A Fast Computation of Complex Convolution Using A Hybrid Transform[1]

I. S. Reed
University of Southern California
Department of Electrical Engineering

T. K. Truong
TDA Engineering Office

In this article, it is shown that the cyclic convolution of complex values can be performed by a hybrid transform. This transform is a combination of a Winograd transform and a fast complex integer transform developed previously by the authors. This new hybrid algorithm requires fewer multiplications than any previously known algorithm.

I. Introduction

Several authors (Refs. 1–9) have shown that transforms over finite fields or rings can be used to compute circular convolutions without round-off error. Recently, Agarwal and Cooley (Ref. 10) used the techniques of Winograd (Refs. 11, 12) to compute cyclic convolutions. These new algorithms for convolutions of a few thousand points require substantially fewer multiplications than the conventional FFT algorithm (Ref. 13).

Previously the authors (Ref. 5) defined a class of Fourier-like transforms over the complex integers modulo q. This was a transform over the Galors field $GF(q^2)$, where $q = 2^p - 1$ is a Mersenne prime for $p = 2, 3, 5, 7, 13, 17, 19, 31, 61, \ldots$ Recently these complex integer transforms were specialized to a transform length of d points, where $d | 8p$ (Ref. 8). The advantage of the latter transform over others is that it can be accomplished completely by circular shifts, i.e., no multiplications are needed (Ref. 8).

In this article, it is shown that Winograd's algorithm (Ref. 11) can be combined with the above-mentioned complex integer transform over $GF(q^2)$ to yield a new algorithm for computing the discrete cyclic convolution of complex number points. By this means a fast method for accurately computing the cyclic convolution of a sequence of complex numbers for long convolution lengths can be obtained. This hybrid algorithm is comparable in speed to that given by Agarwal and Cooley (Ref. 10) and is implemented readily on a digital computer. The dynamic range requirements for this hybrid algorithm are presented here in detail.

II. Cyclic Convolution

The following algorithm for the cyclic convolution of two sequences is based on ideas given by Winograd (Ref. 11). Let the field of rationals be R. Also let $X(u) = x_0 + x_1 u + x_2 u^2 +$

[1] This work was supported in part by the U.S. Air Force Office of Scientific Research under Grant AFOSR-75-2798.

$\cdots + x_n u^{n-1}$, $Y(u) = y_0 + y_1 u + y_2 u^2 + \cdots + y_n u^{n-1}$ be two polynomials over R. The product $T(u) = X(u) \cdot Y(u)$ can be computed by

$$T(u) = X(u) \cdot Y(u) \bmod \prod_{i=0}^{2n-2} (u - \alpha_i) \qquad (1)$$

where $\alpha_i \in R$. It is shown in Ref. 11 that a minimum of $2n - 1$ multiplications are needed to compute (1).

It is readily shown that the cyclic convolution of $X(u)$ and $Y(u)$ is the set of coefficients of the polynominal

$$T'(u) = X(u) \cdot Y(u) \bmod (u^n - 1)$$

Let the polynomial $u^n - 1$ be factored into irreducible relatively prime factors, i.e.,

$$u^n - 1 = \prod_{i=1}^{k} g_i(u)$$

where $(g_i(u), g_j(u)) = 1$ for $i \neq j$. Then $T'(u) \bmod g_i(u)$ for $i = 1, 2, \ldots, k$ can be computed using Eq. (1). Finally, the Chinese remainder theorem is used to evaluate $T'(u)$ from these residues. The above summarizes Winograd's method for performing a cyclic convolution.

The following theorem is due to Winograd (Ref. 11).

Theorem 1: Let a and b be relatively prime positive integers and A be the cyclic $ab \times ab$ matrix, given by

$$A(x, y) = f(x + y \bmod a \cdot b), \qquad 0 \leqslant x, y < ab$$

If π is a permutation of the set of integers $\{0, 1, \ldots, ab - 1\}$, let

$$B(x, y) = A(\pi(x), \pi(y))$$

Then there exists a permutation π such that, if B is partitioned into $b \times b$ submatrices, each submatrix is cyclic and the submatrices form an $a \times a$ cyclic matrix.

It was shown previously (Refs. 10, 12) that the number of multiplications needed to perform a circular convolution of 3, 5, 7, and 9 points of complex numbers is 4, 10, 19, and 22 multiplications, respectively. In order to compute the cyclic convolution of two longer sequences of complex integers, a d-point transform over $GF(q^2)$, where $q = 2^p - 1$ and $d | 8p$, will be utilized here. Since the latter transform can be evaluated without multiplications (Ref. 8) , it can be used with considerable advantage to compute a cyclic convolution of two d-point complex number sequences. The number of complex integer multiplications required to perform this circular convolution over $GF(q^2)$ is precisely d, the number of multiplications needed to multiply together the transforms of the two sequences.

For the moment, let d, the transform length, be an arbitary integer. Next let $d = p_1 \cdot p_2 \ldots p_r$ be the factorization of d into prime integers. If one lets $a_1 = p_1 \cdot p_2 \ldots p_{r-1}$ and $b_1 = p_r$, then by Theorem 1, a $d \times d$ cyclic matrix can be partitioned into $b_1^2 = p_r^2$ matrices of size $a_1 \times a_1$. Next let $a_1 = a_2 \times b_2$, where $a_2 = p_1 \ldots p_{r-2}$ and $b_2 = p_{r-1}$. If a_2 is not a prime, then each $a_1 \times a_2$ cyclic matrix can be partitioned into b_2^2 matrices of size $a_2 \times a_2$. In general, $a_i = a_{i+1} \times b_{i+1}$, where b_{i+1} is a prime. If $a_{i+1} \neq 1$, then each $a_i \times a_i$ cyclic matrix can be partitioned into b_{i+1}^2 matrices of size $a_{i+1} \times a_{i+1}$. Otherwise, the procedure terminates. If the number of multiplications used to compute the cyclic convolution of p_i points is m_i for $i = 1, 2, \ldots, r$, then Winograd has shown (Ref. 10) that the number of multiplications needed to compute a d-point cyclic convolution is equal to $N = m_1 \cdot m_2 \ldots m_r$.

It is necessary to choose only certain values of d as the transform length in order to combine the Winograd transform with the fast complex integer transform over $GF(q^2)$, where $q = 2^p - 1$ is a Mersenne prime. For this purpose, let the number d have the form

$$d = a \cdot 2^m \cdot p \qquad (2)$$

where $m = 0, 1, 2, 3$ and $a = 3, 5, 7$, or 9. For most practical applications, it suffices in (2) to let $p = 31$ or 61.

If d, the transform length of the cyclic convolution, is given by (2), then by Theorem 1, there exists a permutation of rows and columns so that the cyclic $d \times d$ matrix can be partitioned into blocks of $(2^m \cdot p) \times (2^m \cdot p)$ cyclic matrices in such a manner that the blocks form an $a \times a$ cyclic matrix. Now the cyclic convolutions of $a = 3, 5, 7$, or 9 complex number points can be accomplished by Winograd's algorithm. Since $2^m \cdot p | q^2 - 1$ for $m = 0, 1, 2, 3$, a transform of length $2^m \cdot p$ over $GF(q^2)$ can be found and used to compute the cyclic convolution of the $2^m \cdot p$ complex number points. The number of multiplications needed to perform this convolution is $2^m \cdot p$. Using this and the number of multiplications needed for Winograd's algorithm, the total number of multiplications needed to perform a convolution of d complex number points can be computed. The results of this computation are shown in Table 1. The present algorithm, and Agarwal and Cooley's algorithm are compared in this table by giving the total number of complex number multiplications required to perform the different algorithms.

It was shown above that Winograd's algorithm can be combined with a transform over $GF(q^2)$ to yield a new rather fast hybrid algorithm for computing the cyclic convolution of complex values. In this algorithm it was necessary to compute the cyclic convolution of $2^m \cdot p$ complex number points for $m = 0, 1, 2,$ or 3. This cyclic convolution of two d-point sequences of complex number points is

$$c_k = \sum_{n=0}^{d-1} e_n f_{(k-n)} \tag{3}$$

where $d | 8 \cdot p$ and $(k - n)$ denotes the residue of $k - n$ mod d. To compute this convolution the components of the truncated complex numbers e_n and f_n must be converted first to integers a_n and b_n with dynamic ranges, say, A and B, respectively. Previously (Ref. 5), a sufficient dynamic range constraint for A and B was shown to be

$$A \leqslant \frac{q-1}{4Bd} \tag{4}$$

If $A = B$, (4) reduces to

$$A \leqslant \left[\sqrt{\frac{q-1}{4d}}\right] \tag{5}$$

where $[x]$ denotes the greatest integer less than x.

If the circular convolution of a_n and b_n is denoted by c'_k for $k = 0, 1, 2, \ldots, d - 1$, then using the procedure described in the example of Ref. 5, c'_k can be obtained by using fast transforms over $GF(q^2)$. In (3), c_k can be obtained by scaling back c'_k into the scale of the original complex numbers for $k = 0, 1, 2, \ldots d - 1$. Evidently, the only error made in this computation of the c'_ks is the truncation error.

The dynamic range constraint A of the input sequence given in (5) is generally very pessimistic. By an argument similar to that used for integer convolutions (Ref. 14), one can lessen the severity for this dynamic range constraint and still maintain c_k in the interval $\pm(q - 1)/2$ with a small probability of overflowing. This assertion is justified in the Appendix.

To illustrate this new hybrid algorithm, consider the following example.

Example: Let $d = 6$. Next suppose that the input function defined by

$$a_n = 1 + \hat{i}0 \qquad n = 0, 1$$

$$= 0 \qquad 2 \leqslant n \leqslant 5$$

is convolved with itself. This convolution is

$$c_k = \sum_{n=0}^{5} a_n b_{(k-n)}$$

where (x) denotes the residue $k - n$ of modulo 6. This convolution can be written in matrix form as

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{pmatrix} = \begin{pmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 \\ a_1 & a_2 & a_3 & a_4 & a_5 & a_0 \\ a_2 & a_3 & a_4 & a_5 & a_0 & a_1 \\ a_3 & a_4 & a_5 & a_0 & a_1 & a_2 \\ a_4 & a_5 & a_0 & a_1 & a_2 & a_3 \\ a_5 & a_0 & a_1 & a_2 & a_3 & a_4 \end{pmatrix} \begin{pmatrix} b_0 \\ b_5 \\ b_4 \\ b_3 \\ b_2 \\ b_1 \end{pmatrix}$$

By Theorem 1, there exists a permutation π of rows and columns so that the above cyclic matrix can be partitioned into a 2 X 2 block matrix of 3 X 3 cyclic blocks as follows:

$$\begin{pmatrix} c_0 \\ c_4 \\ c_2 \\ c_3 \\ c_1 \\ c_5 \end{pmatrix} = \begin{pmatrix} a_0 & a_4 & a_2 & a_3 & a_1 & a_5 \\ a_4 & a_2 & a_0 & a_1 & a_5 & a_3 \\ a_2 & a_0 & a_4 & a_5 & a_3 & a_1 \\ a_3 & a_1 & a_5 & a_0 & a_4 & a_2 \\ a_1 & a_5 & a_3 & a_4 & a_2 & a_0 \\ a_5 & a_3 & a_1 & a_2 & a_0 & a_4 \end{pmatrix} \begin{pmatrix} b_0 \\ b_2 \\ b_4 \\ b_3 \\ b_5 \\ b_1 \end{pmatrix}$$

This matrix equation has the form

$$\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = \begin{pmatrix} A & B \\ B & A \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}$$

where

$$Y_1 = \begin{pmatrix} c_0 \\ c_4 \\ c_2 \end{pmatrix}, Y_2 = \begin{pmatrix} c_3 \\ c_1 \\ c_5 \end{pmatrix}, X_1 = \begin{pmatrix} b_0 \\ b_2 \\ b_4 \end{pmatrix}, X_2 = \begin{pmatrix} b_3 \\ b_5 \\ b_1 \end{pmatrix},$$

$$A = \begin{pmatrix} a_0 \, a_4 \, a_2 \\ a_4 \, a_2 \, a_0 \\ a_2 \, a_0 \, a_4 \end{pmatrix}, \qquad B = \begin{pmatrix} a_3 \, a_1 \, a_5 \\ a_1 \, a_5 \, a_3 \\ a_5 \, a_3 \, a_1 \end{pmatrix}$$

Thus

$$\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = 2^{-1} \begin{pmatrix} (A+B)(X_1+X_2) + (A-B)(X_1-X_2) \\ (A+B)(X_1+X_2) - (A-B)(X_1-X_2) \end{pmatrix}$$

$$= 2^{-1} \begin{pmatrix} D+E \\ D-E \end{pmatrix} \qquad (6)$$

where $D = (A+B)(X_1+X_2)$, $E = (A-B)(X_1-X_2)$. Now

$$D = \begin{pmatrix} a_0+a_3, a_4+a_1, a_2+a_5 \\ a_4+a_1, a_2+a_5, a_0+a_3 \\ a_2+a_5, a_0+a_3, a_4+a_1 \end{pmatrix} \begin{pmatrix} b_0+b_3 \\ b_2+b_5 \\ b_4+b_1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \qquad (7)$$

Let $x_0 = 1, x_1 = 0, x_2 = 1$ and $y_0 = 0, y_1 = 1, y_2 = 1$. Then the matrix equation defined in (7) can be obtained by computing the convolution of the two sequences a_n and b_n. To do this, use a transform over $GF(q^2)$. In order to avoid overflow, one needs to choose $q = 7$ so that the integer components of a_n, b_n lie in the interval ±1.

Since 2 is a 3rd root of unity, the transform over $GF(7^2)$ of x_n is

$$X_k = \sum_{n=0}^{3-1} x_n 2^{nk} = 2^0 + 2^{2k} \qquad \text{for } k = 0, 1, 2$$

Thus, $X_0 = 2, X_1 = 5, X_2 = 3$.

Similarly, the transform of sequence y_n is

$$Y_k = \sum_{n=0}^{3-1} y_n 2^{nk} = 2^k + 2^{2k} \qquad \text{for } k = 0, 1, 2$$

That is, $Y_0 = 2$, $Y_1 = 6$, $Y_2 = 6$. But $D_k = X_k, Y_k$, i.e., $D_0 = 4$, $D_1 = 2, D_2 = 4$. These are the only complex integer multiplications needed to perform this transform. The inverse transform of D_k is

$$d_n = 3^{-1} \sum_{k=0}^{3-1} D_k \, 2^{-nk}$$

$$= 5(4 + 2 \cdot 2^{-k} + 4 \cdot 2^{-2k}) \bmod 7 \qquad \text{for } k = 0, 1, 2$$

since $3^{-1} \equiv 5 \bmod 7$. Thus finally, $d_0 = 1, d_1 = 2, d_3 = 1$.

In a similar fashion, matrix E, given in (6), can also be obtained as $e_0 = 1$, $e_1 = -2$, $e_2 = 1$. Thus, by (6), one obtains finally $c_0 = 1, c_1 = 2, c_2 = 1, c_3 = 0, c_4 = 0$ and $c_5 = 0$.

Acknowledgement

The authors wish to thank Dr. N. A. Renzetti, Manager of Tracking and Data Acquisition Engineering, and the members of the Advanced Engineering Group in that organization at the Jet Propulsion Laboratory for their early support, suggestions, and encouragement of the research which led to this article.

References

1. Pollar, J. M., "The Fast Fourier Transform in a Finite Field," *Math. Comput.*, 1971, 25, pp. 365-374.
2. Schonhage, A., and Strassen, V., "Schnelle Multiplikation Grosser Zahlen," *Computing*, 1971, 7, pp. 281-292.
3. Rader, C. M., "Discrete Convolution Via Mersenne Transforms," *IEEE Trans. Comp.,* 1972, C-21, pp. 1269-1273.
4. Agarwal, R. C., and Burrus, C. S., "Number Theoretic Transforms to Implement Fast Digital Convolution," *Proc. IEEE,* 1975, 63, pp. 550-560.
5. Reed, I. S., and Truong, T. K., "The Use of Finite Fields to Compute Convolution," *IEEE Trans.*, 1975, IT-21, pp. 208-213.
6. Reed, I. S., and Truong, T. K., "Complex Integer Convolution Over a Direct Sum of Galois Fields," *IEEE Trans.*, 1975, IT-21, pp. 657-661.
7. Vegh, E., and Leibowitz, L. M., "Fast Complex Convolution in Finite Rings," *IEEE Trans.*, 1976, ASSP-24, pp. 343-344.
8. Reed, I. S., Truong, T. K., and Liu, K. Y., "A New Fast Algorithm for Computing Complex Number-Theoretic Transforms," *Electron. Lett.*, 1977, pp. 278-280.
9. Reed, I. S., and Truong, T. K., "Fast Mersenne-Prime Transforms for Digital Filtering," to be published in *Proceedings IEE.*
10. Agarwal, R. C., and Cooly, J. W., "New Algorithm for Digital Convolution," *IEEE Trans. Acoust. Speech Signal Processing*, Vol. ASSP-25, pp. 392-410, Oct. 1977.
11. Winograd, S., "On Computing the Discrete Fourier Transform," *Proc. Nat. Acad. Sci. USA*, 1976, 73, pp. 1005-1006.
12. Winograd, S., *On Computing the Discrete Fourier Transform*, Research Report, Math. Science Dept., IBM Thomas J. Watson Research Center, Yorktown Heights, New York.
13. Cooley, J. W., and Tukey, J. W., "An Algorithm for the Machine Calculation of Complex Fourier Series," *Math. Comput.*, Vol. 19, pp. 297-301, April 1965.
14. Reed, I. S., Kwoh, Y. S., Truong, T. K., and Hall, E. L., "X-Ray Reconstruction by Finite Transforms," *IEEE Transactions on Nuclear Science*, Vol. NS-24, No. 1, February 1977.
15. Papoulis, A., *Probability Random Variable Stochastic Processes*, McGraw-Hill Book Co., New York, 1965.

Table 1. Complexity of New Algorithm for the Convolution of Complex Number Points

d	Factors	No. Complex Number Multiplications	No. Complex Number Multiplications of Agarwal and Cooley's Algorithm
120			560
124	4 × 31	124	
210			1520
244	4 × 61	244	
248	8 × 31	248	
420			3800
488	8 × 61	488	
744	3 × 248	992	
840			10640
1260			20900
1464	3 × 488	1952	
2520			58520
3720	3 × 5 × 248	9920	
7320	3 × 5 × 488	19520	

Appendix A

A Probabilistic Dynamic Range Constraint for the Transform Over $GF(q^2)$

Let $a_n = \alpha_n + \hat{i}\beta_n$ and $b_n = x_n + \hat{i}y_n$. Then the cyclic convolution of a_n and b_n is given by

$$c_k' = \sum_{n=0}^{d-1} (\alpha_n + \hat{i}\beta_n)(x_n + \hat{i}y_n)$$

$$= \sum_{n=0}^{d-1} \mu_n + \hat{i} \sum_{n=0}^{d-1} \nu_n \tag{A-1}$$

where $\mu_n = d_n x_n - \beta_n y_n$, $\nu_n = \alpha_n y_n + \beta_n x_n$. In many applications, the sequences α_n, β_n, x_n, and y_n can be regarded as mutually independent. With this assumption, consider the sum

$$S_\mu = \sum_{n=0}^{d-1} \mu_n$$

in (A-1). The expected value of μ_n is given by

$$E(\mu_n) = E(\alpha_n)\,E(x_n) - E(\beta_n)\,E(y_n)$$

where E denotes the expected value operator. With no loss of generality, the means of α_n, β_n, x_n, and y_n can be assumed to be zero. With these assumptions, $E(\mu_n) = 0$, and the variance σ_μ^2 of μ_n is given by

$$\sigma_\mu^2 = E(\mu_n^2) = E(\alpha_n^2)\,E(x_n^2) + E(\beta_n^2)\,E(y_n^2) \tag{A-2}$$

Finally, assume that α_n and β_n are uniformly distributed over the dynamic range A, and that x_n and y_n are uniformly distributed over the dynamic range B.

$$E(\alpha_n^2) = E(\beta_n^2) = \frac{A^2}{12}$$

and

$$E(x_n^2) = E(y_n^2) = \frac{B^2}{12}$$

Substituting these values in A-2,

$$\sigma_\mu^2 = 2 \cdot \frac{A^2B^2}{12^2} \tag{A-3}$$

Now by the central limit theorem (Ref. 15), the probability of exceeding a threshold λ is

$$P\left\{\frac{S_\mu}{\sqrt{d}\sigma_\mu} > \lambda\right\} = 2(1 - \varphi(\lambda))$$

where $\varphi(\lambda)$ is the standard normal distribution. This equation can be written as

$$P\left\{|S_\mu| > \lambda\sqrt{d}\sigma_\mu\right\} = 2(1 - \varphi(\lambda)) \tag{A-4}$$

To keep S_μ from overflowing, one needs the inequality

$$|S_\mu| \leqslant \frac{q-1}{2}$$

Hence if one sets

$$\lambda\sqrt{d}\sigma_\mu = \frac{q-1}{2} \tag{A-5}$$

then (A-4) is the probability of overflow.

For example, if $\lambda = 3$, then the probability of overflowing is

$$P\left\{ |S_\mu| > \lambda\sqrt{d\mu} \right\} = 2(1 - \varphi(3)) = 0.0026$$

which is very small. Substituting (A-3) into (A-5) yields

$$A \cdot B = \frac{6(q - 1)}{\sqrt{2d}\lambda} \qquad \text{(A-6)}$$

Equation (A-6) is the required relation among the parameters A, B, q, d, and λ. Similarly, one obtains the same result defined in (A-6) for the sum

$$S_\nu = \sum_{n=0}^{d-1} \nu_n \ .$$

Let 2^{k_1} and 2^{k_2} be binary scale factors for a_n and b_n, respectively. Then since

$$-2^{k_1} \leqslant \alpha_n, \beta_n \leqslant 2^{k_1}$$

$$-2^{k_2} \leqslant x_n, y_n \leqslant 2^{k_2}$$

the dynamic ranges A and B are 2^{k_1+1} and 2^{k_2+1}, respectively. For most applications the two Mersenne prime $2^{31} - 1$ and $2^{61} - 1$ will provide enough bit accuracy and dynamic range for computing two $2^m \cdot p$-point sequences of complex numbers. To illustrate this, if d is chosen to be $d = 2^8$, $\lambda = 3$, $q = 2^{31} - 1$ and $k_1 = k_2$, then by (A-6)

$$A = 2^{k_1+1} = \sqrt{\frac{6(q - 1)}{\sqrt{2d}\lambda}} \cong 2^{14}$$

Thus one needs approximately $k_1 = k_2 = 13$ bits to satisfy (A-4) with an overflow probability equal to 0.0026. This is a considerably better bound than one obtains using formula (5) for A. In fact, the dynamic range constraint (5) yields $A \cong 2^{k_1} = 2^{10}$.