

## Research Article

# Iterative Signal Processing for Blind Code Phase Acquisition of CDMA 1x Signals for Radio Spectrum Monitoring

**Ron Kerr and John Lodge**

*Satellite Communications and Radio Propagation, Communications Research Centre Canada, 3701 Carling Avenue, Box 11490, Station H, Ottawa, ON, Canada K2H 8S2*

Correspondence should be addressed to Ron Kerr, ron.kerr@crc.gc.ca

Received 10 February 2010; Revised 14 June 2010; Accepted 2 August 2010

Academic Editor: Peter Hoehner

Copyright © 2010 R. Kerr and J. Lodge. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper addresses the problem of recovering the code phase of the composite spreading sequence for a CDMA 1x signal transmitted from a handset, without the benefit of a priori information from the system. The spreading code is required for the radio spectrum monitoring system for signal detection and measurements rather than for communications. The structure of the CDMA 1x signal is exploited by processing sequential pairs of received samples to form a single soft sample for each pair. The approach models the combination of the long-code generator and the two short-code generators, along with the pair-wise processing, by a single linear system over  $GF(2)$ , with the initial states of the long- and short-code generators forming the input vector. Consequently, a vector of the pair-wise soft samples can be treated as a noisy received codeword that is decoded using iterative soft-in decoding techniques. If the decoder yields the correct candidate “codeword,” the original states of the code generators can be computed. This approach does not require direct access to the transmitted spreading sequence but can be applied to the data modulated signal. Simulation results provide performance estimates of the method with noise, Rayleigh fading, and co-channel interference.

## 1. Introduction

Code division multiple access (CDMA) has become a popular choice for personal communications systems. With the multiple-access scheme multiple users can simultaneously share the same frequency band by using high-rate spreading codes to disperse their transmitted power over the fully-allotted frequency band, thereby keeping the power spectral density for each user at a relatively low level. CDMA represents a new set of challenges for spectrum regulators that need to monitor radio traffic volumes and check license compliance as part of their responsibility to efficiently manage the frequency spectrum in a region. To recover one of the transmitted signals, the receiver generates a local replica of the transmitter’s spreading code and uses it to “despread” the desired signal. If the spreading code for the desired signal is suitably distinct from the spreading codes for the other signals present in the band, the desired signal can be received with relatively little interference. Thus, it is

important that each user is assigned a different spreading code. In CDMA 1x, all users generate their own spreading codes using the same linear feedback shift registers (LFSR). The requirement of assigning distinct spreading codes is achieved by assigning each user a different starting point in the long spreading code. Thus, each user’s code is a distinct time-shifted version of the shared code, and synchronizing to any given user’s code is essentially determining the time reference, or equivalently code phase, for that user.

As the network assigns the starting point in the sequence, a receiving base station knows, within a small uncertainty region, the code phase of all of the users assigned to it. A blind acquisition technique is not required for the base station as it has the users’ code phase information. The work described in this paper is motivated by the application of spectrum monitoring, where it is necessary to perform measurements on the spectrum usage and to locate transmitters. This is a difficult problem as the mobiles’ signals use the same frequency bandwidth and the equipment must operate

without information from the cellular networks. A method to achieve the desired goals is to blindly acquire the code phase of a user in the area to aid in signal detection, signal parameter measurements, direction-finding, and interference cancellation in order to detect other users.

An introduction to acquisitions techniques used for direct sequence spread spectrum (DSSS) is presented in [1]. Three types of acquisition schemes are serial search, parallel search, and sequential estimation. For the sake of discussion, consider the case where there is a spreading sequence which has no underlying structure that can be utilized to acquire the code phase, the sequence has a long period and the receiver has no information about the expected code phase. In a serial search strategy, the receiver would have to test each code phase serially until a correlation with the received signal indicates that the sequences are aligned. The expected number of required tests is approximately half the sequence length [2]. In parallel search techniques, more correlators are used in the receiver to perform multiple tests simultaneously. This approach can reduce the acquisition time compared with the serial approach but with an increase in complexity. In the case of very long code periods, the serial and parallel techniques quickly become impractical due to their expected number of correlations required. It should be noted that there are long spreading sequences that are made up of several shorter component sequences, such as PN ranging codes, which are discussed in [3]. The structure of the shorter component codes can be used to rapidly acquire the code phase of the long spreading code. This type of spreading sequence is not considered here. In this paper, the CDMA 1x cellular standard [4] is used to illustrate the blind acquisition method. The spreading sequence is made up of a long pseudonoise (PN) sequence with period  $2^{42} - 1$  and two short PN sequences with period of  $2^{15}$ . The serial and parallel approaches are computationally impractical for acquiring the sequence quickly due to the expected number of correlations required.

The third class of search strategies for acquisition is sequential estimation. Here, the detector uses information on the chip sequence to estimate the state of the shift register. In [5], hard chip estimates were made and loaded into the spreading generator. The method did not use any of the redundancy in the chip sequence. In [6], orthogonal parity check equations and threshold decoding to improve the chip decisions that were then loaded into the spreading generator were utilized. The  $n$  chips from a spreading sequence from an LFSR was recognized as a truncated codeword of a maximum length code in [7]. Majority logic decoding was used on a number of independent estimates of the bits in the spreading generator's state.

The above methods used hard decisions on the chips in the sequence. Soft decisions were introduced in [2] for majority logic decoding and the RARASE algorithm [8]. In [9], low-density parity check (LDPC) decoding was used to find the state of an LFSR spreading generator. In [10], the method presented used recursive soft sequential estimation (RSSE) and a soft-in soft-out decoder to update the reliabilities associated with the chips in the sequence. A soft-chip register that stored the current reliabilities was

used. The soft-in-soft-out (SISO) decoder was used to update the current chip reliability with information from the channel and the soft-chip register. A differential recursive soft sequential estimator (DRSSE) algorithm was introduced in [11] and eliminated the need for an accurate carrier frequency estimate required in the RSSE algorithm. In [12–14], iterative message passing algorithms were considered for finding the state of spreading sequence generators.

An approach for acquisition of DS-CDMA signals is to treat the spreading code acquisition problem as a decoding problem. In many cases, the spreading code used in the CDMA system is generated by a linear system. The structure of the linear system defines a linear code. In the above references, the problem of acquiring the phase of the spreading sequence using decoding methods was considered. However, there was an assumption that the receiver had access to the CDMA signal modulated only by a spreading sequence (i.e., the chip decisions) and not by data. It is felt that the DRSSE algorithm [11] would not be affected by the data, but the other algorithms would be adversely affected in the presence of data modulation, especially if the spreading factor (i.e., number of bits per chip) is less than the observation interval. The presence of data modulation prevents access to the chip values from the channel and thus the algorithms requiring chip decisions will not work for all cases.

The algorithm that will be presented is capable of acquiring the spreading code when data modulation is present on the signal. In the CDMA 1x system, the effect of the data modulation can be eliminated on the single soft sample that results from each pair of sequential samples. As a result, the algorithm is independent of the data modulated on the signal and can acquire during any phase of the communication. Another benefit of processing sequential pairs of channel samples is the algorithm becomes robust against unknown carrier frequency offsets. In the case of CDMA 1x, the knowledge of the frame structure can be used to quickly eliminate a large number of incorrect spreading generator states without attempting the despreading of the signal.

In this paper, we will present an algorithm that uses iterative soft input decoding to acquire the code phase of a CDMA 1x spread signal without any system information available. This allows the receiver to regenerate the spreading sequence required for despreading of the signal. An application of interest is for use in a spectrum monitoring and signal detection. Once the spreading sequence is found, the signal can be despread for use in monitoring or direction finding.

The outline for the remainder of this paper is as follows. In Section 2, the development of the equations and decoding algorithm will be presented. Performance results will be presented in Section 3, and Section 4 will provide concluding remarks.

## 2. System Model

In several cellular standards [4, 15, 16], a complex-valued spreading sequence is formed from two bipolar sequences as shown in Figure 1, where  $c_1$  and  $c_2$  are bipolar sequences, which take on values of  $\pm 1$ . The real component of the

complex sequence (i.e.,  $\text{Real}(C_n)$ ) is one of the bipolar sequences. The imaginary component is formed by the product of the first sequence, a decimation by two of the second sequence, and an alternating sequence. The resulting complex sequence  $C$  can be described by

$$C_n = c_{1,n}(1 + j(-1)^n c_{2,2\lfloor n/2 \rfloor}), \quad (1)$$

where  $C_n$  is the  $n$ th value of the complex spreading sequence  $C$ ,  $c_{m,n}$  is the  $n$ th value of the  $m$ th bipolar sequence,  $j = \sqrt{-1}$ , and  $\lfloor \cdot \rfloor$  is the floor function where the function returns the nearest integer below the argument.

In acquiring the spreading sequence of an unknown transmitter, it is assumed that the structure of the spreading generators that generate the  $c_1$  and  $c_2$  sequences and the procedures for initializing generators are known. For the purposes of the presentation it is assumed that the spreading factor is even, a minor modification of the technique would be required if the spreading factor was odd. In the following, the signal is sampled at one sample per chip and chip timing is perfect. However, generalization to an oversampled signal with imperfect timing is straightforward.

In this section, the method of processing the signal so that the resulting signal can be considered to be a linear code depending on the initial state of the spreading generator is shown. Consider two adjacent complex samples  $C_n$  and  $C_{n+1}$  of the sequence defined in (1), where  $n$  is an even number. The samples are defined by

$$\begin{aligned} C_n &= c_{1,n} + jc_{1,n}c_{2,n}, \\ C_{n+1} &= c_{1,n+1} - jc_{1,n+1}c_{2,n}. \end{aligned} \quad (2)$$

Note that the contribution to the sequence from  $c_2$  is the same for both chips due to the decimation process. The data associated with the  $n$ th chip is denoted by  $D_n = d_{1,n} + jd_{2,n}$  where the data bits  $d_1$  and  $d_2$  are bipolar valued. The spreading is a complex spreading process, so the two corresponding samples after spreading are given by

$$\begin{aligned} r_n &= (d_{1,n}c_{1,n} - d_{2,n}c_{1,n}c_{2,n}) \\ &\quad + j(d_{1,n}c_{1,n}c_{2,n} + d_{2,n}c_{1,n}), \\ r_{n+1} &= (d_{1,n+1}c_{1,n+1} + d_{2,n+1}c_{1,n+1}c_{2,n}) \\ &\quad + j(d_{1,n+1}c_{1,n+1}c_{2,n} - d_{2,n+1}c_{1,n+1}). \end{aligned} \quad (3)$$

The product of an even index ( $n$ ) sample  $r_n$  and the conjugate of the next sample (i.e.,  $r_n r_{n+1}^*$ ) will be denoted by  $y_n$  and is given by

$$\begin{aligned} y_n &= 2c_{1,n}c_{1,n+1}c_{2,n}((d_{1,n}d_{2,n+1} - d_{1,n+1}d_{2,n}) \\ &\quad + j(d_{1,n}d_{1,n+1} + d_{2,n}d_{2,n+1})). \end{aligned} \quad (4)$$

The assumption that the spreading factor used for the data is an even number and is aligned with the alternating sequence results in the property that  $d_{1,n} = d_{1,n+1}$  and  $d_{2,n} = d_{2,n+1}$  when  $n$  is an even number. With this property, (4) reduces to

$$y_n = 4jc_{1,n}c_{1,n+1}c_{2,n}. \quad (5)$$

Thus, from (5), the value of  $y_n$  is imaginary and proportional to the product of three bipolar sequences and does not depend on the data being transmitted on the channel. The individual sequences can be formed by linear equations on the original state of the spreading generators. Since the product of bipolar elements corresponds to addition in the Galois Field GF(2), this means that the value of  $y_n$  can be expressed as a linear binary equation on the original state of the spreading generators. A truncated sequence composed of

$$\mathbf{y}^b = y_n, y_{n+2}, y_{n+4}, \dots, y_{n+m} \quad (6)$$

can then be generated by a system of  $m$  linear equations based on the original state of the spreading generator. The linear equations can be formed into a binary matrix equation where  $\mathbf{G}$  is the generator matrix and  $\mathbf{x}$  is the state of the spreading generator, given by

$$\mathbf{y} = \mathbf{G}\mathbf{x}, \quad (7)$$

where  $\mathbf{y}$  is the binary image of the bipolar sequence  $\mathbf{y}^b$  (i.e., “4j” is mapped to 0 and “-4j” is mapped to 1), and the arithmetic in (7) is over GF(2). Let the dimension of the state of the shift register be  $k$ . If  $m$  is greater than  $k$ , then (7) is equivalent to the encoding of an  $(m, k)$  code, with  $\mathbf{y}$ ,  $\mathbf{G}$ , and  $\mathbf{x}$  being the  $m$ -dimensional codeword, the generator matrix for the code and the  $k$ -dimensional information bits, respectively. As the truncated sequence can be seen to a codeword of a linear code, a parity matrix  $\mathbf{H}$  can be found such that

$$\mathbf{H}\mathbf{G} = \mathbf{0} \quad (8)$$

over GF(2).

A decoder algorithm can utilize the constraints specified by the code and the noisy received version of the codeword to obtain an estimate of the codeword  $\mathbf{y}$ . Once a codeword estimate is found, then (7) can be used to solve for the  $\mathbf{x}$ . An efficient method is to compute a pseudoinverse matrix,  $\mathbf{P}^\#$  which satisfies the following equation:

$$\mathbf{x} = \mathbf{P}^\#\mathbf{y}. \quad (9)$$

The  $\mathbf{P}^\#$  is constant for a given generator matrix and thus can be computed and stored for a given choice of block size. Thus,  $\mathbf{x}$  can be solved with a matrix-vector multiplication given a codeword estimate.

**2.1. CDMA 1x Information.** A block diagram of the spreading generator for CDMA 1x is shown in Figure 2 [4]. The complex spreading code is generated by two bipolar sequences. The two sequences are generated by three independent linear feedback shift registers (LFSR) sequences denoted as the I and Q channel short-code generators and the long-code generator.

In the CDMA 1x system, the state of the shift registers is a function of the system time [4]. A channel mask is used on the long shift register and adds contributions from the various delays within the shift register to form the output bit. The mask is used to shift the phase of the spreading sequence

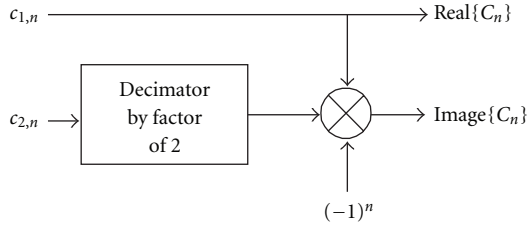


FIGURE 1: Block diagram of a method to form a complex spreading sequence from two bipolar sequences [4].

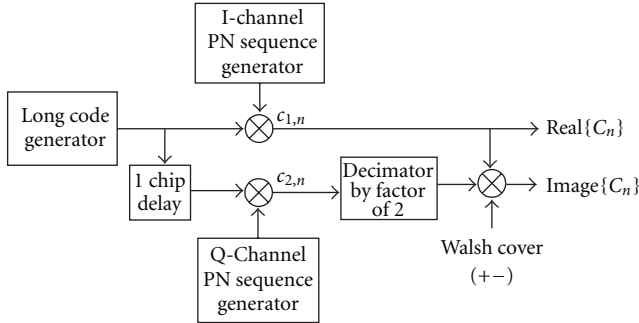


FIGURE 2: Block diagram of CDMA 1x spreading sequence generation.

of the common state of the shift register in the system to a unique phase for each user. The combining of delayed versions of the sequence within the register forms another phase of the same sequence. As a result of this, a state for the original long shift register (without the mask) will produce the same output sequence as the original state of the shift register with the mask. Thus, it suffices to solve for the state of the shift register without the mask since it will produce the desired spreading sequence. To form the generator matrix for the system, the constant multiplier is removed from (5) and converted to work with binary equations as previously described. A logic diagram showing the equivalent binary process is shown in Figure 3. Given

$$\begin{aligned}
 y_n &= c_{1,2n} \oplus c_{1,2n+1} \oplus c_{2,2n} \\
 &= \mathbf{b}_0 \mathbf{x}_{0,2n+1} \oplus \mathbf{b}_1 \mathbf{x}_{1,2n} \oplus \mathbf{b}_0 \mathbf{x}_{0,2n+2} \oplus \mathbf{b}_1 \mathbf{x}_{1,2n+1} \\
 &\quad \oplus \mathbf{b}_0 \mathbf{x}_{0,2n} \oplus \mathbf{b}_2 \mathbf{x}_{2,2n} \\
 &= \mathbf{b}_0 (\mathbf{x}_{0,2n} \oplus \mathbf{x}_{0,2n+1} \oplus \mathbf{x}_{0,2n+2}) \\
 &\quad \oplus \mathbf{b}_1 (\mathbf{x}_{1,2n} \oplus \mathbf{x}_{1,2n+1}) \oplus \mathbf{b}_2 \mathbf{x}_{2,2n} \\
 &= \mathbf{b}_0 (\mathbf{A}_0^{2n} \oplus \mathbf{A}_0^{2n+1} \oplus \mathbf{A}_0^{2n+2}) \mathbf{x}_{0,0} \\
 &\quad \oplus \mathbf{b}_1 (\mathbf{A}_1^{2n} \oplus \mathbf{A}_1^{2n+1}) \mathbf{x}_{1,0} \oplus \mathbf{b}_2 \mathbf{A}_2^{2n} \mathbf{x}_{2,0},
 \end{aligned} \tag{10}$$

where  $\mathbf{x}_{k,n}$  is the state vector for the  $k$ th LFSR at time  $n$ ,  $\mathbf{b}_k$  is the observation vector so that  $\mathbf{b}_k \mathbf{x}_{k,n}$  produces the output at time  $n$  for the  $k$ th LFSR,  $\mathbf{A}_k$  is the transition matrix for the  $k$ th LFSR (i.e.,  $\mathbf{x}_{k,n+1} = \mathbf{A}_k \mathbf{x}_{k,n}$ ), and  $\oplus$  is addition in  $\text{GF}(2)$ . A system state vector can be formed by concatenating the individual LFSR state vectors, that is,

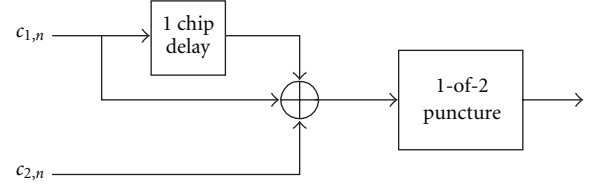


FIGURE 3: Logic diagram for processing the result from the sequential processing of the complex sequence.

$\mathbf{x}_n = [x_{0,n} \ x_{1,n} \ x_{2,n}]^T$ . The matrix equation that provides the  $\mathbf{y}$  is in the form

$$\mathbf{y} = \mathbf{G} \mathbf{x}_0, \tag{11}$$

where  $\mathbf{x}_0$  is the initial state of the spreading generator at a time "0",  $\mathbf{y} = [y_0, y_1, \dots, y_m]^T$ , and the  $n$ th row of  $\mathbf{G}$ , denoted as  $g_n$ , is formed by arranging the terms in (10) to correspond to the system state vector, that is,

$$g_n = [\mathbf{b}_0 (\mathbf{A}_0^{2n} \oplus \mathbf{A}_0^{2n+1} \oplus \mathbf{A}_0^{2n+2}) \ \mathbf{b}_1 (\mathbf{A}_1^{2n} \oplus \mathbf{A}_1^{2n+1}) \ \mathbf{b}_2 \mathbf{A}_2^{2n}]. \tag{12}$$

All of the matrices and vectors in (12) are dependent on the LFSRs used in the system. For the CDMA 1x system, the state of the spreading generator is the sum of the dimensions of the individual shift registers, that is, 72 (i.e., 42 + 15 + 15). Thus, the dimension of  $\mathbf{G}$  is a  $m \times 72$  matrix,  $\mathbf{x}_0$  and  $\mathbf{y}$  are  $72 \times 1$  and  $m \times 1$ , respectively. The vector  $\mathbf{y}$  can be seen as a codeword for a  $(m, 72)$  code defined by the generator matrix  $\mathbf{G}$  with  $\mathbf{x}_0$  being treated as the information bits associated with the codeword. The corresponding parity check matrix,  $\mathbf{H}$ , has dimension  $(m - 72) \times m$  and the solution matrix  $\mathbf{P}^\#$ , which is used to solve for the state of the shift registers given 72 contiguous values from the vector  $\mathbf{y}$ , is a  $72 \times 72$  matrix.

In the CDMA 1x system, the I and Q short shift registers have a known state at the beginning of the frame [4]. The shift registers are clocked simultaneously for 32767 cycles and then a fill zero is added to form a frame of 32768 chips [4]. The development of the equations did not take into consideration the extra fill bit and thus the parity check matrix will not be valid for the  $\mathbf{y}$  vector if the fill bit falls within the samples used to create  $\mathbf{y}$ . The decoding algorithm will not be able to converge to a valid codeword when the parity check matrix is not valid. However as practical block sizes used for decoding are much smaller than the frame size, the number of starting positions within the frame where the equations are not valid is small. For example, if a block size of 1024 is used for the decoder, the receiver needs 2048 samples to form the  $\mathbf{y}$  vector. Thus, there are only 2048 starting positions where the fill bit would be contained in the vector, which leaves 30720 positions within the frame which will have valid equations.

In order to check if the solved state of the shift register is valid, one method is to load a local spreading generator and run the generator to produce the spreading sequence and then correlate this sequence with the received sequence. However, there is a less complex method that can be utilized

that is based on the initialization of the short spreading registers at the beginning of each frame. As the states of the short shift registers are known at the beginning of the frame and the registers are clocked together, the states of both shift registers are known for each offset (in chips) from the start of the frame. Another property that results from this procedure is there is a one-to-one correspondence of the states in the I and Q short shift registers. In other words, there are valid pairs of states. This property can be used to determine if the solved state of the spreading generators is likely to be correct. To use this method, the decoder forms a candidate codeword (described in the next section), and then solves for states of the short shift registers,  $x_1$  and  $x_2$ . As an example, consider the case where the solved state  $x_1$  corresponds to an offset of  $k$  chips from the start of the frame for the I short sequence generator. The state for the Q short sequence generator for the  $k$ th offset from the start of the frame is compared with the solved state  $x_2$ . If the states are not equal, then at least one of the solved states is incorrect which implies that the current candidate codeword is not valid. If the states are equal, then it is highly likely that the solved state is correct.

**2.2. Decoding Algorithm.** The decoding algorithm used to find an estimate of the codeword is the Vector SISO [17]. The algorithm is a soft-in soft-out decoding algorithm. Versions of the algorithm for binary codes have been described in [17–19]. For this application, the algorithm was modified to return hard decisions only. For use with CDMA 1x detection, the decision rule to determine if the estimated codeword is likely to be correct was modified to use the correlations in the I and Q short shift registers as described above.

For the sake of completeness, a brief outline of the decoding algorithm is provided here. For more detail regarding the decoding algorithm, the reader is referred to [17] or [18].

Consider general linear codes for which codewords can be generated from  $\mathbf{y} = \mathbf{G}\mathbf{x}$ , where  $\mathbf{x}$  is a column  $k$ -vector of information symbols,  $\mathbf{y}$  is a column  $n$ -vector of coded bits, and  $\mathbf{G}$  is the  $n \times k$  generator matrix. A parity check matrix,  $\mathbf{H}$ , is any  $(n-k) \times n$  matrix of rank  $(n-k)$  for which  $\mathbf{H}\mathbf{G} = \mathbf{0}$ .

A set of symbol positions (i.e., indices in the codeword vector) are said to be linearly independent if the corresponding columns of a parity check matrix are linearly independent.

A parity check matrix is referred to as being pseudosystematic form relative to a set of  $n-k$  symbol positions if by moving the corresponding  $n-k$  columns, and appropriately ordering them, the matrix can be put in the form  $\mathbf{H} = [\mathbf{I} \ \mathbf{H}']$  where  $\mathbf{I}$  denotes the identity matrix. If bit values are given for the  $n-k$  positions corresponding to the nonidentity portion of the pseudosystematic parity check matrix, it is easy to form the entire codeword by selecting the remaining  $k$  bits to satisfy the parity constraints (e.g., by forming the corresponding pseudo-systematic generator matrix). Forming the codewords in this manner is referred to here as “recoding.”

We can obtain  $\mathbf{d}$  from  $\mathbf{y}$  by mapping 0-valued elements in  $\mathbf{y}$  to 1-valued elements in  $\mathbf{d}$ , and 1-valued elements in  $\mathbf{y}$  to -1-valued elements in  $\mathbf{d}$ . There are  $2^k$  possible vector values

that can be assumed by  $\mathbf{d}$ , and when enumeration is required the  $j$ th possibility is denoted  $d_j$  with  $m$ th element  $d_{m,j}$ .

The general problem can be stated as follows. Given that one of the  $2^k$  possible  $\mathbf{d}$  vectors has been transmitted (each assumed here to be equally likely), and  $\mathbf{r}$  the corresponding vector of noisy samples received over an antipodal additive white Gaussian noise channel, approximate the *a posteriori* log likelihood ratio for each bit in the coded vector. For this application we are only concerned with the corresponding hard decisions.

The required approximate values can be computed using the max-log-APP algorithm (sometimes referred to as the max-log-MAP algorithm) [20]. To briefly explain the max-log-APP algorithm, assume that it is possible to efficiently find the maximum likelihood (ML) codeword,  $j$ , under the constraint that the transmitted bit at time  $k$  is a 1, as well as the ML codeword,  $j'$  under the constraint that the transmitted bit at time  $k$  is a -1.

Consider the difference between the summed metrics

$$\frac{1}{2} \sum_{m=1}^n d_{m,j} r_m - \frac{1}{2} \sum_{m=1}^n d_{m,j'} r_m = r_k + \sum_{\substack{m \neq k \\ d_{m,j} \neq d_{m,j'}}} d_{m,j} r_m \quad (13)$$

$$= llr_k^i + llr_k^e. \quad (14)$$

Note that the composite information given by the right-hand side of the equation only involves the bit positions for which the two codewords differ. The first term of the composite information is the intrinsic information (i.e., the information about the  $k$ th bit's value from the  $k$ th channel sample), while the second term provides an approximation to the extrinsic information (i.e., the information about the  $k$ th bit's value from the structure of the code and the other channel samples).

The steps followed by the basic SISO algorithm are summarized below.

- (1) Form an initial “reliability” vector by taking the absolute values of the elements of  $\mathbf{r}$ .
- (2) This is the first step in the iteration loop. Select the  $n-k$  linearly independent bit positions that are the least reliable in the sense that they correspond to the smallest elements in the reliability vector. We will refer to these as the “least reliable bits” (LRBs). Similarly, the remaining  $k$  bit positions are referred to as the “most reliable bits” (MRBs).
- (3) Using row reduction techniques, put  $\mathbf{H}$  into pseudo-systematic form such that the identity portion of the parity check matrix corresponds to the LRBs. Let  $\mathbf{d}'$  denote the vector of best decisions found to the given point in the algorithm, where “best” means the codeword with the best metric. For the first iteration,  $\mathbf{d}'$  will be the codeword obtained by performing hard decisions on  $\mathbf{y}$  for the MRBs, and then using the pseudo-systematic parity check matrix selecting the LRBs to satisfy the parity constraints.
- (4) Compute the extrinsic information for the MRBs. For the MRBs, a reasonable approximation to the

$k$ th extrinsic information is to take the metric difference between the best decision codeword, and the recoding solution with all of the MRBs remaining the same except for the  $k$ th one. Thus, the extrinsic information from (14) becomes

$$llr_k^{\hat{e}} = d_{m,j} * \sum_{\substack{m \neq k \\ d_{m,j} \neq d_{m,j'}}} d_{m,j} r_m, \quad (15)$$

where each of the  $m$  locations corresponds to an LRB and multiplication by  $d_{k,j}$  accounts for the possibility that the  $k$ th bit may not be  $a + 1$  in the vector of best decisions.

- (5) Compute the MRB metric differences by summing the extrinsic information and the intrinsic information, and compare with the signs of the MRBs in the current best codeword. If any of the signs differ, then a new best path has been found. If the signs differ in several locations, the new best path is the one for which the sign is changed for the bit corresponding to the largest magnitude of composite information (with sign differing from the current best decision). If there are no sign differences or if the maximum allowable number of iterations has been reached, go to Step (7). Otherwise, form the new best path by first changing the sign of the appropriate MRB and then computing the LRBs using recoding.
- (6) Form the new reliability vector by performing element-by-element multiplication between the new best decision vector and  $\mathbf{y}$ . Note that the reliability vector is now a signed vector with the “smallest” ones being negative (i.e., the intrinsic information and the “best” decision differ in sign). Go to Step (2).
- (7) Check if the best codeword is likely to be correct by comparing the states of the short shift registers. If they form a valid pair of states or if the maximum allowable number of iterations has been reached, the algorithm terminates. Otherwise, bias the algorithm away from the current solution by multiplying the current solution by a small scale factor, subtracting it from the received vector, and return to Step (2).

### 3. Simulation Results

This section presents the results from Monte-Carlo simulations for both the additive white Gaussian noise (AWGN) channel and a quasistatic Rayleigh fading channel. The performance plots present the codeword error rate (CER), where the codeword is from a  $(m, 72)$  code. In the simulations, we chose the size of  $m$  to be equal to 512, 1024, and 2048. (Note that there are two chip samples from the channel to produce one codeword sample.)

The simulation has perfect chip timing to the desired user and the simulation operates at 1 sample/chip. The CDMA 1x signal is spread at 1.2288 Mchips/s. When multiple users are present, the users are chip synchronous with the desired user. The desired user has a frequency offset of 3000 Hz from 0 Hz in the simulations. The algorithm is robust

against a frequency offset and the offset was included in the simulations to show the performance with an imperfect carrier frequency estimate. The data signal before spreading includes the pilot channel on the in-phase channel and with random data multiplied by a Walsh code (1 1 1 1 -1 -1 -1 -1) with a relative gain to the pilot of 0.8.

The detector used the biased version of the Vector SISO [18]. The maximum number of bias modifications was set to 20 and a scale factor of 0.5 was used. The codeword was modified and decoding continued, if after solving for the initial state of the shift registers a valid pair for the short shift registers did not occur. A maximum of 50 iterations were allowed for each decoding. In the simulations, a minimum of 10000 codewords were simulated. Once the minimum number of codewords were simulated, the simulation stopped when a minimum of 200 codeword errors were observed.

Equation (9) is used on a candidate codeword (truncated sequence) to solve for the 72 bits necessary to define the state of the shift registers. An error occurs when the decision bits from the decoder do not match the initial state of the shift register.

In Figure 4, the CER performance is presented for the 1 and 2 user cases on the AWGN channel. For the 2 user case, the interfering user has random frequency and phase offsets relative to the desired user and the power is set to  $-0.9$  dB (i.e., less than 1 dB difference) relative to the desired user. The codeword is marked in error if it does not agree with the state of the desired user’s signal. As seen in Figure 4, by increasing the block length of the code from 512 to 2048, improvements in performance of 1.1 dB and 2.3 dB for the 1 and 2 user cases, respectively, are obtained at a CER of  $10^{-2}$ . There is a degradation in performance for the case when there is an interfering user. For the two user case, the performance of the decoder is degraded from the 1 user case by 3 dB, 2.2 dB, and 1.9 dB for block sizes of 512, 1024, and 2048, respectively. The performance of a hard decision decoder that makes hard decisions on the sequence of bits and solves for the state of the spreading generator is provided for comparison. The performance is for the single user on the AWGN channel. The decoder has no coding gain when using 72 bits. The (512,72) code using the soft-decision decoder has a gain in performance of 8.2 dB at a PER of  $10^{-2}$ . The gain is a result of both the coding gain and using soft-decision information for the bits.

As CDMA systems usually work on an  $E_c/N_0$  below 0 dB, it should be noted that the results presented in Figure 4 are still of interest for this application. The application is for detecting users and monitoring spectrum usage. As our application is not involved in communications with the mobiles, there is no strict acquisition time requirement. For this application very poor CERs are tolerable because multiple attempts to recover the code phase are possible. If  $i$  attempts are made, the probability of successful acquisition is  $1 - \text{CER}^i$ . Consequently, values of CER much worse than 0.1 can still be useful in this application.

The performance with Rayleigh fading was considered as the Rayleigh fading channel is common in cellular communications. The channel model that was used was

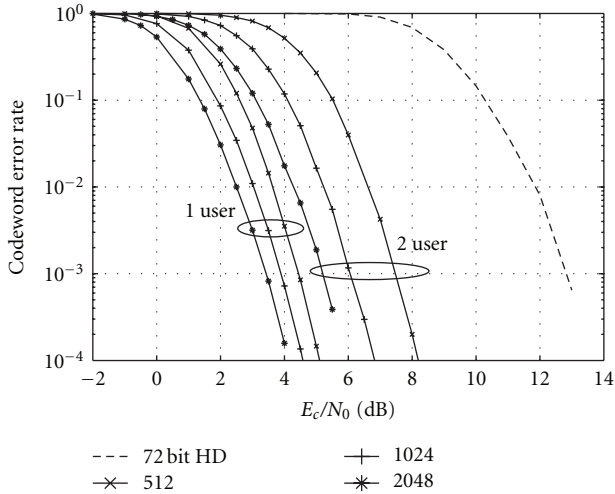


FIGURE 4: Codeword error rate performance results for codeword lengths of 512, 1024, and 2048 for 1 and 2 users on the AWGN channel. Results for the single user case when 72 hard decisions on the processed sequential pairs of channel samples are made and then used to solve for the spreading generator state are shown for comparison.

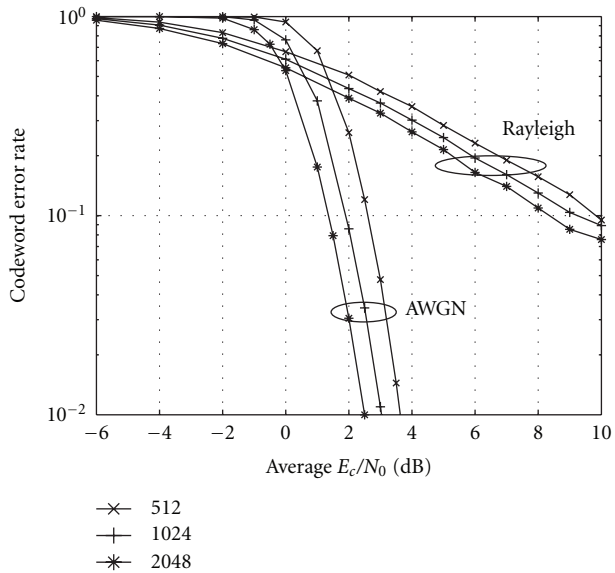


FIGURE 5: Codeword error rate performance results for codeword lengths of 512, 1024, and 2048 for a single user on a quasi-static Rayleigh fading channel. Single user performance on the AWGN channel is included for comparison.

quasistatic Rayleigh fading, where the user’s signal was scaled by a Rayleigh variate for each codeword. The Rayleigh variate was generated by scaling the square root of the sum of the squares of two independent Gaussian random variates with zero-mean and a variance of  $\sigma$ . A scale factor was chosen such that the mean of the Rayleigh variates was 1.0 and the resulting variance was  $4/\pi - 1$ .

For the Rayleigh fading channel simulations with two users, the detected codeword was considered in error only

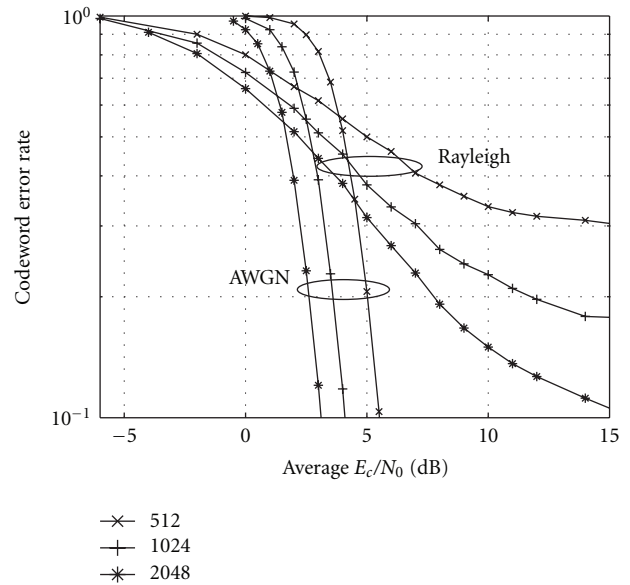


FIGURE 6: Codeword error rate performance results for codeword lengths of 512, 1024, and 2048 for two users on a quasi-static Rayleigh fading channel. The case with two users on the AWGN channel is included for comparison. In the AWGN case, the interfering user’s power is set to 0.9 dB less than the desired user.

if it did not match either of the users. In other words, the detector was successful if it detected either one of the users’ codewords. This measurement was chosen to correspond to typical results for a spectrum monitoring application as we are interested in detecting any user’s signal. We have assumed the signal with the highest power would be the most likely signal detected. In the AWGN case, the interfering user’s signal power was set slightly below that of the desired user’s signal. In the Rayleigh fading case, the users’ signals were set to have an equal power prior to fading, however, after fading either signal could have the highest receive power and considered the “desired” signal.

In Figure 5, the codeword error performance results of the algorithm are shown for the cases when the decoder uses block sizes of 512, 1024, and 2048. As seen in the figures, the detector has a better performance in quasi-static Rayleigh fading at lower  $E_c/N_0$  values. For example, if we consider a detection rate of 1 in 5 trials (e.g., a CER of 0.8) then there are gains of approximately 2 dB for the block sizes tested.

The case for two equal-powered signals is presented in Figure 6. In the low  $E_c/N_0$  region, this case has a higher gain over the two-user AWGN case when comparing the single user cases. The independent fading of the two users is the dominating factor in this result. If due to the fading the interfering signal power is reduced, the algorithm has a better chance to detect the nonfaded signal.

The method presented here is a practical method for blind acquisition of CDMA 1x signals. Speed tests were run on a Pentium IV processor with a clock speed of 3.8 GHz. The results are for 1 sample/chip operation with perfect chip timing and a signal present. The decoding parameters used during the speed tests were that the decoder was allowed 1

iteration, the maximum number of bias modifications was 10 and the extrinsic scale factor was 0.2. The average decoder speeds for block sizes of 512, 1024, and 2048 were 27.2, 9.8, and 3.6 decoding/second, respectively, for an  $E_c/N_0 = 0$  dB. The speeds increase with an increase in  $E_c/N_0$  as fewer average iterations are required.

#### 4. Conclusion

A technique was presented for recovering the code phase of the composite spreading sequence for a CDMA 1x signal transmitted from a handset, without the benefit of any *a priori* information from the system. Key steps in the technique are pair-wise processing to eliminate sensitivity to carrier frequency and phase offsets, followed by soft-in iterative decoding to solve for the state of an equivalent spreading code generator. A novel approach, based upon computing and checking the states of the two short-code generators, for determining if the resulting state is likely to be correct was presented.

The utility of the technique was demonstrated using computer simulation, in both AWGN and Rayleigh fading environments. It was shown that code phase recovery is possible even at very low signal-to-noise ratios and in the presence of significant co-channel interference.

The simulations assumed perfect chip timing synchronization. If chip timing is unknown, as is usually the case, the system can try each of four evenly spaced hypotheses (with a spacing of a quarter of a chip period). This method was found to be robust in low-SNR environments. While the results are not described in this paper, this technique has been used to successfully recover the code phase for off-air signals.

A suggested area of future work is an investigation on the effect of providing reliability information to the decoder that takes into account the non-Gaussian noise terms associated with  $y_n$  in (5). Another area of investigation is an investigation of various methods to reduce the effect of multiple-access interference on the detection algorithm.

#### References

- [1] M. Simon, J. Omura, R. Scholtz, and B. Levitt, *Spread Spectrum Communications*, vol. 3, Computer Science Press, Rockville, Md, USA, 1985.
- [2] G. Stuber, J. W. Mark, and I. F. Blake, "Sequence acquisition using bit estimation techniques," *Information Sciences*, vol. 32, no. 3, pp. 217–229, 1984.
- [3] J. L. Massey, G. Boscagli, and E. Vassallo, "Regenerative pseudo-noise (PN) ranging sequences for deep-space missions," *International Journal of Satellite Communications and Networking*, vol. 25, no. 3, pp. 284–304, 2007.
- [4] 3GPP2 C.S0002-C, Version 2, "Physical layer standard for cdma2000 spread spectrum systems rev C," July 2004.
- [5] R. B. Ward, "Acquisition of pseudo-noise signals by sequential estimation," *IEEE Transactions on Communications*, vol. COM-13, pp. 475–483, 1965.
- [6] H. Pearce and M. Ristenblatt, "The threshold decoding estimator for synchronization with binary linear recursive sequences," in *Proceedings of the IEEE International Conference on Communications (ICC '71)*, pp. 43–50, Montreal, Canada, June 1971.
- [7] C. Kilgus, "Pseudonoise code acquisition using majority logic decoding," *IEEE Transactions on Communications*, vol. 21, no. 6, pp. 772–774, 1973.
- [8] R. B. Ward and K. P. Yiu, "Acquisition of pseudonoise signals by recursion-aided sequential estimation," *IEEE Transactions on Communications*, vol. 25, no. 8, pp. 784–794, 1977.
- [9] P. Guinand and J. Lodge, "Iterative decoding of truncated simplex codes," in *Proceedings of the 2nd Biennial Symposium on Communications*, Kingston, Canada, June 2002.
- [10] L.-L. Yang and L. Hanzo, "Iterative soft sequential estimation assisted acquisition of m-sequences," *Electronics Letters*, vol. 38, no. 24, pp. 1550–1551, 2002.
- [11] L.-L. Yang and L. Hanzo, "Differential acquisition of m-sequences using recursive soft sequential estimation," *IEEE Transactions on Wireless Communications*, vol. 4, no. 1, pp. 128–136, 2005.
- [12] M. Zhu and K. M. Chugg, "Iterative message passing techniques for rapid code acquisition," in *Proceedings of the IEEE Military Communications Conference (MILCOM '03)*, pp. 434–439, Boston, Mass, USA, October 2003.
- [13] K. M. Chugg and M. Zhu, "A new approach to rapid PN code acquisition using iterative message passing techniques," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 5, pp. 884–897, 2005.
- [14] F. Principe, K. M. Chugg, and M. Luise, "Rapid acquisition of gold codes and related sequences using iterative message passing on redundant graphical models," in *Proceedings of the Military Communications Conference (MILCOM '06)*, October 2006.
- [15] 3G TS 25.213 V4.3.0 (2002–2006), "3rd generation partnership project; technical specification group radio access network; spreading and modulation (FDD) release 4," June 2002.
- [16] 3GPP2 C.S.0024-A, "High rate packet data air interface specification version 3.0," September 2006.
- [17] J. Lodge and R. Kerr, "Vector soft-in-soft-out decoding of linear block codes," in *Proceedings of the 22nd Biennial Symposium on Communications*, pp. 373–375, Kingston, Canada, May 2004.
- [18] R. Kerr and J. Lodge, "Near ML performance for linear block codes using an iterative vector SISO decoder," in *Proceedings of the 4th International Symposium on Turbo Codes and Related Topics*, Munich, Germany, April 2006.
- [19] A.-R. Abdul-Shakoor, R. Kerr, J. Lodge, and V. Szwarc, "An FPGA implementation of a soft-in soft-out decoder for block codes," in *Proceedings of the 24th Biennial Symposium on Communications (BSC '08)*, pp. 226–230, Kingston, Canada, June 2008.
- [20] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proceedings of the IEEE International Conference on Communications (ICC '95)*, pp. 1009–1013, June 1995.





**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

