APPLICATIONS OF FOURIER AND BIFORE ANALYSES

TO DISCRETE SIGNALS

by

4891

PEDRO E. MARTINEZ

B. S., Universidad Industrial de Santander, 1967

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1970

Approved by:

Nasir Ahmad
Major Professor

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

1.1 With the increased use of sampled-data systems, the discrete Fourier transform and the BIFORE (Binary Fourier Representation) transform have found a variety of applications which include: (1) image coding [12, 14], (2) spectral analysis of digital systems [3], (3) signal representation and classification [17], and, (4) speech processing [11, 13].

This report has three main objectives: (1) to acquaint the reader with brief introductions to the discrete Fourier and BIFORE transforms, (2) to illustrate some applications of these transforms by means of numerical examples, and, (3) to provide a documentation of the digital computer programs used in (2).

In Chapter II, the discrete Fourier transform is introduced. The corresponding algorithm which enables rapid evaluation of the discrete Fourier transform is discussed in Chapter III. This algorithm is the so-called fast Fourier transform. Applications of the fast Fourier transform for rapid evaluation of the Fourier power and phase spectra, crosscorrelation and convolution are also included.

Chapter IV introduces the BIFORE transform with its associated power and phase spectra. A computer program which facilitates rapid computation of the BIFORE transform and spectra is included. In conclusion, some aspects of the relation between the BIFORE and discrete Fourier spectra are considered.

CHAPTER II

THE DISCRETE FOURIER TRANSFORM

2.1 Definition of the Discrete Fourier Transform (DFT)

Let $\{X(m)\}$ denote a sequence $X(m)$, $m = 0, 1, \ldots, (N-1)$ of N finite valued real or complex numbers. The discrete or finite Fourier transform is defined as

$$C_x(k) = \frac{1}{N} \sum_{m=0}^{N-1} X(m)W^{km}, \quad k = 0, 1, \ldots (N-1)$$

where $W = e^{-i\frac{2\pi}{N}}$ and $i = \sqrt{-1}$. (2 - 1)

The exponential functions W^{km} in (2-1) are orthogonal; that is,

$$\sum_{m=0}^{N-1} W^{km}W^{-\ell m} = \begin{cases} N & \text{if } \frac{k-\ell}{N} \text{ is an integer} \\ 0 & \text{otherwise.} \end{cases} \qquad (2 - 2)$$

Thus, using (2-1) and (2-2), $\{X(m)\}$ can be expressed as the inverse discrete Fourier transform (IDFT)

$$X(m) = \sum_{k=0}^{N-1} C_x(k)\, W^{-km}, \quad m = 0, 1, \ldots, (N-1). \qquad (2 - 3)$$

2.2 Some Properties of the DFT

1. Since the exponential functions W^{km} are N - periodic, it follows that the sequences $\{C_x(k)\}$ and $\{X(m)\}$ in (2-1) and (2-3), respectively, are also N - periodic. That is, the sequences $\{X(m)\}$ and $\{C_x(k)\}$ satisfy the following conditions:

$$X(\pm m) = X(sN \pm m)$$

$$C_x(\pm k) = C_x(sN \pm k) \qquad s = 0, \pm 1, \pm 2, \ldots \qquad (2 - 4)$$

2. If the sequence $\{X(m)\}$ in (2-1) is real, then the DFT coefficients $C_x(k)$ in (2-3) are such that

$$C_x(N/2 + \ell) = \overline{C}_x(N/2 - \ell), \quad \ell = 1, 2, \ldots, (N/2 - 1) \qquad (2 - 5)$$

where $\overline{C}_x(k)$ is the complex conjugate of $C_x(k)$, $k = 1, 2, \ldots, (N - 1)$.

3. $C_x(0)$ and $C_x(N/2)$ are real if the input sequence $\{X(m)\}$ is real. This follows directly from (2-1).

2.3 Computation of the DFT Coefficients Using Matrix Multiplication

For the purpose of discussion, consider the case $N = 8$. Then (2-1) yields the matrix equation

$$\begin{bmatrix} C_x(0) \\ C_x(1) \\ C_x(2) \\ C_x(3) \\ C_x(4) \\ C_x(5) \\ C_x(6) \\ C_x(7) \end{bmatrix} = \frac{1}{N} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & W & W^2 & W^3 & W^4 & W^5 & W^6 & W^7 \\ 1 & W^2 & W^4 & W^6 & W^8 & W^{10} & W^{12} & W^{14} \\ 1 & W^3 & W^6 & W^9 & W^{12} & W^{15} & W^{18} & W^{21} \\ 1 & W^4 & W^8 & W^{12} & W^{16} & W^{20} & W^{24} & W^{28} \\ 1 & W^5 & W^{10} & W^{15} & W^{20} & W^{25} & W^{30} & W^{35} \\ 1 & W^6 & W^{12} & W^{18} & W^{24} & W^{30} & W^{36} & W^{42} \\ 1 & W^7 & W^{14} & W^{21} & W^{28} & W^{35} & W^{42} & W^{49} \end{bmatrix} \begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \\ X(6) \\ X(7) \end{bmatrix}. \qquad (2 - 6)$$

In particular, consider the case when $\{X(m)\}$ is real. Then, using the fact that the function W is N - periodic and applying (2-5) to (2-6), there results

$$\begin{bmatrix} C_x(0) \\ C_x(1) \\ C_x(2) \\ C_x(3) \\ C_x(4) \end{bmatrix} = \frac{1}{8} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & W^1 & W^2 & W^3 & W^4 & W^5 & W^6 & W^7 \\ 1 & W^2 & W^4 & W^6 & W^0 & W^2 & W^4 & W^6 \\ 1 & W^3 & W^6 & W^1 & W^4 & W^7 & W^2 & W^5 \\ 1 & W^4 & W^0 & W^4 & W^0 & W^4 & W^0 & W^4 \end{bmatrix} \begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \\ X(6) \\ X(7) \end{bmatrix} \qquad (2-7)$$

where

$$W = e^{-i\frac{2\pi}{8}} = \frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2}$$

$$W^2 = -i$$

$$W^3 = -\frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2}$$

$$W^4 = -1$$

$$W^5 = -\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}$$

$$W^6 = +i$$

and

$$W^7 = \frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}.$$

In sequel, a numerical example is considered.

Example 2-1. Given the data sequence $X(0) = 1$, $X(1) = 2$, $X(2) = 1$, $X(3) = 1$, $X(4) = 3$, $X(5) = 2$, $X(6) = 1$, $X(7) = 2$, compute the DFT coefficients $\{C_x(k)\}$ by evaluating (2-7). From (2-7) it follows that

$$C_x(0) = (1/8)\left[X(0) + X(1) + X(2) + X(3) + X(4) + X(5) + X(6) + X(7)\right].$$

$C_x(0) = (1/8)(1 + 2 + 1 + 1 + 3 + 2 + 1 + 2) = 13/8$

$C_x(1) = (1/8)\left[X(0) + X(1)W + X(2)W^2 + X(3)W^3 + X(4)W^4 + X(5)W^5 + X(6)W^6 + X(7)W^7\right]$

$C_x(1) = (1/8)\left[1 + 2\frac{\sqrt{2}}{2} - i\ 2\frac{\sqrt{2}}{2} - i - \frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2} + 3(-1) - 2\frac{\sqrt{2}}{2} + i\ 2\frac{\sqrt{2}}{2}\right.$

$\left. + i + 2\frac{\sqrt{2}}{2} + i\ 2\frac{\sqrt{2}}{2}\right]$

$C_x(1) = \frac{-1.293 + i\ 0.707}{8}$

$C_x(2) = (1/8)\left[X(0) + X(1)W^2 + X(2)W^4 + X(3)W^6 + X(4) + X(5)W^2 + X(6)W^4 + X(7)W^6\right]$

$C_x(2) = (1/8)\left[4 + 4(-i) + 2(-1) + 3i\right) = (1/8)(2-i)$

$C_x(3) = (1/8)\left[X(0) + X(1)W^3 + X(2)W^6 + X(3)W + X(4)W^4 + X(5)W^7 + X(6)W^2 + X(7)W^5\right]$

$C_x(3) = (1/8)\left[1 + 2(-\frac{\sqrt{2}}{2}) - i\ 2\frac{\sqrt{2}}{2} + 1i + \frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2} + 3(-1) + 2\frac{\sqrt{2}}{2} + i\ 2\frac{\sqrt{2}}{2} - i\right.$

$\left. - \sqrt{2} + i\sqrt{2}\right]$

$C_x(3) = (1/8)(-2.707 + i\ 0.707)$

$C_x(4) = (1/8)\ X(0) + X(1)W^4 + X(2)W^0 + X(3)W^4 + X(4)W^0 + X(5)W^4 + X(6)W^0$

$+ X(7)W^4$

$C_x(4) = (1/8)\left[1 + 1 + 3 + 1 - (2 + 1 + 2 + 2)\right] = -1/8$

The above $C_x(k)$, $k = 0, 1, \ldots, 4$ yield the remaining coefficients by virtue of (2-5) as follows:

$C_x(5) = \bar{C}_x(3) = (1/8)(-2.707 - i\ 0.707)$

$C_x(6) = \bar{C}_x(2) = (1/8)(2 + i)$

$C_x(7) = C_x(1) = (1/8)(-1.293 - i\ 0.707)$

Thus, in summary, one has

$C_x(0) = \frac{13}{8}$

$C_x(1) = \frac{-1.293 + i\ 0.707}{8}$

$C_x(2) = \frac{2 - i}{8}$

$$C_x(3) = \frac{-2.707 + i\,0.707}{8}$$

$$C_x(4) = -1/8$$

$$C_x(5) = \frac{-2.707 - i\,0.707}{8}$$

$$C_x(6) = \frac{2 + i}{8}$$

$$C_x(7) = -\frac{1.293 - i\,0.707}{8}.$$

The above example leads to the following observations and subsequent generalizations:

1. The total number of arithmetic operations $\sum_1$ required to compute the $C_x(k)$, $k = 0, 1, \ldots \frac{N}{2}$ is the sum of $\frac{N^2}{2}$ multiplications and $(N - 1)\frac{N}{2}$ additions. That is,

$$\sum{}_1 = \frac{N^2}{2} + \frac{N}{2}(N - 1) \simeq N^2, \text{ for sufficiently large N.} \qquad (2 - 8)$$

Thus, if the input is a real N - periodic sequence, the direct matrix multiplication approach requires approximately N^2 arithmetic operations to obtain the DFT coefficients $C_x(k)$, $k = 0, 1, \ldots, (N - 1)$.

2. The general version of (2-7) will consist of an array of $(\frac{N}{2} \times N)$ elements. Consequently the number of storage locations $\hat{\sum}_1$ required if $\{X(m)\}$ is a <u>real</u> N - periodic sequence is such that

$$\hat{\sum}{}_1 \simeq \frac{N^2}{2} \qquad (2 - 9)$$

3. If the input sequence $\{X(m)\}$ is complex, then the complex conjugate property in (2-5) is not valid. Consequently, for complex inputs it follows that

$$\sum{}_1 \simeq 2N^2$$

and

$$\hat{\sum}{}_1 \simeq N^2 \qquad (2 - 10)$$

2.4 Spectrum Considerations

<u>DFT Power Spectrum</u>

Given a data sequence $X(m)$, $m = 0, 1, \ldots (N-1)$, its DFT power spectrum is defined as

$$\left| C_x(k) \right|^2, \qquad k = 0, 1, \ldots (N-1). \qquad (2-11)$$

In particular, if $\{X(m)\}$ is real, then only $(N/2 + 1)$ of the spectrum points in (2-11) are independent, due to the complex conjugate property in (2-5). The independent spectrum points are:

$$\left| C_x(k) \right|^2, \quad k = 0, 1 \ldots N/2. \qquad (2-12)$$

It can be shown that the power spectrum is invariant with respect to shifts of the N - periodic data sequence $\{X(m)\}$.

<u>Phase Spectrum</u>

Given a data sequence $X(m)$, $m = 0, 1, \ldots (N-1)$, the phase spectrum is defined as

$$\psi_x(k) = \tan^{-1}\left\{\frac{B_x(k)}{A_x(k)}\right\}, \; k = 0, 1, \ldots (N-1) \qquad (2-13)$$

where $A_x(k)$ and $B_x(k)$ are respectively the real and imaginary parts of $C_x(k)$.

If the sequence $\{X(m)\}$ is real, then applying (2-5) to (2-13) it can be shown that

$$\psi_x(N/2 + \ell) = -\psi_x(N/2 - \ell), \qquad \ell = 1, 2, \ldots (N/2 - 1) \qquad (2-14)$$

That is, $\psi_x(k)$ in (2-14) is an odd function about $k = N/2$.

A fundamental property of the phase spectrum is that it is invariant with respect to multiplying the sequence $X(m)$, $m = 0, 1, \ldots (N-1)$ by a constant.

CHAPTER III

THE FAST FOURIER TRANSFORM

3.1 Definition of the Fast Fourier Transform (FFT)

The fast Fourier transform is an algorithm which enables rapid computation of

$$C_x(k) = \frac{1}{N}\sum_{m=0}^{N-1} X(m)W^{km} \qquad k = 0, 1, \ldots (N-1) \tag{3 - 1}$$

where $C_x(k)$, $k = 0, 1, \ldots (N-1)$ are the DFT coefficients of $\{X(m)\}$.

3.2 Illustration of the FFT

The algorithm is illustrated for the case N = 8 by the signal flow graph in Fig. (3-1) where

$$A_{2^r} = e^{-i2\pi/2^r} \qquad r = 1, 2, 3. \tag{3 - 2}$$

Inspection of Fig. (3-1) results in the following observations and generalizations:

1. The maximum value of the iteration index r is given by $n = \log_2 N$
2. In the r^{th} iteration, $r = 1, 2, \ldots \log_2 N$, the multipliers $A^s_{2^r}$ are given by

$$A^s_{2^r}, \qquad s = 0, 1, \ldots (2^{r-1}-1) \tag{3 - 3}$$

r = 1 implies that A^0_2 is needed in iteration number 1. Since $A^0_2 = 1$, only additions and substractions are required in this iteration.

r = 2 implies that A^0_4 and A^1_4 are needed in iteration number 2.

r = 3 implies that A^0_8, A^1_8, A^2_8, A^3_8 are needed in iteration number 3, and so on.

3. The r^{th} iteration results in 2^{r-1} groups with $\frac{N}{2^{r-1}}$ members in each

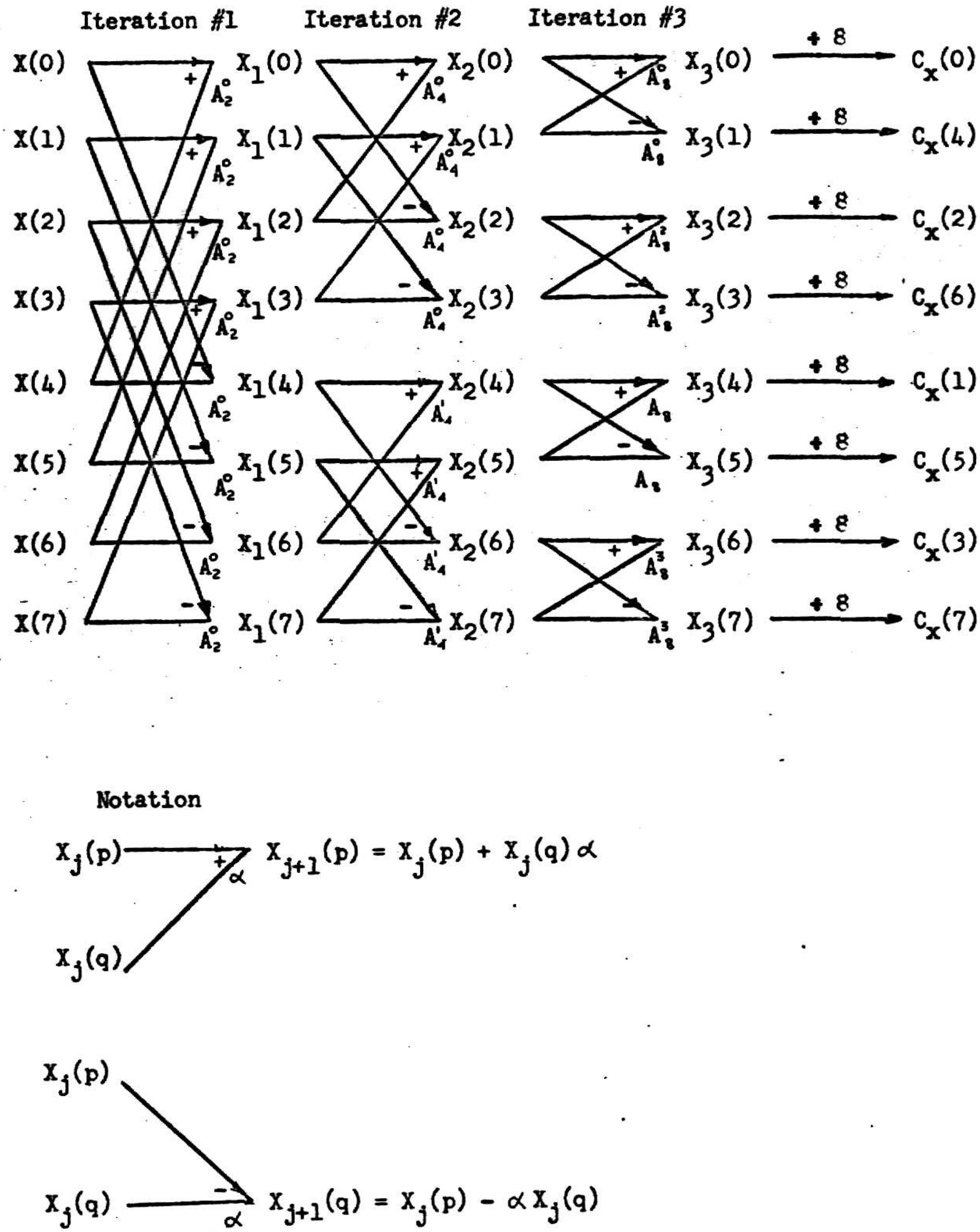


Fig. 3-1. FFT Signal Flow Graph, N = 8

group. Each group takes one multiplier of the form $A_2^s r$ in(3-3). Half the members of each group are associated with $A_2^s r$ while the remaining half is associated with $-A_2^s r$.

4. The first member $X_r(\ )$ of each group to which a multiplier is assigned in the r^{th} iteration can be obtained as shown in Table (3-1) which is the case for N = 8.

Table (3-1)

	$s = s(k_2, k_1, k_0)$			(Bit Reversal)			First member of group which takes $A_2^s r$
s	k_2	k_1	k_0	k_0	k_1	k_2	$X_r(k) = X_r(k_0, k_1, k_2)$
0	0	0	0	0	0	0	$X_r(0)$
1	0	0	1	1	0	0	$X_r(4)$
2	0	1	0	0	1	0	$X_r(2)$
3	0	1	1	1	1	0	$X_r(6)$

5. The $C_x(k)$ corresponding to the last iteration output $X_3(\ )$, are obtained as follows:

a. Express the sequence 1 = 0, 1, 2,..., 7 in binary form to obtain 000, 001, 010, 011, 100, 101, 110, 111.

b. Reverse each of the 3 - bit strings in (a) to obtain the sequence 000, 100, 010, 110, 001, 101, 011, 111.

c. Write the binary sequence in (b) in decimal form to obtain 0, 4, 2, 6, 1, 5, 3, 7. Then there is a one-to-one correspondence between the sequence k = 0, 4, 2, 6, 1, 5, 3, 7 and the sequence 1 = 0, 1, 2, 3, 4, 5, 6, 7.

6. The algorithm is valid regardless of whether the data $X(m)$, $m = 0, 1, \ldots (N-1)$ is real or complex. Thus it can also be used to compute the IDFT in (2-3) with the following minor modifications which follow from (1-3) and (3-1):

a. Replace W by $\overline{W}$.

b. Do not divide the output of the last iteration by N.

This version of the algorithm will be referred to as the inverse fast Fourier transform (IFFT).

3.3 Advantages of the FFT

There are two principal advantages of the FFT with respect to the matrix multiplication method considered in section 2.3. These are discussed in what follows.

1. <u>Saving in Execution Time</u>. Examination of Fig. (3-1) shows that the FFT involves $\log_2 N$ iterations and each iteration requires at most a total of N additions and subtractions in addition to N multiplications. This results in a total of 2N real arithmetic operations per iteration. Since there are $\log_2 N$ iterations, the algorithm requires a total of

$$\Lambda_1 \simeq 2N\log_2 N \qquad (3-4)$$

arithmetic operations to compute the N coefficients $C_x(k)$, $k = 0, 1, \ldots, (N-1)$ in (3-1).

It is recalled that the matrix multiplication approach requires [see (2-8)]

$$\sum_1 \simeq N^2$$

arithmetic operations to compute the DFT coefficients. Thus from (2-8) and (3-4) it follows that the FFT requires far less execution time as N increases.

2. Saving in Storage Locations. Using direct matrix multiplication, the number of storage locations required is given by [see (2-9)]

$$\hat{\Sigma}_1 \simeq \frac{N^2}{2}.$$

In contrast the number of storage locations required by the FFT is given by

$$\hat{\Lambda}_1 \simeq N \qquad (3-5)$$

since the output of one iteration becomes the input to the next as evident from the signal flow graph in Fig. (3-1). A comparison of (2-9) and (3-5) leads to the conclusion that FFT storage requirements are far less severe as N increases.

3.4 Applications of the FFT

Due to the advantages cited in Section 3.3, the FFT has a variety of applications which include: (1) Computation of DFT spectra; (2) Computation of autocorrelation and crosscorrelation functions and related power spectra; (3) Waveform synthesis; (4) Analysis of linear time-invariant systems; and (5) Image classification using the two-dimensional version of the DFT. In this section, some of these applications are illustrated by means of numerical examples.

1. Computation of DFT Spectra

Example 3-1

Let

$$X(t) = e^{-t}, \qquad 0 \leq t \leq 4 \text{ sec.} \qquad (3-6)$$

Consider 32 samples of X(t) in (3-6).

Compute

a. The DFT coefficients $C_x(k)$, k = 0, 1,..., 31

b. Print the output of each of the 5 ($= \log_2 32$) iterations.

c. Compute and plot the DFT power spectrum

$$\left| C_x(k) \right|^2 = \frac{|C_x(k)|^2}{\max\left\{|C_x(k)|^2\right\}}, \quad k = 0, 1, \ldots, 31. \qquad (3 - 7)$$

d. Compute and plot the DFT phase spectrum

$\psi_x(k)$, $k = 0, 1, \ldots, 31$ defined in (2-13).

Solution: The "Fast Fourier Transform" program deck is used as follows:

i. In line 0012 use II = 0 for FFT or II = 1 for IFFT

ii. Punch NUM = 32 (i.e. the number of data points, N) on the first data card

iii. Follow data card in (ii) by the N data points (one per card), X(0), X(1),..., X(31) which are the values of X(t) in (3-6) with the first sample at t = 0.

The program listing and the output are included in what follows. The normalized power spectrum and the phase spectrum are shown in Fig. (3-2). (see page 26).

```
                C
                C      FFT   -  FAST FOURIER TRANSFORM
                C
0001                   DIMENSION IPOWER(10)
0002                   COMPLEX*8 X(32)
0003                   COMPLEX Y(32),ALPHA,ALPH,CMPLX
                C
                C      THIS ROUTINE CALCULATES THE FAST FOURIER
                C      TRANSFORM FOR ANY GIVEN NUMBER  WHICH
                C      IS A POWER OF TWO
                C      NUM  NUMBER OF POINTS
                C
0004               198 READ(1,100) NUM
0005               100 FORMAT (I5)
                C
                C      INPUT THE NUMBER OF DESIRED DATA POINTS.
                C
                C
0006                   READ(1,250) (X(I),I=1,NUM)
0007               250 FORMAT(2F10.7)
                C
                C
0008                   WRITE(3,103)
0009               103 FORMAT (1H0,'ECHO CHECK OF INPUT VALUES'//)
0010                   WRITE (3,1010) (X(I),I=1,NUM)
0011              1010 FORMAT (10X,2F10.5)
                C
                C
                C      CALCULATE NUMBER OF ITERATIONS.
                C
0012                   II=0
0013               65  ITER=0
0014                   IREM=NUM
0015                 1 IREM=IREM/2
0016                   IF (IREM.EQ.0) GO TO 2
0017                   ITER=ITER+1
0018                   GO TO 1
0019                 2 CONTINUE
                C
                C      BEGIN A LOOP FOR (LOG TO BASE TWO OF NUM) ITERATIONS.
                C
0020                   PI=3.141593
0021                   DO 50 M = 1,ITER
                C
                C      CALCULATE NUMBER OF PARTITIONS AND THE VALUE OF
                C      ALPHA FOR EACHITERATION
                C
0022                   IF (M.EQ.1) NUMP = 1
0023                   IF (M.NE.1) NUMP = NUMP * 2
0024                   MNUM =NUM/NUMP
0025                   MNUM2 = MNUM/2
0026                   IF (II.EQ.0) GO TO 52
0027                   ALPHA = CMPLX(COS(PI/NUMP),(SIN(PI/NUMP)))
0028                   GO TO 53
0029                52 ALPHA = CMPLX(COS(PI/NUMP),(SIN(PI/NUMP))*(-1.0))
0030                53 CONTINUE
                C
                C      BEGIN A LOOP FOR THE NUMBER OF PARTITIONS.
```

```
      C
            DO 49 MP = 1,NUMP
            IB = (MP-1) * MNUM
      C
      C     FIND THE POWER OF ALPHA
      C
            IBC = IB
            IL = 1
          4 IBD = IBC/2
            IPOWER(IL) = 1
            IF (IBC.EQ.(IBD*2)) IPOWER(IL) = 0
            IF (IBD.EQ.0) GO TO 5
            IBC = IBD
            IL = IL + 1
            GO TO 4
          5 CONTINUE
            IP = 0
            IFAC = NUM
            DO 6 I = 1,IL
            IFAC = IFAC/2
          6 IP = IP + IPOWER(I) * IFAC
            ALPH = ALPHA**(IP)
      C
      C     BEGIN A LOOP THROUGH THIS PARTITION.
      C
            DO 48 MP2 = 1,MNUM2
            MNUM21 = MNUM2 + MP2 + IB
            IBA = IB + MP2
            Y(IBA) = X(IBA) + ALPH * X(MNUM21)
            Y(MNUM21) = X(IBA) - ALPH * X(MNUM21)
         48 CONTINUE
         49 CONTINUE
      C
      C     BEGIN A ROUTINE TO OUTPUT ITERATION M.
      C
            DO 7 I = 1,NUM
          7 X(I) = Y(I)
            WRITE(3,104)M
        104 FORMAT(1H1,'STATE OF THE DATA AFTER ITERATION',I3,//)
            DO 8 I=1,NUM
          8 WRITE (3,102)X(I)
        102 FORMAT (1H,2F15.5)
         50 CONTINUE
      C
      C     BEGIN A ROUTINE TO CALCULATE THE C(K)
      C
            DO 11 I = 1,NUM
            IB = I - 1
            IL = 1
          9 IBD = IB/2
            IPOWER(IL) = 1
            IF (IB.EQ.(IBD*2)) IPOWER(IL) = 0
            IF (IBD.EQ.0) GO TO 10
            IB = IBD
            IL = IL + 1
            GO TO 9
         10 CONTINUE
            IP = 1
```

```
              IFAC = NUM
              DO 12 I1 = 1,IL
              IFAC = IFAC/2
           12 IP = IP + IFAC * IPOWER(I1)
           11 Y(IP) = X(I)
              WRITE(3,105)NUM
          105 FORMAT (1H1,'FINAL VALUES OF THE C(K) FOR ',I3,
             1'DATA POINTS.',//)
              DO 13 I = 1,NUM
           13 WRITE (3,102) Y(I)
              IF(II.EQ.1) GO TO 999
              WRITE (3,202)
          202 FORMAT (1H1,4X,'AMPLITUDE',6X,'PHASE',//)
              YMAX2 = 0.0
              DO 56 I = 1,NUM
              YMAX = CABS(Y(I))
           56 IF (YMAX.GT.YMAX2) YMAX2 = YMAX
              YMAX = YMAX2
              DO 57 I = 1,NUM
              X(I) = Y(I)
              A=REAL(Y(I))
              B=AIMAG(Y(I))
              OPTION=0.0
              CALL PHASE (A,B,THETA,OPTION)
              THETA=THETA*57.29578
              X2=CABS(Y(I))/YMAX
           57 WRITE(3,102)X2,THETA
              II = II + 1
              IF  (II.EQ.1) GO TO 65
          999 STOP
              END
```

```
      SUBROUTINE PHASE(X,Y,THETA,OPTION)
      PI=3.141593
      IF(X)1,2,3
    1 THETA=SIGN(PI,Y)+ATAN(Y/X)
      GO TO 4
    2 THETA=SIGN(PI/2.0,Y)
      GO TO 4
    3 THETA=ATAN(Y/X)
    4 IF(OPTION)7,7,5
    5 IF(THETA)7,7,6
    6 THETA=THETA-2.0*PI
    7 RETURN
      END
```

ECHO CHECK OF INPUT VALUES

1.00000	0.0
0.88000	0.0
0.77880	0.0
0.69300	0.0
0.60650	0.0
0.53000	0.0
0.47230	0.0
0.41600	0.0
0.36780	0.0
0.32600	0.0
0.28650	0.0
0.25150	0.0
0.22310	0.0
0.19790	0.0
0.17370	0.0
0.15250	0.0
0.13530	0.0
0.12000	0.0
0.10540	0.0
0.09348	0.0
0.08200	0.0
0.07200	0.0
0.06390	0.0
0.05610	0.0
0.04970	0.0
0.04300	0.0
0.03870	0.0
0.03600	0.0
0.03020	0.0
0.02600	0.0
0.02300	0.0
0.02000	0.0

__Remark.__ These are the values of $\{X(m)\}$.

1.13530	0.0
1.00000	0.0
0.88420	0.0
0.78348	0.0
0.68850	0.0
0.60200	0.0
0.53620	0.0
0.47210	0.0
0.41750	0.0
0.36900	0.0
0.32520	0.0
0.28750	0.0
0.25330	0.0
0.22390	0.0
0.19670	0.0
0.17250	0.0
0.86470	0.0
0.76000	0.0
0.67340	0.0
0.59652	0.0
0.52450	0.0
0.45800	0.0
0.40840	0.0
0.35990	0.0
0.31810	0.0
0.28300	0.0
0.24780	0.0
0.21550	0.0
0.19290	0.0
0.17190	0.0
0.15070	0.0
0.13250	0.0

STATE OF THE DATA AFTER ITERATION 2

1.55280	0.0
1.36900	0.0
1.20940	0.0
1.07098	0.0
0.94180	0.0
0.82590	0.0
0.73290	0.0
0.64460	0.0
0.71780	0.0
0.63100	0.0
0.55900	0.0
0.49598	0.0
0.43520	0.0
0.37810	0.0
0.33950	0.0
0.29960	0.0
0.86470	-0.31810
0.76000	-0.28300
0.67340	-0.24780
0.59652	-0.21550
0.52450	-0.19290
0.45800	-0.17190
0.40840	-0.15070
0.35990	-0.13250
0.86470	0.31810
0.76000	0.28300
0.67340	0.24780
0.59652	0.21550
0.52450	0.19290
0.45800	0.17190
0.40840	0.15070
0.35990	0.13250

2.49460	0.0
2.19490	0.0
1.94230	0.0
1.71558	0.0
0.61100	0.0
0.54310	0.0
0.47650	0.0
0.42638	0.0
0.71780	-0.43520
0.63100	-0.37810
0.55900	-0.33950
0.49598	-0.29960
0.71780	0.43520
0.63100	0.37810
0.55900	0.33950
0.49598	0.29960
1.09918	-0.82538
0.96230	-0.72841
0.85562	-0.64314
0.75732	-0.56368
0.63022	0.18918
0.55770	0.16241
0.49118	0.14754
0.43572	0.13268
0.63022	-0.18918
0.55770	-0.16241
0.49118	-0.14754
0.43572	-0.13268
1.09918	0.82538
0.96230	0.72841
0.85562	0.64314
0.75732	0.56368

STATE OF THE DATA AFTER ITERATION 4

4.43690	0.0
3.91048	0.0
0.55230	0.0
0.47932	0.0
0.61100	-0.47650
0.54310	-0.42638
0.61100	0.47650
0.54310	0.42638
0.87301	-1.07053
0.76986	-0.94066
0.56259	0.20014
0.49214	0.18446
0.56259	-0.20013
0.49214	-0.18446
0.87301	1.07053
0.76986	0.94066
1.64355	-1.74700
1.44626	-1.53899
0.55481	0.09624
0.47835	0.08218
0.57857	-0.32107
0.51353	-0.29092
0.68188	0.69943
0.60186	0.61574
0.68188	-0.69943
0.60186	-0.61574
0.57857	0.32107
0.51353	0.29092
0.55481	-0.09624
0.47835	-0.08218
1.64355	1.74700
1.44626	1.53899

8.34737	0.0
0.52642	0.0
0.55230	-0.47932
0.55230	0.47932
0.69353	-1.16202
0.52847	0.20903
0.52847	-0.20903
0.69353	1.16202
1.22429	-2.23420
0.52172	0.09313
0.54468	-0.32513
0.58050	0.72540
0.58051	-0.72540
0.54468	0.32513
0.52173	-0.09313
1.22429	2.23420
2.76178	-3.53857
0.52532	0.04457
0.54208	-0.38895
0.56753	0.58143
0.62198	-0.90969
0.53516	0.26754
0.52353	-0.14691
0.84022	1.54577
0.84022	-1.54577
0.52353	0.14691
0.53516	-0.26754
0.62198	0.90969
0.56753	-0.58143
0.54208	0.38895
0.52532	-0.04457
2.76177	3.53856

FINAL VALUES OF THE C(K) FOR 32DATA POINTS.

8.34737	0.0
2.76178	-3.53857
1.22429	-2.23420
0.84022	-1.54577
0.69353	-1.16202
0.62198	-0.90969
0.58051	-0.72540
0.56753	-0.58143
0.55230	-0.47932
0.54208	-0.38895
0.54468	-0.32513
0.53516	-0.26754
0.52847	-0.20903
0.52353	-0.14691
0.52173	-0.09313
0.52532	-0.04457
0.52642	0.0
0.52532	0.04457
0.52172	0.09313
0.52353	0.14691
0.52847	0.20903
0.53516	0.26754
0.54468	0.32513
0.54208	0.38895
0.55230	0.47932
0.56753	0.58143
0.58050	0.72540
0.62198	0.90969
0.69353	1.16202
0.84022	1.54577
1.22429	2.23420
2.76177	3.53856

Remark. The DFT coefficients are obtained from the above C(k) as follows:

$$C_y(k) = \frac{C(k)}{NUM}.$$

In this example NUM = 32.

AMPLITUDE	PHASE
1.00000	0.0
0.53774	-52.02869
0.30520	-61.27821
0.21077	-61.47316
0.16212	-59.16982
0.13202	-55.63847
0.11130	-51.33130
0.09734	-45.69295
0.08761	-40.95348
0.07993	-35.65942
0.07599	-30.83403
0.07168	-26.56163
0.06808	-21.58038
0.06514	-15.67499
0.06349	-10.12132
0.06316	-4.84985
0.06306	0.0
0.06316	4.84988
0.06349	10.12135
0.06514	15.67505
0.06808	21.58044
0.07168	26.56169
0.07599	30.83411
0.07993	35.65950
0.08761	40.95352
0.09734	45.69305
0.11130	51.33134
0.13202	55.63853
0.16212	59.16986
0.21077	61.47322
0.30520	61.27821
0.53774	52.02873

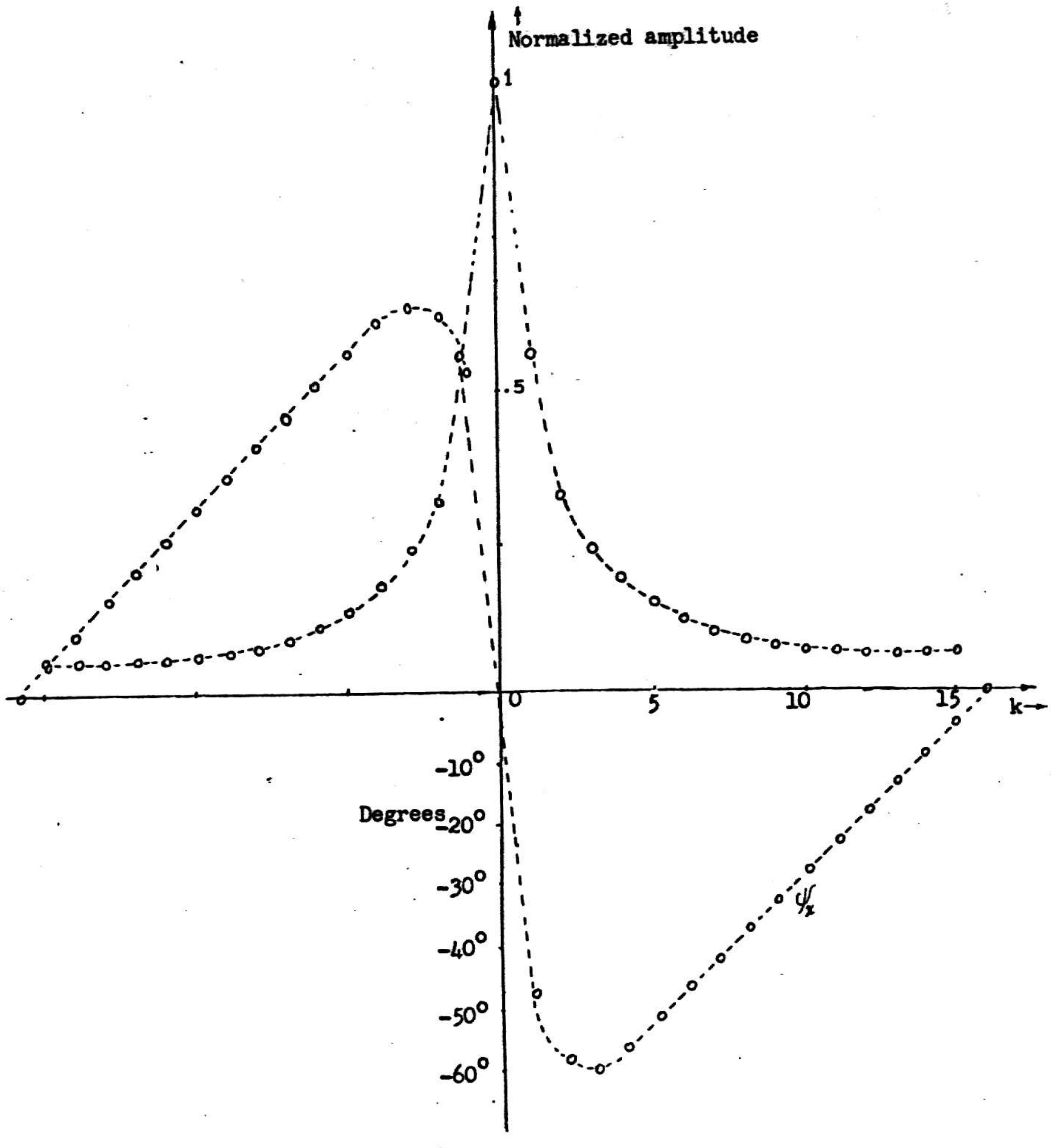


Fig. 3-2. DFT spectra of a 32-periodic real number sequence

2. <u>Convolution and Correlation</u>

Given two real N - periodic sequences $\left\{X(m)\right\}$ and $\left\{Y(m)\right\}$, their cross-correlation $\left\{\hat{Z}(m)\right\}$ and convolution $\left\{Z(m)\right\}$ are defined as

$$Z(m) = \frac{1}{N} \sum_{h=0}^{N-1} X(h)\ Y(m + h), \quad m = 0, 1, \ldots (N-1) \tag{3 - 8}$$

and

$$Z(m) = \frac{1}{N} \sum_{h=0}^{N-1} X(h)\ Y(m-y), \quad m = 0, 1, \ldots (N-1) \tag{3 - 9}$$

respectively.

<u>Computational Considerations</u>

i. Direct Matrix Multiplication: Clearly, (3-8) and (3-9) can be expressed in terms of a product of an (N x N) matrix and an (N x 1) column vector. Thus, direct multiplication requires $\sum_2$ arithmetic operations to compute the sequences $\left\{Z(m)\right\}$ and $\left\{\hat{Z}(m)\right\}$, where

$$\sum\nolimits_2 \simeq 2N^2. \tag{3 - 10}$$

Again, the number of storage locations required is given by

$$\hat{\sum}\nolimits_2 \propto N^2. \tag{3 - 11}$$

ii. Use of the FFT/IFFT: Applying (2-1) and (2-3) to (3-8) and (3-9) it can be shown that [7, 11]

$$C_z(k) = \bar{C}_x(k) C_y(k), \qquad k = 0, 1, \ldots, (N-1) \tag{3 - 12}$$

and

$$C_z(k) = C_x(k) C_y(k), \qquad k = 0, 1, \ldots, (N-1) \tag{3 - 13}$$

Using (3-12) and (3-13), rapid computation of $\left\{Z(m)\right\}$ and $\left\{\hat{Z}(m)\right\}$ is possible as summarized in Fig. (3-2) which shows flow charts depicting the sequence of computations.

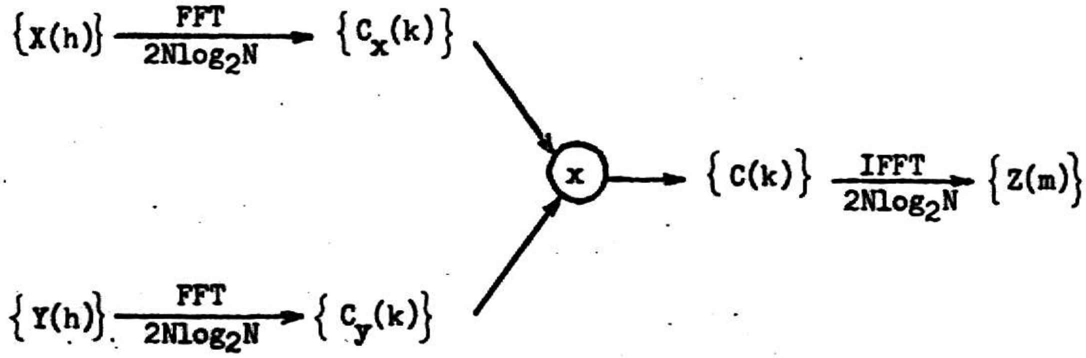


Fig. 3-3a

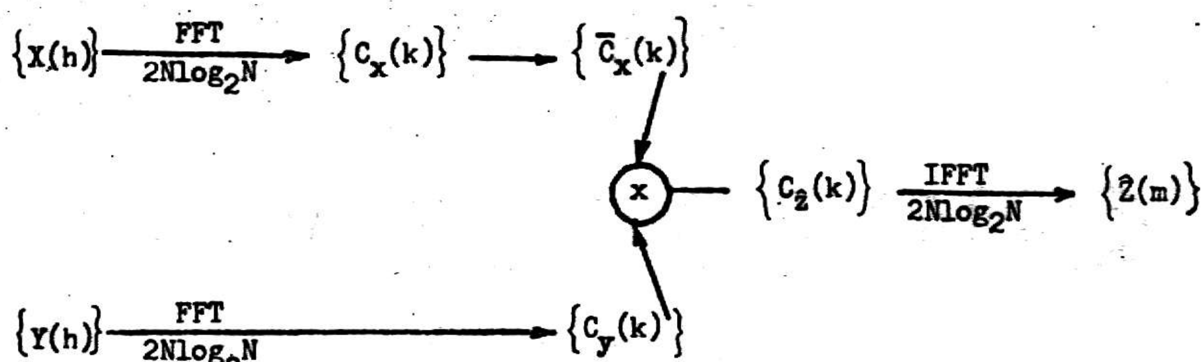


Fig. 3-3b

Fig. 3-3. Flow chart showing sequence of computations to obtain $\{Z(m)\}$ and $\{\hat{Z}(m)\}$

Examination of Fig. (3-3) shows that the number of arithmetic operations Λ_2 and the number of storage locations required to obtain the $\{Z(m)\}$ and $\{\hat{Z}(m)\}$ are

$$\Lambda_2 \simeq 6N\log_2 N \qquad (3 - 14)$$

$$\hat{\Lambda} \propto N. \qquad (3 - 15)$$

Comparing (3-10) with (3-14) and (3-11) with (3-15) it is clear that the FFT/IFFT approach is much more efficient than the straightforward matrix multiplication method.

<u>Example 3-2</u>

Using the 32 - periodic sequence $\{X(m)\}$ considered in Example 3-1,

a. Convolve $\{X(m)\}$ with itself and plot the resulting $\{Z(m)\}$.

b. Find the autocorrelation of $\{X(m)\}$ i.e. the cross-correlation of $\{X(m)\}$ with itself and plot the resulting $\{\hat{Z}(m)\}$.

In (a) and (b), use the FFT/IFFT approach.

Solution: The "Fast Fourier Transform; Convolution and Correlation" deck is used as follows:

i. The program included will yield the convolution. If correlation is desired, then proceed as follows:

Remove card with statement 0094 (see page 32)

Replace the above card with the comment card following 0094.

ii. The first data card must be one with the number of data points Format (I5) . This is followed by N data cards for the first sequence $\{X(m)\}$ and N data Points for the second sequence.

The program listing and output are included in what follows. Figs. (3-4) and (3-5) show plots of $\{Z(m)\}$ and $\{\hat{Z}(m)\}$ on pages 38 and 44 respectively.

```
                C
                C       FFT   -  FAST FOURIER TRANSFORM
                C      CONVOLUTION  AND  CORRELATION
                C
0001                    DIMENSION IPOWER(10)
0002                    COMPLEX*8 X(32)
0003                    COMPLEX Y1(32),Y2(32),ALPHA,ALPH,CMPLX
                C
                C       THIS ROUTINE CALCULATES THE FAST FOURIER
                C       TRANSFORM FOR ANY GIVEN NUMBER  WHICH
                C       IS A POWER OF TWO
                C       NUM  NUMBER OF POINTS
                C
0004                198 READ(1,100) NUM
0005                100 FORMAT (I5)
                C
                C       INPUT THE NUMBER OF DESIRED DATA POINTS.
                C
                C
0006                    JJ=0
0007                 65 READ(1,250) (X(I),I=1,NUM)
0008                250 FORMAT(2F10.7)
                C
                C
0009                    WRITE(3,103)
0010                103 FORMAT (1H0,'ECHO CHECK OF INPUT VALUES'//)
0011                    WRITE (3,1010) (X(I),I=1,NUM)
0012               1010 FORMAT (10X,2F10.5)
                C
                C
                C       CALCULATE NUMBER OF ITERATIONS.
                C
0013                    II=0
0014                 32 ITER=0
0015                    IREM=NUM
0016                  1 IREM=IREM/2
0017                    IF (IREM.EQ.0) GO TO 2
0018                    ITER=ITER+1
0019                    GO TO 1
0020                  2 CONTINUE
                C
                C       BEGIN A LOOP FOR (LOG TO BASE TWO OF NUM) ITERATIONS.
                C
0021                    PI=3.141593
0022                    DO 50 M = 1,ITER
                C
                C       CALCULATE NUMBER OF PARTITIONS AND THE VALUE OF
                C       ALPHA FOR EACHITERATION
                C
0023                    IF (M.EQ.1) NUMP = 1
0024                    IF (M.NE.1) NUMP = NUMP * 2
0025                    MNUM =NUM/NUMP
0026                    MNUM2 = MNUM/2
0027                    IF (II.EQ.0) GO TO 52
0028                    ALPHA = CMPLX(COS(PI/NUMP),(SIN(PI/NUMP)))
0029                    GO TO 53
0030                 52 ALPHA = CMPLX(COS(PI/NUMP),(SIN(PI/NUMP))*(-1.0))
0031                 53 CONTINUE
```

```
               C
               C      BEGIN A LOOP FOR THE NUMBER OF PARTITIONS.
               C
0032                  DO 49 MP = 1,NUMP
0033                  IB = (MP-1) * MNUM
               C
               C      FIND THE POWER OF ALPHA
               C
0034                  IBC = IB
0035                  IL = 1
0036                4 IBD = IBC/2
0037                  IPOWER(IL) = 1
0038                  IF (IBC.EQ.(IBD*2)) IPOWER(IL) = 0
0039                  IF (IBD.EQ.0) GO TO 5
0040                  IBC = IBD
0041                  IL = IL + 1
0042                  GO TO 4
0043                5 CONTINUE
0044                  IP = 0
0045                  IFAC = NUM
0046                  DO 6 I = 1,IL
0047                  IFAC = IFAC/2
0048                6 IP = IP + IPOWER(I) * IFAC
0049                  ALPH = ALPHA**(IP)
               C
               C      BEGIN A LOOP THROUGH THIS PARTITION.
               C
0050                  DO 48 MP2 = 1,MNUM2
0051                  MNUM21 = MNUM2 + MP2 + IB
0052                  IBA = IB + MP2
0053                  IF(JJ.EQ.0) Y1(IBA) = X(IBA) + ALPH * X(MNUM21)
0054                  IF(JJ.EQ.1) Y2(IBA) = X(IBA) + ALPH * X(MNUM21)
0055                  IF(JJ.EQ.0) Y1(MNUM21) = X(IBA) - ALPH * X(MNUM21)
0056                  IF(JJ.EQ.1) Y2(MNUM21) = X(IBA) - ALPH * X(MNUM21)
0057               48 CONTINUE
0058               49 CONTINUE
0059                  DO 7 I = 1,NUM
0060                  IF (JJ.EQ.0) X(I)=Y1(I)
0061                  IF (JJ.EQ.1) X(I)=Y2(I)
0062                7 CONTINUE
0063              102 FORMAT (1H,2F15.5)
0064               50 CONTINUE
               C
               C      BEGIN A ROUTINE TO CALCULATE THE C(K)
               C
0065                  DO 11 I = 1,NUM
0066                  IB = I - 1
0067                  IL = 1
0068                9 IBD = IB/2
0069                  IPOWER(IL) = 1
0070                  IF (IB.EQ.(IBD*2)) IPOWER(IL) = 0
0071                  IF (IBD.EQ.0) GO TO 10
0072                  IB = IBD
0073                  IL = IL + 1
0074                  GO TO 9
0075               10 CONTINUE
0076                  IP = 1
0077                  IFAC = NUM
```

```
0078            DO 12 I1 = 1,IL
0079            IFAC = IFAC/2
0080         12 IP = IP + IFAC * IPOWER(I1)
0081            IF(JJ.EQ.0) Y1(IP)=X(I)
0082            IF (JJ.EQ.1) Y2(IP)=X(I)
0083         11 CONTINUE
0084            WRITE(3,105)NUM
0085        105 FORMAT (1H1,'FINAL VALUES OF THE C(K) FOR ',I3,
               1'DATA POINTS.',//)
0086            DO 13 I = 1,NUM
0087            IF(JJ.EQ.0) WRITE(3,102)Y1(I)
0088            IF(JJ.EQ.1) WRITE(3,102) Y2(I)
0089         13 CONTINUE
0090            IF(II.EQ.1) GO TO 999
0091            JJ=JJ+1
0092            IF (JJ.EQ.1) GO TO 65
0093            DO 30 I=1,NUM
0094            X(I)=(Y1(I)/NUM) * (Y2(I)/NUM)
          C     X(I)=(Y1(I)/NUM) *  (CONJG(Y2(I))/NUM)
0095         30 CONTINUE
0096            II=II+1
0097            JJ=JJ-1
0098            GO TO 32
0099        999 STOP
0100            END
```

ECHO CHECK OF INPUT VALUES

1.00000	0.0
0.88000	0.0
0.77880	0.0
0.69000	0.0
0.60650	0.0
0.53000	0.0
0.47230	0.0
0.41600	0.0
0.36780	0.0
0.32600	0.0
0.28650	0.0
0.25150	0.0
0.22310	0.0
0.19790	0.0
0.17370	0.0
0.15250	0.0
0.13530	0.0
0.12000	0.0
0.10540	0.0
0.09348	0.0
0.08200	0.0
0.07200	0.0
0.06390	0.0
0.05610	0.0
0.04970	0.0
0.04300	0.0
0.03870	0.0
0.03600	0.0
0.03020	0.0
0.02600	0.0
0.02300	0.0
0.02000	0.0

Remark. These are the values of $\{X(m)\}$.

FINAL VALUES OF THE C(K) FOR 32DATA POINTS.

8.34737	0.0
2.76178	-3.53857
1.22429	-2.23420
0.84022	-1.54577
0.69353	-1.16202
0.62198	-0.90969
0.58051	-0.72540
0.56753	-0.58143
0.55230	-0.47932
0.54208	-0.38895
0.54468	-0.32513
0.53516	-0.26754
0.52847	-0.20903
0.52353	-0.14691
0.52173	-0.09313
0.52532	-0.04457
0.52642	0.0
0.52532	0.04457
0.52172	0.09313
0.52353	0.14691
0.52847	0.20903
0.53516	0.26754
0.54468	0.32513
0.54208	0.38895
0.55230	0.47932
0.56753	0.58143
0.58050	0.72540
0.62198	0.90969
0.69353	1.16202
0.84022	1.54577
1.22429	2.23420
2.76177	3.53856

Remark. The DFT coefficients of $\{X(m)\}$ are obtained from the C(k) as follows:

$$C_x(k) = \frac{C(k)}{NUM}$$

In this example, NUM = 32.

ECHO CHECK OF INPUT VALUES

1.00000	0.0
0.88000	0.0
0.77880	0.0
0.69000	0.0
0.60650	0.0
0.53000	0.0
0.47230	0.0
0.41600	0.0
0.36780	0.0
0.32600	0.0
0.28650	0.0
0.25150	0.0
0.22310	0.0
0.19790	0.0
0.17370	0.0
0.15250	0.0
0.13530	0.0
0.12000	0.0
0.10540	0.0
0.09348	0.0
0.08200	0.0

Remark. These are the values of $\{Y(m)\}$.

0.07200	0.0
0.06390	0.0
0.05610	0.0
0.04970	0.0
0.04300	0.0
0.03870	0.0
0.03600	0.0
0.03020	0.0
0.02600	0.0
0.02300	0.0
0.02000	0.0

8.34737	0.0
2.76178	-3.53857
1.22429	-2.23420
0.84022	-1.54577
0.69353	-1.16202
0.62198	-0.90969
0.58051	-0.72540
0.56753	-0.58143
0.55230	-0.47932
0.54208	-0.38895
0.54468	-0.32513
0.53516	-0.26754
0.52847	-0.20903
0.52353	-0.14691
0.52173	-0.09313
0.52532	-0.04457
0.52642	0.0
0.52532	0.04457
0.52172	0.09313
0.52353	0.14691
0.52847	0.20903
0.53516	0.26754
0.54468	0.32513
0.54208	0.38895
0.55230	0.47932
0.56753	0.58143
0.58050	0.72540
0.62198	0.90969
0.69353	1.16202
0.84022	1.54577
1.22429	2.23420
2.76177	3.53856

Remark. The DFT coefficients are obtained from the above C(k) as follows:

$$C_y(k) = \frac{C(k)}{NUM}.$$

In this example NUM = 32.

FINAL VALUES OF THE C(K) FOR 32DATA POINTS.

0.04891	-0.00000
0.07008	-0.00000
0.08574	-0.00000
0.09692	-0.00000
0.10413	-0.00000
0.10799	-0.00000
0.10979	-0.00000
0.10963	-0.00000
0.10802	0.00000
0.10537	0.00000
0.10178	0.00000
0.09750	0.00000
0.09288	0.00000
0.08807	0.00000
0.08305	0.00000
0.07794	0.00000
0.07293	0.00000
0.06806	0.00000
0.06330	0.00000
0.05874	0.00000
0.05435	0.00000
0.05016	0.00000
0.04623	0.00000
0.04251	0.00000
0.03903	0.00000
0.03574	0.00000
0.03272	0.00000
0.03004	-0.00000
0.02743	-0.00000
0.02499	-0.00000
0.02274	-0.00000
0.02066	-0.00000

Remark. These are the values of $\{Z(m)\}$.

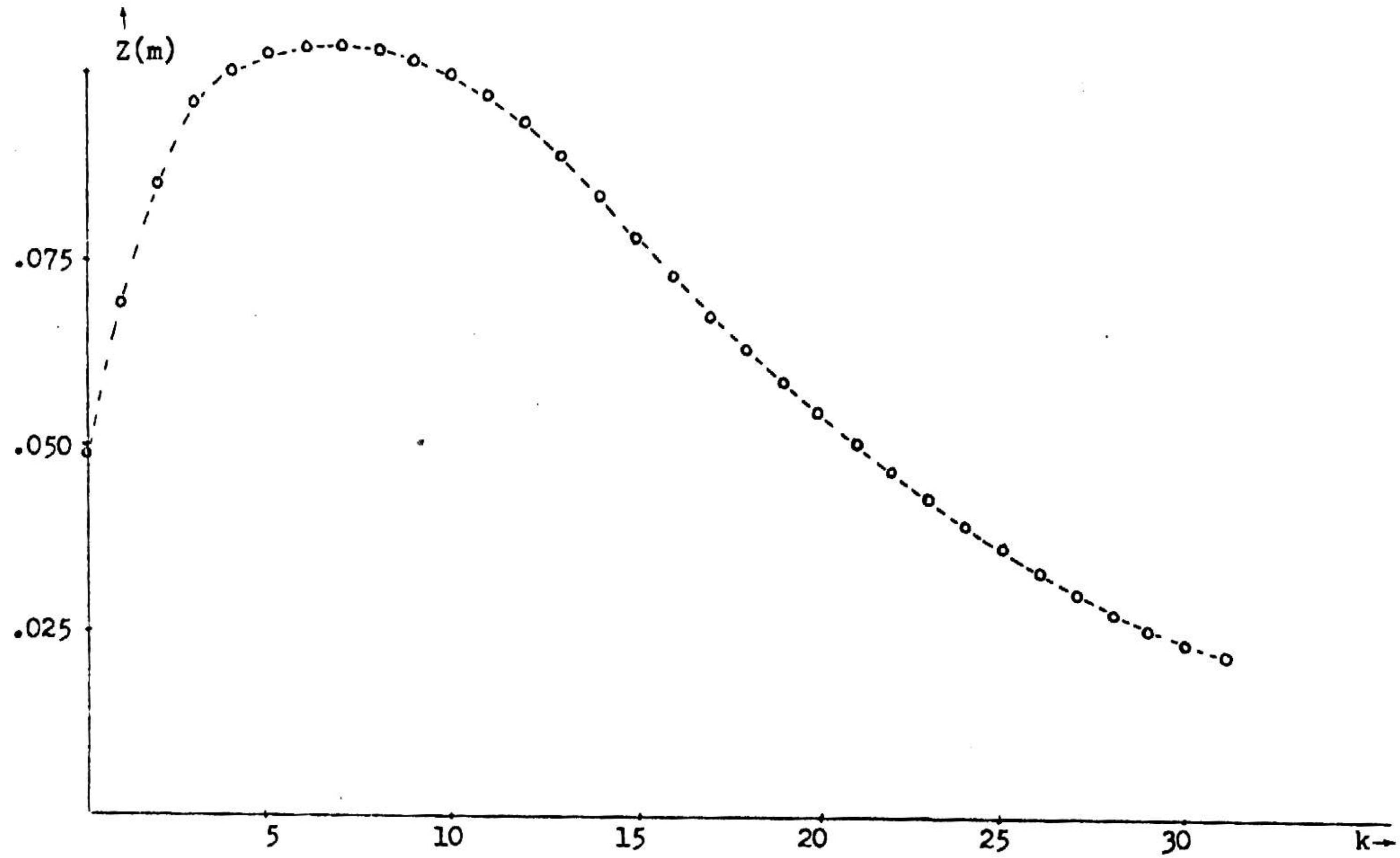


Fig. 3.4. Plot of convolution of $\{X(m)\}$ with itself in Example 3.1

ECHO CHECK OF INPUT VALUES

1.00000	0.0
0.88000	0.0
0.77880	0.0
0.69000	0.0
0.60650	0.0
0.53000	0.0
0.47230	0.0
0.41600	0.0
0.36780	0.0
0.32600	0.0
0.28650	0.0
0.25150	0.0
0.22310	0.0
0.19790	0.0
0.17370	0.0
0.15250	0.0
0.13530	0.0
0.12000	0.0
0.10540	0.0
0.09348	0.0
0.08200	0.0
0.07200	0.0
0.06390	0.0
0.05610	0.0
0.04970	0.0
0.04300	0.0
0.03870	0.0
0.03600	0.0
0.03020	0.0
0.02600	0.0
0.02300	0.0
0.02000	0.0

<u>Remark</u>. These are the values of $\{X(m)\}$.

FINAL VALUES OF THE C(K) FOR 32DATA POINTS.

6.34737	0.0
2.76178	-3.53857
1.22429	-2.23420
0.84022	-1.54577
0.69353	-1.16202
0.62198	-0.90969
0.58051	-0.72540
0.56753	-0.58143
0.55230	-0.47932
0.54208	-0.38895
0.54468	-0.32513
0.53516	-0.26754
0.52847	-0.20903
0.52353	-0.14691
0.52173	-0.09313
0.52532	-0.04457
0.52642	0.0
0.52532	0.04457
0.52172	0.09313
0.52353	0.14691
0.52847	0.20903
0.53516	0.26754
0.54468	0.32513
0.54208	0.38895
0.55230	0.47932
0.56753	0.58143
0.58050	0.72540
0.62198	0.90969
0.69353	1.16202
0.84022	1.54577
1.22429	2.23420
2.76177	3.53856

Remark. The DFT coefficients of $\{X(m)\}$ are obtained from the D(k) as follows:

$$C_x(k) = \frac{C(k)}{NUM}$$

In this example, NUM = 32.

ECHO CHECK OF INPUT VALUES

1.00000	0.0
0.88000	0.0
0.77880	0.0
0.69000	0.0
0.60650	0.0
0.53000	0.0
0.47230	0.0
0.41600	0.0
0.36780	0.0
0.32600	0.0
0.28650	0.0
0.25150	0.0
0.22310	0.0
0.19790	0.0
0.17370	0.0
0.15250	0.0
0.13530	0.0
0.12000	0.0
0.10540	0.0
0.09348	0.0
0.08200	0.0

Remark. These are the values of $\{Y(m)\}$.

0.07200	0.0
0.06390	0.0
0.05610	0.0
0.04970	0.0
0.04300	0.0
0.03870	0.0
0.03600	0.0
0.03020	0.0
0.02600	0.0
0.02300	0.0
0.02000	0.0

FINAL VALUES OF THE C(K) FOR 32DATA POINTS.

8.34737	0.0
2.76178	-3.53857
1.22429	-2.23420
0.84022	-1.54577
0.69353	-1.16202
0.62198	-0.90969
0.58051	-0.72540
0.56753	-0.58143
0.55230	-0.47932
0.54208	-0.38895
0.54468	-0.32513
0.53516	-0.26754
0.52847	-0.20903
0.52353	-0.14691
0.52173	-0.09313
0.52532	-0.04457
0.52642	0.0
0.52532	0.04457
0.52172	0.09313
0.52353	0.14691
0.52847	0.20903
0.53516	0.26754
0.54468	0.32513
0.54208	0.38895
0.55230	0.47932
0.56753	0.58143
0.58050	0.72540
0.62198	0.90969
0.69353	1.16202
0.84022	1.54577
1.22429	2.23420
2.76177	3.53856

<u>Remark</u>. The DFT coefficients are obtained from the above C(k) as follows:

$$C_y(k) = \frac{C(k)}{NUM}$$

In this example, NUM = 32.

FINAL VALUES OF THE C(K) FOR 32DATA POINTS.

0.14101	0.0
0.12503	0.00000
0.11110	0.00000
0.09883	0.00000
0.08804	0.00000
0.07876	0.00000
0.07083	0.00000
0.06382	0.00000
0.05787	0.00000
0.05280	0.00000
0.04851	0.00000
0.04501	0.00000
0.04227	0.00000
0.04017	0.00000
0.03865	0.00000
0.03779	0.00000
0.03752	0.0
0.03779	-0.00000
0.03865	-0.00000
0.04017	-0.00000
0.04227	-0.00000
0.04501	-0.00000
0.04851	-0.00000
0.05280	-0.00000
0.05787	-0.00000
0.06382	-0.00000
0.07083	-0.00000
0.07876	-0.00000
0.08804	-0.00000
0.09883	-0.00000
0.11110	-0.00000
0.12503	-0.00000

<u>Remark</u>. These are the values of $\{\hat{Z}(m)\}$.

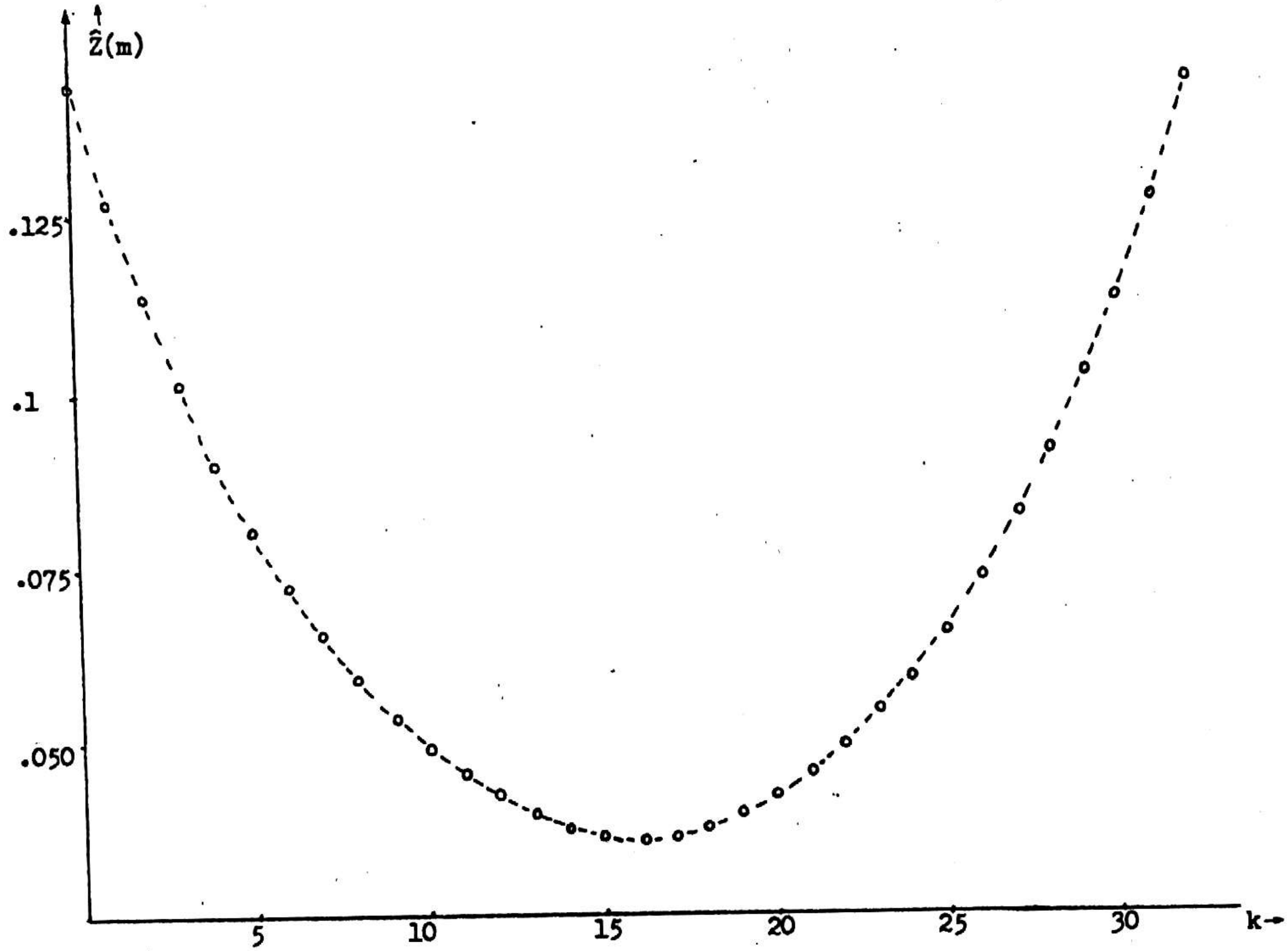


Fig. 3-5. Plot of autocorrelation of $\{X(m)\}$ in Example 3-1.

CHAPTER IV

BIFORE TRANSFORM

4.1 Definition of the BIFORE (Binary Fourier Representation) Transform* (BT)

Let $X(m)$, $m = 0, 1, \ldots, (N-1)$ be a real N - periodic sequence. Then its BT is defined as

$$\left\{\underline{B}_x(n)\right\} = \frac{1}{N}\left[H(n)\right]\left\{\underline{X}(n)\right\} \tag{4 - 1}$$

where

$n = \log_2 N$.

$\left\{\underline{X}(n)\right\}' = \left\{X(0), X(1)\ldots\ldots X(N-1)\right\}$, is a $(1 \times N)$ vector representation of the sequence $\left\{X(m)\right\}$, prime indicating transpose,

$\left\{\underline{B}_x(n)\right\}' = \left\{B_x(0), B_x(1), \ldots B_x(N-1)\right\}$, is a $(1 \times N)$ vector whose elements are the BT coefficients and,

$\left[H(n)\right]$ is a $(N \times N)$ Hadamard matrix.

The Hadamard matrices $\left[H(n)\right]$ in (4-1) can be generated by the recursive relation [10]

$$\left[H(k+1)\right] = \left[\begin{array}{c|c} H(k) & H(k) \\ \hline H(k) & -H(k) \end{array}\right], \quad k = 0, 1, \ldots, n \tag{4 - 2}$$

$H(0) = 1$.

Since the Hadamard matrices in (4-2) are orthogonal, the inverse BIFORE transform (IBT) is defined as

$$\left\{\underline{X}(n)\right\} = \left[H(n)\right]\left\{B_x(n)\right\}. \tag{4 - 3}$$

4.2 The Fast BIFORE Transform

The fast BIFORE transform (FBT) is an algorithm which enables

---

*Also known as the Hadamard transform (HT) or the Walsh-Fourier transform.

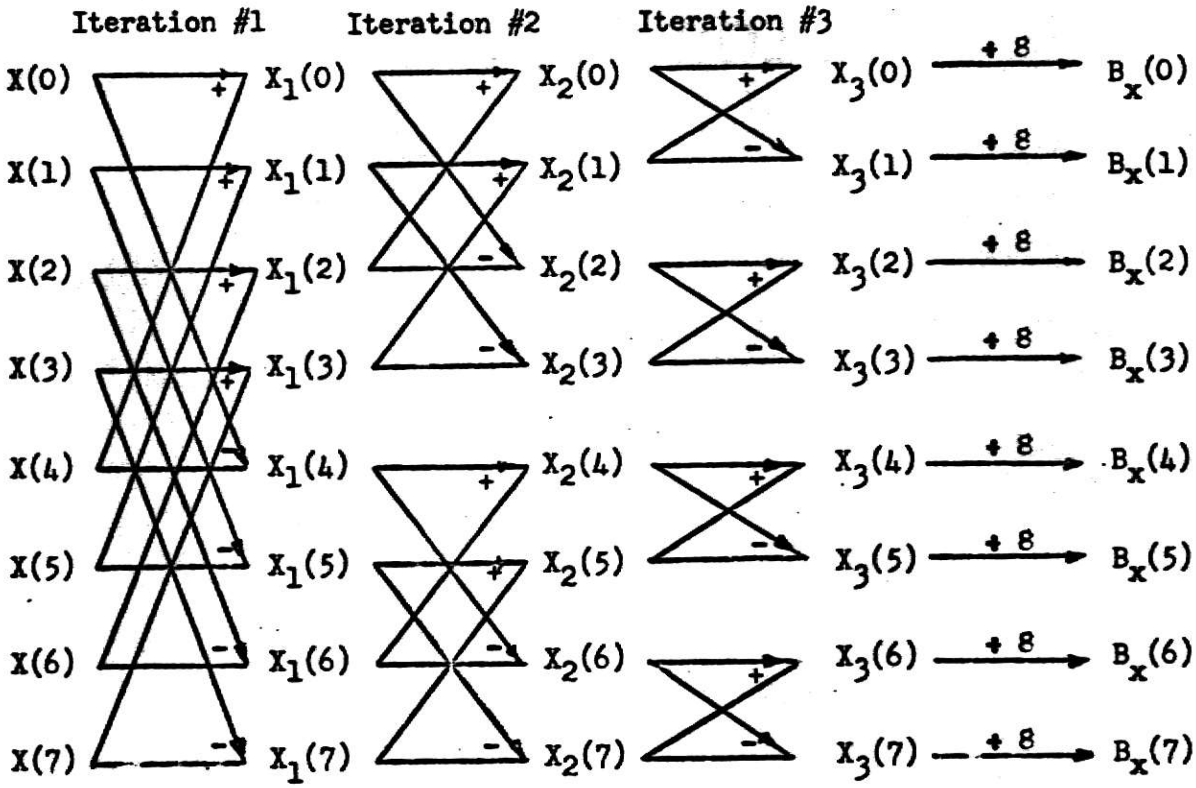
rapid computation of the BIFORE coefficients $B_x(0)$, $B_x(1) \ldots B_x(N-1)$ in (4-1). The algorithm can be developed using elementary matrix partitioning and the recursive properties of the Hadamard matrices in (4-2) [3]. Fig. (4-1) shows the signal flow graph for the case N = 8. Inspection of Fig. (4-1) results in the following observations and generalizations:

1. The total number of iterations is given by $n = \log_2 N$. Thus if r is an iteration index, then, $r = 1, 2, \ldots \log_2 N$.
2. The r^{th} iteration results in 2^{r-1} groups with $N/2^{r-1}$ members in each group. Half the members in each group are associated with an addition operation while the remaining half are associated with a subtraction operation.
3. The total number of arithmetic operations to compute all the transform coefficients $B_x(k)$, $k = 0, 1 \ldots (N-1)$ is approximately $N \log_2 N$.
4. The output data of an iteration becomes the input data to the following iteration. Thus no additional storage locations are required as the data is being processed. Consequently the number of storage locations required is proportional to N.
5. The algorithm can also be used to compute the IBT in (3-2) by merely omitting the division by N, of the last iteration output. We shall refer to this version of the algorithm as the inverse fast BIFORE transform.

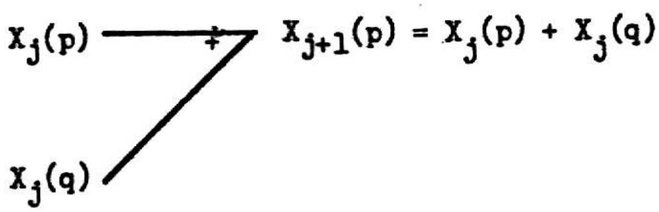
4.3 Advantages of the FBT

As in the case of the FFT, the FBT also yields substantial savings in execution time and storage.

1. Saving in Execution Time. The direct approach to compute $\{B_x(n)\}$ in (4-1) involves a (N x N) Hadamard matrix whose elements are ± 1.



Notation:



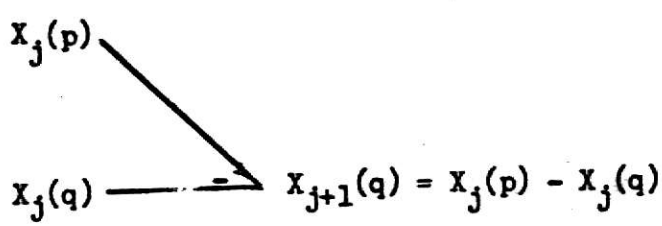


Fig. 4-1. FBT Signal Flow Graph, N = 8.

Therefore, direct matrix multiplication requires $\sum_3$ real number additions and subtractions, where

$$\sum_3 \simeq N^2 \qquad (4 - 4)$$

Using the FBT, a total of $\log_2 N$ iterations are involved. Each iteration requires N/2 real number additions and subtractions. Thus the total number of arithmetic operations is given by

$$\Lambda_3 \simeq N \log_2 N \qquad (4 - 5)$$

From (4 - 4) and (4 - 5) it follows that with increasing N, the number $N \log_2 N$ becomes small compared to N^2 and hence the FBT is more economical to use.

2. Saving in Storage Locations. In Fig. (4-1) it is observed that the output of the r^{th} iteration becomes the input to the $(r + 1)^{th}$ iteration, $r = 1, 2, \ldots \log_2 N$. This implies that no additional storage locations are required as the data is being processed. Consequently the total number of storage locations required in the FBT is given by

$$\hat{\Lambda}_3 \simeq N \qquad (4 - 6)$$

In the direct approach it is necessary to store (N x N) Hadamard matrices; thus the number of storage locations $\hat{\sum}_3$ required is such that

$$\hat{\sum}_3 \simeq N^2 \qquad (4 - 7)$$

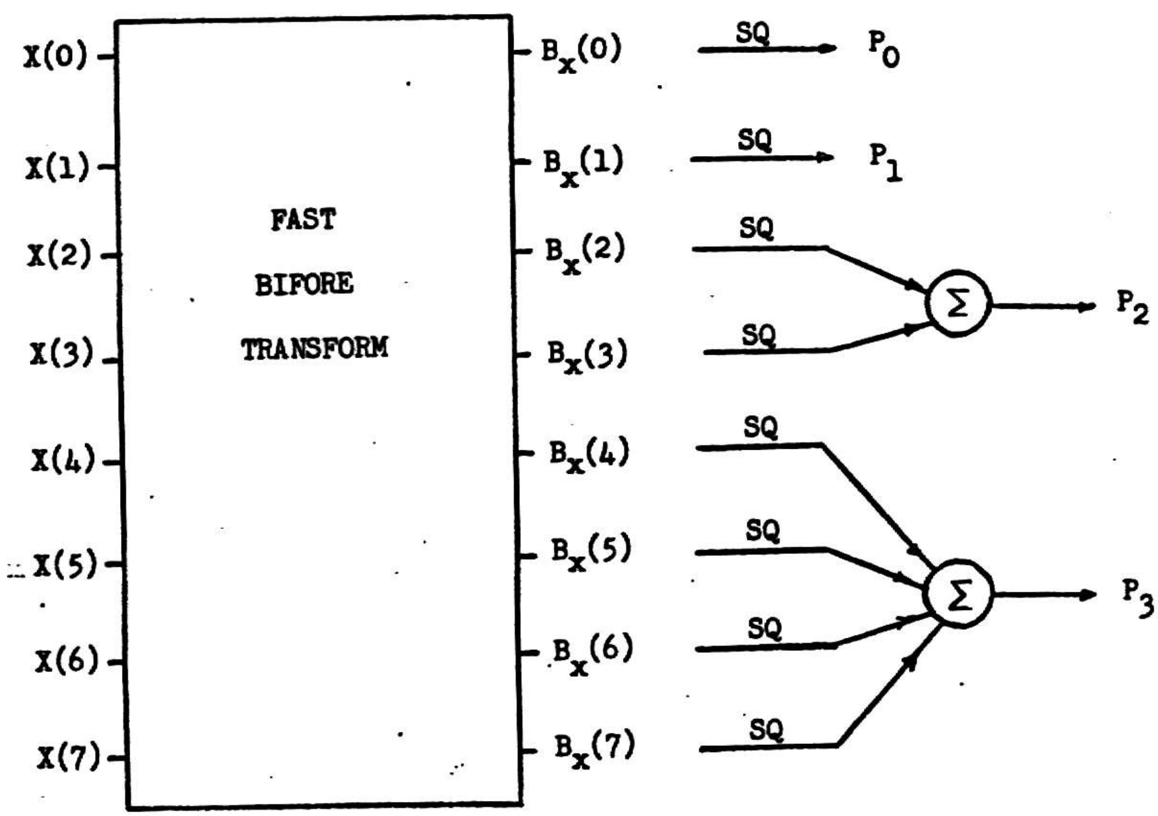
Comparison of (4-6) and (4-7) leads to the conclusion that the FBT yields a substantial saving in storage as N increases.

4.4 BT Power Spectrum

The power spectrum is defined by a set of $(\log_2 N + 1)$ points $\{P_i\}$ as follows:

$$P_0 = B_x^2(0)$$

$$P_s = \sum_{k = 2^{s-1}}^{2^s - 1} B_x^2(k), \quad s = 1, 2, 3, \ldots n, \quad n = \log_2 N \qquad (4 - 8)$$



Notation: SQ denotes "square."

Fig. 4-2. Computation of the Power spectrum, N = 8.

For example, with N = 8, (4-8) yields

$$P_0 = B_x^2\,(0)$$
$$P_1 = B_x^2\,(1)$$
$$P_2 = B_x^2\,(2) + B_x^2\,(3)$$
$$P_3 = B_x^2\,(4) + B_x^2\,(5) + B_x^2\,(6) + B_x^2\,(7). \qquad (4-9)$$

The BT power spectrum is invariant to shifts in the input sequence $\{X(m)\}$ similar to the DFT power spectrum. However, the BT power spectrum provides data compression since the DFT power spectrum consists of $(\frac{N}{2} + 1)$ independent spectrum points when $\{X(m)\}$ is real.

4.5 Computation of the BT Power Spectrum

The FBT can be used to compute the power spectrum as illustrated in Fig. (4-2) for the case N = 8. It can be shown, [7] , however, that by suitably modifying the FBT approach, the spectrum can be computed without having to actually compute all the coefficients $B_x(k)$, $k = 0, 1 \ldots (N-1)$ in (4-8). This is possible due to the orthogonal properties of the Hadamard matrix. For the case N = 8, it can be shown that [7]

$$P_0 = \frac{1}{8^2}\left\{X_3^2\,(0)\right\}$$
$$P_1 = \frac{1}{8^2}\left\{X_3^2\,(1)\right\}$$
$$P_2 = \frac{2}{8^2}\left\{\sum_{m=2}^{3} X_2^2\,(m)\right\}$$
$$P_3 = \frac{2^2}{8^2}\left\{\sum_{m=4}^{7} X_1^2\,(m)\right\}. \qquad (4-10)$$

The sequence of computations in (4-10) are shown in the signal flow graph in Fig. (4-3).

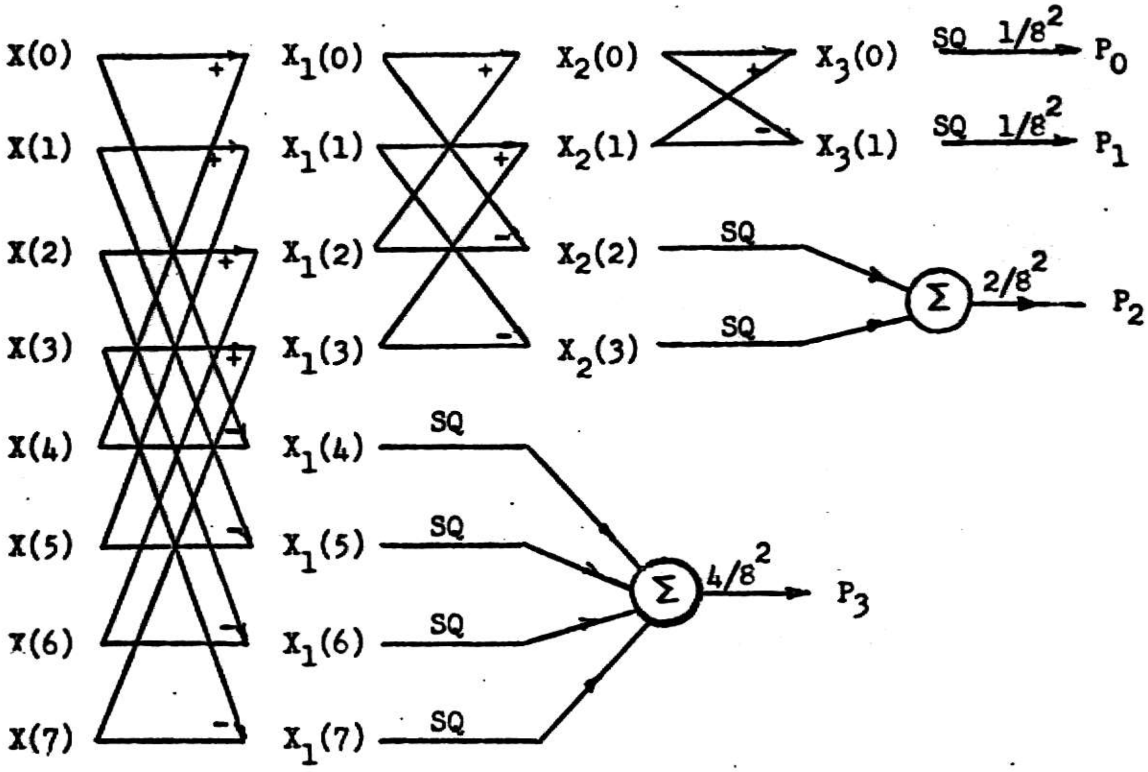


Fig. 4-3. Power spectrum computation, N = 8.

<u>Generalizations</u>. A generalization of the FBT modification to compute the power spectrum is as follows:

$$P_0 = \frac{1}{N^2} X_n^2(0)$$

$$P_s = \frac{2^{s-1}}{N^2} \sum_{m=2^{s-1}}^{2^s-1} X_{n+1-s}^2(m), \quad s = 1, 2, \ldots n \quad n = \log_2 N. \qquad (4 - 11)$$

In conclusion, the following points pertaining to the signal flow graph associated with (4-11) are summarized:

1. For a specified N, the total number of iterations is given by $n = \log_2 N$. Thus, if r is an iteration index, then $r = 1, 2, \ldots n$.
2. The r^{th} iteration results in two groups with $N/2^{r-1}$ members in each group. The sum of squares of the members in the lower group divided by $N^2/2^{n-r}$ yield the spectrum point P_{n-r}, $r = 1, 2, \ldots (n-2)$.
3. The spectrum points P_0 and P_1 are obtained by computing $X_n^2(0)/N^2$ and $X_n^2(1)/N^2$ where $X_n(0)$ and $X_n(1)$ are the zeroth and first data points of the last iteration.
4. This modified approach is more efficient than a direct application of the FBT since it is not necessary to compute the transform coefficients $B_x(k)$, $k = 0, 1 \ldots (N-1)$ in order to obtain the spectrum.

4.6 Relation between the BT and DFT Power Spectra

Since the BT power spectrum affords data compression, the question naturally arises whether one can obtain the BIFORE spectrum from the DFT power spectrum. It can be shown that [7] the BT power spectrum can be obtained from the DFT power spectrum as follows:

$$P_o = C_x^2(0)$$

$$P_1 = C_x^2\,(N/2)$$

and $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ (4 - 12)

$$P_s = 2\left[\sum_{k=0}^{2^{s-2}-1} \left| C_x\left[2^{n-s}(2k+1)\right]\right|^2\right], \quad \begin{array}{l} s = 2, 3, \ldots n \\ n = \log_2 N. \end{array}$$

From (4-12) it is clear that the power spectrum points $\{P_i\}$ represent the power in groups of frequencies rather than individual frequencies which is the case with the DFT spectrum. However, the power in individual frequencies is combined in such a way that the shift in variance (i.e. with respect to the input sequence $\{X(m)\}$) is not lost. It is emphasized that (4-12) is valid for real input sequences. In closing it is noted that (4-12) also yields the "frequency content" of a BT power spectrum point P_i. If $F(P_i)$ denotes the frequency content of P_i, then, from (4-12) it follows that

$$F(P_0) = 0$$

$$F(P_1) = N/2 \qquad\qquad (4 - 13)$$

and

$$F(P_s) = 2^{n-s}(2k+1), \quad s = 2, 3, \ldots n;\ k = 0, 1, \ldots (2^{s-2}-1).$$

For example, with $N = 32$, $n = 5$ and hence (4-13) yields:

$$F(P_0) = 0$$

$$F(P_1) = 16$$

$$F(P_2) = 8$$

$$F(P_3) = 4,\ 12$$

$$F(P_4) = 2,\ 6,\ 10,\ 14$$

and

$$F(P_5) = 1,\ 3,\ 5,\ 7,\ 9,\ 11,\ 13,\ 15. \qquad\qquad (4 - 14)$$

4.7 BT Phase Spectrum

It is recalled that the DFT phase spectrum of an N - periodic sequence $\{X(m)\}$ is defined as [see (2-13)]

$$\psi_x(k) = \tan^{-1}\left[\frac{B_x(k)}{A_x(k)}\right], \quad k = 0, 1, \ldots, (N-1) \qquad (4-15)$$

where $A_x(k)$ and $B_x(k)$ are the real and imaginary parts of the DFT coefficient $C_x(k)$. From (4-15) it follows that the DFT phase spectrum has the following properties:

1. $\psi_x(k)$ is invariant with respect to multiplying the sequence $\{X(m)\}$ by a constant; that is, the phase spectrum does not change when $\{X(m)\}$ is "amplified" or "attenuated."
2. As the sequence $\{X(m)\}$ is shifted, the $\psi_x(k)$ change. In other words, the DFT phase spectrum characterizes the shift in the input sequence $\{X(m)\}$.

A BT phase spectrum which has properties analogous to those cited above has been developed [5]. It is defined as follows:

$$\cos\theta_0 = \frac{B_x(0)}{|P_0^{\frac{1}{2}}|}$$

$$\cos\theta_s = \frac{\sum_{k=2^{s-1}}^{2^s-1} B_x(k)}{2^{\left(\frac{s-1}{2}\right)}\,|P_s^{\frac{1}{2}}|}, \quad s = 1, 2, \ldots, n;\ n = \log_2 N. \qquad (4-16)$$

where P_i, $i = 0, 1, \ldots, n$ are the BT power spectrum points.

Examination of (4-16) shows that the $\cos\theta_s$ do not change when $\{X(m)\}$ is multiplied by a constant. Again, as $\{X(m)\}$ is shifted, the $\cos\theta_s$ change. In fact, the $\cos\theta_s$ are 2^{s-1}, $s = 1, 2, \ldots, n$

anti-periodic as illustrated in Example (4-1) which follows.

Example 4-1

For the 8-periodic sequence

$$\{X(m)\} = \{1\ 2\ 1\ 1\ 3\ 2\ 1\ 2\}$$

compute the BT phase spectrum of $X(m)$ for all possible shifts.

Solution: Using (4-16) it is straightforward to obtain the BT spectrum. The results are summarized in Table (4-1), where r is the shift-parameter.

Table 4-1. Anti-periodic structure of phase spectrum, N = 8.

SPECTRUM POINT \ SHIFT	0	1	2	3	4	5	6	7
$\cos\theta_0$	1	1	1	1	1	1	1	1
$\cos\theta_1$	-1	1	-1	1	-1	1	-1	1
$\cos\theta_2$	$\frac{2}{5}$	$\frac{1}{5}$	$-\frac{2}{5}$	$-\frac{1}{5}$	$\frac{2}{5}$	$\frac{1}{5}$	$-\frac{2}{5}$	$-\frac{1}{5}$
$\cos\theta_3$	$-\frac{2}{5}$	0	0	$-\frac{1}{5}$	$\frac{2}{5}$	0	0	$\frac{1}{5}$

Observation. In Table (4-1) it is observed that:

i. $\cos\theta_0$ is 1 - periodic.

ii. $\cos\theta_s$ is 2^{s-1}- anti-periodic, s = 1, 2, 3.

It can be shown that the phase spectrum points have the anti-periodic structure expressed in (ii). That is, $\cos\theta_s$ is 2^{s-1}-anti-periodic, s = 1, 2,..., n, where $n = \log_2 N$.

Computational Considerations. Rapid computation of the BT phase spectrum is possible if the cos θ_s in (4-16) are expressed in the following equivalent form [7]:

$$\cos \theta_0 = \frac{X_n(0)}{N|P_0^{\frac{1}{2}}|}$$

$$\cos \theta_s = \frac{2^{(\frac{s-1}{2})}}{N|P_s^{\frac{1}{2}}|} X_{n+1-s}(2^{s-1}), \quad s = 1, 2, \ldots, n \qquad (4 - 17)$$

where

$$n = \log_2 N$$

and

$X_k(1)$ is the 1^{st} output point of the k^{th} iteration.

The signal flow graph associated with (4-17) is shown in Fig. (4-4) for the case N = 8.

In sequel, two numerical examples are considered.

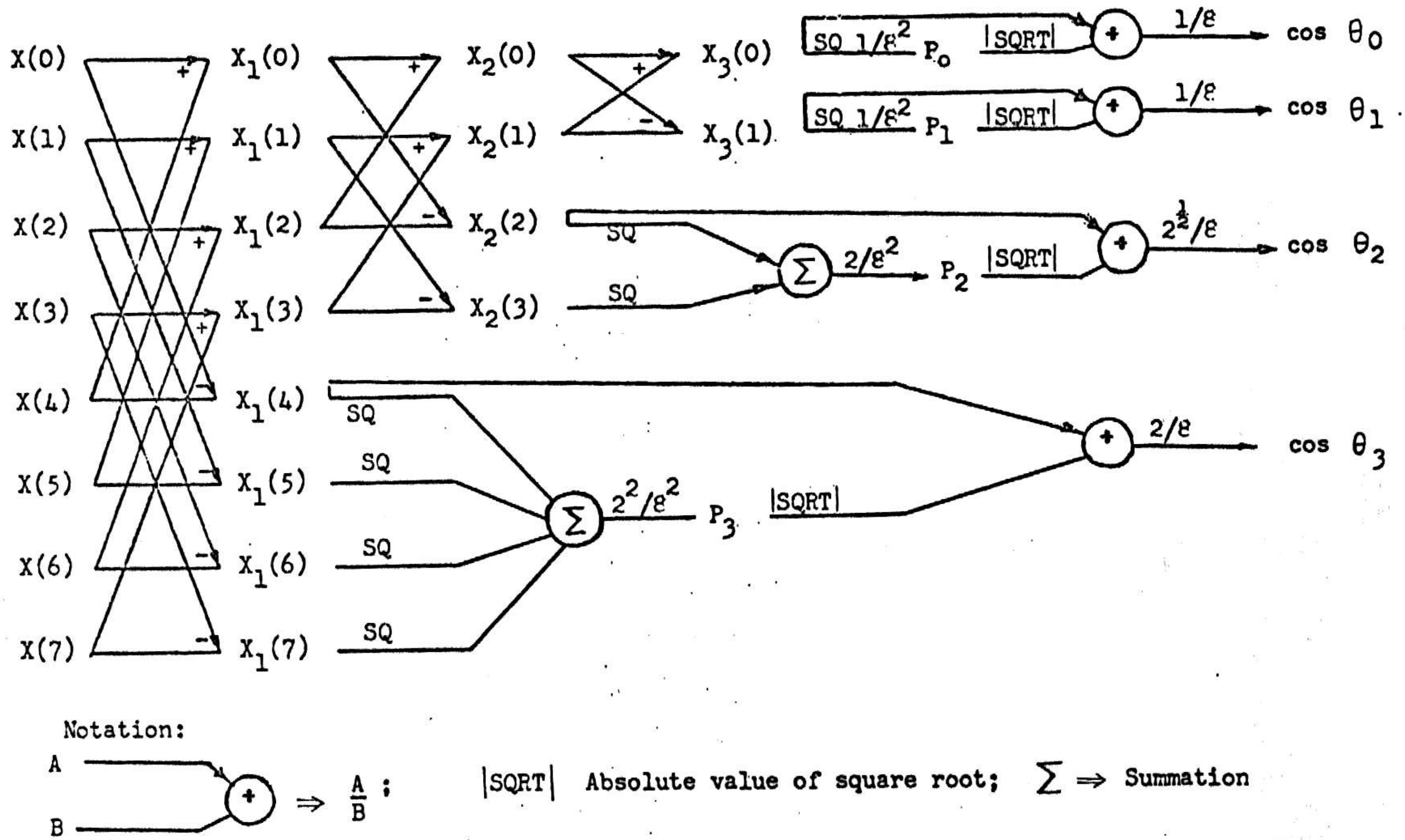


Fig. 4-4. BT Phase Spectrum Signal Flow Graph, N = 8.

Example 4-2

Find the BIFORE transform, power and phase spectra, of the 32-periodic sequence $\{X(m)\}$ considered in Examples (3-1) and (3-2).

Solution: The "BIFORE TRANSFORM" program deck is used as follows:

i. The number of data points, NUM, has to be a power of two. Punch NUM in the first ten columns of the first data card. Again, in column 20 of the same card, punch any one of the numbers 1, 2, 3 or 4 which respectively yield the following outputs:

1 The BIFORE transform.

2 The inverse BIFORE transform.

3 The power spectrum.

4 The power and phase spectra.

ii. The above first card is followed by the NUM (i.e. $N = 2^n$) data cards.

The program listing and output are included in what follows. Fig. (4-5) shows plots of the BT power and phase spectra. The frequency content of each spectrum point is also indicated using the information in (4-14).

```
$JOB          PM
C    THIS PROGRAM COMPUTES THE BIFORE TRANSFORM
C       OR THE INVERSE BIFORE TRANSFORM
C        THE DATA SET. NUM MUST BE A POWER OF TWO
C       LESS THAN OR EQUAL TO 1024
C      AN INTEGER DISP WHICH WILLBE
C       1 IF THE BIFORE TRANSFORM IS TO BE CALCULATED
C       2 IF TE INVERSE BIFORE TRANSFORM IS TO BE USED
C       3 IF THE POWER SPECTRUM IS TO BE COMPUTED
C       4 IF THE PHASE SPECTRUM IS TO BE FOUND
C       NUM AND DISP WILL BE RIGHT JUSTIFIED
C       TO CC 10 AND CC 20 RESPECTIVELY
  999 FORMAT (2I10)
  998 FORMAT (8G10.6)
  997 FORMAT ('1','NUMBER OF DATA POINTS:
     1',I5,5X,' DISPOSITION CODE:',I3,
     2///' ','ECHO CHECK OF INPUT DATA')
  996 FORMAT (' ',8G16.7)
  995 FORMAT ('-','FINAL VALUES OF BIFORE TRANSFORM'///)
  994 FORMAT ('-','FINAL VALUES OF
     1INVERSE BIFORE TRANSFORM'///)
  993 FORMAT (' ',G15.7)
  992 FORMAT ('-','CHECK: RESULT OF IBT AND
     1SHOULD EQUAL INPUT'///)
  991 FORMAT ('-','POWER SPECTRUM OF X(M)'///)
  990 FORMAT ('-','POWER AND PHASE SPECTRA OF
     1  X(M)'/'-',14X,'POWER',33X,
     2'PHASE'/38X,'DEG-RADIANS',15X,'COS(DEG)'//)
  989 FORMAT (' ',10X,G15.7,10X,G15.7,10X,G15.7)
      DIMENSION X(1024)
    1 READ (1,999,END=1000) NUM,IDISP
      WRITE (3,997) NUM,IDISP
      READ (1,998) (X(M),M=1,NUM)
      WRITE (3,996) (X(M),M=1,NUM)
      ISIG = NUM
C
C     CALCULATE THE NUMBER OF ITERATIONS:
C
      ITER = 0
      N = NUM
   10 N = N / 2
      IF (N .EQ. 0) GO TO 20
      ITER = ITER + 1
      GO TO 10
C
C    LOOP THROUGH EACH ITERATION;
C         DIVIDE THE ARRAY INTO 2**(ITER-1) PARTITIONS
C     WITH NGRP ELEMENTS IN EACH PARTITION
C    WHEN POWER SPECTRUM IS CALCULATED
C     EACH ITERATION WILL OPERATE ON ONLY HALF
C    THE REMAINING ARRAY,  ISIG IS A SIGNAL
C     TO POINT TO THE LAST ELEMENT
C
   20 DO 50 I = 1,ITER
      IF (I .EQ. 1) NGRP = NUM
      IF (I .NE. 1) NGRP = NGRP / 2
      IHAF = NGRP / 2
C
C    LOOP THROUGH EACH PARTITION
C         REPLACE  ELMENTS HALF WAY APART
```

```
C        IN THE PARTITION WITH THEIR SUM AND
C           DIFFERENCE RESPECTFULLY
C
         J = 1
      25 DO 30 K = 1,IHAF
         ISUB1 = J + K - 1
         ISUB2 = IHAF + ISUB1
         TEMP = X(ISUB1)
         X(ISUB1) = TEMP + X(ISUB2)
         X(ISUB2) = TEMP - X(ISUB2)
      30 CONTINUE
         J = J + NGRP
         IF (J .LE. ISIG) GO TO 25
         IF (IDISP .GE. 3) ISIG = ISIG / 2
      50 CONTINUE
         IF (IDISP .LT.0) GO TO 90
         IF (IDISP .EQ. 1) GO TO 70
         IF (IDISP .GE. 3) GO TO 100
C
C       FOR BT OUTPUT WILL BE DIVIDED BY N,
C        PRINTED, THEN THE ARRAY WILL BE
C       OPERATED ON BY THE IBT AS A CHECK PROCEDURE
C
         IDISP = -1
         WRITE (3,995)
         DO 60 I = 1,NUM
         X(I) = X(I) / NUM
      60 WRITE (3,993) X(I)
         GO TO 20
C
C       FOR IBT OUTPUT WILL BE PRINTED,
C        DIVIDED AND SENT TO THE BT AS PART OF THE CHECK
C
      70 IDISP = -1
         WRITE (3,994)
         DO 80 I = 1,NUM
         WRITE (3,993) X(I)
      80 X(I) = X(I) / NUM
         GO TO 20
      90 WRITE (3,992)
         WRITE (3,996) (X(M),M=1,NUM)
         GO TO 1
C
C       FORM SUMS FOR POWER AND PHASE SPECTRA,
C       SUM OF SQUARES OF DATA POINTS WILL BE
C        STORED IN EVEN POWERS OF THE MEMORY
C           LOCATIONS:  IE. X(1),X(2),X(4),X(8),. . .
C       XN0 CONTAINS THE VALUE OF THE FIRST  ELEMENT
C        BEFORE POWER SPECTRUM CALCULATION
C       XN1 CONTAINS THE VALUE OF THE SECOND ELEMENT
C        BEFORE POWER SPECTRUM CALCULATION
C
     100 XN0 = X(1)
         XN1 = X(2)
         X(1) = X(1)**2 * 2
         X(2) = X(2)**2
         J = 2
         N = 4
     110 SUM = 0
     120 J = J + 1
```

```
      SUM = SUM + X(J)**2
      IF (J .NE. N) GO TO 120
      X(N) = SUM
      N = N * 2
      IF (N .LE. NUM) GO TO 110
C
C     POWER SPECTRUM WILL BE CALCULATED BY
C      MULTIPLYING EACH SUM OF SQUARE BY
C     AN APPROPRIATE CONSTANT
C
      ITER = ITER + 1
      J = 1
      FACT = NUM * NUM
      DO 130 I = 1,ITER
      X(J) = X(J) / FACT /2
      FACT = FACT / 2
  130 J = J * 2
C
C     OUTPUT POWER SPECTRUM ONLY
C
      IF (IDISP .EQ. 4) GO TO 150
      WRITE (3,991)
      J = 1
      DO 140 I = 1,ITER
      WRITE (3,993) X(J)
  140 J = J*2
      GO TO 1
C
C     CALCULATE PHASE SPECTRUM
C     DIVIDE X(L) BY CORRECT POWER SPECTRUM POINT,
C      INCLUDE CORRECT CONSTANT,
C     TAKE ARCOS OF RESULT AND PRINT POWER
C      AND PHASE POINTS. (L=1,2,3,5,9,....)
C
  150 WRITE (3,990)
      FACT = 1.0 / 1.414213
      I = 1
  160 IF (I .LE. 2) J = I
      IF (I .EQ. 1) TEMP = XN0/(SQRT(X(1))*NUM)
      IF (I .EQ. 2) TEMP = XN1/(SQRT(X(2))*NUM)
      IF (I .LE. 2) GO TO 165
      J = J*2
      TEMP = FACT*X(J-(J/2-1))/(SQRT(X(J))*NUM)
  165 RAD = ARCOS (TEMP)
      WRITE (3,989) X(J),RAD,TEMP
      I = I + 1
      FACT = FACT * 1.414213
      IF (I .LE. ITER) GO TO 160
      GO TO 1
 1000 STOP
      END
```

NUMBER OF DATA POINTS: 32

ECHO CHECK OF INPUT DATA

1.000000	0.8800000	0.7788000	0.6900000
0.3678000	0.3260000	0.2865000	0.2515000
0.1353000	0.1200000	0.1054000	0.9347999E-01
0.4970000E-01	0.4300000E-01	0.3870000E-01	0.3600000E-01

0.6065000	0.5300000	0.4723000	0.4160000
0.2231000	0.1979000	0.1737000	0.1525000
0.8200002E-01	0.7200003E-01	0.6389999E-01	0.5610000E-01
0.3020000E-01	0.2600000E-01	0.2300000E-01	0.2000000E-01

FINAL VALUES OF INVERSE BIFORE TRANSFORM

8.347374
0.5264206
1.031620
0.7297897E-01
2.056977
0.1180198
0.2512204
0.1777977E-01
3.856178
0.2468185
0.4680188
0.4097897E-01
0.9513797
0.5281943E-01
0.1196197
0.6579936E-02
6.357817
0.4031792
0.7883797
0.5142021E-01
1.560218
0.9477997E-01
0.1967787
0.9820223E-02
2.933020
0.1899796
0.3495801
0.4021996E-01
0.7274202
0.3837985E-01
0.8437997E-01
0.9819925E-02

CHECK: RESULT OF IBT AND BT SHOULD EQUAL INPUT

0.9999991	0.8799994	0.7787994	0.6899995
0.3677996	0.3259994	0.2864996	0.2514994
0.1352996	0.1199998	0.1053998	0.9347963E-01
0.4970010E-01	0.4299997E-01	0.3869991E-01	0.3599970E-01

0.6064997	0.5299999	0.4722999	0.4159999
0.2230999	0.1978999	0.1736999	0.1524999
0.8199978E-01	0.7199985E-01	0.6390005E-01	0.5609991E-01
0.3020005E-01	0.2599990E-01	0.2300006E-01	0.1999994E-01

NUMBER OF DATA POINTS: 32

ECHO CHECK OF INPUT DATA

1.000000	0.8800000	0.7788000	0.6900000
0.3678000	0.3260000	0.2865000	0.2515000
0.1353000	0.1200000	0.1054000	0.9347999E-01
0.4970000E-01	0.4300000E-01	0.3870000E-01	0.3600000E-01

0.6065000	0.5300000	0.4723000	0.4160000
0.2231000	0.1979000	0.1737000	0.1525000
0.8200002E-01	0.7200003E-01	0.6389999E-01	0.5610000E-01
0.3020000E-01	0.2600000E-01	0.2300000E-01	0.2000000E-01

POWER AND PHASE SPECTRA OF X(M)

POWER	PHASE DEG-RADIANS	PHASE COS(DEG)
0.6804556E-01	0.0000000	1.000000
0.2706225E-03	0.0000000	1.000000
0.1044496E-02	0.7147734	0.7552418
0.4207529E-02	0.9413253	0.5887172
0.1569728E-01	1.039803	0.5063903
0.5174896E-01	1.075670	0.4751423

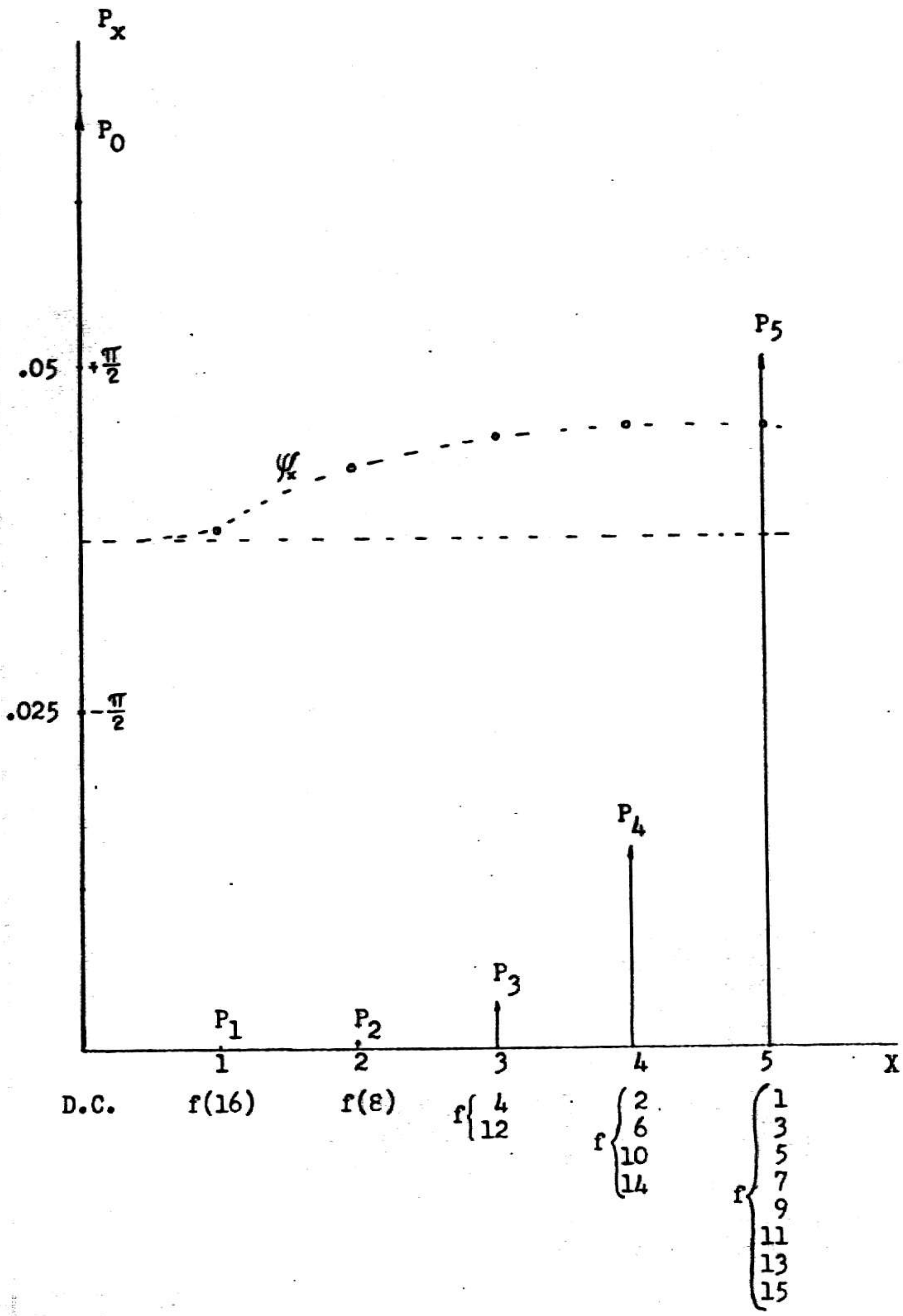


Fig. 4-5. BIFORE power and phase spectra.

<u>Example 4-3</u>

Using the DFT coefficients of $\{X(m)\}$ in Example (3-1) (see page 12), and the BT spectrum computed in Example (4-2), verify that the BT spectrum can be constructed from the DFT spectrum using (4-12).

Solution: With N = 32, and n = 5, (4-12) yields

$$P_0 = C_x^2(0)$$

$$P_1 = C_x^2(16)$$

$$P_2 = 2|C_x(8)|^2$$

$$P_3 = 2 \quad |C_x(4)|^2 + |C_x(12)|^2$$

$$P_4 = 2 \quad |C_x(2)|^2 + |C_x(6)|^2 + |C_x(10)|^2 + |C_x(14)|^2$$

and

$$P_5 = 2 \quad |C_x(1)|^2 + |C_x(3)|^2 + |C_x(5)|^2 + |C_x(7)|^2 + |C_x(9)|^2 + + |C_x(11)|^2 + |C_x(13)|^2 + |C_x(15)|^2. \qquad (4-18)$$

The BT power spectrum was found to be [see Example (4-2)]

$$P_0 = 0.6804 \times 10^{-1}$$
$$P_1 = 0.2706 \times 10^{-3}$$
$$P_2 = 0.1044 \times 10^{-2}$$
$$P_3 = 0.4207 \times 10^{-2}$$
$$P_4 = 0.1569 \times 10^{-1}$$
$$P_5 = 0.5174 \times 10^{-1}. \qquad (4-19)$$

Now, using the $C_x(k)$, k = 0, 1,... 16 obtained in Example (3-1) (see page 24), one has

$$|C_x(0)|^2 = (8.3473/32)^2 = 0.6804 \times 10^{-1}$$

$$\left|c_x(16)\right|^2 = \left(\frac{0.0630 \times 8.3473}{32}\right)^2 = 0.2706 \times 10^{-3}$$

$$2\left|c_x(8)\right|^2 = 2\left(\frac{0.08761 \times 8.3473}{32}\right)^2 = 0.1044 \times 10^{-2}$$

$$2\left[\left|c_x(4)\right|^2 + \left|c_x(12)\right|^2\right] = 2\,(0.1621^2 + 0.0680^2)\,(8.3473/32)^2$$
$$= 0.4207 \times 10^{-2}$$

$$2\left[\left|c_x(2)\right|^2 + \left|c_x(6)\right|^2 + \left|c_x(10)\right|^2 + \left|c_x(14)\right|^2\right] = 2\,(0.3052^2 + 0.1113^2$$
$$+ 0.0759^2 + 0.0634^2) \times (8.3473/32)^2 = 0.1569 \times 10^{-1}$$

$$2\left[\left|c_x(1)\right|^2 + \left|c_x(3)\right|^2 + \left|c_x(5)\right|^2 + \left|c_x(7)\right|^2 + \left|c_x(9)\right|^2 + \left|c_x(11)\right|^2\right.$$
$$\left. + \left|c_x(13)\right|^2 + \left|c_x(15)\right|^2\right] = 2\left[(0.5374^2 + 0.2107^2 + 0.1320^2\right.$$
$$\left. + 0.0973^2 + 0.0799^2 + 0.0716^2 + 0.0651^2 + 0.0631^2)(8.3473/32)^2\right]$$
$$= 0.5170 \times 10^{-1}. \qquad (4 - 20)$$

Comparing (4-19) and (4-20) it follows that (4-18) is verified.

REFERENCES

[1] Ahmed, N., and Rao, K. R., "Discrete Fourier and Hadamard Transforms," Electronics Letters, April 2, 1970, pp. 221-224.

[2] Ahmed, N., and Cheng, S. M., "On Matrix Partitioning and a Class of Algorithms," accepted for publication in IEEE Transactions on Education.

[3] Ahmed, N., and Rao, K. R., "Spectral Analysis of Linear Time-Invariant Systems Using BIFORE (Binary Fourier Representation)," Electronics Letters, vol. 6, No. 2, 22nd Jan., 1970, pp. 43-44.

[4] Ahmed, N., and Rao, K. R., "Convolution and Correlation Using Binary Fourier Representation," Proceedings of the First Annual Houston Conference on Circuits, Systems, and Computers, May 22-23, 1969, Houston, Texas.

[5] Ahmed, N., Rao, K. R. and Fisher, P. S., "BIFORE (Binary Fourier Representation) Phase Spectrum," Presented at the 13th Midwest Symposium on Circuit Theory, University of Minnesota, Mpls, Minn., May 7-8, 1970. (Also submitted to Electronics Letters).

[6] Ahmed, N., Bates, R. M., and Rao, K. R., "Multi-dimensional Binary Fourier Representation," Proc. Second Southeastern Symposium on System Theory, March 1970, pp. J3-0 to J3-10.

[7] Ahmed, N., Class Notes, EE 826.

[8] Cochran, W. T., Cooley, J. W., et. al., "The Fast Fourier Transform," Proc. IEEE, October 1967, pp. 1664-1674.

[9] Glassman, J. A., "A Generalization of the Fast Fourier Transform," IEEE Transactions on Computers, vol. C-19, February 1970, pp. 105-116.

[10] Ohnsorg, F. R., "Binary Fourier Representation," Spectrum Analysis Techniques Symposium, Sept. 1966, Honeywell Research Center, Hopkins, Minnesota.

[11] Oppenheim, A. V., "Speech Spectrograms Using the Fast Fourier Transform," IEEE Spectrum, August 1970, pp. 57-62.

[12] Pratt, W. K., et. al., "Hadamard Transform Image Coding," Proc. IEEE, vol. 57, pp. 58-68, January 1969.

[13] Robinson, G. S. and Campanella, S. J., "Digital Sequency Decomposition of Voice Signals," Proceedings of the Walsh Function Symposium (sponsored by the Navy Research Laboratory and the University of Maryland), 1970, pp. 230-237.

[14] Rosenfeld, A., "Picture Processing by Computer," Academic Press, 1969, Chapter 4.

[15] Special Issues on Fast Fourier Transform, IEEE Transactions on Audio and Electroacoustics, June 1969.

[16] Walsh, J. L., "A Closed Set of Orthogonal Functions," Am J. Math., vol. 55, pp. 5-24, 1923.

[17] Whelchel, J. E. and Guinn, D. F., "The Fast Fourier-Hadamard Transform and its Use in Signal Representation and Classification," EASCON '68. Record, pp. 561-573.

ACKNOWLEDGMENTS

The author wishes to express his appreciation to his advisor, Dr. Nasir Ahmed, for his guidance and patience in the preparation of this report.

He also wishes to express thanks to his wife for her constant encouragement and for her effort in typing this work.

APPLICATIONS OF FOURIER AND BIFORE ANALYSES

TO DISCRETE SIGNALS

by

PEDRO E. MARTINEZ

B. S., Universidad Industrial de Santander, 1967

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1970

This report provides brief introductions to the discrete Fourier transform and the BIFORE (Binary Fourier Representation) transform. Certain applications of these transforms in the area of signal processing are illustrated by means of numerical examples. Documentations of the digital computer programs used to work the numerical examples are also