A Linear Filtering Approach to the Computation of Discrete Fourier Transform

LEO I. BLUESTEIN, Member, IEEE
Electronic Systems Laboratory
General Telephone and Electronics Laboratories, Inc.
Waltham, Mass. 02154

Abstract

It is shown in this paper that the discrete equivalent of a chirp filter is needed to implement the computation of the discrete Fourier transform (DFT) as a linear filtering process. We show further that the chirp filter should not be realized as a transversal filter in a wide range of cases; use instead of the conventional FFT permits the computation of the DFT in a time proportional to $N \log_2 N$ for any N, N being the number of points in the array that is transformed. Another proposed implementation of the chirp filter requires N to be a perfect square. The number of operations required for this algorithm is proportional to $N^{3/2}$.

Manuscript received October 15, 1968; revised August 20, 1970.

Introduction

There is currently a good deal of interest [1], [2] in a technique known as the fast Fourier transform (FFT), which is a method for rapidly computing the discrete Fourier transform of a time series (discrete data samples). This transform is the array of N numbers, A_n, $n=0, 1, \cdots, N-1$ which are defined by the relation

$$A_n = \sum_{k=0}^{N-1} X_k \exp(-2\pi jnk/N) \qquad (1)$$

where X_k is the kth sample of a time series consisting of N (possibly complex) samples and $j=\sqrt{-1}$. Quite obviously, computing the A_n in a brute force manner requires N^2 operations,[1] a number which is too large to make the use of the discrete transform attractive. The fast Fourier transform discovered by Cooley and Tukey [3], on the other hand, requires about $N \log_2 N$ operations (if N is chosen to be a power of 2) a saving which is so great when N is large that it makes the use of the discrete transform attractive in such fields as digital filtering, computation of power spectra and autocorrelation functions and the like. If N is not chosen optimally, that is, if N is a composite number of the form

$$\prod_{i=1}^{s} p_i^{k_i}$$

where the p_i are prime numbers, the number of operations required by the Cooley–Tukey algorithm is proportional to

$$N \cdot \sum_{i=1}^{s} k_i p_i.$$

These results may be derived by viewing the computation of the DFT as a multistage operation on the array of X's.

We view the computational problem here rather as one of linear filtering. It is first shown that the discrete version of a chirp filter is an essential part of a linear filtering procedure which converts the X array to the A array. We then show that it is usually inadvisable to realize the chirp filter in transversal filter form since little saving in the number of operations would result. If instead the FFT is used to realize this filter, it is possible to compute the DFT in a time proportional to $N \log_2 N$ for any N. We have also devised another synthesis procedure for the chirp filter when N is a perfect square; in this case the number of computations is proportional to $N^{3/2}$.

An Algorithm Suggested by Chirp Filtering

This work was motivated by the observation that chirp filtering a waveform is nearly equivalent to taking its Fourier transform. For the sake of argument, assume that

[1] We usually take an operation to mean multiplication by a complex number plus all attendant computations, since multiplication of this sort on a computer requires an inordinate amount of time.

a waveform $f(t)$ is applied to the input of a chirp filter. Because a chirp filter's delay characteristic is proportional to frequency, the low frequencies contained in $f(t)$ appear in the filter's output sooner than components at the high frequencies. This is then roughly a Fourier analysis of the waveform $f(t)$. Hence it appears reasonable to try to use a sampled equivalent of a chirp filter to generate the discrete Fourier transform.

For analog waveforms, a chirp filter is one whose frequency response is the bandpass equivalent of $\exp(-jf^2)$ over a range of frequencies. The impulsive response of this filter is proportional to $\exp[j(\pi t)^2]$. Now let us enter the realm of sampled systems. Motivated by the above discussion, we consider the effect of a sampled filter whose impulsive response is

$$h_r = \exp(+j\pi r^2/N) \tag{2}$$

on a sampled time function whose values are y_n. The y_n are assumed to be nonzero only for $0 \leq n \leq N-1$. Use of the convolution summation shows that the output of the filter at time $N+n$, u_{N+n} is

$$u_{N+n} = \sum_{k=0}^{N-1} y_k e^{j\pi(N+n-k)^2/N} \tag{3}$$

or

$$u_{N+n}e^{-j\pi n^2/N - j\pi N} = \sum_{k=0}^{N-1} (y_k e^{j\pi k^2/N}) e^{-2\pi jnk/N}. \tag{4}$$

Therefore, if we force

$$X_k = y_k e^{+j\pi k^2/N} \tag{5}$$

a comparison of (1) and (5) shows that

$$A_n = u_{N+n} e^{-j\pi n^2/N - j\pi N}. \tag{6}$$

Therefore, as Fig. 1 shows, the problem reduces to one of efficiently realizing a sampled chirp filter, whose impulsive response is given by (2). Moreover, we need only match (2) for $2N-1 \geq r \geq 0$. In addition, it is easily seen that the function $e^{-j\pi n^2/N}$ can be computed recursively to form the quantities y_k and A_k from X_k and u_{N+k}, respectively. This recursion is shown in Fig. 2 and depends on the observation that

$$e^{-j\pi(n+1)^2/N} = e^{-j\pi n^2/N} e^{-j2\pi n/N} e^{-j\pi/N}. \tag{7}$$

In Fig. 2 the lower loop computes $e^{-j2\pi n/N}$; the upper loop performs the remaining operations indicated by (7). In using this recursion one must take care to avoid accumulation of round-off error.

Chirp Filter Synthesis Procedures

A. Transversal Filter Techniques

It is readily apparent that the sampled chirp filter essential to our algorithm may be synthesized as a transversal filter, i.e., as the sampled equivalent of a tapped delay line as shown in Fig. 3. The tap gains in this realization are of the form $e^{-j\pi r^2/N}$, $r=0, 1, 2, \cdots, N-1$. It is also clear that the number of distinct multipliers required in this realization can be reduced by observing that for certain values of r, say r_1 and r_2,

$$e^{-j\pi r_1^2/N} = \pm e^{-j\pi r_2^2/N} \tag{8}$$

or

$$e^{-j\pi r_1^2/N} = \pm je^{-j\pi r_2^2/N}. \tag{9}$$

If (8) holds, the quantity at the r_2th tap is multiplied by ± 1, added to the number on the r_1th tap and the sum multiplied by $e^{-j\pi r_1^2/N}$; if (9) holds, the r_2th quantity is multiplied by $\pm j$ (interchange of real and imaginary parts followed by possible negation) and summed as before.

The appropriate question now is, "How many multipliers are actually needed?" This question can be answered by the following considerations. If (8) holds,

$$r_1^2 = r_2^2 \bmod N \tag{10}$$

because (8) implies that

$$e^{-j\pi r_1^2/N} = e^{-j\pi(r_2^2+\alpha N)/N} \qquad \alpha = 0 \quad \text{or} \quad 1. \tag{11}$$

If (9) holds,

$$e^{-j\pi r_1^2/N} = e^{-j\pi(r_2^2+\beta(N/2))/N} \qquad \beta = +1 \quad \text{or} \quad -1. \tag{12}$$

Since $r_2^2+\beta(N/2)$ must be an integer, it follows that $N/2$ must be an integer, i.e., N is even. Thus (9) implies

$$r_1^2 = r_2^2 \bmod N/2. \tag{13}$$

If N is odd, we may disregard (9) and therefore (13) and concentrate on (10). The number of multipliers required is the number of distinct values of a such that

$$x^2 = a \bmod N \tag{14}$$

has a solution among the values $x=1, 2, 3, \cdots, N-1$. When this is so, and a and N are relatively prime, a is referred to as a quadratic residue of N.

If N is even, (10) implies (13). By the same considerations as above, we find that when N is even, the number of required multipliers is the number of distinct values of a, such that

$$x^2 = a \bmod N/2 \tag{15}$$

has a solution. Again, if a and $N/2$ are relatively prime, a is referred to as the quadratic residue of $N/2$.

Although we have not been able to find a complete solution to these questions, the following partial results may be obtained:

1) The number of distinct nonzero values of a satisfying (14) is equal to or greater than the largest integer less than $\sqrt{N}$, since values of x less than or equal to $\sqrt{N}$, when squared, generate distinct values of a. A similar consideration holds for (15).

2) When N is prime, a is always a quadratic residue. It then follows from number theory that the number of multipliers required is $(N-1)/2$.

3) When $N=2^{\alpha+1}$, the number of distinct values of a satisfying (15) can be computed by means of the following considerations. We divide the integers less than $N/2$ into

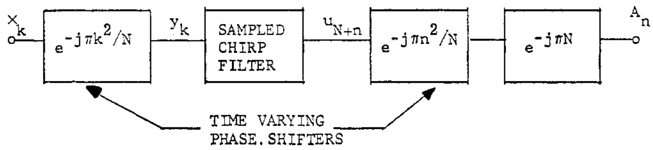


Fig 1. A Fourier transformer.

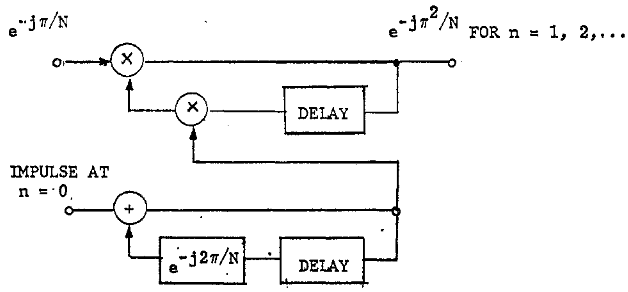
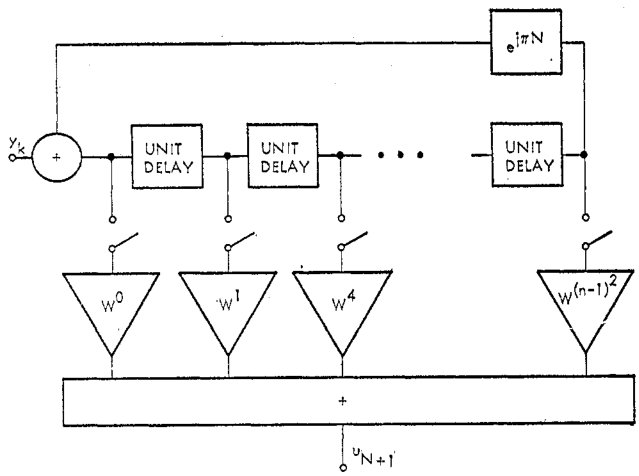


Fig. 2. A recursive method for generating $\exp(-j\pi n^2/N)$.

Fig. 3. A tapped delay-line version of a sampled chirp filter. $W \equiv e^{j\pi/N}$. Multiplications take place after the y_k's are read in.



two sets, one comprised of odd numbers and one made up of the remaining even ones. The quadratic residues of 2^α are derived from the set of odd integers; these number $2^{\alpha-3}$ (if $\alpha \geq 3$).[2] Since each odd integer corresponds to some quadratic residue of 2^α, only the set of even integers need now be considered. We note that if an even value of x generates a distinct value of a, this value must be a multiple of 4. These values can therefore be found by taking the distinct values of a_1 which satisfy

$$x^2 = a_1 \bmod 2^{\alpha-2} \tag{16}$$

[2] This follows from two problem exercises on p. 39 of MacDuffee [4] and a result given on p. 94 by Vinagradov [5]. These are, respectively:

"Ex. 3. If $x^2 \equiv 1 \bmod n$ has exactly k solutions, and if $(c, n) = 1$, then $x^2 \equiv c \bmod n$ has no solution or exactly k solutions.

"Ex. 4. If c is a quadratic residue modulo n, and if $x^2 \equiv c \bmod n$ has k roots, there are exactly $\varphi(n)/k$ quadratic residues of n."

and

"The necessary conditions for the solvability of the congruence
$x^2 \equiv a (\bmod 2^\alpha)$; $(a, 2) = 1$
are: $a \equiv 1 (\bmod 4)$ for $\alpha = 2$, $a \equiv 1 (\bmod 8)$ for $\alpha \geq 3$. If these conditions are satisfied, then the number of solutions is: 1 for $\alpha = 1$; 2 for $a = 2$; 4 for $\alpha \geq 3$."

The number of residue classes relatively prime to 2^α, $\varphi(2^\alpha)$, is just the number of odd integers less than 2^α and so is equal to $2^{\alpha-1}$. Since k (see Ex. 3) is 4, the number of quadratic residues is therefore $2^{\alpha-1}/4 = 2^{\alpha-3}$.

and multiplying them by 4. Once again we divide the integers less than $2^{\alpha-2}$ into odd and even groups; the group of odd numbers yield $2^{\alpha-5}$ distinct values of a_1 (if $\alpha - 2 \geq 3$). The remaining group of even integers will generate distinct values of a_1 corresponding to distinct values of a_2 which satisfy the new relation

$$x^2 = a_2 \bmod 2^{\alpha-4}. \tag{17}$$

The process continues until we have the relation

$$x^2 = a \bmod 4 \tag{18}$$

or

$$x^2 = a \bmod 2. \tag{19}$$

Equation (18) has only one nontrivial value of a which satisfies it, $a = 1$. Equation (19) has only a trivial value of a satisfying it, which does not require another multiplier. Thus, the number of distinct multipliers required to achieve a chirp response in transversal filter is

$$2^{\alpha-3} + 2^{\alpha-5} + 2^{\alpha-7} + \cdots \beta \tag{20}$$

where

$$\beta = \begin{cases} 0 & \alpha \text{ even} \\ 1 & \alpha \text{ odd.} \end{cases} \tag{21}$$

For large α, (20) is very nearly

$$2^{\alpha-3}\left(\frac{4}{3}\right)=\frac{2^{\alpha+1}}{12} \tag{22}$$

so that the number of multipliers needed is of the order of $N=2^{\alpha+1}$; combining terms prior to multiplying them only reduces the number of required multiplications by a factor of 12.

Thus, the transversal filter approach is not very fruitful for large N. The first observation implies that the best that one can hope for using this technique is a computational procedure with the number of computations proportional to $N^{3/2}$. The other two observations show that even this is not achievable for N a prime or a power of 2.

B. FFT Techniques

However, it should be observed[3] that we may realize *any* filter by means of the fast Fourier transform technique as suggested by Stockham [6]. In this method both the input to a filter and the filter's impulsive response are properly augmented by zeros and fast Fourier transformed. The results are multiplied together and the inverse transform applied. The number of operations is of the order of 3 $N'\log_2 N'$ where N' is both the length of the augmented impulsive response and the augmented input. The number N' is selected for this case to be the smallest power of 2 greater than or equal to $2N-1$. Thus, if a fast Fourier transform algorithm requiring N to be a power of 2 is available, it can easily be converted to one which accepts any N. The number of operations for the converted program will also be proportional to $N \log N$. This idea has been extended further by Rabiner, Shafer, and Rader [7].

C. Recursive Techniques

In this section we show that it is possible to achieve the performance promised by observation (1) using recursive techniques. We shall assume in what follows that N is a perfect square, i.e., that

$$N=m^2. \tag{23}$$

We will assume that the impulsive response given by (2) is zero for $r>2N-1$. The z transform, $H(z)$, of the impulsive response h_r is

$$H(z)=\sum_{r=0}^{2N-1} e^{j\pi r^2/m^2}z^{-r}. \tag{24}$$

Let us write

$$r=t+im \tag{25}$$

where

[3] This elegant observation is due to C. Rader of the M.I.T. Lincoln Laboratory.

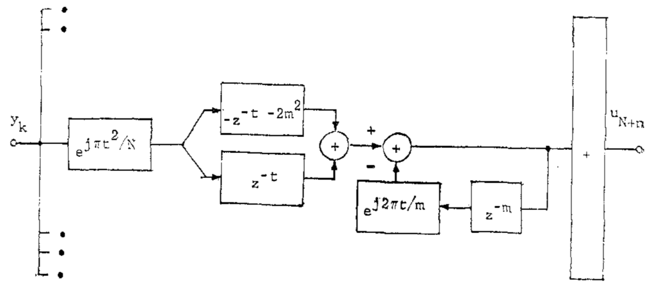


Fig. 4. A realization of $H(z)$.

$$t\le m-1. \qquad i\le 2m-1. \tag{26}$$

It then follows that

$$H(z)=\sum_{t=0}^{m-1} z^{-t}e^{j\pi t^2/N}\sum_{i=0}^{2m-1} e^{j2\pi ti/m}z^{-im}e^{j\pi i^2}. \tag{27}$$

but since

$$e^{j\pi i^2}=\begin{cases}-1 & i \text{ odd}\\ +1 & i \text{ even}\end{cases} \tag{28}$$

it follows that

$$e^{j\pi i^2}=(-1)^i \tag{29}$$

and

$$\begin{aligned}H(z)&=\sum_{t=0}^{m-1} z^{-t}e^{j\pi t^2/N}\sum_{i=0}^{2m-1}\left[-e^{j2\pi tm}z^{-m}\right]^i\\ &=\sum_{t=0}^{m-1} z^{-t}e^{j\pi t^2/N}\,\frac{1-z^{-2m^2}}{1+e^{j2\pi t/m}z^{-m}}.\end{aligned} \tag{30}$$

Thus, $H(z)$ may be realized by a bank of m filters as shown in Fig. 4. (A box labeled z^{-x} represents a delay of x units.) Since the impulsive response after $2m^2-1$ units does not concern us, we may open circuit the link connected to the box labeled z^{t-2m^2}. The number of operations required is about $3N\ m$, since we need only count operations which take place during the crucial interval between 0 and $2N-1$. This number is of the same form (within a multiplicative constant) as that required by the Cooley–Tukey algorithm for $N=m^2$ and m a prime.

Additional Comments

We have shown that by viewing the computation of the DFT as a linear filtering problem one is led naturally into algorithms which depend crucially upon the discrete version of a chirp filter. For arrays of large size, realizing the chirp filter by means of the FFT is appropriate, i.e., linear filtering cannot replace the FFT but can only extend its usefulness. For arrays of moderate size, and for wired program machines, algorithms based on the realizations shown in either Fig. 1 or Fig. 4 may be preferable for reasons of simplicity. Our results would tend to indicate that of these choices, the synthesis given in Fig. 4 is probably more desirable since asymptotically with N it behaves as well as any chirp transversal filter can ever be expected to perform. For special situations, and when hardware considerations are important, this tentative conclusion may not hold true.

References

[1] J. W. Cochran, J. W. Cooley, D. L. Favin, H. D. Helms, R. A. Kaenel, W. W. Lang, G. C. Maling, Jr., D. E. Nelson, C. M. Rader, and P. D. Welch, "What is the fast Fourier transform?" *Proc. IEEE*, vol. 55, pp. 1664–1674, October 1967.

[2] E. O. Brigham and R. E. Morrow, "The fast Fourier transform," *IEEE Spectrum*, vol. 4, pp. 63–70, December 1967.

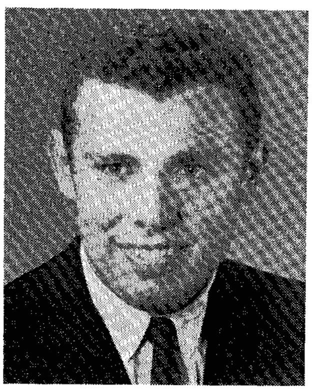
[3] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comput.*, vol. 19, pp. 297–301, April 1965.

[4] C. C. MacDuffee, *An Introduction to Abstract Algebra*. New York: Wiley, 1940.

[5] I. M. Vinagradov, *Elements of Number Theory*. New York: Dover, 1958.

[6] T. G. Stockham, "High speed convolution and correlation," *1966 Spring Joint Comput. Conf., AFIPS Proc.*, vol. 28. Washington, D. C.: Spartan, 1966, pp. 229–233.

[7] L. R. Rabiner, R. W. Schafer, and C. M. Rader, "The chirp z-transform algorithm and its application," *Bell Syst. Tech. J.*, vol. 48, no. 5, pp. 1249–1292, May–June 1969.



Leo I. Bluestein (S'60–M'63) was born in Bronx, N. Y., on April 24, 1936. He received the B.S.E.E. degree from the City College of New York, New York, N. Y., in 1957, and the M.S.E.E. and Ph.D. degrees from Columbia University, New York, N. Y., in 1959 and 1962, respectively.

From 1962 to 1966 he was with the M.I.T. Lincoln Laboratory, Lexington, Mass., where he was concerned with problems in signal design and analog modulation. Since 1966 he has been with the General Telephone and Electronics Corporation, and is currently managing the Information Processing Group of the GT & E Laboratories' Communication and Information Sciences Department, Waltham, Mass. His present research interests include feedback communication and digital filter design.