A Recursive All-Lag Reference-Code Correlator for Generating Odd Correlations

Tung-Sang Ng, Kun-Wah Yip and Chin-Long Cheng

Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong,China
Fax. ++ 852 + 2559 8738. Tel.: ++ 852 + 2857 8406.
Email: {tsng, kwyip, clcheng}@eee.hku.hk

Abstract — An all-lag reference-code correlator generates an all-lag even- or odd-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. In a previous paper, a recursive form for generating all-lag even correlations was derived. In this paper, we derive the recursive form for generating all-lag odd correlations. It is shown that the corresponding 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.

I. Advantages of all-lag correlators and the need for efficient implementation

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 even-correlation vectors, $\{\mathbf{r}_n\}$, or a stream of all-lag odd-correlation vectors, $\{\bar{\mathbf{r}}_n\}$, at a rate equal to the rate of incoming data samples. In this context, $\mathbf{r}_n = [r_{0,n}, r_{1,n}, \cdots, r_{N-1,n}]^T$ and $\bar{\mathbf{r}}_n = [\bar{r}_{0,n}, \bar{r}_{1,n}, \cdots, \bar{r}_{N-1,n}]^T$ are given by

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

and

$$\bar{\mathbf{r}}_n = \bar{\mathbf{C}}\mathbf{d}_n, \tag{2}$$

respectively, where

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

and

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

are $N \times N$ matrices, and

$$\mathbf{d}_n = \left[d_{n-(N-1)}, d_{n-(N-2)}, \cdots, d_{n-1}, d_n\right]^T \tag{5}$$

is a data vector containing N most-recent data samples. The subindex m of $r_{m,n}$ and $\bar{r}_{m,n}$ refers to a lag of the reference code sequence while the second subindex n is time.

A. Advantages

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 [1]-[9]. We shall illustrate the advantages of all-lag correlators by considering the acquisition process of a direct-sequence spread-spectrum (DSSS) signal [10]. Consider first the case of using a parallel correlator that correlates a sequence of DSSS signal samples $\{d_n\}$ with a reference code $\{c_n\}$ and generates a sequence of correlation values, $\{u_n\}$, at a rate equal to the rate of incoming signal samples, where

$$u_n = c_0 d_{n-(N-1)} + c_1 d_{n-(N-2)} + c_2 d_{n-(N-3)} + \cdots + c_{N-2} d_{n-1} + c_{N-1} d_n \tag{6}$$

is the correlation result obtained at the nth sampling instant. The reference-code length N is normally selected such that it is equal to the length of the spreading sequence multiplied by the number of samples per chip. Before acquisition, the DSSS signal is not code-aligned with the receiver's copy of the reference code sequence. Since the reference-code length is N, we can code-align, or acquire, the incoming DSSS signal at the receiver by computing N

This work was supported by the Hong Kong Research Grants Council and by the University Research Committee of The University of Hong Kong, Hong Kong. The authors are with The University of Hong Kong. Fax: ++ 852 + 2559 8738. Tel.: ++ 852 + 2857 8406. (Email: {tsng, kwyip, clcheng}@eee.hku.hk)

correlation values corresponding to the correlation of the signal with 0, 1, $\cdots$, $N-1$ lags (or delays) of the reference code. The receiver is therefore required to compute

$$\begin{aligned} u_n &= c_0 d_{n-(N-1)} + c_1 d_{n-(N-2)} + \cdots + c_{N-1} d_n \\ u_{n+1} &= c_0 d_{n-(N-2)} + c_1 d_{n-(N-3)} + \cdots + c_{N-1} d_{n+1} \\ &\vdots \\ u_{n+N-2} &= c_0 d_{n-1} + c_1 d_n + \cdots + c_{N-1} d_{n+N-2} \\ u_{n+N-1} &= c_0 d_n + c_1 d_{n+1} + \cdots + c_{N-1} d_{n+N-1} \end{aligned} \tag{7}$$

and the acquisition circuit determines which one of these values has the largest magnitude. Acquisition is declared on the time position where the largest magnitude occurs.

In the absence of data modulation embedded in the DSSS signal, the signal is a periodic repetition of the reference code sequence. The intended information contained in signal samples d_{n+1}, d_{n+2}, $\cdots$, d_{n+N-1} is also contained in $d_{n-(N-1)}$, $d_{n-(N-2)}$, $\cdots$, d_{n-1}, respectively, so that (7) can be expressed as

$$\begin{aligned} u_n &= c_0 d_{n-(N-1)} + c_1 d_{n-(N-2)} + \cdots + c_{N-1} d_n \\ u_{n+1} &= c_0 d_{n-(N-2)} + c_1 d_{n-(N-3)} + \cdots + c_{N-1} d_{n-(N-1)} \\ &\vdots \\ u_{n+N-2} &= c_0 d_{n-1} + c_1 d_n + \cdots + c_{N-1} d_{n-2} \\ u_{n+N-1} &= c_0 d_n + c_1 d_{n-(N-1)} + \cdots + c_{N-1} d_{n-1} \end{aligned} \tag{8}$$

Thus, computation of u_n, u_{n+1}, $\cdots$, u_{n+N-1} is equivalent to computing $\mathbf{r}_n$ given by (1). Rapid acquisition of the incoming DSSS signal is achieved by locating the time position having the largest magnitude among $r_{m,n}$, $m = 0, 1, \cdots, N-1$. It is apparent that acquisition of a DSSS signal by using $\mathbf{r}_n$ can be achieved in a duration of N consecutive sampling periods while acquisition using a parallel correlator involves a larger data block of $2N-1$ samples.

In the presence of antipodal data modulation, that is, when the symbols are either +1 or –1, successive symbols may or may not have a transition in polarity. When successive data symbols contained in DSSS signal samples $d_{n-(N-1)}$, $d_{n-(N-2)}$, $\cdots$, d_{n+N-1} have the same sign, it is easy to show that computation of u_n, u_{n+1}, $\cdots$, u_{n+N-1} is equivalent to the computation of $\mathbf{r}_n$. Acquisition is declared at the time position having the largest magnitude among $r_{m,n}$, $m = 0, 1, \cdots, N-1$. When successive data symbols are opposite in sign, using only the information contained in $\mathbf{r}_n$ is not sufficient for acquisition unless data transition occurs at the zeroth-lag position, a condition that does not occur frequently. To achieve rapid acquisition, we make use of the information provided by both $\mathbf{r}_n$ and $\bar{\mathbf{r}}_n$. In case successive data symbols are opposite in sign, the correlation peak among $\bar{r}_{m,n}$, $m = 0, 1, \cdots, N-1$, is located where data transition occurs because of an intentional reversal of sign during correlation of the signal as seen from (2). Therefore, a data-modulated DSSS signal can be acquired by locating the time position having the largest magnitude among the elements of $\mathbf{r}_n$ and $\bar{\mathbf{r}}_n$. Note that acquisition can be accomplished when $\mathbf{r}_n$ and $\bar{\mathbf{r}}_n$ are available, that is, after N signal samples are obtained. On the other hand, acquisition using a parallel correlator requires a time of $2N-1$ sampling periods.

B. Implementation problems

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 $\mathbf{C}$ or $\bar{\mathbf{C}}$ according to whether even- or odd-correlation values are to be generated, and produces a sequence of correlation results $\{r_{m,n}\}$ or $\{\bar{r}_{m,n}\}$ at a rate of one result per sampling instant. A practical method to implement a parallel correlator is based on the systolic array architecture [1], [2]. Table 1 lists the required numbers of multipliers, adders, etc., for implementing a parallel correlator, wherein the results are taken from [11]. 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.

Recently, the authors [11] have derived a recursive form for generating all-lag even-correlation vectors. It has been shown that the resultant correlator 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. For reference, we indicate that the correlator requires N multipliers, $N+1$ adders, $2N$ storage elements and 1 negator (Table 1). This result enables system designers to utilize all-lag correlation information while keeping the implementation cost low. In this paper, we complement the work of [11] and derive the recursive form for an all-lag reference-code correlator that generates odd correlations. Implementation aspects are also discussed.

II. Recursive all-lag reference-code correlator for generating $\{\bar{r}_n\}$

We proceed to derive the recursive formula for computing $\bar{\mathbf{r}}_n$ based on the same steps as in deriving the one for $\mathbf{r}_n$ in [11]. Define an $N \times N$ shift matrix

$$\bar{\mathbf{S}} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \\ -1 & 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}, \tag{9}$$

which performs a linear transform on a length-N column vector by cyclically shifting up the elements by one step and reversing the sign of the resultant lowest element. This transform is realized in practice by an inverting end-around shift register. It is easy to show that

$$\bar{\mathbf{S}}^N = -\mathbf{I}. \tag{10}$$

Let $\bar{\mathbf{c}}_m$ denote the mth column[1] of $\bar{\mathbf{C}}$ for $m = 0, 1, \cdots, N-1$, namely,

$$\bar{\mathbf{c}}_m = [c_m, c_{m-1}, \cdots, c_1, c_0, -c_{N-1}, -c_{N-2}, \cdots, -c_{m+2}, -c_{m+1}]^T. \tag{11}$$

It can be easily shown that

$$\begin{aligned} \bar{\mathbf{c}}_{N-1} &= -\bar{\mathbf{S}}\bar{\mathbf{c}}_0 \\ \bar{\mathbf{c}}_{m-1} &= \bar{\mathbf{S}}\bar{\mathbf{c}}_m, \qquad m = 1,2,\cdots,N-1. \end{aligned} \tag{12}$$

Since $\bar{\mathbf{C}} = [\bar{\mathbf{c}}_0\ \bar{\mathbf{c}}_1 \cdots \bar{\mathbf{c}}_{N-1}]$, we can express (2) as

$$\bar{\mathbf{r}}_n = \sum_{m=0}^{N-1} d_{n+m-(N-1)}\bar{\mathbf{c}}_m, \tag{13}$$

so that

$$\bar{\mathbf{r}}_n = d_n\bar{\mathbf{c}}_{N-1} + \sum_{m'=1}^{N-1} d_{n-1+m'-(N-1)}\bar{\mathbf{c}}_{m'-1}. \tag{14}$$

Applying (12) to the last expression yields

$$\bar{\mathbf{r}}_n = d_n\bar{\mathbf{c}}_{N-1} + \bar{\mathbf{S}}\sum_{m'=1}^{N-1} d_{n-1+m'-(N-1)}\bar{\mathbf{c}}_{m'} + d_{n-N}(\bar{\mathbf{S}}\bar{\mathbf{c}}_0 + \bar{\mathbf{c}}_{N-1}). \tag{15}$$

It follows that the recursive relationship is given by

$$\bar{\mathbf{r}}_n = \bar{\mathbf{S}}\bar{\mathbf{r}}_{n-1} + (d_n + d_{n-N})\bar{\mathbf{c}}_{N-1}, \tag{16}$$

which enables generation of $\bar{\mathbf{r}}_n$ based on $\bar{\mathbf{r}}_{n-1}$. Notice that $\bar{\mathbf{S}}\bar{\mathbf{r}}_{n-1}$ is an inverting end-around rotation of $\bar{\mathbf{r}}_{n-1}$. The initial condition is derived as follows. Repeated application of (16) for N times followed by an application of (10) and (12) yields

$$\bar{\mathbf{r}}_n = -\bar{\mathbf{r}}_{n-N} + \bar{\mathbf{C}}(\mathbf{d}_n + \mathbf{d}_{n-N}). \tag{17}$$

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

Again, assume that signal samples, d_n's, are only available for $n = 1, 2, 3, \cdots$ and that we want to generate $\bar{\mathbf{r}}_N$, $\bar{\mathbf{r}}_{N+1}$, $\bar{\mathbf{r}}_{N+2}$, $\cdots$. It is easy identify the desired initial condition to be $\bar{\mathbf{r}}_0 = \mathbf{0}$ and $d_0 = d_{-1} = \cdots = d_{-(N-1)} = 0$, so that (17) and (2) becomes identical for $n = N$. The recursive formula (16) can be used thereafter to generate $\bar{\mathbf{r}}_n$, $n > N$.

Fig. 1 shows a recursive all-lag reference-code correlator that generates $\{\bar{\mathbf{r}}_n\}$ and that is constructed according to (16). It is apparent that implementation of this correlator requires a length-N shift register for storing the input signal sequence, an output storage for storing the N correlation results, a negator, N multipliers and $N + 1$ two-input adders. Values in the shift register and the output storage are reset to zero prior to operation. Table 1 lists the required numbers of components for realizing this correlator, alongside with those results for other correlators. Results of Table 1 indicate that the implementation complexity of all three correlators is approximately the same and is of the order N.

III. Conclusions

A recursive form for generating all-lag odd-correlation sequences $\{\bar{\mathbf{r}}_n\}$ has been derived. It has been shown that using this recursive form, all-lag reference-code correlators can be implemented with a complexity approximately the same as that of a conventional parallel correlator. The degree of implementation complexity for the recursive all-lag reference-code correlator has therefore been reduced substantially from order N^2 to order N.

References

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

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

[3] 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.

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

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

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

[7] 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.

[8] 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.

[9] 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,

[10] J. Li and S. Tantaratana, "Optimal and suboptimal coherent acquisition schemes for PN sequences with data modulation," *IEEE Trans. Commun.*, vol. 43, pp. 554-564, Feb./Mar./Apr. 1995.

[11] T.-S. Ng, K.-W. Yip and C.-L. Cheng, "Recursive All-Lag Reference-Code Correlator and its Efficient Implementation," *Proc. IEEE ISCAS 2000*, pp. IV473-IV476, May 28-31, 2000, Geneva, Switzerland.

Table 1. Implementation complexity of various correlators. Results for the parallel correlator and the correlator that generates $\{\mathbf{r}_n\}$ are taken from [11].

	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
Recursive all-lag reference-code correlator that generates $\{\bar{\mathbf{r}}_n\}$	N	$N+1$	$2N$	1

Fig. 1. A recursive all-lag reference-code correlator that generates a sequence of all-lag odd-correlation vectors $\{\bar{\mathbf{r}}_n\}$.