A Combined Soft-Decision Deinterleaver/Decoder for the IS95 Reverse Link

A. Mark Earnshaw, *Member, IEEE,* and Steven D. Blostein, *Senior Member, IEEE*

***Abstract*—The reverse link encoding steps of the IS95 cellular code-division multiple-access (CDMA) standard consist of convolutional encoding, block interleaving, and orthogonal Walsh function encoding. Deinterleaving individual symbol metrics obtained from the Walsh function matched filters followed by conventional Viterbi decoding produces suboptimal results, as unwanted intersymbol Walsh function correlation is introduced. We propose a combined deinterleaver/decoder with improved performance over existing decoders with little added overhead and no extra decoding delay. Applied to the IS95 reverse link, the proposed decoder has about 1.0-dB gain over soft-decision decoding with interleaved symbol metrics at a bit error rate (BER) of 10^{-3}.**

***Index Terms*—Code-division multiple-access, decoding with soft decisions, Viterbi decoding.**

I. Introduction

BASED on a number of design considerations and performance comparisons [1]–[4], the reverse link of the IS95 cellular code-division multiple-access (CDMA) standard [5] operating at a voice rate of 9600 bps, data is divided into 192-b frames, and rate 1/3 convolutional encoding with a constraint length of 9 is applied. The result is block-interleaved using a 32 × 18 array. The symbols[1] are then grouped into sets of six which index one of 64 orthogonal Walsh functions, each corresponding to a sequence of 64 Walsh chips. The Walsh chips are then spread by a pseudonoise sequence. This encoding process is summarized in Fig. 1.

To decode the received signal, matched filtering with each of the 64 possible Walsh chip sequences takes place at six-symbol intervals. The outputs of these matched filters are then used as soft-decision decoding metrics. Unfortunately, since the Walsh function encoding takes place *after* block interleaving in order to increase the free distance of the overall coding [1], a deinterleaver followed by a Viterbi decoder [2]–[4] using the deinterleaved symbol metrics produces suboptimal results [6]. Other techniques for convolutional decoding do exist, such as Fano's sequential decoding algorithm [7], [8], the stack algorithm [7], [9], and feedback decoding [7]. However, these alternative methods are applicable to the case of large constraint length codes, but do not solve the tandem deinterleaving/decoding problem specific to the IS95 reverse link. In Section II, we review existing decoding metric generation techniques and then propose a combined soft-decision deinterleaver/decoder with improved performance. A discussion of relative algorithm complexity is also included. In Section III, the expected decoder performance of IS95 cellular CDMA with no interleaving is derived for verification purposes and is then compared to simulation results in Section IV.

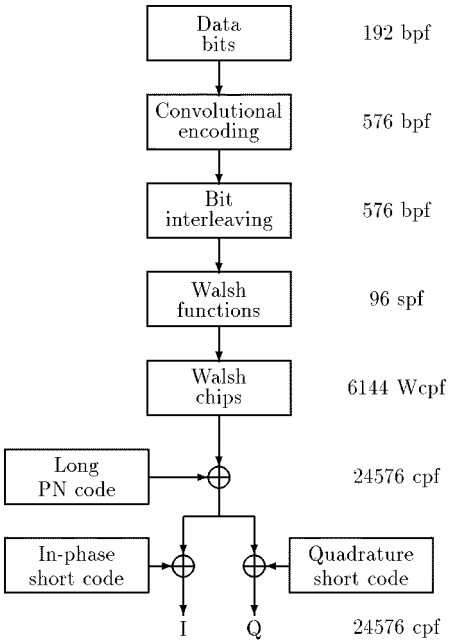

Fig. 1. Data encoding process for IS95.

II. Signal Decoding

A. No Interleaving

For noninterleaved data, standard Viterbi decoding [2] may be applied directly to the correlated Walsh sequences by using the matched filter outputs as metrics[2] to evaluate surviving paths in the decoding trellis. In the following, we refer to $C_W(w, j)$ as the matched filter correlation output for Walsh function w in the jth Walsh function position of an IS95 frame, where $0 \leq w \leq 63$ and $1 \leq j \leq 96$. In addition, we also assume that the entire frame of data has been received before decoding commences. This is a required assumption for situations involving

Manuscript received May 1, 1997; revised April 23, 1999. This work was supported by the Canadian Institute for Telecommunications Research under the NCE program of the Government of Canada.

A. M. Earnshaw is with Nortel Networks, Nepean, Ont., Canada.

S. D. Blostein is with the Department of Electrical and Computer Engineering, Queen's University, Kingston, Ont., K7L 3N6, Canada.

Publisher Item Identifier S 0018-9545(00)02545-7.

[1]Throughout this paper, we shall use *bits* to refer to the original data bits and *symbols* to refer to the bit values obtained at the output of the convolutional encoder.

[2]Without loss of generality, we assume that a higher metric value is preferred over a lower one.

TABLE I
SYMBOL METRIC CALCULATION EXAMPLE FOR INTERLEAVED SOFT-DECISION DECODING

Walsh Function	Binary Representation	C_W
0	000	6
1	001	2
2	010	7
3	011	4
4	100	1
5	101	5
6	110	8
7	111	3

Symbol Position	Symbol Value	Symbol Location	Symbol Metric
1	0	0−−	7
	1	1−−	8
2	0	−0−	6
	1	−1−	8
3	0	−−0	8
	1	−−1	5

interleaving. Since each Walsh function is derived from two of the original data bits, each state transition in the decoding trellis outputs two data bits instead of the usual one and a double state transition is used in the decoding trellis: each input node is connected to four output nodes and the $C_W(w, j)$ are used as branch metrics with specific values of w depending on the Walsh function output corresponding to a given trellis branch and j depending on the current stage within the decoding trellis.

B. Hard-Decision Decoding with Deinterleaved Symbol Metrics

If a block interleaver is added, it is necessary to generate metrics for individual convolutionally encoded symbols to make deinterleaving possible. These deinterleaved symbol metrics can either be summed in groups of three to generate the branch metrics for standard Viterbi decoding of a rate 1/3 encoder, or, in this case, in groups of six to match the noninterleaved IS95 reverse link Walsh function encoding scheme. The standard decoder in Section II-A may then be used without modification.

One method to generate individual symbol metrics is by hard-decision decoding [10]–[12]. At the jth six-symbol Walsh function boundary, $M_W(j)$ is recorded where $M_W(j)$ is defined as the maximum observed Walsh function correlation value for position j over all of the possible Walsh function values

$$M_W(j) = \max_{0 \le w \le 63} [C_W(w, j)]. \tag{1}$$

Individual metrics for each of the six symbols within a Walsh function are assigned as follows: a binary symbol value which matches the corresponding symbol in Walsh function j is assigned a metric value $M_W(j)$, while the opposite symbol value for the same symbol position is assigned a metric of zero.

Such a technique is suboptimal for two reasons. First, the binary metric quantization within a Walsh function discards information. Second, since the Walsh functions are mutually orthogonal, the summed M_W for incorrectly decoded Walsh functions should be zero mean. However, hard-decision decoding will result in a number of incorrect Walsh functions having nonzero-mean metrics and may cause an incorrect deinterleaved sequence of symbols to have a correlation value close to that of the correct symbol sequence. This will increase the probability of selecting an incorrect path, thus raising the number of decoding errors.

C. Soft-Decision Decoding with Deinterleaved Symbol Metrics

As it is widely accepted that soft-decision decoding generally outperforms hard-decision decoding [12], soft-decision symbol metrics have also been used in recent IS95 performance studies [13]–[15]. Rather than performing binary quantization within a group of six symbols, finer quantization is achieved by exploiting the fact that each binary output symbol has 32 Walsh functions which match the same symbol value at that particular position. For the jth Walsh function position, the value of $C_W(w, j)$ which is maximal over the set of 32 matching Walsh functions is the soft-decision metric for that particular symbol position and binary value. These metric values may then be deinterleaved and processed as discussed in Section II-B. This is sometimes referred to as the dual-max algorithm.

Table I shows an example of a symbol metric assignment for the case of eight different three-symbol Walsh functions. The left-hand table lists the C_W at some arbitrary position within a frame. The rightmost column of the right-hand table is the corresponding soft-decision metric for each pair of symbol positions and values. This approach, although an improvement over hard-decision quantization, is still suboptimal for the reasons described in Section II-B.

D. Combined Deinterleaving/Decoding

1) Concept: At the start of decoding an IS95 frame, no data bits are initially known. If there is no interleaving as in Section II-A, the Viterbi decoder performs its usual time-sequential progression through the trellis, forming hypotheses for the original data bits for each surviving trellis path until a single optimal path is obtained at the end of the frame. Without interleaving, there is a temporal correspondence between the jth pair of data bits and $C_W(w, j)$ so we can use the $C_W(w, j)$ values to define *branch metrics* for the decoding trellis.

When interleaving is present, rather than using the suboptimal approaches for metric generation outlined in Sections II-B and C, we propose a new approach which combines the deinterleaving and decoding operations.

First, observe that after interleaving, the IS95 convolutionally encoded binary symbols appear in the order

$$b_0 b_{32} b_{64} \cdots b_1 b_{33} b_{65} \cdots b_{543} b_{575} \tag{2}$$

where the subscript denotes time. However, in Viterbi decoding, the convolutionally encoded symbols for each surviving path must be hypothesized in time-sequential order $(b_0, b_1, b_2, \cdots)$.

Therefore, as we progress through the decoding trellis, the six convolutionally encoded symbols corresponding to each successive pair of data bits are distributed among six different received Walsh functions. This makes it difficult to define branch metrics for decoding without resorting to individual deinterleaved symbol metrics as previously discussed.

Instead of a deinterleaver, we employ the concept of a *path metric* which we assign to each hypothesized surviving path through the entire trellis. A path metric is updated as each new set of six symbols is hypothesized. Due to the interleaving, not all symbols of each received Walsh function will be simultaneously defined at a given time during decoding. The path metric must thus be sequentially evaluated based on *partially specified* Walsh functions. At each stage of the decoding process, each surviving path is extended and its path metric is reevaluated based on the newly hypothesized additional symbols within that path's set of partially defined Walsh functions. We note that computational efficiency and low delay is critical for practical implementation. An efficient algorithm for implementing such a decoding strategy is presented in the following section.

In a conventional Viterbi decoder, only a single surviving path at each node is retained due to the additivity of branch metrics that form the overall path metric. In this situation, the best surviving path at each trellis node is known to be optimal. However, since the proposed deinterleaver/decoder reevaluates path metrics over time, it is possible that an incorrectly hypothesized path may temporarily have the largest path metric value. The proposed decoder should therefore retain multiple surviving paths at each trellis node (typically two is sufficient for good performance) in order to reduce the possibility of accidentally deleting the best path before decoding is complete.

2) Implementation: As in the other methods, we group the hypothesized interleaved symbols by six to match Walsh function boundaries. We remark that for this rate 1/3 decoder, we could alteratively employ groups of three symbols, but for comparison purposes we wish to maintain compatibility with the approaches described earlier.

Initially, there are 2^6 possible M_W for each set of six binary symbols composing a Walsh function. If the first k of these six symbols have already been decoded, then there would be 2^{6-k} distinct Walsh functions sharing the first k defined symbol values. A metric can be formed by taking the maximum C_W from among these 2^{6-k} partial matches. In contrast to Section II-C, this is actually a type of soft-decision metric for *multiple* received symbols. For example, in Fig. 2, if the first four symbols have been hypothesized to be 1011_2, then Walsh functions 44 through 47 are potential matches and the maximum correlation is $M_W(1011{-}{-}_2) = 7$. An example using 3-b Walsh functions is shown in Table II where the differential metric values shown will be discussed shortly.

Let $v_j = s_0 s_1 \cdots s_{k_j-1}$ denote the first k_j hypothesized symbols of the jth Walsh function position. We define the metric at the jth Walsh function position along path P_i as

$$M_F(P_i, j, k_j, v_j) \equiv \begin{cases} \max\limits_{w \in \mathcal{S}_{P_i}} [C_W(w, j)], & k_j > 0 \\ 0, & k_j = 0. \end{cases} \tag{3}$$

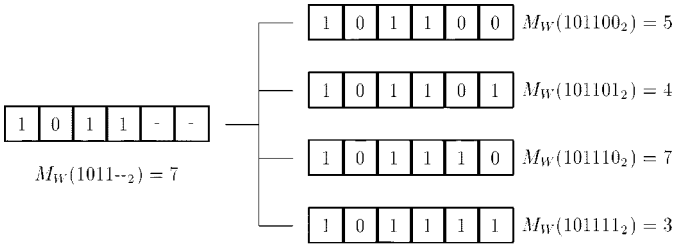

Fig. 2. Illustration of partially known Walsh function metric calculation.

In (3), maximization is over the set of C_W values of the 2^{6-k_j} Walsh functions which share the same first k_j symbols of the j^{th} Walsh function position in path P_i. Mathematically, this set can be expressed as

$$\mathcal{S}_{P_i} \equiv \{w | 0 \le w \le 63, \lfloor w/2^{6-k_j} \rfloor = v_j\} \tag{4}$$

where $\lfloor \cdot \rfloor$ denotes floor function and integer division is used.

The overall path metric for a surviving path P_i is calculated as

$$M_{P_i} = \sum_{j=1}^{96} M_F(P_i, j, k_j, v_j) \tag{5}$$

where we are summing up the current metric values for each Walsh function position within the frame.

The decoding proceeds in stages. As each new set of six symbols is hypothesized and the corresponding Walsh function values are updated, the path metrics are recalculated. Clearly, it is computationally inefficient to recalculate the overall path metric at each trellis decoding stage. A more efficient method for updating the path metrics is to somehow recreate the additivity of conventional Viterbi decoders. We define the *differential metric* for Walsh function j of path P_i with k_j hypothesized bits $v_j = s_0 s_1 \cdots s_{k_j-1}$ as

$$D_F(P_i, j, k_j, v_j) = M_F(P_i, j, k_j, v_j) - M_F(P_i, j, k_j - 1, v_j). \tag{6}$$

These $D_F(P_i, j, k_j, v_j)$ can actually be *precalculated* before decoding commences. An example is shown in Table II where the generation of metrics for partially specified Walsh functions proceeds from the left table to the right table. Following this, the differential metrics are computed from right to left.

As the symbols for a particular Walsh function are decoded, the $M_F(P_i, j, k_j, v_j)$ must be modified accordingly. Since only six Walsh functions in each path are updated at each decoding stage, it is only necessary to modify the six $M_F(\cdot)$ values which have changed rather than reevaluating all of the terms in (5). This can be accomplished most efficiently by adding the relevant precalculated differential metric values to the current path metric.

TABLE II
ILLUSTRATION OF COMPUTATION OF PARTIALLY KNOWN WALSH FUNCTION METRIC VALUES FOR COMBINED DEINTERLEAVING/DECODING

Binary Form	Metric Value	Diff. Met.
000	6	+0
001	2	−4
010	7	+0
011	4	−3
100	1	−4
101	5	+0
110	8	+0
111	3	−5

Binary Form	Metric Value	Diff. Met.
00−	6	−1
01−	7	+0
10−	5	−3
11−	8	+0

Binary Form	Metric Value	Diff. Met.
0−−	7	+7
1−−	8	+8

TABLE III
SAMPLE WALSH FUNCTION VALUES FOR CURRENT HYPOTHESIZED SURVIVING PATH

Walsh Func. Index	1	2	3	4	5	6
Walsh Func. Value	110−−−	010−−−	101−−−	110−−−	111−−−	001−−−

The basic steps of the proposed decoding process are summarized as follows:

Let N_s denote the number of surviving paths to be retained.

At each decoding trellis stage and node, do the following:

1. Extend each surviving path P to form four new paths P_i $(1 \leq i \leq 4)$.
2. For each new path, update the appropriate six Walsh functions with the newly hypothesized symbols, and reevaluate the path metric in Equation (5) by adding the precalculated differential metrics.
3. Keep the best N_s paths arriving at each node at the next stage in the decoding trellis.

3) Example: The following example, for illustration purposes, uses actual IS95 reverse link parameters. In IS95, the constraint length of the convolutional code is 9, and 2 b are shifted per decoding stage (with decoding proceeding on a Walsh function basis). This results in $2^7 = 128$ nodes in the trellis, each with four input and four output paths to be computed per decoding stage. Suppose we have a partially decoded frame and let us consider one of the 128 nodes in the decoding trellis. As explained in Section II-D2, only six Walsh functions will be affected in this decoding stage and let us assume that their decoded bits are as given in Table III. There are four possible combinations of the next two input bits which we can use to extend this surviving path. Let the corresponding output symbols from the convolutional encoder be as given in Table IV.

We then update the path metric for each of the extended paths. Table V shows an example set of differential metrics that are calculated prior to decoding the frame. We update the current metric of each partially defined Walsh function (consisting of three known and three unknown symbols) by hypothesizing the fourth decoded symbol to be either zero or one.

TABLE IV
INPUT DATA BITS AND ENCODED OUTPUT SYMBOLS AT SAMPLE TRELLIS NODE

Hypothesized Input Data Bits	Expected Encoder Output Symbols
00	011011
01	111101
10	001100
11	100100

The contents of Tables III through V are then used to revise the current path metric M_{P_i} for each of the path extensions shown in Table VI. We have assumed that the previous path metric has a value of 421. For each path extension, there are two hypothesized data bits, the corresponding expected convolutional encoder output symbols, the updated M_F corresponding to those output symbols, and the original path metric M_{P_i} updated by the appropriate differential values from Table V.

E. Algorithm Complexity

In the following, we discuss the dominant per-frame computational requirements of the different decoding methods, by considering the number of additions and comparisons.

The computations for a conventional Viterbi decoder with no symbol interleaving are common to all algorithms. In IS95, there are 96 decoding stages, 128 nodes per stage, and four path extensions per node, there are approximately (neglecting the beginning and end of the decoding trellis) a total of $96 \times 128 \times 4 = 49152$ additions and $96 \times 128 \times 3 = 36864$ comparisons.

In addition to the above computations, both the hard and soft-decision algorithms require additional computations for branch metric generation as well as deinterleaving. With hard-decision decoding, it is necessary to search among all 64 matched filter

TABLE V
SAMPLE DIFFERENTIAL METRIC VALUES

Walsh Function Index	New Walsh Function Value	Differential Metric
1	1100−−	+00
	1101−−	−67
2	0100−−	−40
	0101−−	+00
3	1010−−	+00
	1011−−	−07
4	1100−−	+00
	1101−−	−30
5	1110−−	+00
	1111−−	−46
6	0010−−	−43
	0011−−	+00

outputs to identify the maximum within each of the 96 Walsh function positions per frame. Consequently, there are $96 \times 63 = 6048$ comparisons, plus the deinterleaving of $576 \times 2 = 1152$ symbol metrics. The deinterleaved symbol metrics must then be summed to form branch metrics which requires $96 \times 64 \times 5 = 30720$ additions.

When soft-decision decoding is used, the maximum value among 32 Walsh function correlations must be identified for each of the two binary symbol values for each of 576 symbols per frame. This requires $576 \times 2 \times 31 = 35712$ comparisons, plus the same deinterleaving and additions as for the hard-decision algorithm.

For the proposed combined deinterleaver/decoder, we infer from Table II that each of 96 Walsh function positions require $32+16+\cdots+2 = 62$ comparisons to assign metric values for a total of $96 \times 62 = 5952$ comparisons. An equal number (5952) of difference operations must be performed in order to precalculate the differential metrics. Each of six differential metrics must also be added to the path metric to update each path extension (rather than one for the case of the standard Viterbi algorithm) for an approximate total of $96 \times 128 \times 4 \times 6 \times N_s = 294912N_s$ additions, with N_s denoting the number of retained surviving paths, which is typically set to be one or two. Also, roughly $96 \times 128 \times 4 \times N_s = 49152N_s$ comparisons are required.

Table VII summarizes the number of operations required per frame for each algorithm.

III. PERFORMANCE BOUND APPROXIMATION

We now estimate the BER for uncorrelated Rayleigh fading with no interleaving. This can be used as a performance bound for the case of perfect (infinite) interleaving.

The Walsh function correlation statistics have been derived in [16] and [17], with the correct and incorrect Walsh function correlations being Gaussian $N(N(\mu_F, \sigma_F^2)$ and $N(N(\mu_{\overline{F}}, \sigma_F^2)$ random variables, respectively, where

$$\mu_F = 128\sqrt{\pi}T_c\sqrt{E[P]} \tag{7}$$

$$\mu_{\overline{F}} = 0 \tag{8}$$

$$\sigma_F^2 = 512\left\{\frac{(N_M-1)T_c^2\{(\nu-1)\psi+1\}E[P]}{6\nu} + \frac{1}{2}T_c\sigma_n^2\right\}. \tag{9}$$

In (7) and (9), T_c is the PN chip period, N_M is the number of mobiles, ν is the factor by which transmission power is reduced when no data is being transmitted, ψ is the voice activity factor, and σ_n^2 is the background noise variance.

It can be shown [16] that the expected received power in (7) and (9) is given by

$$E[P] = \left(\frac{E_b}{N_0}\right)\sigma_n^2\left\{\frac{1}{R_B} - \frac{N_M-1}{B}\left(\frac{E_b}{N_0}\right) + \left(\psi + \frac{1-\psi}{\nu}\right)\right\}^{-1} \tag{10}$$

where E_b/N_0 is the desired received power ratio, R_B is the data bit rate (9600 bps), and B is the spread-spectrum bandwidth (1.2288 MHz).

A. Path Metrics

Consider the correct path P_1 and an incorrect path P_2 which differ in d Walsh functions. We need to statistically characterize the Walsh functions which are not common to both paths. Let X_1 and X_2 represent the metric values for the dissimilar portions of the two paths. Since we are summing Gaussian random variables, X_1 and X_2 will have distributions $N(\mu_1, \sigma_1^2) = N(d\mu_F, d\sigma_2^2)$ and $N(\mu_2, \sigma_2^2) = N(0, d\sigma_F^2)$, respectively. The difference, $X_3 = X_2 - X_1$, is Gaussian with parameters

$$\mu_3 = \mu_2 - \mu_1 = -d\mu_F \tag{11}$$

$$\sigma_3^2 = \sigma_2^2 + \sigma_1^2 = 2d\sigma_F^2. \tag{12}$$

We are interested in the probability

$$P(X_2 > X_1) = P(X_3 > 0) = \frac{1}{2}\,\text{erfc}\left(\frac{\mu_F}{2\sigma_F}\sqrt{d}\right). \tag{13}$$

B. Bit Error Rate (BER) Evaluation

The above statistics can be propagated through a standard Viterbi decoder to yield an estimated BER. We consider the noninterleaved case, since an analytical investigation of the combined deinterleaver/decoder appears intractable.

TABLE VI
UPDATED PATH METRIC VALUES AFTER SAMPLE SURVIVING PATH HAS BEEN EXTENDED WITH ALL FOUR POSSIBILITIES

Data Bits	Encoder Symbols	Updated Walsh Function Values	Revised Path Metric
00	011011	1100−− 0101−− 1011−− 1100−− 1111−− 0011−−	$421 + 00 + 00 - 07 + 00 - 46 + 00 = 368$
01	111101	1101−− 0101−− 1011−− 1101−− 1110−− 0011−−	$421 - 67 + 00 - 07 - 30 + 00 + 00 = 317$
10	001100	1100−− 0100−− 1011−− 1101−− 1110−− 0010−−	$421 + 00 - 40 - 07 - 30 + 00 - 43 = 301$
11	100100	1101−− 0100−− 1010−− 1101−− 1110−− 0010−−	$421 - 67 - 40 + 00 - 30 + 00 - 43 = 241$

We use the approach in [1] and [7] for obtaining an upper bound to the BER, although we obtain an actual approximate expectation by refining this method. From (13) and [7]

$$P_B \approx \frac{1}{2} \sum_{d=d_{\text{free}}}^{\infty} \beta_d \operatorname{erfc}\left(\frac{\mu_F}{2\sigma_F}\sqrt{d}\right). \tag{14}$$

In (14), the minimum free distance is $d_{\text{free}} = 5$ for IS95, β_d is a weighting coefficient based on the number of paths and output bit errors, and d represents the distance of an incorrect path from the correct path in terms of the number of incorrect Walsh functions. The determination of β_d is described in the Appendix.

It should be noted that (14) is only an approximation due to the finite frame length and also due to the assumed statistical independence of distinct path metrics, which is not the case since trellis branches are shared among a number of paths. In practice, usually a maximum of only 25 terms of the summation in (14) are required due to the exponential decrease of the erfc$(\cdot)$ term.

IV. RESULTS AND DISCUSSION

We have evaluated the proposed algorithm in a chip-level reverse link simulation of IS95 [16], [17] consisting of one mobile in a circular cell of radius 500 m. This simulator is capable of modeling a correlated or uncorrelated Rayleigh fading channel complete with path loss and shadowing effects. We have also derived and verified the probability distributions for the inputs to the Viterbi decoder in a memoryless channel [16]. It is therefore possible, in this instance, to generate these random values directly without requiring PN chip-level simulation.

Fig. 3 plots the simulated BER of memoryless Rayleigh fading for hard-decision deinterleaved symbol metrics (Section II-B), soft-decision deinterleaved symbol metrics (Section II-C), and the proposed decoder (Section II-D) with one and two surviving paths being retained, respectively. Simulated noninterleaved results (Section II-A) and the analytically calculated BER (Section III-B) are included for comparison. At low E_b/N_0, incorrect path selection reduces interleaving performance, while at high E_b/N_0, the correlated Walsh functions reduce the relative performance of the noninterleaved system. From Fig. 3, the deinterleaver/decoder with one surviving path provides about a 1-dB improvement over soft-decision (Section II-C) and a 2-dB improvement over hard-decision (Section II-C) at a BER of 10^{-3}. Retaining two surviving paths instead of one yields another 0.5-dB improvement.

When comparing the analytical and simulated BER for no interleaving, good agreement is obtained over the E_b/N_0 range from 3.5 to 5.5 dB. An exact correspondence is not expected since there may be significant branch overlap between the longer trellis paths which were assumed to be statistically independent in the analysis. At lower E_b/N_0, longer paths have a nonnegligible contribution to the summation in (14), and the assumed statistical independence breaks down.

Time-correlated Rayleigh fading represents a more realistic evaluation environment for interleaving in IS95. These simula-

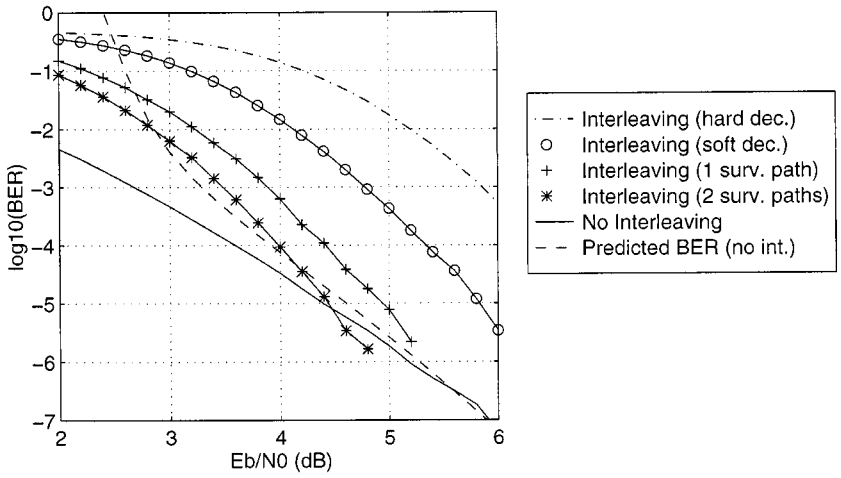

Fig. 3. BER for one mobile with uncorrelated Rayleigh fading.

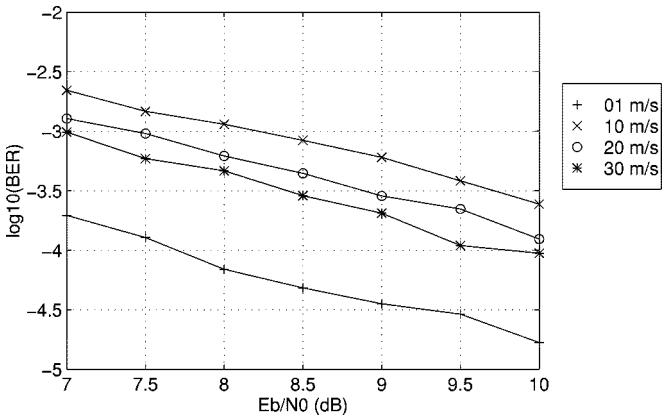

Fig. 4. BER for one mobile with correlated Rayleigh fading and interleaving (combined deinterleaver/decoder).

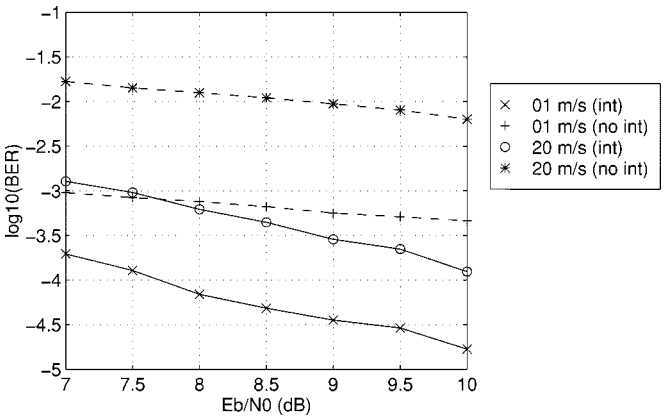

Fig. 5. BER for one mobile with correlated Rayleigh fading (with combined deinterleaver/decoder and without interleaving).

tions were conducted at the PN chip-level using correlated shadowed fading [18]. Fig. 4 shows the simulated BER for user velocities ranging from 1 to 30 m/s when the combined deinterleaver/decoder is used with one surviving path being retained. As expected, a lower BER is observed at higher velocities since the mobile remains in fades for shorter periods of time. At low mobile velocities, however, the IS95 closed-loop power control algorithm is rapid enough to compensate for the slow fading. Fig. 5 illustrates the improvement in BER through the introduction of interleaving for user velocities of 1 and 20 m/s in correlated fading.

Fig. 6 shows the improvement obtained by increasing the number of retained paths in the proposed decoder. However, added path computations trade off against the improved performance. From Fig. 6, retaining two paths per node seems beneficial, while from Table VII, retaining additional paths provides diminishing returns with increased computational expense.

In summary, the IS95 reverse link BER performance can be improved by about 1.2 dB using the proposed deinterleaver/decoder over that proposed in [1] with additional computation with no additional decoding delay. A performance bound was derived for the IS95 reverse link for the case of no interleaving. Simulation results have shown agreement between observations and predictions, and have indicated that the modified decoder performs relatively well in correlated fading.

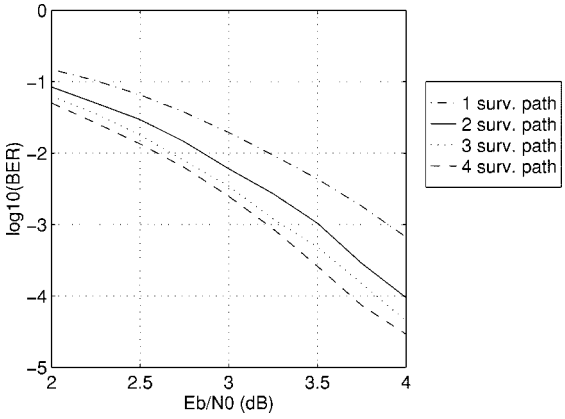

Fig. 6. BER for one mobile with uncorrelated Rayleigh fading (interleaving with combined deinterleaver/decoder).

TABLE VII
SUMMARY OF DECODING ALGORITHM COMPLEXITY

Algorithm	Additions	Comparisons	Deinterleaving
No Interleaving	49152	36864	N
Hard-Decision	79872	42912	Y
Soft-Decision	79872	72576	Y
Deint/Decoder	$5952 + 294912N_s$	$5952 + 49152N_s$	N

APPENDIX I
CALCULATION OF β_d

We first obtain the transfer function [7], $T(D, N)$, for IS95 encoding without interleaving, where the exponent of D, $f(d)$, is the distance of the current path from the correct path in terms of incorrect Walsh functions, and the exponent of N is the number of output bit errors. $T(D, N)$ can be obtained by deriving a system of equations from the encoding trellis, which can then be solved using symbolic computation software.

An infinite series representation of the transfer function is

$$T(D,N) = \sum_{d=d_{\text{free}}}^{\infty} a_d D^d N^{f(d)} \tag{15}$$

where a_d is the number of paths of distance d from the correct path with the specified number of output bit errors. From (15)

$$\left.\frac{dT(D,N)}{dN}\right|_{N=1} = \sum_{d=d_{\text{free}}}^{\infty} f(d) a_d D^d = \sum_{d=d_{\text{free}}}^{\infty} \beta_d D^d \tag{16}$$

where β_d is a weighting term which combines the number of paths and the number of output bit errors for the current probability term.

Since (16) specifies the β_d coefficients in (13), the approximate BER in (14) can be obtained. For numerical purposes, (16) can also be expressed as the ratio of two polynomials which, for the case of IS95 encoding with no interleaving, can be derived as

$$\begin{aligned} G = {} & d^{51} - 16d^{50} + 118d^{49} - 532d^{48} + 1664d^{47} - 4003d^{46} \\ & + 8338d^{45} - 16\,528d^{44} + 30\,644d^{43} - 47\,518d^{42} \\ & + 52\,593d^{41} - 24\,631d^{40} - 43\,887d^{39} + 125\,350d^{38} \\ & - 161\,892d^{37} + 108\,602d^{36} + 17\,501d^{35} - 139\,070d^{34} \\ & + 178\,294d^{33} - 117\,113d^{32} + 6252d^{31} + 76\,146d^{30} \\ & - 83\,526d^{29} + 34\,115d^{28} + 14\,432d^{27} - 27\,526d^{26} \\ & + 19\,847d^{25} - 15\,612d^{24} + 11\,283d^{23} + 3658d^{22} \\ & - 18\,527d^{21} + 18\,427d^{20} - 7280d^{19} - 1594d^{18} \\ & + 2691d^{17} - 169d^{16} - 667d^{15} - 386d^{14} \\ & + 1214d^{13} - 847d^{12} - 43d^{11} + 168d^{10} \\ & + 42d^{9} + 17d^{8} - 41d^{7} + 11d^{6} + 4d^{5} \end{aligned}$$

$$\begin{aligned} H = {} & 4d^{50} - 52d^{49} + 289d^{48} - 864d^{47} + 1350d^{46} - 436d^{45} \\ & - 2491d^{44} + 5576d^{43} - 6912d^{42} + 7382d^{41} \\ & - 5944d^{40} - 3360d^{39} + 20\,574d^{38} - 33\,378d^{37} \\ & + 30\,938d^{36} - 14\,968d^{35} - 6965d^{34} + 28\,584d^{33} \\ & - 44\,096d^{32} + 45\,376d^{31} - 25\,127d^{30} - 11\,006d^{29} \\ & + 41\,533d^{28} - 45\,134d^{27} + 19\,584d^{26} + 12\,746d^{25} \\ & - 25\,958d^{24} + 17\,018d^{23} - 174d^{22} - 10\,366d^{21} \\ & + 8565d^{20} - 1402d^{19} - 2548d^{18} + 2762d^{17} \\ & - 850d^{16} - 314d^{15} - 29d^{14} - 26d^{13} - 22d^{12} \\ & + 110d^{11} + 297d^{10} - 24d^{9} - 132d^{8} - 54d^{7} \\ & - 21d^{6} + 22d^{5} + 54d^{4} + 14d^{3} + d^{2} - 6d + 1. \end{aligned}$$

The values of β_d in (16) can then be calculated by dividing H into G.

REFERENCES

[1] A. J. Viterbi, *CDMA: Principles of Spread Spectrum Communication*. Reading, MA: Addison-Wesley, 1995.

[2] ——, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[3] G. D. Forney, Jr., "The Viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[4] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*. New York: McGraw-Hill, 1979.

[5] Qualcomm Inc., *Mobile Station-Base Compatibility Standard for Dual-Mode Wideband Spread System*, 1993.

[6] L. F. Chang, F. Ling, D. D. Falconer, and N. R. Sollenberger, "Comparison of two convolutional orthogonal coding techniques for CDMA radio communications systems," *IEEE Trans. Commun.*, vol. 43, no. 6, pp. 2028–2037, 1995.

[7] J. G. Proakis, *Digital Communications*. New York: McGraw-Hill, 1989.

[8] R. M. Fano, "A heuristic discussion of probabilistic decoding," *IEEE Trans. Inform. Theory*, vol. 9, pp. 64–74, 1963.

[9] F. Jelinek, "Fast sequential decoding algorithm using a stack," *IBM J. Res. Develop.*, vol. 13, no. 6, pp. 675–685, 1969.

[10] F. Ling and D. D. Falconer, "Combined orthogonal/convolutional coding for a digital cellular system," in *1992 IEEE 42nd Veh. Technol. Conf.*, pp. 63–66.

[11] L. F. Chang and N. R. Sollenberger, "Comparison of two interleaving techniques for CDMA radio communications systems," in *1992 IEEE 42nd Veh. Technol. Conf.*, pp. 275–278.

[12] M. J. Miller, B. Vucetic, and L. Berry, *Satellite Communications: Mobile and Fixed Services*. Norwell, MA: Kluwer, 1993.

[13] A. Jalali and P. Mermelstein, "Effects of diversity, power control, and bandwidth on the capacity of microcellular CDMA systems," *IEEE J. Select. Areas Commun.*, vol. 12, no. 5, pp. 952–961, 1994.

[14] A. F. Naguib and A. Paulraj, "Performance of DS/CDMA with M-ary orthogonal modulation cell site antenna arrays," in *1995 Int. Conf. Commun.*, pp. 697–702.

[15] ——, "Performance of wireless CDMA with M-ary orthogonal modulation and cell site antenna arrays," *IEEE J. Select. Areas Commun.*, vol. 14, no. 9, pp. 1770–1783, 1996.

[16] A. M. Earnshaw and S. D. Blostein, "A chip-level IS95-compliant cellular CDMA simulator: Design, implementation, and analysis," Queen's Univ. Tech. Rep., Canada, Available: http.//ipcl.ee.queensu.ca, 1996.

[17] A. M. Earnshaw, "An investigation into improving performance of cellular CDMA communication systems with digital beamforming," Ph.D. dissertation, Queen's Univ., Canada, 1997.

[18] C. Loo, "A statistical model for a land mobile satellite link," *IEEE Trans. Veh. Technol.*, vol. 34, no. 3, pp. 122–127, 1985.

A. Mark Earnshaw (M'97) received the B.A.Sc. and M.A.Sc. degrees in systems design engineering from the University of Waterloo, Canada, in 1989 and 1991, respectively, and the Ph.D. degree from Queen's University, Kingston, Ont., Canada, in 1997.

While at Queen's University, he was an Adjunct Instructor, Research Assistant, and Teaching Assistant. Since 1997, he has been with Nortel Networks, Nepean, Ont., as a Radio Systems Designer, performing research and development on next-generation CDMA systems.

Dr. Earnshaw was the recipient of a Postgraduate Scholarship from the Natural Sciences and Engineering Research Council of Canada.

Steven D. Blostein (S'83–M'88–SM'96) received the B.S. degree in electrical engineering from Cornell University, Ithaca, NY, in 1983 and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Illinois, Urbana-Champaign, in 1985 and 1988, respectively.

He has been on the Faculty at Queen's University, Kingston, Ont., Canada, since 1988 and currently holds the position of Associate Professor in the Department of Electrical and Computer Engineering. He has been a Consultant to both industry and government in the areas of document image compression, motion estimation, and target tracking and was a Visiting Associate Professor in the Department of Electrical Engineering, McGill University, Canada, in 1995. His current interests lie in statistical signal processing, wireless communications, and video image communications. He is currently an Associate Editor for the IEEE IMAGE PROCESSING

Dr. Blostein served as Chair of the IEEE Kingston Section from 1993 to 1994. He currently leads the Multirate Wireless Data Access Major Project sponsored by the Canadian Institute for Telecommunications Research. He is a registered Professional Engineer in Ontario, Canada.