Recursive All-Lag Reference-Code Correlator and its Efficient Implementation

Tony T. S. Ng, Kun-Wah Yip and Chin-Long Cheng

{tsng, kwyip, clcheng}@eee.hku.hk

Department of Electrical and Electronic Engineering
The University of Hong Kong
Pokfulam Road
Hong Kong

Fax: ++ 852 + 2559 8738   Tel.: ++ 852 + 2859 2710

Abstract — An all-lag reference-code correlator generates an all-lag correlation vector at a rate equal to the rate of incoming data samples. Direct implementation of an all-lag reference-code correlator requires N parallel correlators, and the resultant degree of complexity is of the order N^2, where N is the length of the reference code. This paper derives a recursive form for the all-lag reference-code correlator. It is shown that the proposed recursive all-lag reference-code correlator can be implemented with a complexity approximately equal to that of a single parallel correlator. That is, the degree of complexity of the proposed recursive all-lag reference-code correlator is of the order N. Thus, substantial reduction in the implementation complexity is achieved.

Index Terms — All-lag reference-code correlator, Recursive relationship, Low-complexity implementation, Spread spectrum, Parallel correlator.

I. INTRODUCTION

Correlators are widely used in applications involving signals that are formed by periodic repetition of reference codes with or without data modulation. The signals may be further corrupted by noise and various kinds of interference. Depending on applications, the reference code can be a pseudo-noise sequence, a sampled sinusoidal wave or, in fact, any arbitrary sequence of data. As a particular example, the reference code in a direct-sequence spread-spectrum (DSSS) system is a pseudo-noise spreading sequence. DSSS techniques [1], [2] have applications in many areas such as multiple-access data communications, secure communications, channel sounding, ranging and target identification using radars or sonars, and navigation using Global Positioning System (GPS). A correlator is required in a DSSS receiver to initially acquire the incoming DSSS signal. It is also used to perform other functions such as code tracking, symbol and carrier clock recovery, demodulation of information symbols embedded in a DSSS signal, and channel estimation.

In this paper, we are concerned with all-lag reference-code correlators. An all-lag reference-code correlator correlates a stream of data samples, $\{d_n\}$, with 0, 1, $\cdots$, $N-1$ lags of a length-N reference code sequence, $\{c_0, c_1, \cdots, c_{N-1}\}$, and thereby produces a stream of all-lag correlation vectors, $\{\mathbf{r}_n\}$, at a rate equal to the rate of incoming data samples. In this context, $\mathbf{r}_n = [r_{0,n}, r_{1,n}, \cdots, r_{N-1,n}]^T$ is given by

$$\mathbf{r}_n = \mathbf{C}\mathbf{d}_n \tag{1}$$

where

$$\mathbf{C} = \begin{bmatrix} c_0 & c_1 & c_2 & \cdots & c_{N-2} & c_{N-1} \\ c_{N-1} & c_0 & c_1 & \cdots & c_{N-3} & c_{N-2} \\ c_{N-2} & c_{N-1} & c_0 & \cdots & c_{N-4} & c_{N-3} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ c_2 & c_3 & c_4 & \cdots & c_0 & c_1 \\ c_1 & c_2 & c_3 & \cdots & c_{N-1} & c_0 \end{bmatrix} \tag{2}$$

is an $N \times N$ matrix, and

$$\mathbf{d}_n = \left[d_{n-(N-1)}, d_{n-(N-2)}, \cdots, d_{n-1}, d_n \right]^T \tag{3}$$

is a data vector containing N most-recent data samples. Note that the subindex m of $r_{m,n}$ refers to a lag of the reference code sequence while the second subindex n is time.

Interest in all-lag reference-code correlators arises because they provide more correlation information than serial correlators, parallel correlators and banks of serial correlators, the latter three types of correlators being commonly used in practical situations [3]-[11]. The

TABLE 1. IMPLEMENTATION COMPLEXITY OF VARIOUS CORRELATORS.

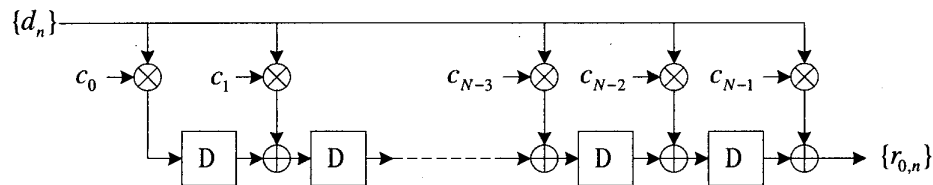|	Required number of:			
	multipliers	adders	storage units	negator
Parallel correlator	N	$N-1$	$N-1$	—
Recursive all-lag reference-code correlator that generates $\{\mathbf{r}_n\}$	N	$N+1$	$2N$	1

Fig. 1. A parallel correlator.

additional correlation information provided by an all-lag reference-code correlator, when processed, can be utilized for various purposes. Some applications that can be benefited from this additional information are listed as follows.

- Acquisition of a DSSS signal can be accomplished by locating the element of $\mathbf{r}_n$ having the largest magnitude. In this regard, acquisition based on all-lag correlation can be achieved in a duration of N sampling periods. On the other hand, using a parallel correlator to determine acquisition requires a longer duration of $2N - 1$ sampling periods.
- Code tracking and automatic frequency control in DSSS receivers can be made easier and can be enhanced by more frequent adjustments.
- In channel estimation, the impulse response of a multipath fading channel can be estimated directly and more conveniently from a knowledge of $\mathbf{r}_n$. More frequent update of channel estimates is also made possible, which improves RAKE receiver performance in response to rapidly varying channels.
- By processing the additional correlation information provided by an all-lag reference-code correlator, both acquisition and channel estimation can be made more robust to noise and interference than using a parallel correlator.

Direct implementation of an all-lag reference-code correlator is by means of N parallel correlators, where the mth parallel correlator, $m = 0, 1, \cdots, N - 1$, correlates a block of data samples given by $\mathbf{d}_n$ with the sequence taken from the mth row of $\mathbf{C}$, and produces a sequence of

correlation results $\{r_{m,n}\}$ at a rate of one result per sampling instant. A practical method to implement a parallel correlator is based on the systolic array architecture [3], [4]. Fig. 1 depicts such a parallel correlator for generating, as an example, $\{r_{0,n}\}$, and Table 1 lists the required numbers of multipliers, adders, etc., for implementing a parallel correlator. It is apparent that a parallel correlator comprises N multipliers, $N - 1$ adders and $N - 1$ storage elements, so that the degree of implementation complexity is of the order N. Since an all-lag reference-code correlator implemented using the direct approach comprises N parallel correlators, the degree of implementation complexity is of the order N^2. The resultant implementation complexity is especially significant when the reference-code length N is large.

Previous research effort has been devoted to minimize the implementation complexity of a parallel correlator for some special cases [12]-[14]. However, techniques directed to the reduction of implementation complexity for an all-lag reference-code correlator in general have not appeared in the previous literature. The objective of the present work is to develop a low-complexity architecture for an all-lag reference-code correlator, which is based on a recursive form derived in this paper. The all-lag reference-code correlator that is realized by this recursive relationship is referred to as a recursive all-lag reference-code correlator. In this paper, we show that it can be efficiently realized with a complexity approximately equal to that of a single parallel correlator. That is, the resultant implementation complexity is of the order N. This result enables system

designers to utilize all-lag correlation information while keeping the implementation cost low.

This paper is organized as follows. The recursive form for an all-lag reference-code correlator that generates $\{r_n\}$ is derived in Section II. Efficient implementation of recursive all-lag reference-code correlators are also discussed. Conclusions are drawn in Section III.

II. RECURSIVE ALL-LAG REFERENCE-CODE CORRELATOR

Define an $N \times N$ shift matrix

$$S = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}. \tag{4}$$

This shift matrix performs a linear transformation on an N-length column vector by cyclically shifting up the elements in the vector by one step. This transformation can be realized in practice by using an end-around shift register. Since cyclically shifting an N-length vector for N steps reproduces the original vector, it follows that

$$S^N = I \tag{5}$$

where I is an $N \times N$ identity matrix.

The desired recursive form for r_n is obtained by expressing r_n in terms of r_{n-1}. Let c_m be the mth column[1] of C where $m = 0, 1, \cdots, N - 1$. That is,

$$c_m = [c_m, c_{m-1}, \cdots, c_1, c_0, c_{N-1}, c_{N-2}, \cdots, c_{m+2}, c_{m+1}]^T. \tag{6}$$

It is easy to verify that

$$c_{N-1} = Sc_0$$
$$c_{m-1} = Sc_m, \qquad m = 1, 2, \cdots, N-1. \tag{7}$$

Since $C = [c_0\ c_1 \cdots c_{N-1}]$, it follows that (1) can be expressed as

$$r_n = \sum_{m=0}^{N-1} d_{n+m-(N-1)} c_m. \tag{8}$$

Setting $m' = m + 1$ in (8), we find that

$$r_n = d_n c_{N-1} + \sum_{m'=1}^{N-1} d_{n-1+m'-(N-1)} c_{m'-1}. \tag{9}$$

Applying (7) to this expression gives

---

[1] Throughout this paper, the left uppermost element of a matrix is assigned an index (0,0) rather than the usual index (1,1).

$$r_n = d_n c_{N-1} + S \sum_{m'=1}^{N-1} d_{n-1+m'-(N-1)} c_{m'}$$
$$+ d_{n-N}(Sc_0 - c_{N-1}). \tag{10}$$

Noting that $r_{n-1} = \sum_{m'=0}^{N-1} d_{n-1+m'-(N-1)} c_{m'}$ as seen from (8), we arrive at the desired recursive form

$$r_n = Sr_{n-1} + (d_n - d_{n-N})c_{N-1}. \tag{11}$$

Based on a knowledge of r_{n-1}, one can generate r_n by this recursive relationship. Note that Sr_{n-1} is an end-around rotation of r_{n-1}. As (11) is a recursive equation only, it remains to find the initial condition that makes (11) and (1) yield the same result. Repeated application of (11) for N times followed by applying (5) and (7) gives

$$r_n = r_{n-N} + C(d_n - d_{n-N}). \tag{12}$$

Without loss of generality, we assume that signal samples, d_n's, are only available for $n = 1, 2, 3, \cdots$ and that it is desired to generate $r_N, r_{N+1}, r_{N+2}, \cdots$. If we provide an initial condition that $r_0 = 0$ and $d_0 = d_{-1} = \cdots = d_{-(N-1)} = 0$, then (12) becomes identical to (1) for $n = N$. Based on a valid result of r_N, one can compute r_n, $n > N$, by using (11). Note that intermediate results $r_1, r_2, \cdots, r_{N-1}$ are not valid.

The number of arithmetic operations of the recursion for each iteration can easily be observed to be 1 subtraction, N additions, N multiplications and $2N$ shifts. Based on the recursive relationship of (11), the all-lag reference-code correlator that generates $\{r_n\}$ can be constructed as depicted in Fig. 2. It requires an N-length shift register to store the input signal samples, and output storage to store the N correlation results for the previous sampling instant, a negator, N multipliers and $N + 1$ two-input adders. Notice that prior to operation, the values stored in the shift register and in the output storage are initialized to zero. Table 1 summarizes the required numbers of components for implementing this all-lag reference-code correlator. Comparing corresponding numbers of components for realizing a parallel correlator as listed also in Table 1, one immediately finds that implementation complexity of an all-lag reference-code correlator that generates $\{r_n\}$ is approximately the same as that of a conventional parallel correlator. In particular, degree of complexity of both correlators are of the order N.

III. CONCLUSIONS

A recursive form for generating an all-lag correlation sequence $\{r_n\}$ has been derived. It has been shown that using this recursive form, an all-lag reference-code correlator can be implemented with a complexity
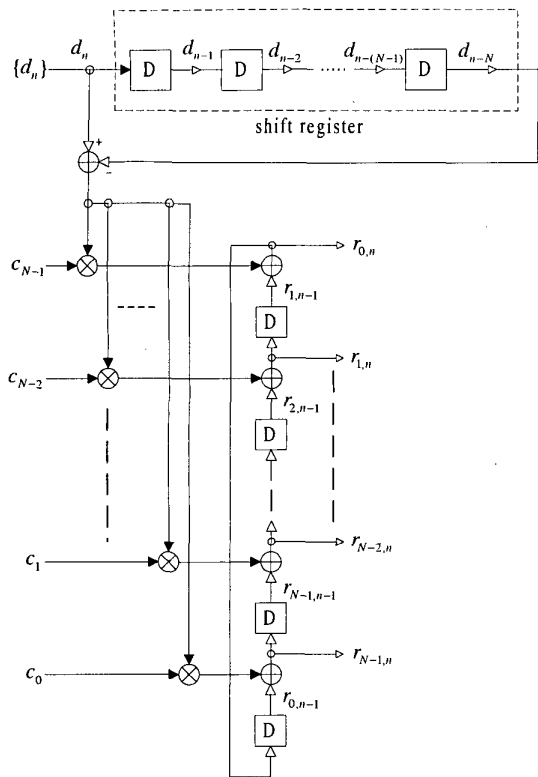
Fig. 2. A recursive all-lag reference-code correlator that generates a sequence of all-lag correlation vectors $\{\mathbf{r}_n\}$.

approximately the same as that of a conventional parallel correlator. The degree of implementation complexity for a recursive all-lag reference-code correlator is therefore reduced substantially from order N^2 to order N.

REFERENCES

[1] R. L. Peterson, R. E. Ziemer and D. E. Borth, *Introduction to Spread Spectrum Communications*, 1995, Prentice-Hall: New Jersey.

[2] M. K. Simon, J. K. Omura, R. A. Scholtz and B. K. Levitt, *Spread Spectrum Communications Handbook*, revised ed., 1994, McGraw-Hill: New York.

[3] H. T. Kung, "Why systolic architectures?" *Computer*, vol. 15, pp. 37-46, Jan. 1982.

[4] D. T. Magill and G. Edwards, "Digital matched filter ASIC," *Proc. IEEE MILCOM'90*, pp. 235-238, Sep. 30 - Oct. 3, 1990.

[5] R. S. Mowbray and P. M. Grant, "Simplified matched filter receiver designs for spread spectrum communications applications," *IEE Electronics and Communication Engineering Journal*, pp. 59-64, Apr. 1993.

[6] R. C. Dixon and J. S. Vanderpool, "Spread spectrum correlator," U. S. Pat. No. 5,022,047, Jun. 4, 1991.

[7] R. C. Dixon and J. S. Vanderpool, "Dual-threshold spread spectrum correlator," U. S. Pat. No. 5,719,900, Feb. 17, 1998.

[8] R. Price and P. E. Green, Jr., "A communication technique for multipath channels," *Proc. IRE*, vol. 46, pp. 555-570, Mar. 1958.

[9] M. Luise and R. Reggiannini, "Carrier recovery in all-digital modems for burst-mode transmission," *IEEE Trans. Commun.*, vol. 43, pp. 1169-1178, Feb./Mar./Apr. 1995.

[10] U. Fawer, "A coherent spread-spectrum diversity receiver with AFC for multipath fading channels," *IEEE Trans. Commun.*, vol. 42, pp. 1300-1311, Feb./Mar./Apr. 1994.

[11] A. Q. Hu, P. C. K. Kwok and T. S. Ng, "MPSK DS/CDMA carrier recovery and tracking based on correlation technique," *IEE Electron. Lett.*, vol. 35, pp. 201-203, Feb. 1999.

[12] J. Zhang and Q. Zhang, "An approach to realize the parallel real-time correlator based on parallel pipeline from addition network," *Proc. IEEE TENCON'93*, pp. 849-852, Oct. 21, 1993, Beijing, China.

[13] W. C. Lin, K. C. Liu and C. K. Wang, "Differential matched filter architecture for spread spectrum communication system," *IEE Electron. Lett.*, vol. 32, pp. 1539-1540, Aug. 1996.

[14] B. S. E. Tan and G. J. R. Povey, "Low complexity spread spectrum correlator," *IEE Electron. Lett.*, vol. 33, pp. 1204-1205, Jul. 1997.