partial parallel interference cancellation (PIC) algorithm of [4]. This algorithm starts off with no prior knowledge of the bit estimates, i.e $\hat{b}_k = 0$, and calculates LLRs for the entire symbol before applying (20) to obtain the revised bit estimate. Then additional stages proceed to repeat the process. On the other hand, in Dunyak's Mixed Gaussian Demodulator (MGD) [6], initially all of the symbol LLRs are 0. After each chip, the new chip LLR replaces 0 in (30) leading to an updated symbol LLR after each chip. A new bit estimate is computed at each chip as the hyperbolic tangent of half the current state of $L_l[m]$. The process then repeats at each chip. Before considering these algorithms and their serial interference cancellation (SIC) counterparts in more detail, we study some convergence properties.

A. Convergence Dynamics

As described in detail below, the symbol estimate and interference cancellation is applied in an iterative fashion. Any practical interference cancellation algorithm must provide its benefit in the first few iterations, but a more careful examination of the training dynamics still provides substantial insight. The reader should keep in mind, though, that limiting behavior is not observed in practice due to the small number of iterations.

Assuming that the demodulation function is given by (29), and can be plugged into (30) then the symbol likelihood can be expressed as

$$L_l^*[m] = \frac{u_l[m]}{\sigma^2} - \frac{A_l^{\dagger}}{\sigma^2} \sum_{k \neq l} A_k \sum_{n \in \Omega_l[m]} \rho_{lk}[n] \hat{b}_k^*[n], \tag{31}$$

where

$$u_l[m] \doteq \sum_{n \in \Omega_l[m]} A_l^{\dagger} h_l^{\dagger}[n] y[n] \tag{32}$$

is the unnormalized matched filter, and

$$\rho_{lk}[n] \doteq h_l^{\dagger}[n] h_k[n], \tag{33}$$

is simply the product of the two spreading waveforms at time n. Furthermore, the converged solution also requires

$$\hat{b}_l^*[m] = \tanh \frac{L_l^*[m]}{2}. \tag{34}$$

This results in a fixed point problem which, in principle, provides a solution to the bit estimation problem, with training resulting from iterating the fixed point map. Obviously, from the finite range of the hyperbolic tangent, any training sequence contains at least one accumulation point. Unfortunately, we were unable to prove analytical convergence, uniqueness, or existence. When using "hard" bit estimates, convergence often fails when a single error is traded cyclically between several different users. In fact, this property is often used to improve performance by averaging bit estimates for the last several iterations, so single error cycles of length greater than three can be corrected. We hoped that use of a soft estimate would preclude this type of failed convergence. Numerical experience suggests this is the case, but the question of convergence remains open.

Using continuous time variables and replacing the summations with integrals, we can also obtain continuous time simultaneous equations. One way to find the optimum might be to build an analog circuit with the corresponding relationships and then have the system automatically converge. Indeed, early work by Abrams, *et. al.* [1] devised an analog interference cancellation system along these lines.

Despite lack of a convergence proof, we have still developed significant insight into the training dynamics. Using analytical viewpoints, we developed examples where the innovation has more than one minima in the $\hat{b}_k$ space. This occurred when two of the spreading functions were almost parallel, while the corresponding true bit values were of opposite sign. In this case, a local minimum occurs near both (+1,-1) and (-1,+1) with a saddlepoint near the origin. Depending on the additive noise, the iterated map converged near one or the other of (+1,-1) or (-1,+1). Obviously, the resulting bit error was sometimes large. We further observed that, in some cases, the convergence could be slow when the estimate shot back and forth across a canyon in the innovation function. Both of these behaviors frequently occur in optimization problems solved by descent, and interference cancellation is no exception. However, these two geometries are quite unlikely in random channels.

As can be seen in (31), as expected, the dependence on the received signal is provided by the unnormalized matched filter. Much has been made about the fact that the matched filter is a sufficient statistic for multiuser detection [17], and indeed almost all other algorithms that we are familiar with take this point of view. Nevertheless, while this quantity serves as a sufficient statistic, it is not necessarily true that directly calculating this quantity leads to the most efficient algorithm. The basis for the chip-level algorithms that will be discussed is that it is possible to perform interference cancellation "on-line" before we have observed $y[n]$ over the support of the symbol. The interpretation of the symbol likelihoods as being the sum of the chip likelihoods over the support of the symbol tells us how to construct algorithms that can begin to cancel interference before observing the received signal over the entire support of the symbol, and thus converging in fewer iterations as demonstrated in [6].

B. The Innovation Variance

A word should be mentioned on use of the innovation variance in (31). Solving the set of simultaneous equations for the LLRs and the bit estimates requires knowledge of the innovation variance at each step. A method of maintaining a variance estimate must be used for each algorithm. This problem has been addressed by several authors. In [4], the choice of the scale factor in the partial PIC algorithm amounts to an implicit variance estimate. For the MGD algorithm [6] Dunyak derives both a recursive estimate and a simpler approximation. In earlier work on using a Kalman filter for the interference cancellation, maintenance of the variance is implicitly done because calculation of the Kalman gain requires maintaining the solution to the Riccati equations [5], [9]- [11]. We choose not to specify the specific variance estimation to be used as it does not affect the main points of this paper.

V. Four Canonical Interference Cancellation Algorithms

What follows is a series of four canonical algorithms that iterate the simultaneous equations. They can be thought of as being of the class of Expectation-Maximization algorithm [3], with the calculation of the bit estimates being the Expectation step, and the calculation of the new LLR being the Maximization step. In the algorithms that follow, it will really be a Maximization-Expectation algorithm in the sense that for a chunk of data, the Maximization step will be run on the chunk followed by the Expectation step.

The differences between the four algorithms is the order in which bits are estimated and LLRs updated. These algorithms are multistage algorithms; for implementation on a parallel computer, calculations on one stage can begin as soon as all of the dependent quantities on the previous stage are complete. The "symbol"-level algorithms complete the Maximization step or the Expectation step on an entire symbol before proceeding to the next step. In contrast, the "chip"-level algorithms perform the Maximization step for a single chip followed by the Expectation step for a single chip before proceeding onto the next chip. There are both parallel and serial versions of each type. In the parallel version, the innovation is computed, either for the whole chip or symbol, and then the LLRs are updated, followed by the bit estimates. In the serial version, the users are ordered, typically by magnitude. The LLRs are updated one user at a time for either a chip or a symbol period, and then the bit for that user is updated for the chip or symbol period. The next user's interference cancellation incorporates these updated bit estimates of the same iteration.

A. Symbol Parallel Interference Cancellation (SPIC)

A version of SPIC is described in [4]. In our canonical algorithm, the iteration proceeds as follows:

Initialization: For all users l and for all symbols starting at all m,

$$\hat{b}_l^{(0)}[m] = 0 \tag{35}$$
$$L_l^{(0)}[m] = 0. \tag{36}$$

Iteration Loop: Starting with iteration $i = 0$:
- **User Loop:** Initialize $l = 0$, run for all users l:
  - **Symbol Loop:** For symbol $m = 0$:
  - **Initialize Symbol LLR:** Set $L_l[m] = 0$.
    - **Chip Loop for LLRs:** For each chip $n \in \Omega_l[m]$:
      - **Calculate the innovation** according to (8) with the most up to date bit estimates, $\hat{b}_k^{(i-1)}[m]$.
      - **Calculate the Interference Cancelled Signal** (10) using the most up to date bit estimate $\hat{b}_l^{(i-1)}[m]$.
      - **Calculate Chip LLR:** Calculate the chip LLR $\lambda_l^{(i)}[n]$ according to (29).
      - **Accumulate the Symbol LLR** $L_l[m] := L_l[m] + \lambda_l^{(i)}[n]$.
    - **Update the Bit Estimates** $\hat{b}_l^{(i)}[m]$ from $L_l^{(i)}[m]$ using (34).
    - **Next chip:** m := m+1.
  - **Next User:** l := l+1.
- **Next Iteration:** i := i+1.

Notice that all of the LLRs are computed from bit estimates obtained on the previous iteration. Therefore, the first iteration of symbol PIC is the conventional demodulator.

B. Symbol-Level Serial Interference Cancellation (SSIC)

In SSIC, which is similar to [14] and [18], the iteration proceeds after ordering the users, typically by amplitude.

Initialization: For all users l and for all symbols starting at all m,

$$\hat{b}_l^{(0)}[m] = 0 \tag{37}$$
$$L_l^{(0)}[m] = 0. \tag{38}$$

Iteration Loop: Starting with iteration $i = 0$:
User Loop: For $l = 0$:
Symbol Loop: For symbol $m = 0$:
Initialize Symbol LLR: Set $L_l[m] = 0$.
Chip Loop for LLRs: For each chip $n \in \Omega_l[m]$:
Calculate the innovation according to (8) with the most up-to-date bit estimates. For users $k \geq l$, the bit estimate from the previous iteration, $\hat{b}_k^{(i-1)}[m]$, is used. For users $k < l$, the bit estimate from the current iteration, $\hat{b}_k^{(i)}[m]$, is used.
Calculate the Interference Cancelled Signal according to (10) using the most up-to-date bit estimate (either $\hat{b}_l^{(i-1)}[m]$ or $\hat{b}_l^{(i)}[m]$) .
Calculate Chip LLR: Calculate the chip LLR $\lambda_l^{(i)}[n]$ according to (29).
Accumulate the Symbol LLR $L_l[m] := L_l[m] + \lambda_l^{(i)}[n]$.
Update the Bit Estimates $\hat{b}_l^{(i)}[m]$ from $L_l^{(i)}[m]$ using (34).
Next chip: m := m+1.
Next User: l := l+1.
Next Iteration: i := i+1.

The main difference between SPIC and SSIC is which bit estimate is used in calculating the innovation, with SSIC using bit estimates from the current iteration where possible.

C. Chip-Level Parallel Interference Cancellation (CPIC)

The new chip-level algorithms pioneered by the parallel MGD algorithm of Dunyak [6] organize the calculations with many more frequent updates. The rationale for using a chip-level algorithm is that the calculations for the interference cancellation and the update of the LLRs require exactly the same amount of calculation as for the symbol PIC algorithm. The difference here is that the estimation of the bits is updated after each chip. While it may seem that this is computationally burdensome, in practice a saturating piecewise linear approximation to the hyperbolic tangent function provides excellent performance, so there really isn't a significant increase in calculation.

In the original MGD algorithm of [6], iterations were combined by maintaining an accumulation of chip LLRs across iterations with the scale factor being the reciprocal of the latest estimate for the innovation variance. Doing so takes the point of view that the new innovation seen at each chip for each iteration is a different independent observation than the earlier one, and by weighting it with the reciprocal of a presumably smaller variance, the chip LLRs of the later iterations will be given more weight.

The CPIC algorithm described below assumes that the symbol LLR is always the sum of the chip LLRs across the support of the symbol. Therefore, for a later iteration, the new chip LLR replaces the old, in the local calculation of the symbol LLR.

The CPIC algorithm proceeds as follows:

Initialization: For all users l and for all symbols starting at all m, and for all chips n,

$$\hat{b}_l^{(0)}[n] = 0 \tag{39}$$
$$L_l^{(0)}[m] = 0 \tag{40}$$
$$\lambda_l^{(0)}[n] = 0 \tag{41}$$

Iteration Loop: For iteration $i = 0$:
Chip Loop: For chip $n = 0$:
Calculate the innovation according to (8) with the most up to date bit estimates, $\hat{b}_k^{(i)}[n-1]$ for users in which chip $n-1$ is part of the same symbol, and $\hat{b}_k^{(i-1)}[n+N_c-1]$ for users in which the symbol starts at n, where N_c is the number of chips in the symbol. In other words, for interior chips, use the bit estimate updated at the last chip on the same iteration, and for the starting chip in a symbol, use the final bit estimate for the last chip in the symbol computed on the previous iteration.
User Loop: For user $l = 0$:
Calculate the Interference Cancelled Signal according to (10) using the most up-to-date bit estimate (either $\hat{b}_l^{(i)}[n-1]$ or $\hat{b}_l^{(i-1)}[n+N_c-1]$).
Calculate the Chip LLR: $\lambda_l^{(i)}[n]$ according to (29).
Update the Symbol LLR for the symbol m such that $n \in \Omega_l[m]$ chip m using the update.

$$L_l^{(i)}[m] = L_l^{(i-1)}[m] + \lambda_l^{(i)}[n] - \lambda_l^{(i-1)}[n], \tag{42}$$

which calculates (30).
Update the Bit Estimate: Use (34) to update the bit estimate $\hat{b}_l^{(i)}[n]$.
Repeat for next user: $l := l + 1$.
Repeat for next chip:$n := n + 1$.
Repeat for next iteration: $i := i + 1$.

Notice that if the bit estimate calculation was performed only at the end of a symbol instead of at every chip, then the result would be identical to SPIC. However, the advantage of CPIC is that in the case of users of unequal power, once there is sufficient processing gain so that the LLR begins to have the correct sign, interference cancellation can begin without waiting until the end of the symbol. It should be pointed out that (20) provides a soft bit estimate, so early in the process, when the reliability is very small, the amount of a user's signal that is used for cancellation is also small. Because the algorithm begins to converge before the end of a symbol period, the innovation variance decreases for all users much more rapidly than in SPIC.

D. Chip-Level Serial Interference Cancellation (CSIC)

Another interesting choice for iteration sequence, and one not previously seen in the literature, is chip-level serial interference cancellation. In this algorithm, the LLRs and bit estimates are updated each chip and used for the next user in the list on the same chip. The algorithm is as follows:

Initialization: For all users l and for all symbols starting at all m, and for all chips n,

$$\hat{b}_l^{(0)}[n] = 0 \tag{43}$$
$$L_l^{(0)}[m] = 0 \tag{44}$$
$$\lambda_l^{(0)}[n] = 0 \tag{45}$$

Iteration Loop: For iteration $i = 0$:
Chip Loop: For chip $n = 0$:
User Loop: For user $l = 0$:
Calculate the innovation: The innovation is calculated according to (8) with the most up-to-date bit estimates. For users $k \geq l$ the quantity $\hat{b}_k^{(i)}[n-1]$ is used for users in which chip $n-1$ is part of the same symbol, and $\hat{b}_k^{(i-1)}[n+N_c-1]$ for users in which the symbol starts at n, and N_c is the number of chips in the symbol. However, for users $k < l$, the bit estimate is the one computed for the current chip $\hat{b}_k^{(i)}[n]$.
Calculate the Interference Cancelled Signal according to (10) using the most up-to-date bit estimate (either $\hat{b}_l^{(i)}[n-1]$ or $\hat{b}_l^{(i-1)}[n+N_c-1]$).
Calculate the Chip LLR: $\lambda_l^{(i)}[n]$ according to (29).
Update the Symbol LLR for the symbol m such that $n \in \Omega_l[m]$ chip m using the update.

$$L_l^{(i)}[m] = L_l^{(i-1)}[m] + \lambda_l^{(i)}[n] - \lambda_l^{(i-1)}[n]. \tag{46}$$

which calculates (30).
Update the Bit Estimate: Use (34) to update the bit estimate $\hat{b}_l^{(i)}[n]$.
Repeat for next user: $l := l + 1$.
Repeat for next chip:$n := n + 1$.
Repeat for next iteration: $i := i + 1$.

The key difference between the CPIC and the CSIC is that in CSIC, calculation of the innovation is inside the User Loop since earlier users on the same chip use bit estimates derived on the current chip and later users use bit estimates from the prior chip, whereas in CPIC, the innovation is calculated once for all users since they all use the bit estimates from the previous chip.

E. Hybrid Solutions

While the chip-level algorithms typically require slightly more computation than the symbol-level algorithms, this drawback is mitigated by the fact that it would require fewer iterations to achieve the same performance. However, the memory requirements for the algorithms are different.

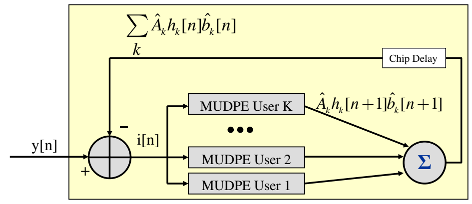

Fig. 1. Single Stage of the CPIC Algorithm Viewed as a One-Step Ahead Predictor

SPIC does not require maintaining either the innovation signal or the chip LLR between iterations, since after the chip LLR is added to the symbol LLR, it is no longer needed. CPIC on the other hand needs to store the chip LLRs for each iteration so that they can be subtracted on the next iteration to maintain the symbol LLR as the sum of the updated chip LLRs. SSIC does not require the maintenance of the chip LLRs; however, the innovations must be stored after being updated by each successive user. This is not a huge drawback because it can replace the actual input data as long as the difference between an old and new bit estimate is stored in addition to the bit estimate itself. This would allow the innovations to be updated at the next iteration by using the difference. As in CPIC, CSIC must store the chip LLRs for use on the next iteration.

A very practical MUD algorithm can be derived by combining the rapid convergence of CPIC on the first iteration with the low memory requirements of SPIC on later iterations.

VI. Interpretation of the CPIC as a One-Step-Ahead Predictor

In this section we explore the interpretation of CPIC as a One-Step-Ahead predictor. Analogous to Kalman filtering, the processing for each user can be totally decoupled. We can reorganize the algorithm so that for each user, a Multiuser Detection Processing Element (MUDPE) can be constructed whose input is the innovation signal and whose output is the one-step ahead prediction of that user's contribution to the next chip. These predictions are then added to cancel the interference for the next chip. The macroscopic structure of a single iteration of the CPIC algorithm can be seen in Fig. 1.

The internal structure of the MUDPE can be seen in Fig. 2. Basically, the one-step-ahead prediction of User k's own signal is added to the innovation to create the interference cancelled signal $z_k[n]$. This signal is then passed through a conventional demodulator to calculate both the chip LLR and the accumulation of the symbol LLR. The current symbol LLR is then passed through the hyperbolic tangent operation to obtain the bit estimate. This updated bit estimate is then used to create the prediction for the next chip.

The key to the low complexity and practicality of the CPIC algorithm is that each user's MUDPE can operate independently on the innovations signal to compute its own contribution to the next chip.

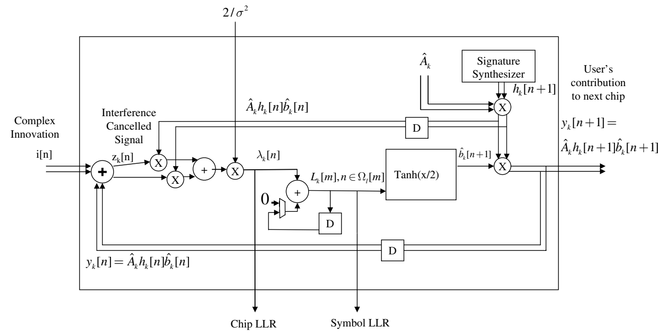

Fig. 2. Multiuser Detection Processing Element

VII. Conclusion

After deriving the relationship between the LLRs and MMSE bit estimates, we've shown for CDMA how the symbol LLRs can be viewed as the summation of the chip LLRs over the support of the symbol. While indeed a "converged" solution shows that the results can be obtained by computing a matched filter over the entire symbol, our viewpoint indicates several possible iteration approaches to obtain the LLRs and bit estimates. We then proceeded to describe four possible canonical interference cancellation algorithms that alternately update the LLR and then update the bit estimates. The four variations, serial vs. parallel interference cancellation and chip vs. symbol level interference calculation, differ in the order in which they apply the same basic iterations. Although it has long been well understood that the calculation of a matched filter should be the first step in a MUD algorithm, the value of the chip-level algorithms can be seen in that they start converging even before the end of the symbol and differ from the symbol-level algorithms in that bit estimates are updated every chip. The result is that interference cancellation can begin in the chip-level algorithms much sooner and with much less data than in the symbol-level algorithms, notably as soon as the sign of the LLR begins to be correct.

Acknowledgment

The authors gratefully acknowledge the valuable interactions with Emily Krzysiak and her staff at Air Force Research Lab, Rome, NY.

References

[1] B. Abrams, A. Zeger, and T. Jones, "Efficiently structured CDMA receiver with near-far immunity," *IEEE Trans. Vehicular Technology*, vol. 44 no. 1:1-13, 1995.

[2] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Soft-output decoding algorithms in iterative decoding of turbo codes," *Jet Propulsion Lab TDA Progress Report 42-124*, February 15, 1996, pp. 63-87.

[3] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B,* 39(1):1-38, 1977.

[4] D. Divsalar, M. K. Simon, and D. Raphaeli, "Improved parallel interference cancellation," *IEEE Trans. Commun.* vol. 46, Feb. 1998, pp 258-268.

[5] J. Dunyak, C. Suprin, S. Kumaresan, and B. Flanagan, "A decoupled Kalman filter technique for multiuser detection of pulse amplitude modulation CDMA," *2nd IASTED International Conference Wireless and Optical Communications*, July 2002.

[6] J. Dunyak, C. Suprin and S. Betz, "Multiuser detection using interference cancellation at each chip," *IEEE International Symposium on Advances in Wireless Communications*, Sept. 2002. pp. 101-102.

[7] J. Dunyak and C. Suprin, "Multiuser CDMA with a chip-Level interference cancellation technique," IEEE MILCOM, vol. 1, Oct. 2003, pp. 358-362.

[8] J. Dunyak, "Chip level interference cancellation techniques for multiuser detection," under review for *IEEE Trans on Comm.*, 2004.

[9] B. Flanagan, J. Dunyak, C. Suprin, S. Betz,"A Kalman-PIC multistage technique to reduce multiaccess interference," *IEEE International Symposium on Advances in Wireless Communications*, Sept. 2002, pp. 55-56.

[10] B. Flanagan, C. Suprin, S. Kumaresan, and J. Dunyak, "Performance of a joint Kalman demodulator for multiuser detection," Proceeding of the Vehicular Technology Conference, Sept. 2002, vol.3, pp.1525-1529.

[11] B. Flanagan, and J. Dunyak, "A steady state decoupled Kalman filter technique for multiuser detection," IEEE MILCOM 2003, vol 1. Oct 2003, pp. 347-352.

[12] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, March 1996, pp. 429-445.

[13] R. LaPat, "A Kalman joint-demodulator," unpublished work, circa 1998.

[14] P. Patel and J. Holtzman, "Analysis of a simple successive interference cancellation scheme in a DS/CDMA system," *IEEE J. Select. Areas Commun.*, vol 12, pp. 796-807, June 1994.

[15] M. K. Varanasi and B. Aazhang, "Multistage detection in asynchronous code-division multiple-access," *IEEE Trans. Comm,* vol. 38, pp. 509-519, Apr. 1990.

[16] M. K. Varanasi and B. Aazhang, "Near optimal detection in code-division multiple-access communications," *IEEE Trans. Comm,* vol. 39, pp. 725-736, May 1991.

[17] S. Verdu, *Multiuser Detection*, Cambridge University Press, 1998.

[18] A. J. Viterbi, "Very low rate convolutional codes for maximum theoretical performance of spread-spectrum multiple-access channels," *IEEE J. Select. Areas Commun.*, vol 8, pp. 641-649, May 1990.

[19] X. Wang and H. V. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Trans. Comm,* vol. 47, pp. 1047-1061, July 1999.

[20] X. Wang and H. V. Poor, *Wireless Communications Systems: Advanced Techniques for Signal Reception*, Prentice-Hall, 2004.

[21] Y.C. Yoon, R. Kohno, and H. Imai, "A spread-spectrum multiaccess system with cochannel interference cancellation, *IEEE J. Select. Areas Commun.*, vol 11. pp. 1067-1075, Sept. 1993.