CALCULATING LINEAR FROM CIRCULAR CONVOLUTIONS/CORRELATIONS

S Boussakta (1), A.G.J. Holt (2).

Electrical and Electronic Engineering Department, University of Newcastle Upon Tyne, Newcastle Upon Tyne, NE1 7RU.

ABSTRACT

The Fermat number transform (FNT) is one of the most useful number theoretic transforms. This paper investigates its application to the calculation of convolutions and correlations with emphasis on the transform length constraints. A technique is introduced to increase the sequence length that may be convolved through an FNT of specified size while preserving other advantages.

1. INTRODUCTION

If an input data sequence is denoted by x(n), the time domain response of filtering this sequence with an impulse response h(n) is the linear convolution $y_L(n)$:

$$y_L(n) - \sum_{m-0}^{N-1} x(m)h(n-m) \qquad (1)$$

Which is usually abbreviated as:

$$y_L(n) - x(n)*h(n) \qquad (2)$$

Similarly, correlating two sequences x(n) and h(n) yields

$$y(n)- \sum_{m-0}^{N-1} x(m)h(n+m) \qquad (3)$$

It can be seen that, in both the convolution and correlation, the calculation of an output point depends on many input samples. The time domain or direct calculation is computationally expensive.

An alternative technique for calculating convolutions and correlations uses orthogonal transforms having the cyclic convolution property (CCP), where the multiplication of the transforms of two N-point sequences corresponds to the transform of their cyclic convolution [1,2].

The use of the discrete transforms to compute convolution and correlation is aimed at the reduction in the number of arithmetic operations needed. Using the fast Fourier transform (FFT) to rapidly calculate the (DFT), made this a most practical method. The advantage of this (block processing) over the direct method increases with block size N. However, the computationally efficient FFT method still involves significant round off error, and requires the storage or generation of the sine and cosine functions which will be rounded. This motivated the study of other transforms which retain the (CCP), while reducing rounding errors and computational load.

The cyclic convolution property, known in the complex field for the (DFT), has been extended to other families of transforms defined in finite fields and known as number theoretic transforms (NTTs). Early references to the subject are Knuth [7], Good [8], Pollard [9] and Rader [10,11]. Agarwal and Burrus studied these transforms and their application to digital signal processing [12,13].

This work revealed that, in finite fields there is a whole family of transforms each with properties depending on the chosen modulus and kernel. VLSI designs for FNT transformers have been developed [17],[18]. In this paper, the problem of calculating the convolution is investigated when a transform length is needed which is longer than that available from existing VLSI designs or software. A technique is presented allowing the calculation of convolutions and correlations for data sequences of length greater than N when the available VLSI design will accommodate only length N.

2. DEFININTION OF NTTs

Let F be any prime or composite number (the product of mutual primes) and α be a primitive root of order N, then a NTT in this ring is defined as:

$$X(k) = \sum_{n=0}^{N-1} x(n)\alpha^{nk} \bmod F \qquad (5)$$

$$k=0,1,2,...,N-1$$

where N is the transform length and

$$\alpha^N = 1 \quad \text{Mod } F \quad \text{and } \alpha^p = 1 \quad \text{for } 0 < p < N$$

If N and F have no common factor then N^{-1} exists and by analogy to the DFT an inverse can be defined

$$x(n) = N^{-1} \sum_{k=0}^{N-1} X(k)\alpha^{-nk} \quad \text{Mod } F \qquad (6)$$

$$n=0,1,2,...,N-1$$

where N^{-1} is the inverse of N mod F

Since α is a primitive root of order N, the exponents of α are calculated modulo N. Replacing n by -n, Eq.6 and Eq.5 become similar except for a scale factor N. This means that the inverse transform can be calculated by time reversing the input and applying the same forward transform, so simplifying implementation.

The NTTs in general have the cyclic convolution property and can be used to calculate the convolutions by the same method as the DFT. The NTTs have some advantages over the FFT:

1 NTTs do not need any manipulation of the trigonometric functions (sine and cosine);
2 They are calculated modulo an integer and hence they are error free making exact results possible;
3 Multiplication free transforms can be achieved through an appropriate choice of the transform parameters.

3. NUMBER THEORETIC TRANSFORMS AND THE TRANSFORM LENGTH

For NTTs a rigid relationship exists between the modulus, the transform length and the kernel chosen. For binary processors, especially those using application specific circuits (ASICs) to accommodate the required number of bits, the Fermat numbers F3-F6 are good choices for defining NTTs (F5,F6 are not primes). From [12], the maximum length for a multiplication free transform for $\alpha=\surd 2$ is 256 . This is well suited for moderate length convolutions and image filtering.

A problem, however, arises when a longer transform length is needed. This can happen in one dimensional processing where the sequences to be processed are naturally long.

Agarwall and Burrus [14] used the scheme proposed by Rader [15] of mapping one dimensional convolution to two dimensions. With this scheme the sequence length which could be filtered via the multiplierless FNTs is proportional to the square of the word length . Other authors proposed the use of mixed radix NTTs with 2 as a root of unity and composite transform lengths at the expense of using moduli with more complex arithmetic [16].

Another simple solution is the use of a kernel which is different from 2, for example $\alpha=3$. Using this as the basis function all the transform lengths up to the modulus (for prime numbers) are possible and the transform length problem is considerably alleviated. Values of N up to 65536 are possible with $\alpha=3$; this requires the use of multiplication. Compared with the FFT, the FNT provides a simpler butterfly structure involving one single integer multiplier per butterfly against four real multipliers for the FFT.

4. THE RELATIONSHIP BETWEEN THE LINEAR AND CIRCULAR CONVOLUTIONS:

Let the third Fermat number F_3 be the modulus, $N=32$ and $\alpha=\surd 2$; (however, the explanation applies for all NTTs). For this choice of α, and modulus, the NTTs are:

$$X(k) = \sum_{n=0}^{31} x(n)\surd 2^{nk} \text{ Mod } F_3 \qquad (7)$$

$$k=0,1,2,...,31$$

Where $F_3 = 2^8 + 1 = 257$;
and the inverse

$$x(-n)= \frac{1}{32} \sum_{k=0}^{31} X(k)\surd 2^{nk} \text{ Mod } F_3 \qquad (8)$$

$$n=0,1,...31$$

The 1-D convolution operation using FNTs as, illustrated in Fig.1, is given by:

$$y_c(n) = \text{INFNT}(\text{FNT}[x(n)] \times \text{FNT}[h(n)]) \qquad (9)$$

Where x is point by point multiplication and $y_c(n)$ is equivalent to:

$$\bar{y}_c(n)= \sum_{m=0}^{N-1} x(m)h(<n-m>N) \tag{10}$$

< >N means the indices are calculated modulo N. This results in a circular convolution with the same length as the convolved sequence. This is usually written as:-

$$y_c(n)=x(n)xh(n) \tag{11}$$

In many digital signal processing problems the desired convolution is linear (Eq.1) rather than circular.The direct application of any transform, to the calculation of the convolution function of discrete data leads to a circular convolution which is different from the desired linear convolution leading to erroneous results. This is inherent in the operation of transform algorithms which assume that the function being transformed is periodic, and hence the resulting N point transform is also periodic.In order to bring the efficiency of fast transforms to the calculation of aperiodic convolutions, a relationship is needed between these convolutions . A well known method is to increase the size of the transformed sequences by including sufficient zero values to prevent the individual periods of the convolution from overlapping. The difference between a circular and a linear convolution is shown in figure 2.

The linear convolution of a sequence of length M_1 and a filter of filter length L is an ($M=M_1$+L-1) point linear convolution, is shown in figure 2a (for $L=M_1=N$). A circular convolution figure 2b can be equivalent to a linear one, only if the sequences M1 and L are sufficiently padded with zero values so that the calculation of a linear convolution is completed before any value of the circular convolution begins to wrap around. This is the case if the period of the circular convolution is at least equal to the linear convolution length:-

$$N \geq M_1+L-1 \tag{12}$$

Therefore, the two sequences should be padded with zeros to give an extended form as follows:-

$$x(n)=\begin{cases} x(n) & \text{for } n=0,1,2,\dots,M-1 \\ 0 & \text{for } M \leq n \leq N-1 \end{cases}$$

and

$$h(n) \text{ for } n=0,1,2,\dots,L-1$$
$$h(n) = 0 \quad \text{for } L \leq n \leq N-1$$

Convolving these sequences using NTTs will lead to the desired linear convolution.

5. CALCULATION OF LINEAR CONVOLUTION USING SHORTER CIRCULAR CONVOLUTION

The cross-over point where the efficiency of frequency domain calculation exceeds that of the time domain calculation depends on the problem, the transform and the hardware available. However, a common criterion is that transform algorithms are more efficient for large filter lengths than for short ones [2,3].

When the multiplierless FNTs are applied, however, only short or medium data sequence lengths are accommodated with currently designed hardware. Therefore, methods which extend the sequences being convolved through FNTs of a given length, while maintaining the other advantages, are of interest. Some of these methods such as the use of multidimensional mapping and higher order roots have already been introduced [14].

In this section a method is introduced that calculates a linear convolution using circular convolution of the same or shorter length than the sequences being convolved, allowing the increase of multiplication free FNT lengths.

Usually, the linear convolution of length (N_1+N_2-1) is calculated through a circular convolution of length $(M \geq N_1+N_2-1)$. A method that calculates the linear convolution using a shorter circular one has been formulated and is outlined below.

Consider as an example the calculation of the linear convolution $y_L(n)$ of two arbitrary sequences x(n) and h(n) given by:

```
x(n)= 2  3  4  1  5  4  3  2  1  4  5  5  4  3
      2  1

h(n)= 1  2  4  6  2  0  0  0  1  2  3  3  2  1
      4  5
```

Calculating by means of FNTs after sufficiently padding both sequences by zero values, the linear convolution is given by:

```
yL(n) = 2    7    18    33    45    48    45    56    53    47
        51   65   92   106    112   114   104   80    67    80
        77   73   63    62   70     64   51    36    24    14
        5    0
```

Their circular convolution when calculated using a circular convolver program of length 16 is found to be:

```
yc(n): 106   87    85    113   122   121   108   118   123   111   102
       101   116   120   117   114
```

Examining this example and figure 2, it can be seen that the circular convolution can be obtained from the linear one by taking the set of samples which pass the maximum shift allowable (the circular convolver length) and adding them to the first part, point to point. These parts, for this particular example , are given by:

$y_L(n)$ = 2 7 18 33 45 48 45 56
53 47 51 65 92 106 112 114

and

$y_L(N+n)$=104 80 67 80 77 73 63 62
70 64 51 36 24 14 5

It is clear that adding these sequences gives yc(n).
Hence $yc(n) = y_L(n) + y_L(N+n)$ (13)

$n=0,1,\ldots,N-1$

This equation answers the question how a circular convolution of two sequences can be calculated from their linear counterpart.
This can also be proved by writing the FNT of the linear convolution y_L:

$$Y_L(k) = \sum_{n=0}^{2N-1} y_L(n)\alpha^{nk} \text{ Mod Ft} \qquad (14)$$

$$Y_L(k) = \sum_{n=0}^{N-1} y_L(n)\alpha^{nk} + \alpha^{Nk}\sum_{n=0}^{N-1} y_L(n)\alpha^{nk} \text{ Mod ft} \qquad (15)$$

The transformed circular convolution is obtained from $Y_L(k)$ by taking:

$$\alpha^N = 1 \text{ Mod Ft} \qquad (16)$$

The effect of taking $\alpha^N = 1$ in Eq.15 results in adding the samples which are beyond N and the first samples of the linear convolution to yield the circular one.

Equation 14 can be written in more general form as

$$Y_L(k) = \sum_{n=0}^{N-1} y_L(n)\alpha^{nk} + \alpha^{nk}\sum_{n=0}^{N_1-1} y_L(n+N)\alpha^{nk} \quad \text{Mod Ft} \qquad (17)$$

For $N_1 < N$

These results can now be used to show how a linear convolution of length greater than N can be calculated using a circular convolution of length N.

First the circular convolution is divided into its first and second parts $y_1c(n)$ and $y_2c(n)$ respectively.
where:

$$y_{1c} = y_c \text{ for } 0 \le n \le N/2-1 \qquad (18)$$

and $$y_{2c} = y_c \text{ for } N/2 \le N1-1 \qquad (19)$$

Let $N_1 = N$ (Results have also been derived for $N_1 = N/2$).

Substituting Eq.18 and Eq.19 into Eq.13 gives:

$$y_{1c}(n) = y_L(n) + y_L(N+n) \text{ for } 0 \le n \le N/2-1 \qquad (20)$$

$$y_{2c}(n) = y_L(n) + y_L(N+n) \text{ for } N/2 \le n \le N-1 \qquad (21)$$

Eq.20 and Eq.21 relate the circular and linear convolutions. If the linear convolution is known, the circular one can be obtained. Otherwise, if the circular is known and the $y_L(n)$ is known for:

$$0 \le n \le N/2-1 \qquad (22)$$

and

$$3N/2 \le n \le 2N-1 \qquad (23)$$

the other parts of the linear convolution can be obtained.

These intervals present the linear convolution over the first and the last N/2 points and can be deduced from the partial linear convolutions x_1*h_1 and x_2*h_2 where:

$$x_1(n) = \begin{cases} x(n) \text{ for } 0 \le n \le N/2-1 \\ 0 \quad \text{Otherwise} \end{cases} \qquad (24)$$

and

$$x_2(n) = \begin{cases} 0 & \text{for } 0 \le n \le N/2-1 \\ x(n) & \text{for } N/2 \le n \le N-1 \end{cases} \qquad (25)$$

The same is done for h(n):

$$h_1(n) = \begin{cases} h(n)\ 0 \le n \le N/2-1 \\ 0 \quad \text{Otherwise} \end{cases} \qquad (26)$$

$$h_2(n) = \begin{cases} 0 \quad \text{Otherwise} \\ h(n) \quad \text{for } N/2 \le n \le N-1 \end{cases} \qquad (27)$$

Now calculate the linear convolution of $x_1(n)$, $x_2(n)$ with $h_1(n)$ and $h_2(n)$ respectively.

$$y_{1L}(n) = x_1(n) * h_1(n) \qquad (28)$$
$$y_{2L}(n) = x_2(n) * h_2(n) \qquad (29)$$

Writing the equations of $y_{1L}(n)$ and $y_{2L}(n)$ and comparing them with Eq.1 leads to:

$$y_{1L}(n) \qquad = y_L(n)$$

for $0 \le n \le N/2-1$ for $0 \le n \le N/2-1$ (30)

$y_{2L}(n)$ $= y_L(n)$
for $N/2 \le n \le N-1$ for $3N/2 \le n \le 2N-1$ (31)

Using these equations, with Eq.18 and Eq.19, the linear convolution can be deduced as:

$$y_L(n) = \begin{cases} y_{1L}(m) \quad 0 \le m \le N/2-1 & \text{for } 0 \le n \le N/2-1 \\ y_{2c}(m+N/2) - y_{2L}(m+N/2) \quad m=0,1,2,\ldots,N/2-1 & \text{for } N/2 \le n \le N-1 \\ y_{2c}(m) - y_{1L}(m) \quad m=0,1,2,\ldots,N/2-1 & \text{for } N \le n \le 3N/2 \\ y_{2L}(m+N/2) \quad m=0,1,2,\ldots,N/2-1 & \text{for } 3N/2 \le n \le 2N-1 \end{cases} \qquad (31)$$

Thus, (2N-1) points linear convolution can be carried out using three N point circular convolutions and N additions as shown in figure 3.

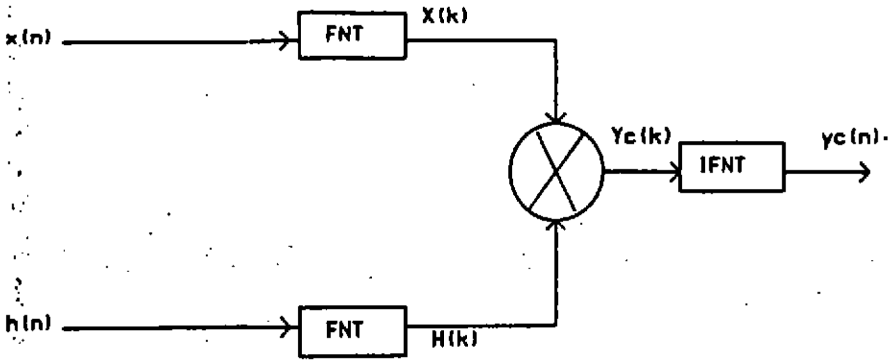
6. CONCLUSION

A method is introduced that calculates a linear convolution using a circular convolution of a relatively shorter length and if used in conjunction with the FNTs allows the doubling of the linear convolution length that can be calculated using the multiplication free FNT transforms. This method is also compared with the conventional method and found to be more efficient in terms of speed and memory requirement.

7. REFERENCES

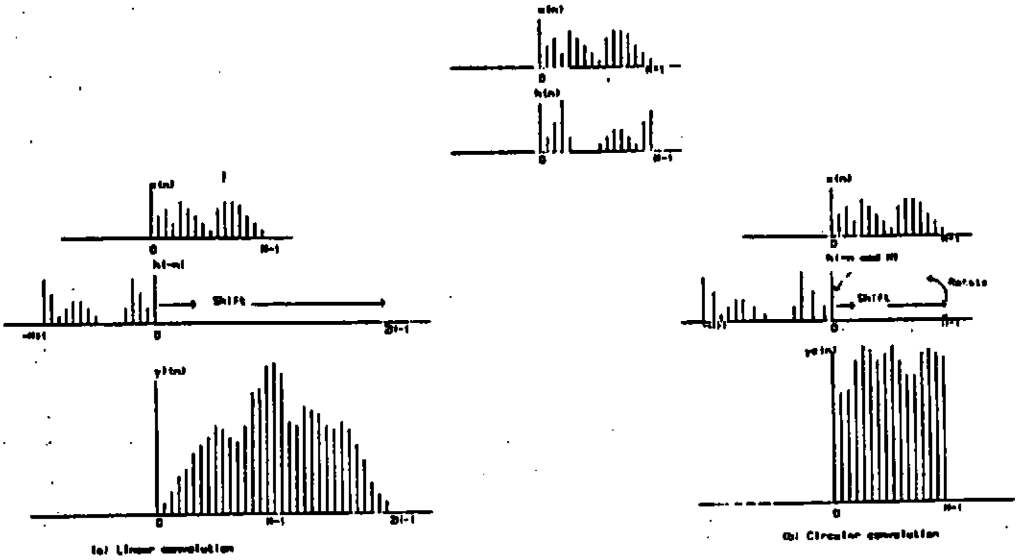
1. Rabiner, L.R., and Gold, B.: "Theory and application of digital signal processing", Prentice Hall, London, 1975, pp 419-433.
2. Stockham, T.G.,:"High speed convolution and correlation." In spring Joint Comput. Conf., AFIPS Co/nf. Proc.28: 229-233. Washington, D.C: Spartan, 1966,pp.
3. Cooley J.W. and Tukey J.W.: "An Algorithm for Machine Computation of Complex Fourier Series", Math. Comp. Vol.19, April 1965, pp. 297-301
4. Hall E.L: "A comparison of computations for spatial frequency filtering", Proc. IEEE, Vol.60, No.7, July 1972,pp.
5. Brigham, E.O.: "The fast Fourier transform", Englewood Clifts, Nj: Prentice-Hall, Inc. 1974
6. Meyer, R.: "Error analysis and comparison of FFT implementation structures", IEEE INT. Conference, on Acoustics, Speech and Signal processing, ICASSP.89, Glasgow, Scotland, pp.888-8917.
7. Knuth D.E.: "The art of computer programming", Vol.2, Semi numerical Algorithms". Proc., Mass., Addisson-Wesly, 1969 p.555 and pp.259-262

8. Good I.J. : "The relationship between two fast Fourier transforms", IEEE Trans. Comput., Vol. C-20, March 1971, pp. 310-317.
9. Pollard, J.M : "The Fast Fourier transform in a finite field", Mathematics of Computation, Vol.25, No.114, April, 1971, pp. 365-374
10. Rader, C.M : "The number theoretic DFT and exact discrete convolution", IEEE Arden House Workshop on digital signal processing, Harriman, N.Y., Jan.11, 1972.
11. Rader, C.M.: "Discrete convolution via Mersenne transforms", IEEE Trans. Comput. Vol. C-21, Dec. 1972, pp. 1269-1273.
12. Agarwal, R.C., and Burrus, C.S.: "Fast convolution using Fermat number transforms with applications to digital filtering", IEEE Trans. On Acoustics, Speech and Signal Processing, ASSP-22, No.2, 1974 pp. 87-97.
13. Agarwal, R.C., and Burrus, C.S.: "Number theoretic transforms to implementfast digital convolution", Proc. IEEE, 1974, 63(4). pp. 550-560.
14. Agarwal R.C. and Burrus C.S.: "Fast one dimensional digital convolution by multidimensional techniques", IEEE Trans. Acoust., Speech, and Signal Processing, ASSP-22, No.1, Feb. 1974, pp. 1-10.
15. Burrus. S.C.: "Index mapping for Multidimensional formulation of the DFT and convolutions" IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP, No 3, June 1977, pp. 239-242.
16. Duhamel P. and Hollman H.: "Number theoretic transforms with 2 as a root of unity", Elec. Lett. Vol 18, 28th October 1982, pp. 978-980.
17. Truong T.K. Reed I.S., Yeh, C.S and Shao H.M.: "Parallel VLSI architecture for a digital filter of arbitrary length using the Fermat number transforms". IEEE Proc. Int. Conference on Circuits and Computers (ICCC-82) Sept. 28-Oct.1, 1982, New York, pp.574-578
18. Boussakta, S. and Holt, A.G.J.: "Calculation of the discrete Hartley transform via the Fermat number transform using a VLSI chip", IEE Proc. Vol. 135, Pt. G, June 1988 pp.101-103.



Circular convolution calculated using the Fermat number transform

Figure 1

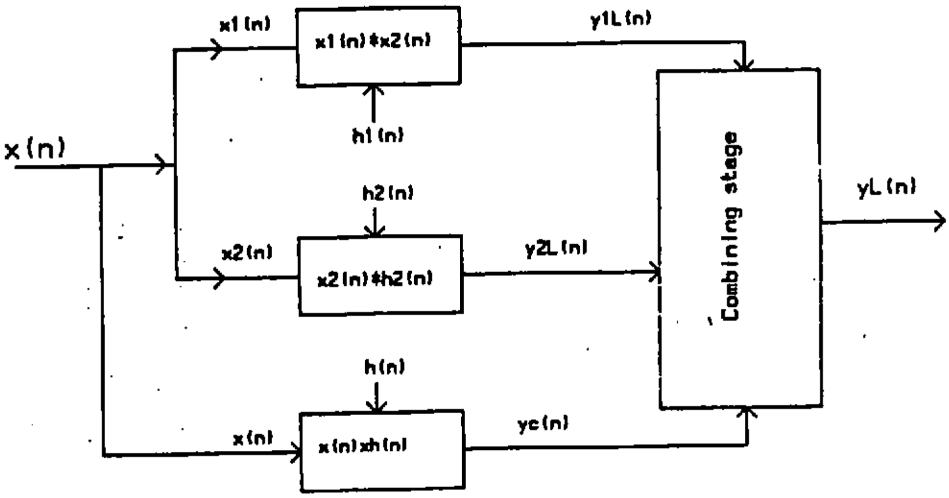


(a) Linear convolution

(b) Circular convolution

The difference between linear and circular convolutions

Figure 2



The calculation of the linear convolution of two N-point sequences using N-pont circular convolver

Figure (3)