INTERNATIONAL ATOMIC ENERGY AGENCY
UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION
INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
I.C.T.P., P.O. BOX 586, 34100 TRIESTE, ITALY, CABLE: CENTRATOM TRIESTE

UNITED NATIONS INDUSTRIAL DEVELOPMENT ORGANIZATION
INTERNATIONAL CENTRE FOR SCIENCE AND HIGH TECHNOLOGY
c/o INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS 34100 TRIESTE (ITALY) VIA GRIGNANO, 9 (ADRIATICO PALACE) P.O. BOX 586 TELEPHONE 040-224572 TELEFAX 040-224575 TELEX 460449 APH I

H4.SMR/537-14

SECOND COLLEGE ON THEORETICAL AND EXPERIMENTAL RADIOPROPAGATION PHYSICS
(7 January - 1 February 1991)

Co-sponsered by ICTP, URSI, ICSU and with the participation of ICS

THE DISCRETE AND FAST FOURIER TRANSFORMS

S. C. Dutta Roy
Indian Institute of Technology, Delhi
New Delhi, India

MAIN BUILDING STRADA COSTIERA, 11 TEL. 22401 TELEFAX 224163 TELEX 460392 ADRIATICO GUEST HOUSE VIA GRIGNANO, 9 TEL. 224241 TELEFAX 224531 TELEX 460449
MICROPROCESSOR LAB. VIA BEIRUT, 31 TEL. 224471 TELEFAX 224163 TELEX 460392 GALILEO GUEST HOUSE VIA BEIRUT, 7 TEL. 22401 TELEFAX 224559 TELEX 460392

THE DISCRETE AND FAST FOURIER TRANSFORMS

S C DUTTA ROY

Department of Electrical Engineering
Indian Institute of Technology
New Delhi 110 016
India

Text of a series of lectures to be delivered at the Second College on Theoretical and Experimental Radio-propagation Physics, sponsored by the URSI and the ICTP, Trieste, 7 January - 1 February 1991

1. THE DISCRETE FOURIER TRANSFORM

One way of implementing a discrete time system is based on the fact that the output sequence {y(n)} is the convolution of the input sequence {x(n)} with the impulse response sequence {h(n)} of the system as given by

$$y(n) = \sum_{r=0}^{n} h(r)x(n-r) = \sum_{r=0}^{n} h(n-r)x(r) \qquad (1)$$

Recall that in the continuous signal case, the convolution of the input signal x(t) with the impulse response h(t) gives the output y(t) and that in the frequency domain, this amounts to a multiplication of the transforms (Laplace or Fourier) of x(t) and h(t) to give the transform of y(t). y(t) can then be obtained by the inverse transform operation. A similar operation can be performed with discrete time systems if we have a suitable transform. As is well known, the z-transform does provide such a vehicle; however, for numerical computation, a modified version of it, called the Discrete Fourier Transfo (DFT) has been found most suitable. The signal processing operation then simpl boils down to the following sequence of computations:

1. Compute the DFT of {x(n)}
2. Compute the DFT of {h(n)}
3. Multiply the two
4. Compute the inverse DFT (IDFT) of the product.

Let, for simplicity, the notation x_k be used for $x(k) \equiv x(kT)$, and consider a sequence $\{x_k\}$ of length N i.e. k = 0, 1, 2, ... N - 1. Then the DFT of $\{x_k\}$ is defined by

$$A_r = \sum_{k=0}^{N-1} x_k e^{-j2\pi rk/N}, \quad r=0,1,2,\ldots N-1 \qquad (2)$$

The DFT is thus also a sequence $\{A_r\}$ of length N. The x_k's may be complex numbers; the A_r's are almost always complex. For notational convenience, let

$$W = e^{-j2\pi/N} \qquad (3)$$

so that

$$A_r = \sum_{k=0}^{N-1} x_k W^{rk}, \quad r=0,1,\ldots N-1 \qquad (4)$$

If one compares (4) with the continuous Fourier transform $A(\omega)$ of a signal $x(t)$ viz.

$$A(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt, \qquad (5)$$

then one way of interpreting the DFT is that it gives the N-point discrete spectrum of the N-point time series $\{x(kT)\}$ at the frequency points $\frac{r}{NT}$, r=0,1,...N-1; the fundamental frequency, obviously, is $f_o=1/(NT)$.

The inverse DFT (IDFT) of the complex sequence $\{A_r\}$, r=0,1,...N-1, is given by

$$x_k = \frac{1}{N} \sum_{r=0}^{N-1} A_r W^{-rk}, \quad k=0,1,\ldots N-1 \qquad (6)$$

That this exists and is unique can be easily established by substituting (4) in (6) and carrying out some elementary manipulations.

Since $e^{j\theta}$ is periodic with a period 2π, it follows from (4) and (6) that

$$A_r = A_{r+mN} \qquad\qquad m=0,\pm1,\pm2,\ldots \qquad (7)$$
$$x_k = x_{k+mN}$$

i.e. both DFT and IDFT yield sequences which are periodic, with periods $Nf_o = \frac{1}{T} = f_s$ and NT respectively.

2. THE FAST FOURIER TRANSFORM

The Fast Fourier Transform (FFT) is a highly efficient method for computing the DFT of a time series. A direct computation from (4) would require N^2 complex multiplications; in contrast, application of FFT can reduce this number to $(N/2)\log_2 N$. For example, for N=512, the ratio $(N/2)\log_2 N \div N^2$ becomes less than 1 percent. This drastic reduction in computation time through FFT has made the FFT an important tool in many signal processing applications.

The DFT, given by (4) and its inverse, given by (6), are of the same form so that any algorithm capable of computing one may be used for computing the other by simply exchanging the roles of x_k and A_r, and making appropriate scale factor and sign changes. There are two basic forms of FFT; the first, due to Cooley and Tukey [1], is known as <u>decimation in time</u>, while the other, obtained by reversing the roles of x_k and A_r, gives the form called <u>decimation in frequency</u>, and was proposed by Gentleman and Sande [2]. Clearly, they should be equivalent; it is however worth distinguishing between them and discussing them separately.

Let N be even and the sequence $\{x_k\}$ be decomposed as

$$\{x_k\} = \{u_k\} + \{v_k\} \qquad (8)$$

where

$$u_k = x_{2k} \qquad k=0,1,2,\ldots\frac{N}{2}-1 \qquad (9)$$
$$v_k = x_{2k+1}$$

Thus $\{u_k\}$ contains the even numbered points and $\{v_k\}$ contains the odd numbered points of $\{x_k\}$ and each has N/2 points. The DFT's of $\{u_k\}$ and $\{v_k\}$ are, therefore,

$$B_r = \sum_{k=0}^{\frac{N}{2}-1} u_k e^{-j2\pi rk/(N/2)}$$
$$= \sum_{k=0}^{\frac{N}{2}-1} v_k e^{-j4\pi rk/N} \qquad r=0,1,2,\ldots\frac{N}{2}-1 \qquad (10)$$
$$C_r = \sum_{k=0}^{\frac{N}{2}-1} u_k e^{-j4\pi rk/N}$$

The DFT we want is

$$A_r = \sum_{k=0}^{N-1} x_k e^{-j2\pi rk/N}$$
$$= \sum_{k=0}^{\frac{N}{2}-1} x_{2k} e^{-j4\pi rk/N} + \sum_{k=0}^{\frac{N}{2}-1} x_{2k+1} e^{-j2\pi r(2k+1)/N}$$
$$r=0,1,2,\ldots N-1$$
$$= B_r + e^{-j2\pi r/N} C_r, \quad 0 \le r < N/2, \qquad (11)$$

because B_r and C_r are defined for r=0 to $\frac{N}{2}-1$. Further, B_r and C_r are periodic with period $\frac{N}{2}$ so that

$$B_{r+N/2} = B_r \text{ and } C_{r+N/2} = C_r. \qquad (12)$$

Thus

$$A_{r+N/2} = B_r + e^{-j2\pi(r+N/2)/N} C_r$$
$$= B_r - e^{-j2\pi r/N} C_r, \quad 0 \le r < N/2 \qquad (13)$$

Finally, using (3), (11) and (13), we get

$$A_r = B_r + W^r C_r \qquad 0 \le r < N/2 \qquad (14)$$
$$A_{r+N/2} = B_r - W^r C_r$$

A direct calculation of B_r and C_r from (10) requires $(N/2)^2$ complex multiplications each. Another N such multiplications are required to compute A_r's from (14), thus making a total of $2(N/2)^2 + N = N^2/2 + N$, which is less than N^2 if $N > 2$. This is illustrated in Fig. 1 by a signal flow diagram for N=8, where we have used the fact that $W^{N/2} = -1$, so that $-W^r = W^{r+N/2}$.

The DFT's of $\{u_k\}$ and $\{v_k\}$, $k=0,1,\ldots\frac{N}{2}-1$, can now be computed through a similar decomposition if $\frac{N}{2}$ is even; thus the computation of $\{B_r\}$ and $\{C_r\}$ reduces to the task of finding the DFT's of four sequences, each of N/4 samples These reductions can be continued as long as each sequence has an even number of samples. Thus if $N = 2^n$, one can make n such reductions by applying (9) and (14), first for N, then for N/2, and so on, and finally for a two-point function. The DFT of a one-point function is, of course, the sample itself. The successive reduction of an 8-point DFT, which began in Fig. 1 , is continued in Figs. 2 and 3. In Fig. 3, the operation has been completely reduced to complex multiplications and additions. The number of summing nodes is (8)(3)=24 and 24 complex additions are therefore required; the number of complex multiplications needed are also 24 = (no. of rows)(no. of multiplications in each row) = (3)(8). Half of these multiplications are easily eliminated by noting

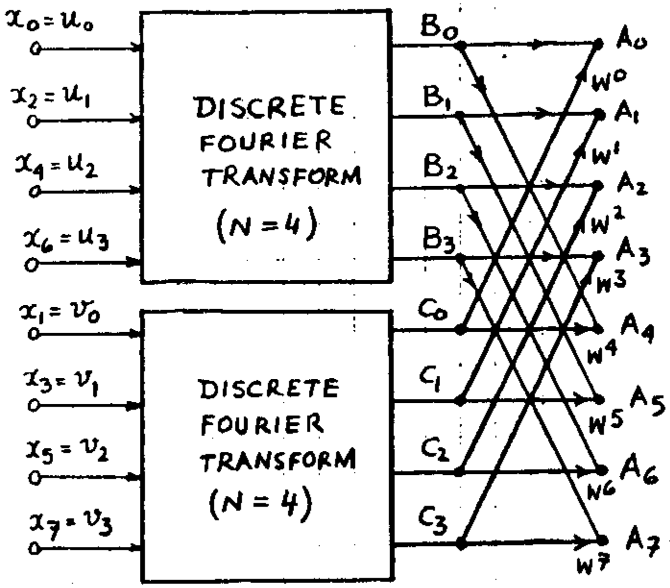

Figure 1

Illustrating the first step in decimation in time form of FFT for N=8.

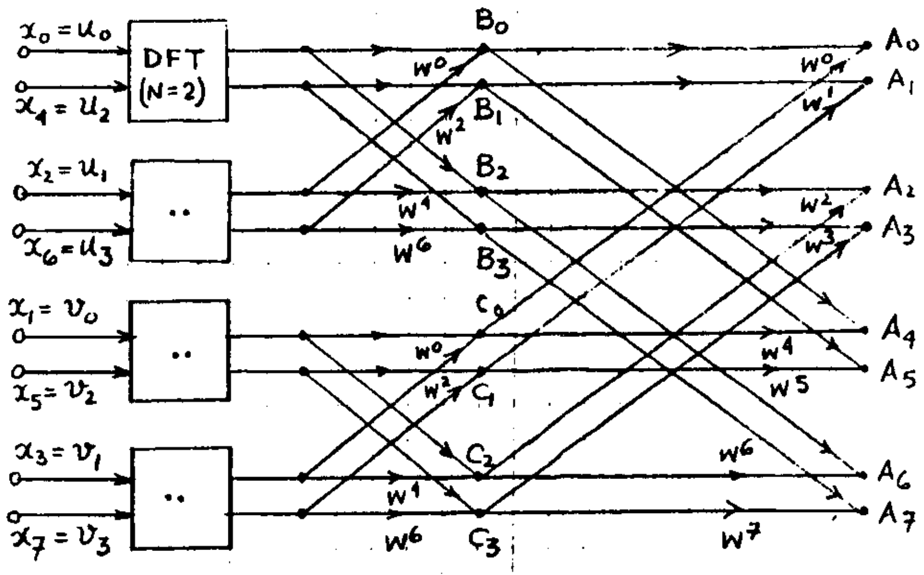

Figure 2

Illustrating two steps of decimation in time form of FFT for N=8

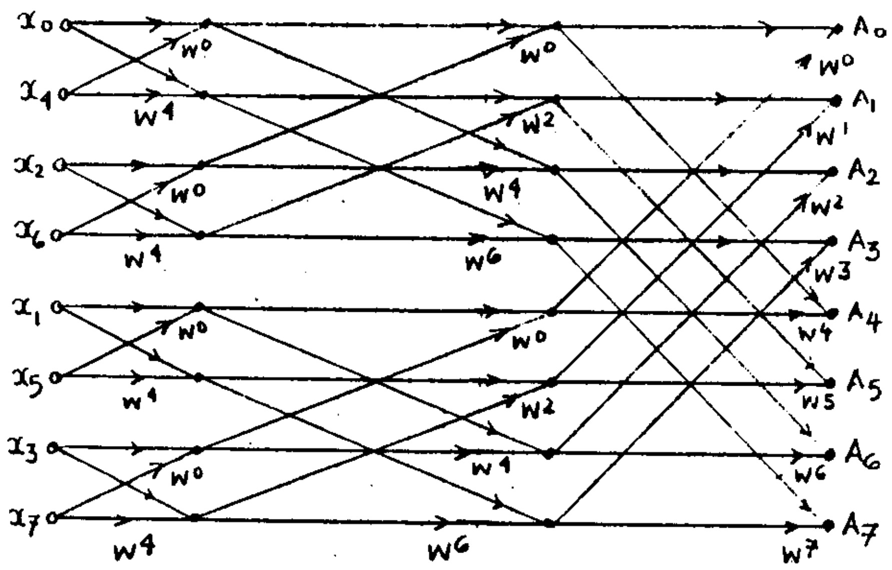

Figure 3

Illustrating decimation in time form of FFT for N = 8.

that $W^7 = -W^3$, $W^6 = -W^2$, $W^5 = -W^1$ and $W^4 = -W^0$. Thus, in general, N $\log_2$ N complex additions and, at most, $\frac{1}{2}$ N $\log_2$ N complex multiplications are required for the computation of an N-point DFT, when N is a power of 2.

When N is not a power of 2, but has a factor of p, one can develop equations analogous to (9) through (14) by forming p different sequences, $\{u_k^{(i)}\}=\{x_{pk+i}\}$, i=0 to p-1, each having N/p samples. For example, if N = 14, having a factor 3, we can form three sequences

$$\{u_k^{(0)}\} = \{x_0, x_3, x_6, x_9, x_{12}\}$$

$$\{u_k^{(1)}\} = \{x_1, x_4, x_7, x_{10}, x_{13}\} \qquad (15)$$

$$\{u_k^{(2)}\} = \{x_2, x_5, x_8, x_{11}, x_{14}\}$$

Each of these sequences has a DFT $B_r^{(i)}$, and the DFT of $\{x_k\}$ can be computed from p simpler DFT's. Further simplification occurs if N has additional prime factors.

In the decimation in frequency form of FFT, the sequence $\{x_k\}$, k=0,1,...,N-1 and N even, is decomposed as

$$u_k = x_k \qquad k=0,1,\ldots,\frac{N}{2}-1 \qquad (16)$$
$$v_k = x_{k+n/2}$$

i.e. $\{u_k\}$ is composed of the first N/2 points and $\{v_k\}$ is composed of the last N/2 points of $\{x_k\}$. Then one can write

$$A_r = \sum_{k=0}^{\frac{N}{2}-1} [u_k e^{-j2\pi rk/N} + v_k e^{-j2\pi r(k+\frac{N}{2})/N}] = \sum_{k=0}^{\frac{N}{2}-1} (u_k + e^{-j\pi r} v_k) e^{-j2\pi rk/N},$$

$$r=0,1,\ldots N-1. \qquad (17)$$

Consider the even-numbered and odd-numbered points of the DFT separately;

let

$$R_r = A_{2r}$$

$$0 \leq r < N/2 \qquad (18)$$

$$S_r = A_{2r+1}$$

If is this step that may be called the decimation in frequency. Note that for computing R_r, (17) becomes

$$R_r = A_{2r} = \sum_{k=0}^{\frac{N}{2}-1} (u_k+v_k)e^{-j2\pi rk/(N/2)} \qquad (19)$$

which we recognize as the N/2 point DFT of the sequence $\{u_k + v_k\}$. Similarly,

$$S_r = A_{2r+1} = \sum_{k=0}^{\frac{N}{2}-1} [u_k+v_k e^{-j\pi(2r+1)}]$$

$$\cdot e^{-j2\pi(2r+1)k/N}$$

$$= \sum_{k=0}^{\frac{N}{2}-1} (u_k-v_k)e^{-j2\pi k/N} e^{-j2\pi rk/(N/2)} \qquad (20)$$

which we recognize as the N/2 point DFT of the sequence $\{(u_k-v_k)e^{-j2\pi k/n}\}$.

Thus, the DFT of an N-sample sequence $\{x_k\}$, N even, can be computed as the N/2 point DFT of a simple combination of the first N/2 and the last N/2 simples of $\{x_k\}$ for even numbered points, and a similar DFT of a different combination of the same samples of $\{x_k\}$ for the odd numbered points. This is illustrated in Fig. 4 for N=8.

As was the case with decimation in time, we can replace each of the DFT's indicated in Fig. 4 by two 2-point DFT's, and each of the 2-point DFT's by two 1-point transforms, these last being equivalency operations. These steps

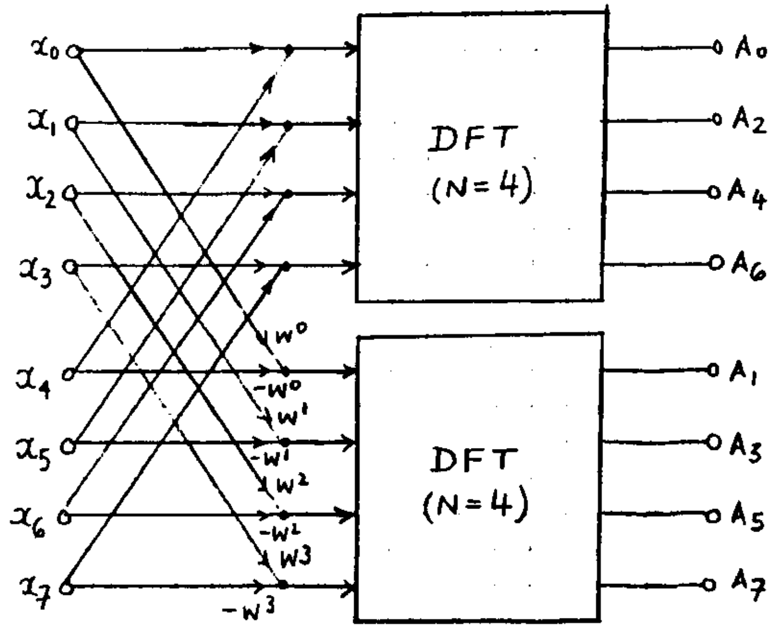

Figure 4

Illustrating the first step in decimation in frequency form of FFT for N=8.

are indicated in Figs. 5 and 6.

There are many variations and modifications of the two basic FFT schemes, which we would not discuss here.

3. APPLICATIONS OF FFT TO COMPUTE CONVOLUTION AND CORRELATION

It may be recalled that our motivation for introducing the DFT and FFT was to convert the convolution relation (1) viz.

$$y_n = \sum_{r=0}^{n} h_r x_{n-r} = \sum_{r=0}^{n} h_{n-r} x_r \qquad (21)$$

into a product form, through the DFT. To this end, assume that both the impulse response $\{h_n\}$, and the input $\{x_n\}$ are bandlimited to $\frac{1}{2T}$ Hz. Then the output $\{y_n\}$ is also frequency bandlimited. Also, if both $\{h_n\}$ and $\{x_n\}$ are defined for the range $0 \leq n \leq N-1$, then $\{y_n\}$ is defined for the range $0 \leq n \leq 2N-1$. For example, if $\{h_n\} = \{h_0, h_1\}$ and $\{x_n\} = \{x_0, x_1\}$, then $\{y_n\} = \{h_0 x_0, h_0 x_1 + h_1 x_0, h_1 x_1\}$. Let the DFT's of $\{x_n\}$ and $\{h_n\}$ be $\{A_r\}$ and $\{H_r\}$ respectively. Then the n-th sample in the IDFT of the product $\{A_r H_r\}$ is

$$y'_n = \frac{1}{N}\sum_{r=0}^{N-1} A_r H_r W^{-rn} \quad , n=0,1,\ldots N-1 \qquad (22)$$

Substituting, in (22),

$$A_r = \sum_{k=0}^{N-1} x_k W^{rk}, \; H_r = \sum_{\ell=0}^{N-1} h_\ell W^{r\ell} \qquad (23)$$

and carrying out some elementary manipulations, it is not difficult to show that (22) simplifies to

$$y'_n = \sum_{k=0}^{N-1} x_k h_{n-k} = \sum_{k=0}^{n} x_k h_{n-k} + \sum_{k=n+1}^{N-1} x_k h_{N+n-k} \qquad (24)$$

$$= y_n + \text{perturbation term} \qquad (25)$$

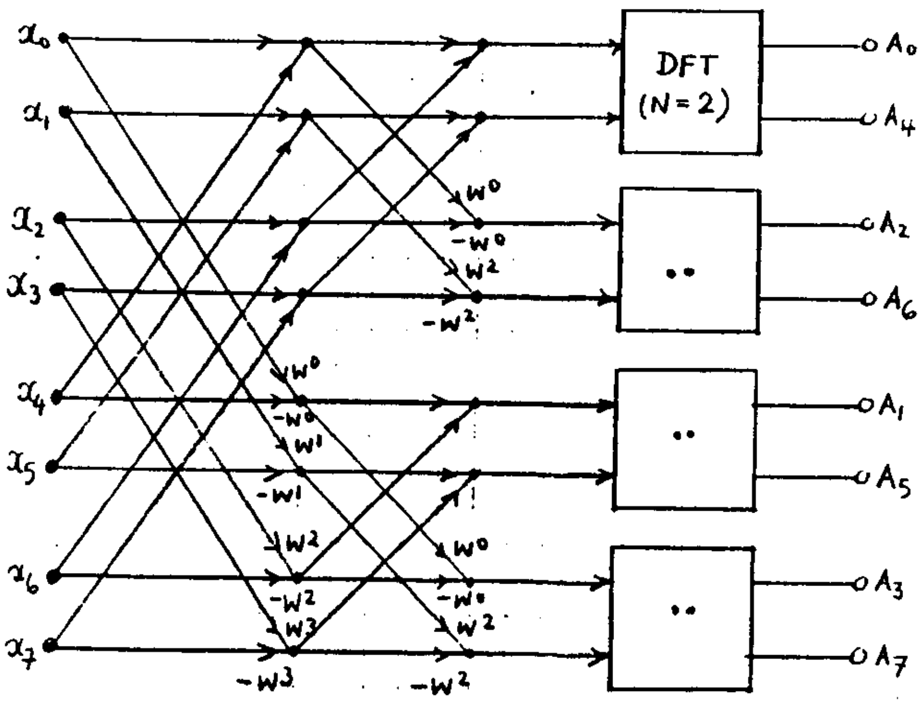

Figure 5

Illustrating two steps of decimation in frequency form of FFT for N=8.

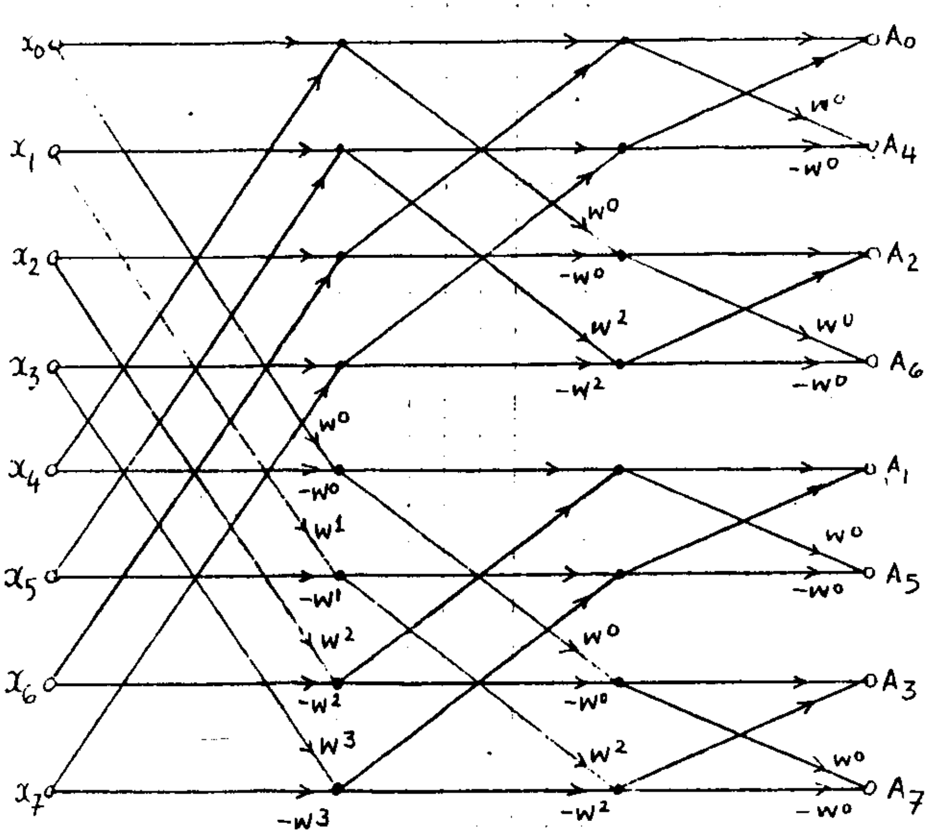

Figure 6

Illustrating decimation in frequency form of FFT for N=8

The last form is obtained by comparison with (21) while the last term in (24) represents the "cyclical" part of the convolution, arising out of the periodicity of DFT and IDFT; h is the cyclical variable passing from h_o to h_{N-1} as k passes from n to n+1. The convolution can be made cyclical in x instead of h by interchanging x and h in (24).

The procedure outlined at the beginning of Section 1 for implementing a digital signal processor viz. taking the DFT's of $\{x_n\}$ and $\{h_n\}$, multiplying them, and taking IDFT of the product, does not, therefore, give the desired output sequence $\{y_n\}$ unless the perturbation term in (24) can be made zero. This term arises due to the fact that the DFT assumes both $\{x_n\}$ and $\{h_n\}$ to be periodic. Further, $\{y_n'\}$ is of length N instead of 2N-1. Note that if we extend both $\{x_n\}$ and $\{h_n\}$ to a length 2N by adding N zeros to each, i.e. if we change $\{x_n\}$ to $\{\hat{x}_n\} = \{x_0, x_1, \ldots x_{N-1}, 0, \ldots 0\}$ and similarly for $\{h_n\}$, then the perturbation term becomes zero. Further, the sequence $\{y_n\}$ will be N + N - 1 = 2N-1 terms long i.e. y_{2N-2} will be the last non-zero term in $\{y_n\}$. As an example, let N=4 i.e.

$$\{x_n\} = \{x_0, x_1, x_2, x_3\}$$
$$\{h_n\} = \{h_0, h_1, h_2, h_3\} \qquad (26)$$

The true convolution of $\{x_n\}$ with $\{h_n\}$ gives

$$y_0 = x_0h_0$$
$$y_1 = x_0h_1+x_1h_0$$
$$y_2 = x_0h_2+x_1h_1+x_2h_0$$
$$y_3 = x_0h_3+x_1h_2+x_2h_1+x_3h_0$$
$$y_4 = x_1h_3+x_2h_2+x_3h_1 \qquad (27)$$
$$y_5 = x_2h_3+x_3h_2$$
$$y_6 = x_3h_3$$

On the other hand, the DFT procedure, leading to (24) gives

$$y'_n = \sum_{k=0}^{3} x_k h_{n-k} \tag{28}$$

so that

$$\begin{aligned} y'_0 &= x_0h_0+x_1h_{-1}+x_2h_{-2}+x_3h_{-3} \\ &= x_0h_0+(x_1h_3+x_2h_2+x_3h_1) \\ y'_1 &= x_0h_1+x_1h_0+(x_2h_3+x_3h_2) \\ y'_2 &= x_0h_2+x_1h_1+x_2h_0+(x_3h_1) \\ y'_3 &= x_0h_3+x_1h_2+x_2h_1+x_3h_0 \end{aligned} \tag{29}$$

where the perturbation terms are bracketed. Also $\{y'_n\}$ consists of only 4 terms. Now let

$$\begin{aligned} \{\hat{x}_n\} &= \{x_0,x_1,x_2,x_3,0,0,0,0\} \\ \{\hat{h}_n\} &= \{h_0,h_1,h_2,h_3,0,0,0,0\} \end{aligned} \tag{30}$$

Then the DFT procedure gives

$$y'_n = \sum_{k=0}^{7} x_k h_{n-k} \tag{31}$$

so that

$$\begin{aligned} y'_0 &= x_0h_0 \\ y'_1 &= x_0h_1+x_1h_0 \\ y'_2 &= x_0h_2+x_1h_1+x_2h_0 \\ y'_3 &= x_0h_3+x_1h_2+x_2h_1+x_3h_0 \\ y'_4 &= x_1h_3+x_2h_2+x_3h_1 \\ y'_5 &= x_2h_3+x_3h_2 \\ y'_6 &= x_3h_3 \end{aligned} \tag{32}$$

By comparing with (27), we see that $\{y'_n\} = \{y_n\}$, $n=0,1,2,\ldots 7$. Thus, the modification does give correct results.

Before stating this simple remedy in formal terms, we would like to emphasize that blind use of FFT for computing the convolution of two sequences will lead to incorrect results, because the DFT introduces a periodic extension of both data and processor impulse response. This results in cyclic or periodic convolution, rather than the desired noncyclic or aperiodic convolution. If $\{x_n\}$ and $\{h_n\}$ contain N samples each, then the true convolution should result in 2N-1 samples for $\{y_n\}$. If DFT is used, then $\{A_r\}$ and $\{H_r\}$ each consist of N samples, so does $\{A_rH_r\}$ and hence its IDFT. Hence $\{y'_n\}$ found by DFT is not the same as $\{y_n\}$ because of folding (or aliasing or cycling) occuring in the time domain. This can be corrected, as demonstrated by the example, by adding zeros to both $\{x_n\}$ and $\{h_n\}$ and thereby increase their lengths sufficiently so that no overlap occurs in the resultant convolution.

We now state formally the steps for computing convolution by DFT:

1. Let $\{x_n\}$ be defined for $0 \leq n \leq M-1$ and $\{h_n\}$ be defined for $0 \leq n \leq P-1$
2. Select N such that

   $N \geq P + M - 1$

   $N = 2^k$
3. Form the new sequences $\{\hat{x}_n\}$ and $\{\hat{h}_n\}$ such that

$$\hat{x}_n = \begin{cases} x_n, & 0 \leq n \leq M-1 \\ 0, & M \leq n \leq N-1 \end{cases}$$

$$\hat{h}_n = \begin{cases} h_n, & 0 \leq n \leq P-1 \\ 0, & P \leq n \leq N-1 \end{cases}$$

4. Compute the DFT's $\{\hat{A}_r\}$ and $\{\hat{H}_r\}$ of $\{\hat{x}_n\}$ and $\{\hat{h}_n\}$ by FFT.

5. Compute

$$\{\hat{B}_r\} = \{\hat{A}_r \hat{H}_r\}$$

6. Find the IDFT of $\{\hat{B}_r\}$ by FFT; the result is $\{y_n\}$.

This technique is referred to as <u>select-saving</u>.

Next, we consider the application of FFT to compute the cross-correlation sequence $\{R_{xy}(k)\}$ of two given sequences $\{x_n\}$ and $\{y_n\}$, each of length N where

$$R_{xy}(k) \triangleq \frac{1}{N} \sum_{n=0}^{N-1} x_n y_{n-k} \tag{33}$$

and the auto-correlation sequence $\{R_{xx}(k)\}$ of a sequence $\{x_n\}$, where

$$R_{xx}(k) \triangleq \frac{1}{N} \sum_{n=0}^{N-1} x_n x_{n-k} \tag{34}$$

Note that the essential difference between convolution, as given by (21) and correlation, as given by (33) and (34) is that one of the sequences is reversed in direction for one operation as compared with the other. Thus, if FFT is to be used to compute correlation, the same kind of precautions, as discussed for convolution, are to be exercised. The procedure, here, is based on the fact that if $DFT\{x_n\} = \{A_r\}$, and $DFT\{y_n\} = \{B_r\}$, then

$$DFT\{\sum_{n=0}^{N-1} x_n y_{n-k}\} = \{A_r \bar{B}_r\} \tag{35}$$

where bar denotes complex conjugate. Thus applied to (33) and (34) one obtains

$$\{R_{xy}(k)\} = IDFT\ \{A_r \bar{B}_r/N\} = IDFT\ \{S_{xy}(r)\} \tag{36}$$

$$\{R_{xx}(k)\} = IDFT\ \{|A_r|^2/N\} = IDFT\ \{S_{xx}(r)\} \tag{37}$$

where $\{S_{xy}(r)\}$ and $\{S_{xx}(r)\}$ are the cross power spectrum sequence and auto power-spectrum sequences respectively.

4. APPLICATION OF FFT TO FIND THE SPECTRUM OF A CONTINUOUS SIGNAL

The DFT, as we have seen, is specifically concerned with the analysis and processing of discrete periodic signals, and that it is a zero-order approximation of the continuous Fourier transform. It is therefore tempting to apply the DFT directly to privide, through FFT, a numerical spectral analysis of sampled versions of continuous signals. This would be a perfectly valid application, if the continuous signal is periodic, band-limited and sampled in accordance with the sampling theorem. Deviations from these cause errors, and most of the problems in using the DFT to approximate the CFT (C for continuous) are caused by a misunderstanding of what this approximation involves.

There are, essentially, three phenomena, which contribute to errors in relating the DFT to the CFT. The first, called <u>aliasing</u>, occurs due to inadequate sampling rate. The solution to this problem is to ensure that the sampling rate is high enough to avoid any spectral overlap. This requires some prior knowledge of the nature of the spectrum, so that the appropriate sampling rate may be chosen. In absence of such prior knowledge, the signal must be prefiltered to ensure that no components higher than the folding frequency appear.

The second problem is that of <u>leakage</u>, arising due to the practical requirement of observing the signal over a finite interval. This is equivalent to multiplying the signal by a window function. The simplest window is a rectangular function as shown in Fig. 7 (b), and its effect on the spectrum of a sine signal, shown in Fig. 7 (a) is displayed in Fig. 7 (c). Note that there occurs a spreading or leakage of the spectral components away from the correct frequency; this results in an undesirable modification of the total spectrum.

The leakage effect cannot always be isolated from the aliasing effect because leakage may also lead to aliasing if the highest frequency of the

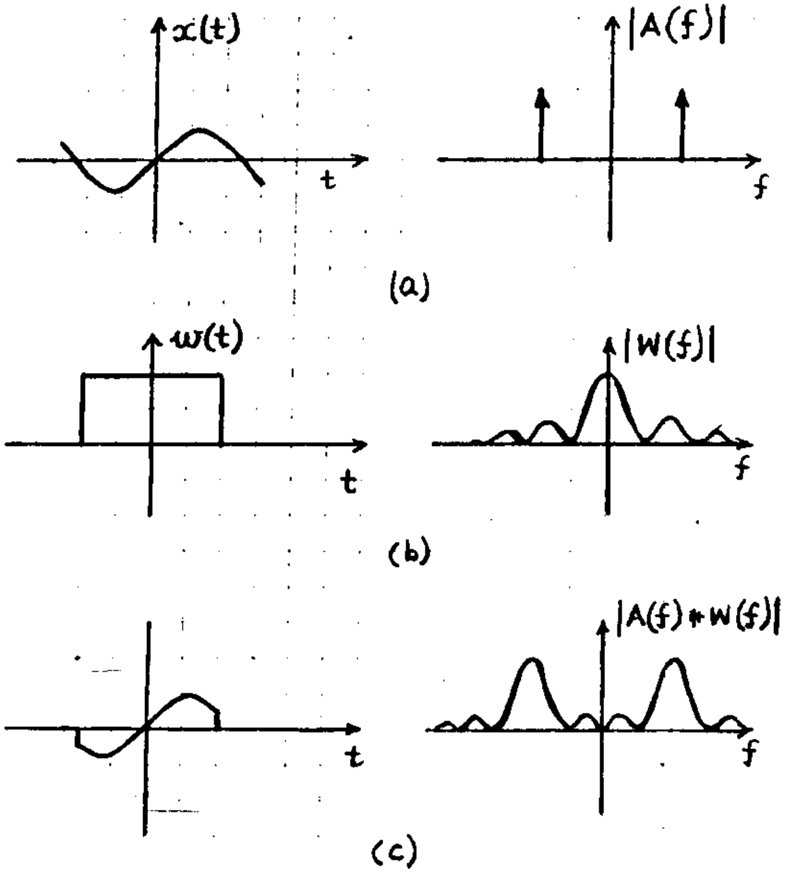

Figure 7

Illustrating 'leakage' due to finite observation time

composite spectrum moves beyond the folding frequency. This possibility is particularly significant in the case of a rectangular window, because the tail of the window spectrum does not converge rapidly.

The solution to the leakage problem is to choose a window function that minimizes the spreading. One example is the so-called "raised cosine" window in which a raised cosine wave is applied to the first and last 10 percent of the data and a weight of unity is applied in between. Since only 20 percent of the terms in the time series are given a weight other than unity, the computation required to apply this window in the time domain is relatively small, as compared to other continuously varying weight windows e.g. the Hamming window.

The third problem in relating the DFT to the CFT is the <u>picket-fence effect</u>, resulting from the inability of the DFT to observe the spectrum as a continuous function, since the computation of the spectrum is limited to integer multiples of the fundamental frequency $f_o = 1/(NT)$. In a sense, the observation of the spectrum with the DFT is analogous to looking at it through a sort of "picket-fence" since we can observe the exact behavior only at discrete points. It is possible that a major peak lies between two of the discrete transform lines, and this will go undetected without some additional processing.

One procedure for reducing the picket-fence effect is to vary the number of points N in a time period by adding zeros at the end of the original record, while maintaining the original record intact. This process artificially changes the period, which, in turn, changes the locations of the spectral lines without altering the continuous form of the original spectrum. In this manner, spectral components originally hidden from view may be shifted to points where they may be observed.

REFERENCES

[1] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series", Mathematics of Computation, vol. 19, pp. 297-301, April 1965.

[2] W. M. Gentleman and G. Sande, "Fast Fourier transforms - for fun and profit", 1966 Fall Joint Computer Conference, pp. 563-578.