

CONTROL SYSTEM ANALYSIS

VIA

FAST FOURIER TRANSFORM

---

A Thesis

Presented to

the Faculty of the Department of Electrical Engineering

The University of Houston

---

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Electrical Engineering

---

by

Ran-Fun Chiu

May, 1969

501192

## ACKNOWLEDGMENT

I wish to express my sincere gratitude to my advisor, Professor C. F. Chen, for his advice and guidance. I would like to express my grateful thanks to Professor W. P. Schnider and Professor G. F. Paskuze for their valuable comments, corrections, and discussions. Loving and heartfelt thanks due my wife, Lucy, who not only provided me with continuous encouragement, but also typed this thesis.

CONTROL SYSTEM ANALYSIS

VIA

FAST FOURIER TRANSFORM

---

An Abstract

Presented to

the Faculty of the Department of Electrical Engineering

The University of Houston

---

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Electrical Engineering

---

by

Ran-Fun Chiu

May, 1969

## ABSTRACT

The Fast Fourier Transform has demonstrated its power in the fields of communication, optics etc. It has never been applied to control system analysis.

This thesis presents the fundamentals of the Fast Fourier Transform first; then develops several related theorems for Laplace transforms. It is believed that this is the first time to use the Fast Fourier Transform to perform the inverse Laplace transform of irrational transfer functions and transcendental transfer functions. Sixteen computer programs are included.

## TABLE OF CONTENTS

SECTION	PAGE
I. FUNDAMENTAL OF FOURIER TRANSFORM . . . . .	1
II. DISCRETE FOURIER TRANSFORM . . . . .	12
III. INVERSE DISCRETE FOURIER TRANSFORM . . . . .	21
IV. FAST FOURIER TRANSFORM- DECIMATION IN TIME . . . . .	24
V. FAST FOURIER TRANSFORM- DECIMATION IN FREQUENCY . . . . .	45
VI. FOURIER SERIES EVALUATION . . . . .	64
VII. CONVOLUTION, CROSS-CORRELATION AND AUTO-CORRELATION . . . . .	74
VIII. INVERSE LAPLACE TRANSFORM OF RATIONAL TRANSFER FUNCTIONS . . . . .	85
IX. INVERSE LAPLACE TRANSFORM OF IRRATIONAL TRANSFER FUNCTIONS . . . . .	91
X. INVERSE LAPLACE TRANSFORM OF TRANSCENDENTAL TRANSFER FUNCTIONS . . . . .	100
REFERENCES . . . . .	107

## I. Fundamentals of Fourier Transform

### 1. Fourier Transform Formula

Given a function,  $x(t)$ , the integral

$$X_1(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt \quad (1)$$

is called the Fourier integral or Fourier Transform of  $x(t)$

The inversion formula of (1) is

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X_1(\omega) e^{j\omega t} d\omega \quad (2)$$

Substituting the basic relation,

$$\omega = 2\pi f$$

into (1) and (2) respectively, we obtain

$$X_1(2\pi f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi f t} dt \quad (1a)$$

$$x(t) = \int_{-\infty}^{\infty} X_1(2\pi f) e^{j2\pi f t} df \quad (2a)$$

Let

$$X_1(2\pi f) = X(f); \quad (3)$$

we have finally:

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi f t} dt \quad (1b)$$

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi f t} df \quad (2b)$$

Equations (1b) and (2b) are our two basic relations representing the Fourier Transform.

### 2. Exponential Function

Let us take an exponential function as an example to show

how to use the transform.

First, we substitute the function  $x(t) = e^{-at} u(t)$  shown in Fig. 1, into (1b)

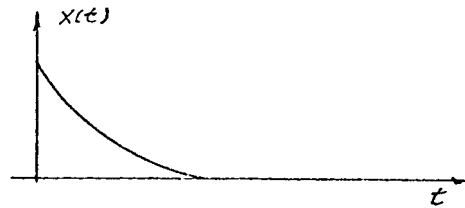


Fig. 1.

$$X(f) = \int_{-\infty}^{\infty} e^{-at} u(t) e^{-j2\pi ft} dt \triangleq \mathcal{F} \{ e^{-at} u(t) \} = \frac{1}{a + j2\pi f} \quad (4)$$

The symbol " $\mathcal{F}$ " corresponds the transformation:

$$\mathcal{F} \{ (*) \} = \int_{-\infty}^{\infty} (*) e^{-j2\pi ft} dt$$

It means "taking the Fourier Transform of (\*)".

or we can write into the following table form

$\frac{x(t)}{e^{-at} u(t)}$	$\frac{X(f)}{\frac{1}{a + j2\pi f}}$
-----------------------------	--------------------------------------

Secondly, we take the inverse Fourier Transform of (4), or

$$\mathcal{F}^{-1} X(f) = x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi ft} df \quad (5)$$

which is the application of (2b).

Substitution yields

$$x(t) = \int_{-\infty}^{\infty} \frac{e^{j2\pi ft}}{a + j2\pi f} df$$

Rearrangement gives

$$\begin{aligned} &= \int_{-\infty}^{\infty} \frac{e^{j2\pi ft}}{2\pi j \left( f - j \frac{a}{2\pi} \right)} df \\ &= \frac{1}{2\pi j} \int_{-\infty}^{\infty} \frac{e^{j2\pi ft}}{f - j \frac{a}{2\pi}} df \end{aligned} \quad (5a)$$

For evaluating (5a), we usually apply the contour integration of

$$\frac{1}{2\pi j} \oint \frac{e^{jt2\pi f}}{f - j\frac{a}{2\pi}} df \quad (6)$$

For  $t > 0$ , we take  $\Gamma_1 \sim \Gamma_2$  contour for the integration.

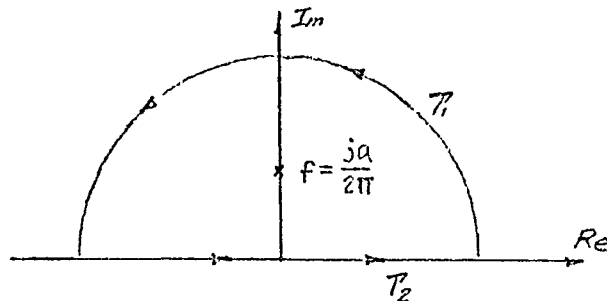


Fig. 2.

Function (6) can be easily determined by the fundamental residue theorem.

$$\begin{aligned} \frac{1}{2\pi j} \oint_{\Gamma_1 \Gamma_2} \frac{e^{jt2\pi f}}{f - j\frac{a}{2\pi}} df & \quad (7) \\ &= \frac{1}{2\pi j} (2\pi j \sum \text{Res.}) \end{aligned}$$

$$= e^{-at} \quad (7a)$$

However, (7) can be decomposed into two parts:

$$\frac{1}{2\pi j} \int_{\Gamma_1} \frac{e^{jt2\pi f}}{f - j\frac{a}{2\pi}} df + \frac{1}{2\pi j} \int_{\Gamma_2} \frac{e^{jt2\pi f}}{f - j\frac{a}{2\pi}} df \quad (8)$$

The first term is equal to zero\* while the second term is identical to (5)

Then, we obtain

$$\frac{1}{2\pi j} \int_{-\infty}^{\infty} \frac{e^{jt2\pi f}}{f - j\frac{a}{2\pi}} df = e^{-at} \quad (9)$$

\* see section VIII Jordan's Lemma



On the other hand, for  $t < 0$ , we take the contour of integration,  $\Gamma_3 \cup \Gamma_2$  as shown in Fig. 3.

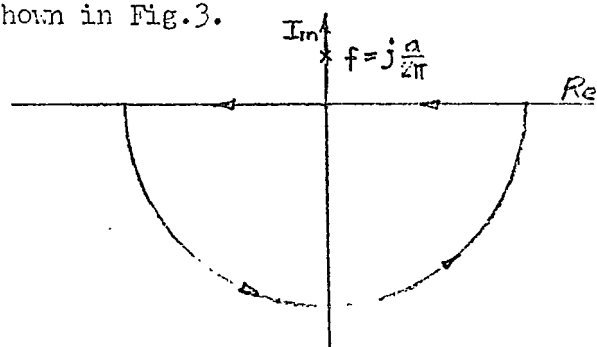


Fig. 3.

It is obvious that the result is equal to zero. Therefore, we proved Pair (4a).

### 3. Symmetrical exponential function

Verify the following transform pair:

$x(t)$	$X(f)$	(10)
$e^{-a t }$	$\frac{2a}{a^2 + 4\pi^2 f^2}$	

The time function is shown in Fig. 4

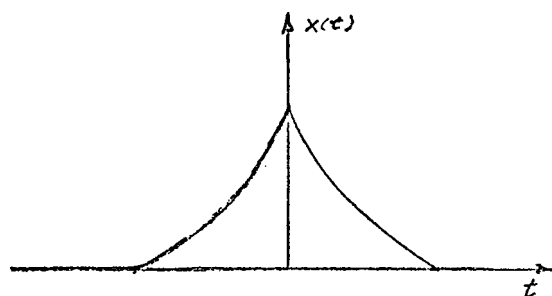


Fig. 4.

Its Fourier transform can be evaluated by direct substitution.

$$\begin{aligned}
 X(f) &= \int_{-\infty}^{\infty} e^{-a|t|} e^{-j2\pi ft} dt \\
 &= \int_{-\infty}^0 e^{at} e^{-j2\pi ft} dt + \int_0^{\infty} e^{-at} e^{-j2\pi ft} dt \\
 &= \frac{2a}{a^2 + 4\pi^2 f^2}.
 \end{aligned} \tag{11}$$

The inverse Fourier Transform of the frequency function is relatively involved. We write:

$$\begin{aligned}
 x(t) &= \int_{-\infty}^{\infty} \frac{2ae^{j2\pi ft}}{a^2 + 4\pi^2 f^2} df \\
 &= \frac{a}{2\pi^2} \int_{-\infty}^{\infty} \frac{e^{j2\pi ft}}{(f + j\frac{a}{2\pi})(f - j\frac{a}{2\pi})} df
 \end{aligned} \tag{12}$$

Again we use contour integration.

For  $t > 0$ , applying the same reasoning used in the last example shown in Fig. 2, we obtain:

$$\begin{aligned}
 x(t) &= \frac{a}{2\pi^2} \left[ 2\pi j \frac{e^{j2\pi \cdot \frac{a}{2\pi}}}{j\frac{a}{2\pi} \cdot 2} \right] \\
 &= e^{-at}
 \end{aligned} \tag{13}$$

For  $t < 0$ , if we take the contour as shown in Fig. 5,

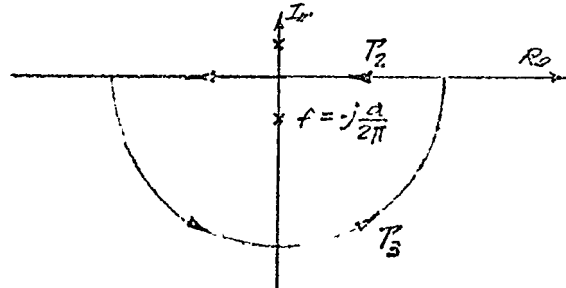


Fig. 5.

we have

$$\oint X(f) e^{2\pi f t} df = \frac{a}{2\pi^2} \left[ 2\pi j \frac{e^{(jt2\pi)(-j\frac{a}{2\pi})}}{-j \frac{a}{2\pi} \cdot 2} \right] = -e^{at} \quad (14)$$

The same integration can also be evaluated as follows

$$\begin{aligned} & \oint X(f) e^{2\pi f t} df \\ &= \int_{\Gamma_2} X(f) e^{jt2\pi f} df + \int_{\Gamma_3} X(f) e^{jt2\pi f} df \end{aligned}$$

Considering the second term being zero, we obtain

$$\begin{aligned} &= \int_{-\infty}^{\infty} X(f) e^{jt2\pi f} df \\ &= - \int_{-\infty}^{\infty} X(f) e^{jt2\pi f} df \\ &= -x(t). \end{aligned} \quad (15)$$

Comparing (14) and (15), the following relation is established,

$$x(t) = e^{at} \quad (16)$$

Based on the results shown in (13) and (16), Pair (10) is proved.

4. Derive the pair for a unit impulse

$x(t)$	$X(f)$
$\delta(t-a)$	$e^{-j2\pi fa}$

(17)

When  $\delta(t-a)$  is defined as follows:

$$\delta(t-a): \begin{cases} =0, & t \neq a \\ \int_{-\infty}^{\infty} f(t)\delta(t-a)dt=f(a) \end{cases} \quad (18)$$

The Fourier Transform is obtained as

$$\begin{aligned} X(f) &= \int_{-\infty}^{\infty} \delta(t-a)e^{-j2\pi ft} dt \\ &= e^{-j2\pi fa} \end{aligned}$$

It is seen that Pair (17) is almost verified by definition.

### 5. A Train of Unit Impulses

A very useful pair which is for an infinite pulse train is given below:

$x(t)$	$X(f)$
$\sum_{n=-\infty}^{n=\infty} \delta(t-n\Delta t)$	$\sum_{n=-\infty}^{n=\infty} e^{-j2\pi fn\Delta t}$

(19)

Repeatedly using (12), we can easily prove the pair.

### 6. Weighted impulse train

The Fourier Transform of a weighted pulse train plays an important role in signal analysis. The pair is as follow:

$x(t)$	$X(f)$
$\Delta t \sum_{n=-\infty}^{\infty} \delta(t-n\Delta t)$ $=q(t)$	$\sum_{n=-\infty}^{\infty} \delta(f - \frac{n}{\Delta t})$ $=Q(f)$

(20)

The proof of this pair is not very straightforward but can be shown rigorously:

Starting with the left hand side of (20),

$$\begin{aligned}
 \mathcal{F} \Delta t \sum_{n=-\infty}^{\infty} \delta(t-n\Delta t) &= \int_{-\infty}^{\infty} \Delta t \sum_{n=-\infty}^{\infty} \delta(t-n\Delta t) e^{-j2\pi f t} dt \\
 &= \Delta t \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} \delta(t-n\Delta t) e^{-j2\pi f t} dt \\
 &= \Delta t \sum_{n=-\infty}^{\infty} e^{-j2\pi n \Delta t f}
 \end{aligned}
 \tag{21}$$

by using (17),

Then we consider the right hand side of (20);

$$\sum_{n=-\infty}^{\infty} \delta(f - \frac{n}{\Delta t})$$

It is noted that the function is periodic with the period,  $\frac{1}{\Delta t}$ , as shown in Fig. 6,

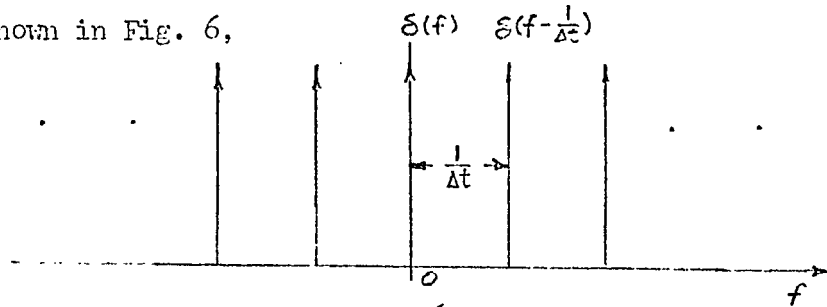


Fig. 6.

Because the periodic nature, we can naturally represent it as a Fourier series, or

$$\delta\left(f - \frac{n}{\Delta t}\right) = \sum_{n=-\infty}^{\infty} C_n e^{-j2\pi n \Delta t f} \quad (22)$$

where

$$C_n = \frac{1}{\Delta t} \int_0^{\Delta t} \delta(f) e^{j2\pi n \Delta t f} df \quad (23)$$

using (18),

$$\begin{aligned} &= \Delta t \cdot e^{j2\pi n \Delta t \cdot 0} \\ &= \Delta t (1) \\ &= \Delta t \end{aligned} \quad (24)$$

therefore,

$$\sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{\Delta t}\right) = \sum_{n=-\infty}^{\infty} \Delta t e^{-j2\pi n \Delta t f} \quad (22a)$$

The right hand sides of (21) and (22a) are identical. Their left hand sides must be equal to each other, then we have

$$\int \Delta t \sum_{n=-\infty}^{\infty} \delta(t - n \Delta t) = \sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{\Delta t}\right) \quad (20)$$

So pair (20) is proved.

## 7. Weighted sampling function

The product of an arbitrary function,  $x(t)$ , and the weighted impulse train function,  $q(t)$ <sup>2</sup>, is particularly useful in our later derivations of the Fourier method. We would investigate the nature of the product in detail.

Let

$$x(t) \cdot q(t) = y(t) \quad (23)$$

where  $y(t)$  can be called a weighted sampling function.

Substituting  $q(t)$  gives

$$x(t) \cdot \sum_{n=-\infty}^{\infty} \Delta t \delta(t - n \Delta t) = y(t) \quad (23a)$$

Fourier transforming (23a), we have

$$X(f) * Q(f) = Y(f) \quad (24)$$

where "\*" denotes Fourier convolution.

Substitution (20) into (24) yields

$$X(f) * \sum_{n=-\infty}^{n=\infty} \delta\left(f - \frac{n}{\Delta t}\right) = Y(f) \quad (25)$$

If  $X(f)$  is known as shown in Fig. 8.

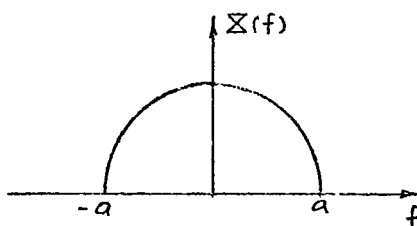


Fig. 8.

and we know that  $Q(f)$  is expressed by Fig. 6,  $Y(f)$  can be obtained by the convolution of the curves shown in Fig. 8 and Fig. 6. The result,  $Y(f)$ ,

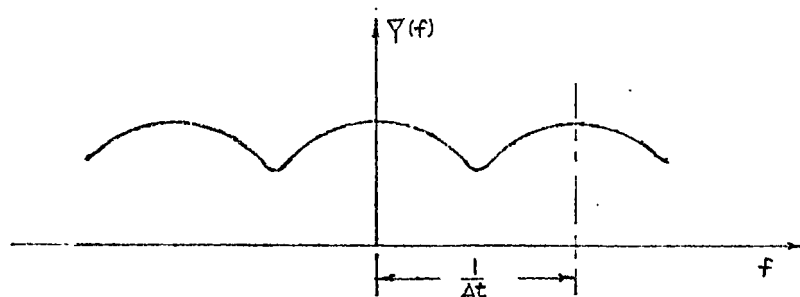


Fig. 9.

would look like the curve shown in Fig. 9. If  $\Delta t$  decreases, the period of the wave will increase accordingly. When  $\frac{1}{\Delta t}$  of Fig. 6 is greater than  $2a$  of Fig. 8, function  $Y(f)$  becomes separate pulses as shown in Fig. 10.

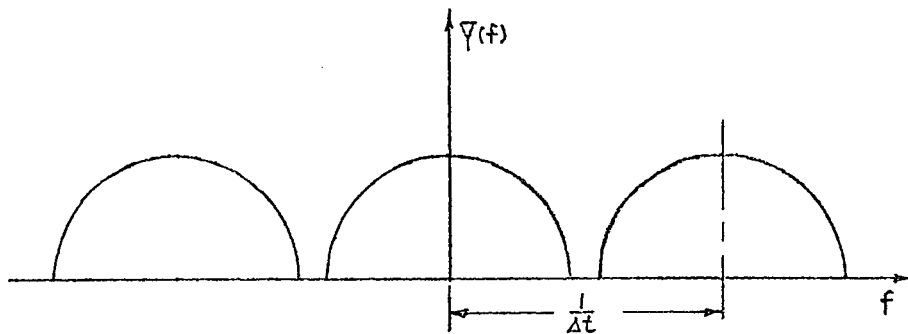


Fig. 10.

Then the relationship between  $X(f)$  and  $Y(f)$  would be as follows

$$\begin{cases} X(f)=Y(f), & \frac{1}{2\Delta t} \leq f \leq \frac{1}{2\Delta t} \\ X(f)=0, & \text{otherwise} \end{cases} \quad (26)$$

In other words,  $Y(f)$  is a periodic function, while  $X(f)$  is an aperiodic one. However, in the interval  $\frac{1}{2\Delta t} \leq f \leq \frac{1}{2\Delta t}$ ,  $X(f)$  and  $Y(f)$  are identical. In the Fourier transform method, we take the advantage of this relation. Instead of finding  $X(f)$  directly, we evaluate  $Y(f)$  and take one period of  $Y(f)$  as the answer,  $X(f)$ .



## II. Discrete Fourier Transform

### 1. Derivation of Discrete Formula

We rewrite equation (23) of section I,

$$y(t) = x(t) \cdot \sum_{n=-\infty}^{\infty} \Delta t \delta(t - n\Delta t) \quad (1)$$

$$= \sum_{n=-\infty}^{\infty} x(t) \Delta t \delta(t - n\Delta t) \quad (1a)$$

Since

$$x(t) \delta(t - n\Delta t) = x(n\Delta t) \delta(t - n\Delta t)$$

$$y(t) = \sum_{n=-\infty}^{\infty} x(n\Delta t) \Delta t \delta(t - n\Delta t) \quad (2)$$

If  $x(t)$  belongs to a special class which has the following properties:

$$\left. \begin{aligned} x(t) &= 0, & t < 0 \\ x(t) &= \text{negligible}, & t \geq T, \text{ where } T \text{ is a constant,} \\ \text{and } x(t) &= \text{finite values, } & 0 \leq t \leq T \end{aligned} \right\} \quad (3)$$

Graphically, this class looks like the curve shown in Fig. 11.

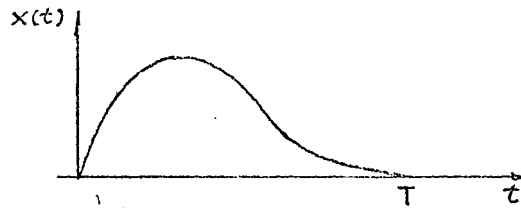


Fig. 11.

Equation (2) can be approximated by

$$y(t) = \sum_{n=0}^{N-1} x(n\Delta t) \Delta t \delta(t - n\Delta t) \quad (4)$$

where

$$N = \frac{T}{\Delta t} \quad (5)$$

Fourier transforming (4), we obtain

$$Y(f) = \int_{-\infty}^{\infty} \sum_{n=0}^{N-1} x(n\Delta t) \Delta t \delta(t - n\Delta t) e^{-j2\pi f t} dt$$

or

$$Y(f) = \sum_{n=0}^{N-1} x(n\Delta t) \Delta t e^{-j2\pi f n \Delta t} \quad (6)$$

As we mentioned before,  $Y(f)$  is a periodic function, with period  $\frac{1}{\Delta t}$ . when  $Y(f)$  is determined, one period of  $Y(f)$  is the same as the aperiodic function,  $X(f)$ .

If we only ask the numerical  $X(f)$ , equation (6) then can be used in the following way:

Let

$$f = \frac{k}{T} \quad \text{and} \quad k=0 \quad \text{first and substitute them into (6).}$$

The value  $Y(0)$  is found:

$$Y(0) = \sum_{n=0}^{N-1} x(n\Delta t) \Delta t e^0 \quad (7)$$

then let  $k=1$ , repeating the process yields

$$Y\left(\frac{1}{T}\right) = \sum_{n=0}^{N-1} x(n\Delta t) \Delta t e^{-j2\pi \frac{1}{T} n \Delta t} \quad (8)$$

In general

$$Y\left(\frac{k}{T}\right) = \sum_{n=0}^{N-1} x(n\Delta t) \Delta t e^{-j2\pi \frac{k}{T} n \Delta t} \quad (9)$$

noting that  $\Delta t = \frac{T}{N}$

Equation (7)~(9) can also be written as

$$Y\left(\frac{k}{T}\right) = \sum_{n=0}^{N-1} x(n\Delta t) \Delta t e^{-j2\pi \frac{k}{N} n} \quad k=0,1,\dots,N-1. \quad (10)$$

Equation (10) is called the Discrete Fourier Transform.

## 2. Discrete Formula in Matrix Form

For example, taking  $N=3$ , the expansion of (10) should be as follows:

$$\begin{aligned}
Y\left(\frac{0}{T}\right) &= x(0\Delta t)\Delta t e^{-j2\pi \cdot 0 \cdot 0 \cdot \frac{1}{3}} + x(1\Delta t)\Delta t e^{-j2\pi \cdot 0 \cdot 1 \cdot \frac{1}{3}} + x(2\Delta t)\Delta t e^{-j2\pi \cdot 0 \cdot 2 \cdot \frac{1}{3}} \\
Y\left(\frac{1}{T}\right) &= x(0\Delta t)\Delta t e^{-j2\pi \cdot 1 \cdot 0 \cdot \frac{1}{3}} + x(1\Delta t)\Delta t e^{-j2\pi \cdot 1 \cdot 1 \cdot \frac{1}{3}} + x(2\Delta t)\Delta t e^{-j2\pi \cdot 1 \cdot 2 \cdot \frac{1}{3}} \\
Y\left(\frac{2}{T}\right) &= x(0\Delta t)\Delta t e^{-j2\pi \cdot 2 \cdot 0 \cdot \frac{1}{3}} + x(1\Delta t)\Delta t e^{-j2\pi \cdot 2 \cdot 1 \cdot \frac{1}{3}} + x(2\Delta t)\Delta t e^{-j2\pi \cdot 2 \cdot 2 \cdot \frac{1}{3}}
\end{aligned}
\tag{11}$$

A corresponding matrix form is

$$\begin{bmatrix} Y\left(\frac{0}{T}\right) \\ Y\left(\frac{1}{T}\right) \\ Y\left(\frac{2}{T}\right) \end{bmatrix} = \begin{bmatrix} e^{-j2\pi \cdot 0 \cdot 0 \cdot \frac{1}{3}} & e^{-j2\pi \cdot 0 \cdot 1 \cdot \frac{1}{3}} & e^{-j2\pi \cdot 0 \cdot 2 \cdot \frac{1}{3}} \\ e^{-j2\pi \cdot 1 \cdot 0 \cdot \frac{1}{3}} & e^{-j2\pi \cdot 1 \cdot 1 \cdot \frac{1}{3}} & e^{-j2\pi \cdot 1 \cdot 2 \cdot \frac{1}{3}} \\ e^{-j2\pi \cdot 2 \cdot 0 \cdot \frac{1}{3}} & e^{-j2\pi \cdot 2 \cdot 1 \cdot \frac{1}{3}} & e^{-j2\pi \cdot 2 \cdot 2 \cdot \frac{1}{3}} \end{bmatrix} \begin{bmatrix} x(0\Delta t)\Delta t \\ x(1\Delta t)\Delta t \\ x(2\Delta t)\Delta t \end{bmatrix}
\tag{12}$$

For clarity, we change (10) to the following notations

$$\begin{bmatrix} S_0 \\ S_1 \\ S_2 \end{bmatrix} = \begin{bmatrix} \underline{0 \alpha} & \underline{1 \alpha} & \underline{2 \alpha} \\ \underline{0 \alpha} & \underline{1 \alpha} & \underline{2 \alpha} \\ \underline{0 \alpha} & \underline{2 \alpha} & \underline{1 \alpha} \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \\ A_2 \end{bmatrix}
\tag{12a}$$

where

$$\begin{aligned}
S_k &= Y\left(\frac{k}{T}\right) \\
\alpha &= -\frac{2\pi}{N} \quad \text{or} \quad \underline{\alpha} = e^{-j\frac{2\pi}{N}}
\end{aligned}
\tag{13}$$

$$A_k = x(k\Delta t)\Delta t$$

We can write a general formula of (12a).

$$\begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ \vdots \\ S_{n-1} \end{bmatrix} = \begin{bmatrix} \underline{0 \alpha} & \underline{0 \alpha} & \underline{0 \alpha} & \dots & \underline{0 \alpha} \\ \underline{0 \alpha} & \underline{1 \alpha} & \underline{2 \alpha} & \dots & \underline{(n-1) \alpha} \\ \underline{0 \alpha} & \underline{2 \alpha} & \underline{4 \alpha} & \dots & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \underline{0 \alpha} & \underline{(n-1) \alpha} & \dots & \dots & \underline{(n-1)^2 \alpha} \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \\ A_2 \\ \vdots \\ A_{n-1} \end{bmatrix} \quad (14)$$

First of all, we should find  $A_n$ ,

$$A_n = \Delta t \times (n\Delta t) \quad (13)$$

we use  $t=0.5$ , and  $N=8$ ,  $T=4$ , substituting the given data into (14), we have

$$\begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -\frac{2\pi}{8} & -\frac{4\pi}{8} & -\frac{6\pi}{8} & -\frac{8\pi}{8} & -\frac{10\pi}{8} & -\frac{12\pi}{8} & -\frac{14\pi}{8} \\ 1 & -\frac{4\pi}{8} & -\frac{8\pi}{8} & -\frac{12\pi}{8} & -\frac{16\pi}{8} & \dots & \dots & \dots \\ 1 & -\frac{6\pi}{8} & -\frac{12\pi}{8} & -\frac{18\pi}{8} & \dots & \dots & \dots & \dots \\ 1 & -\frac{8\pi}{8} & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & -\frac{10\pi}{8} & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & -\frac{12\pi}{8} & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & -\frac{14\pi}{8} & \dots & \dots & \dots & \dots & -\frac{98\pi}{8} & \dots \end{bmatrix} \begin{bmatrix} 0 \\ 1.199 \\ 1.163 \\ 0.867 \\ 0.585 \\ 0.377 \\ 0.286 \\ 0.146 \end{bmatrix} \quad (15)$$

or

$$S_k = \sum_{n=0}^7 A_n \angle n\alpha \quad k=0,1,\dots,7. \quad \text{where}$$

$$\alpha = -\frac{2\pi}{8}$$

The first equation of (15) reads:

$$S_0 = 0 + 1.199 + 1.163 + 0.867 + 0.585 + 0.377 + 0.286 + 0.146 = 4.62$$

which is the magnitude of the frequency spectrum when  $f=0$ .

Theoretically, it should be found as

$$\left. \frac{10}{j2\pi f+1} - \frac{10}{j2\pi f+2} \right|_{f=0} = 5. \quad (16)$$

$S_1, S_2, \dots$  and  $S_7$  can be evaluated similarly.

#### 4. Vector Diagram Interpretation

Equation (14) in general, or equation (15) in particular can be geometrically interpreted as follows:

We rewrite the equation of (14a) in its expansion form, for example,

$$S_1 = A_0 e^{-j\frac{2\pi}{N} \cdot 0 \cdot 1} + A_1 e^{-j\frac{2\pi}{N} \cdot 1 \cdot 1} + \dots + A_7 e^{-j\frac{2\pi}{N} \cdot 1 \cdot 7}$$

$$= A_0 \underline{0 \alpha} + A_1 \underline{1 \alpha} + \dots + A_7 \underline{7 \alpha}$$

If each term is considered as a vector as shown in Fig. 13a,  $S_1$  becomes a summation of many vectors or the resultant vector.

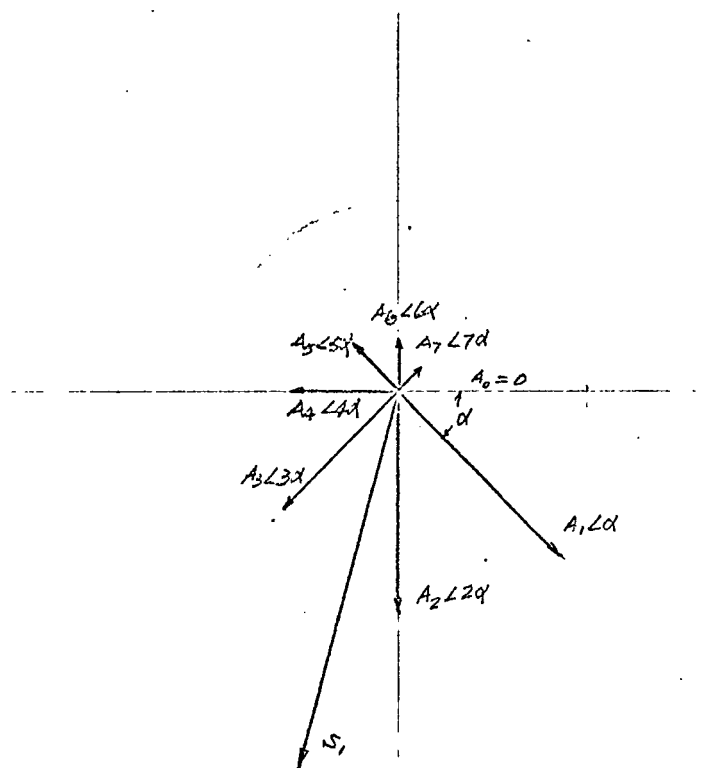
If  $S_2$  is desired, we simply double the angle of each vector and then take the summation. The geometrical correspondence is shown in Fig. 13b.

Therefore, from the geometrical viewpoint, equation (14) can be stated as follows:

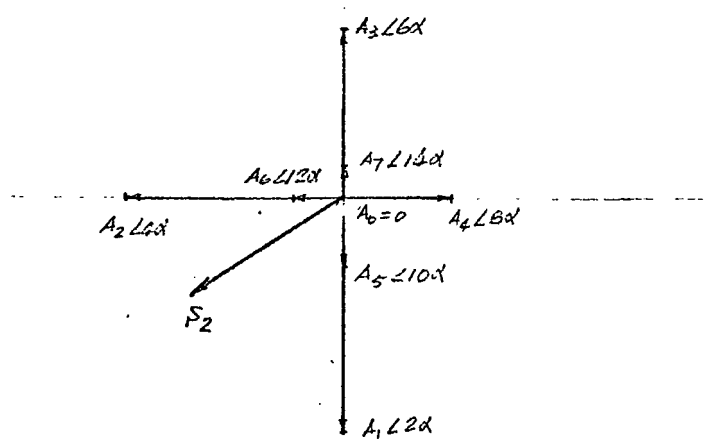
The frequency spectrum matrix,  $[S_k]$ , is equal to the "angular" matrix  $[\underline{nk\alpha}]$  times the "magnitude" matrix  $[A_n]$ . Each term of the square matrix is a pure angle while each term of the last column matrix is a pure scalar. However, each term of the spectrum matrix is a complex number in general.

#### 5. Computer Program for Discrete Fourier Transform

Based on equation (14), a computer program is written and an example is tested. In the first run, we use  $N=8$ , the answer coincides with that obtained from the graphical method. In the second run, we use  $N=1024$ . The comparison of the results is shown in table 1.



(a)



(b)

Fig. 13

Table 1

F	Absolute Value	
	N=8	N=1024
-1.00000	0.59835	0.238267
-0.75000	0.67242	0.405277
-0.50000	0.98687	0.813938
-0.25000	2.07848	2.110291
0.00000	4.56685	4.996576
0.25000	2.07848	2.110291
0.50000	0.98688	0.405279
0.75000	0.67243	0.405279

Fig. 14 is the frequency spectrum of the example problem.

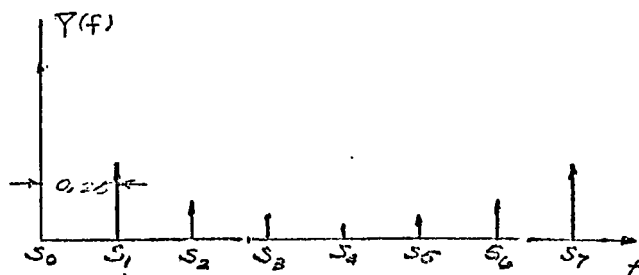


Fig. 14.

Comparing this spectrum with that shown in Fig. 10, we know that the second half starting  $S_4$  to  $S_7$  of Fig. 14, should be moved ahead of  $S_0$ , or be shown as in Fig. 15. This means that we obtain  $X(f)$  from  $Y(f)$ .

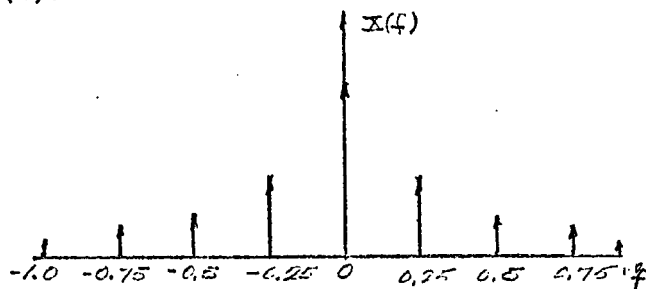


Fig. 15.



```

C      PROGRAM #1
C      DISCRETE FOURIER TRANSFORM
C      IF ND EQUAL 0 FOR FUNCTION, OTHERWISE FOR INPUT DATA
C      T IS THE TOTAL TIME, 2**M IS THE TOTAL POINTS
C      FUNCTION
      FUN(T)=10.*(EXP(-T)-EXP(-2.*T))
C      MAIN PROGRAM
      COMPLEX S,DES
      DIMENSION A(5000)
      READ(5,400) M,T,ND
400    FORMAT(I10,F15.5,I5)
      N=2**M
      DET=T/N
      IF(ND)10,19,10
10     READ(5,402) (A(I),I=1,N)
402    FORMAT(5F16.6)
      GO TO 29
19     CONTINUE
      DO 20 I=1,N
20     A(I)=FUN((I-1)*DET)
29     WRITE(6,403)
403    FORMAT(15X,'FREQUENCY',8X,'REAL PART',6X,'IMAGINARY PART',5X,'ABSOLUTE VALUE')
      MN=N/2
      DO 30 I=1,MN
      S=(0.,0.)
      DO 40 J=1,N
40     S=S+DET*CMPLX(A(J),0.)*CEXP((0.,-1.)*2.*3.1416*(MN+I-1)*(J-1)/N)
      F=(-MN+I-1)/T
      ABL=CABS(S)
30     WRITE(6,404) F,S,ABL
404    FORMAT(8X,4F16.6)
      DO 50 I=1,MN
      S=(0.,0.)
      DO 60 J=1,N
60     S=S+DET*CMPLX(A(J),0.)*CEXP((0.,-1.)*2.*3.1416*(I-1)*(J-1)/N)
      F=(I-1)/T
      ABL=CABS(S)
50     WRITE(6,404) F,S,ABL
      STOP
      END

```

### III. Inverse Discrete Fourier Transform

#### 1. The Inverse Formula and its Proof

We rewrite the discrete Fourier formula,

$$S_k = \sum_{n=0}^{N-1} A_n e^{-j \frac{2\pi nk}{N}} \quad k=0,1,2,\dots,N-1. \quad (1)$$

and we would like to claim that the inverse discrete Fourier formula is as follows:

$$A_n = \frac{1}{N} \sum_{k=0}^{N-1} S_k e^{j \frac{2\pi nk}{N}} \quad n=0,1,2,\dots,N-1. \quad (2)$$

A proof of equation (2) is offered below:

Starting from the right hand side of (2)

$$\frac{1}{N} \sum_{k=0}^{N-1} S_k e^{j \frac{2\pi nk}{N}} \quad (3)$$

we substitute  $S_k$  of (1) into it.

$$= \frac{1}{N} \sum_{k=0}^{N-1} \left[ \sum_{m=0}^{N-1} A_m e^{-j \frac{2\pi mk}{N}} \right] e^{j \frac{2\pi nk}{N}}.$$

noting that we changed the subscript  $n$  into  $m$ , then interchange the summations

$$\begin{aligned} &= \frac{1}{N} \sum_{m=0}^{N-1} A_m \sum_{k=0}^{N-1} e^{-j \frac{2\pi mk}{N}} e^{j \frac{2\pi nk}{N}} \\ &= \frac{1}{N} \sum_{m=0}^{N-1} A_m \sum_{k=0}^{N-1} e^{-j \frac{2\pi (m-n)k}{N}} \end{aligned} \quad (4)$$

$$= \frac{1}{N} N A_n \quad (5)$$

$$= A_n \quad (6)$$

From (4) to (5), we used the following relation:

$$\sum_{k=0}^{N-1} e^{-j \frac{2\pi(m-n)k}{N}} = \begin{cases} 0 & , \text{ if } m \neq n \\ N & ; \text{ if } m = n \end{cases} \quad (7)$$

Therefore, equation (2) is proved.

2. The matrix form of (2) is written as follows

$$\begin{bmatrix} A_0 \\ A_1 \\ A_2 \\ \vdots \\ A_{N-1} \end{bmatrix} = \begin{bmatrix} \underline{-0 \cdot \alpha} & \underline{-0 \cdot \alpha} & \underline{-0 \cdot \alpha} & \dots & \underline{-0 \cdot \alpha} \\ \underline{-0 \cdot \alpha} & \underline{-1 \cdot \alpha} & \underline{-2 \cdot \alpha} & \dots & \underline{-(N-1) \cdot \alpha} \\ \underline{-0 \cdot \alpha} & \underline{-2 \cdot \alpha} & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \underline{-0 \cdot \alpha} & \underline{-(N-1) \cdot \alpha} & \dots & \dots & \underline{-(N-1)^2 \cdot \alpha} \end{bmatrix} \begin{bmatrix} S_0/N \\ S_1/N \\ S_2/N \\ \vdots \\ S_{N-1}/N \end{bmatrix} \quad (8)$$

The square matrix of (8) is similar to that of (2); but each term is with a negative sign.

We also can use vectors to express the operation involved in (8); however, because  $S_k$  is a complex number, the picture is not as simple as before.

3. Computer Program for Inverse Discrete Fourier Transform

```

C      PROGRAM #2
C      INVERSE DISCRETE FOURIER TRANSFORM
C      IF ND EQUAL 0 FOR FUNCTION, OTHERWISE FOR INPUT DATA
C      T IS THE TOTAL TIME, 2**M IS THE TOTAL SAMPLE POINT
C      MAIN PROGRAM
      COMPLEX S, DES, A, CFUN, SL
      DIMENSION S(5000)
      READ(5,400) M, T, ND
400  FORMAT(110,F15.5,I5)
      N=2**M
      DET=T/N
      DES=(0.,1.)*2.*3.1416/T
      IF(ND) 10,19,10
10  READ(5,402) (S(I),I=1,N)
      WRITE(5,401)
401  FORMAT(15X,'INPUT DATA')
      WRITE(6,404) (S(I),I=1,N)
402  FORMAT(8F10.5)
      GO TO 29
      GO TO 29
19  MN=N/2
      DO 20 I=1,MN
      S(I)=CFUN((I-1)*DES)
20  S(MN+I)=CFUN((-MN+I-1)*DES)
29  WRITE(6,403)
403  FORMAT(15X,'TIME',10X,'X(T)')
      DO 30 I=1,N
      A=(0.,0.)
      DO 40 J=1,N
40  A=A+S(J)*CEXP((0.,1.)*2.*3.1416*(I-1)*(J-1)/N)
      TN=(I-1)*DET
      AR=REAL(A)/T
30  WRITE(6,404) TN,AR
404  FORMAT(10X,2F16.6)
      STOP
      END

```

```

C      FUNCTION SUBROUTINE
      COMPLEX FUNCTION CFUN(SL)
      COMPLEX SL
      CFUN=((10.,0.)/((SL+(1.,0.))*(SL+(2.,0.)))
      RETURN
      END

```

TIME	X(T)
0.000000	0.665416
0.500000	2.484723
1.000000	2.392231
1.500000	1.772028
2.000000	1.206113
2.500000	0.746917
3.000000	0.523303
3.500000	0.209283

## IV. Fast Fourier Transform — Decimation in Time

## 1. Basic Algorithm

The Fast Fourier Transform is an algorithm that makes possible the computation of the Discrete Fourier Transform of a time series more rapidly than other algorithms available. The two basic forms of the Fast Fourier Transform are (1) Decimation in time and (2) Decimation in frequency. We will explain the first technique as follows:

Consider (1/a) of section II again:

$$S_k = \sum_{n=0}^{N-1} A_n e^{-j \frac{2\pi}{N} \cdot nk} \quad (1)$$

For convenience, let  $N$  be even. Expanding yields

$$\begin{aligned} S_k = & A_0 e^{-j \frac{2\pi \cdot 0 \cdot k}{N}} + A_1 e^{-j \frac{2\pi \cdot 1 \cdot k}{N}} + A_2 e^{-j \frac{2\pi \cdot 2 \cdot k}{N}} + \dots \\ & + A_{N-2} e^{-j \frac{2\pi(N-2)k}{N}} + A_{N-1} e^{-j \frac{2\pi(N-1)k}{N}} \end{aligned} \quad (1a)$$

Regrouping gives

$$\begin{aligned} S_k = & \left[ A_0 e^{-j \frac{2\pi \cdot 0 \cdot k}{N}} + A_2 e^{-j \frac{2\pi \cdot 2 \cdot k}{N}} + \dots \right] \\ & + \left[ A_1 e^{-j \frac{2\pi \cdot 1 \cdot k}{N}} + A_3 e^{-j \frac{2\pi \cdot 3 \cdot k}{N}} + \dots \right] \\ = & \sum_{r=0}^{N/2-1} A_{2r} e^{-j \frac{2\pi 2rk}{N}} + \sum_{r=0}^{N/2-1} A_{2r+1} e^{-j \frac{2\pi(2r+1)k}{N}} \\ \text{or} \quad = & \sum_{r=0}^{N/2-1} A_{2r} e^{-j \frac{2\pi rk}{N/2}} + e^{-j \frac{2\pi k}{N}} \sum_{r=0}^{N/2-1} A_{2r+1} e^{-j \frac{2\pi rk}{N/2}} \end{aligned} \quad (2)$$

Note that the first term is with the standard Discrete Fourier Transform form while the second term is a factor  $e^{-j \frac{2\pi k}{N}}$  times

that standard form. Therefore, when we let:

$$P_k = \sum_{r=0}^{N/2-1} A_{2r} e^{-j \frac{2\pi r k}{N}} \quad (3)$$

and

$$Q_k = \sum_{r=0}^{N/2-1} A_{2r+1} e^{-j \frac{2\pi r k}{N}} \quad (4)$$

Equation (2) becomes

$$S_k = P_k + e^{-j \frac{2\pi k}{N}} Q_k, \quad k=0,1,\dots,N-1. \quad (5)$$

The periodic nature of  $P_k$  and  $Q_k$  should be observed.

$$P_k = P_{k+\frac{N}{2}} \quad (6a)$$

$$Q_k = Q_{k+\frac{N}{2}} \quad (6b)$$

These two relations (6a) and (6b) are easily proved by substitution.

By using (6a) and (6b), we have the following reasoning:

$$S_k = P_k + e^{-j \frac{2\pi k}{N}} Q_k, \quad (7a)$$

When  $0 \leq k < \frac{N}{2}$

let

$$k = \frac{N}{2} + k'$$

$$S_{\frac{N}{2}+k'} = P_{\frac{N}{2}+k'} + e^{-j \frac{2\pi (\frac{N}{2}+k')}{N}} Q_{\frac{N}{2}+k'} \quad (7b)$$

when

$$N > k \geq \frac{N}{2}$$

$$S_{\frac{N}{2}+k'} = P_{k'} - e^{-j \frac{2\pi k'}{N}} Q_{k'}$$

Therefore, equation (5) can be written as two equations.

$$S_k = P_k + e^{-j\frac{2\pi k}{N}} Q_k \quad ? \quad (8)$$

$$S_{\frac{N}{2}+k} = P_k - e^{-j\frac{2\pi k}{N}} Q_k, \quad k=0,1,\dots,\frac{N}{2}-1.$$

Equation (8) is the Fast Fourier Transform algorithm.

The computation of the Discrete Fourier Transform of  $N$  samples can be reduced to computing the Discrete Fourier Transforms of two sequences of  $\frac{N}{2}$  samples each; the computation of  $P_k$  (or  $Q_k$ ) can be reduced to computation of sequences of  $\frac{N}{4}$  samples. If we let  $N=2^M$ , these reductions can be carried out continuously.

## 2. Signal Flow Graph Illustration

Suppose  $N=8$ , Equation (8) becomes

$$\begin{cases} S_k = P_k + e^{-j\frac{2\pi k}{8}} Q_k \\ S_{4+k} = P_k - e^{-j\frac{2\pi k}{8}} Q_k \end{cases} \quad k=0,1,2,3 \quad (9)$$

Equation (9) can be illustrated as a signal flow graph shown in Fig. 16.

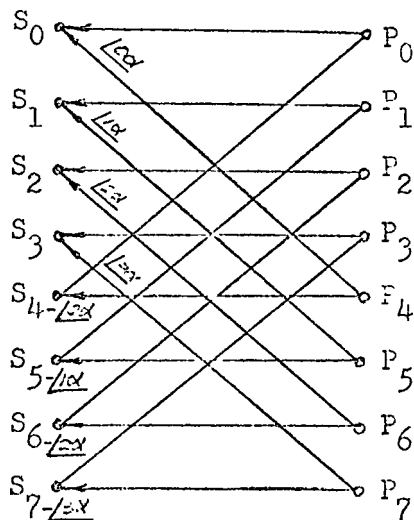


Fig. 16.

Referring to the basic definitions in (3) and (4), we can consider  $\{P_k\}$  and  $\{Q_k\}$  to be generated from the samples  $A_0, A_2, A_4, A_6$ , and the samples  $A_1, A_3, A_5, A_7$ , respectively.

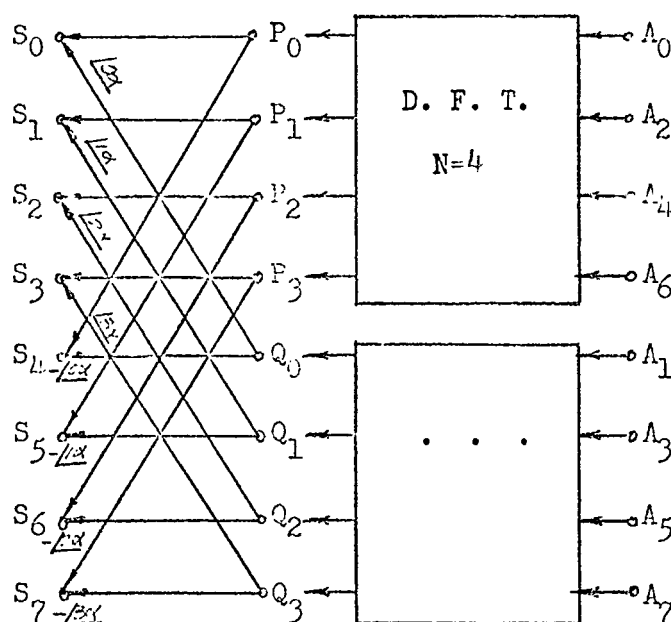


Fig. 17.

Graphically, we express the generation process in Fig. 17.

### 3. Vector-Diagram Interpretation of FFT

We have used vectors to interpret DFT in the last section. A question naturally arises : What is the vector picture which corresponds FFT?

Let us repeat Fig. 12. and reshow it as Fig. 18. We have explained that the resultant of Fig. (18a) is  $S_1$ . Now, decompose the set of vectors into two sets: namely,  $A_0, A_2, A_4$ , and  $A_6$  as a group and  $A_1, A_3, A_5$ , and  $A_7$  as another group. Then equation (9) can be regarded as the combination of the two graphs: vectors



shown in Fig. (18b) are superimposed by the vectors shown in (18d).

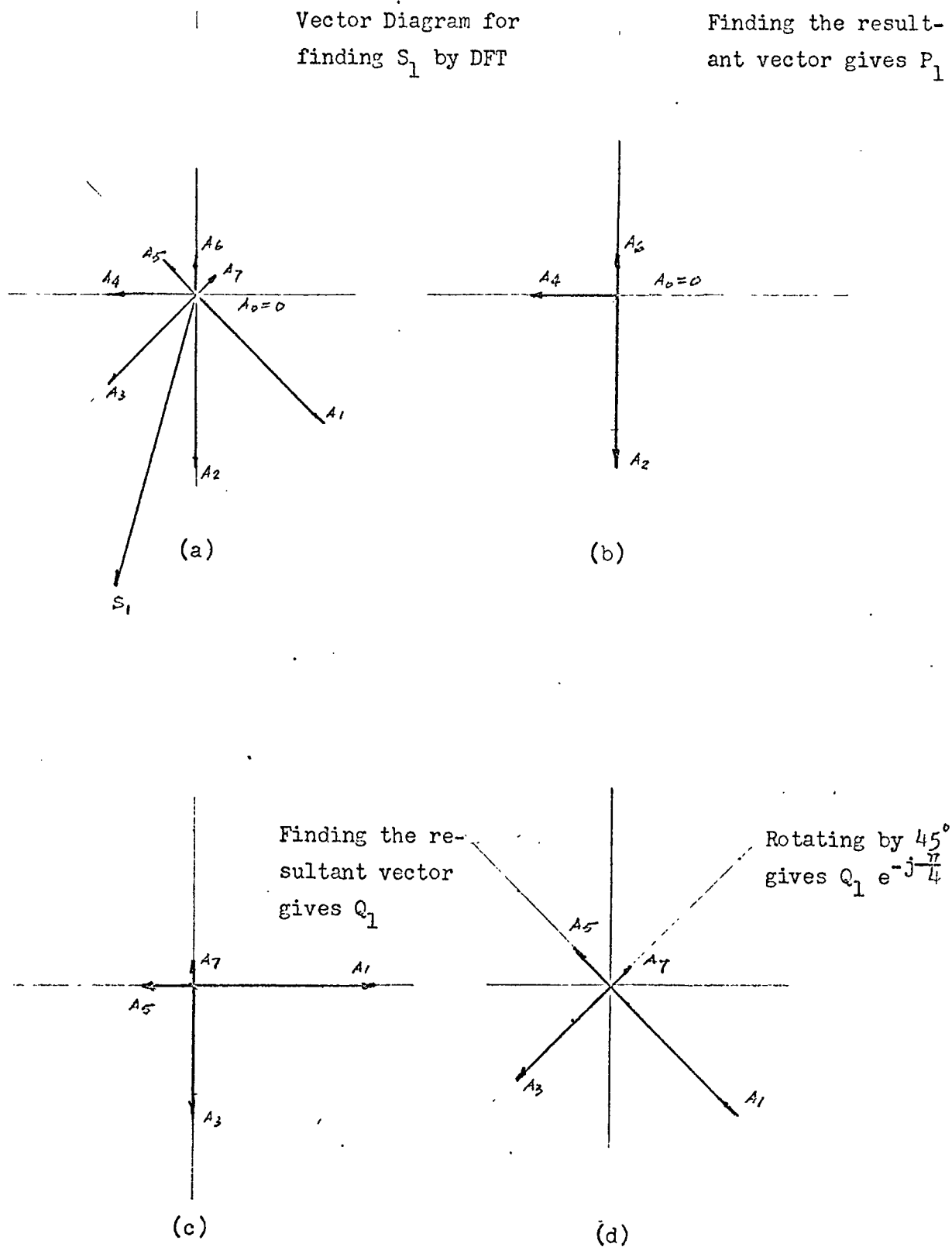


Fig. 18.

#### 4. Matrix Explanation

Assuming  $N=8$ , equation (14) of section II becomes:

$$\begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \underline{1\alpha} & \underline{2\alpha} & \underline{3\alpha} & \underline{4\alpha} & \underline{5\alpha} & \underline{6\alpha} & \underline{7\alpha} \\ 1 & \underline{2\alpha} & \underline{4\alpha} & \underline{6\alpha} & \underline{8\alpha} & \underline{10\alpha} & \underline{12\alpha} & \underline{14\alpha} \\ 1 & \underline{3\alpha} & \underline{6\alpha} & \underline{9\alpha} & \underline{12\alpha} & \underline{15\alpha} & \underline{18\alpha} & \underline{21\alpha} \\ 1 & \underline{4\alpha} & \underline{8\alpha} & \underline{12\alpha} & \underline{16\alpha} & \underline{20\alpha} & \underline{24\alpha} & \underline{28\alpha} \\ 1 & \underline{5\alpha} & \underline{10\alpha} & \underline{15\alpha} & \underline{20\alpha} & \underline{25\alpha} & \underline{30\alpha} & \underline{35\alpha} \\ 1 & \underline{6\alpha} & \underline{12\alpha} & \underline{18\alpha} & \underline{24\alpha} & \underline{30\alpha} & \underline{36\alpha} & \underline{42\alpha} \\ 1 & \underline{7\alpha} & \underline{14\alpha} & \underline{21\alpha} & \underline{28\alpha} & \underline{35\alpha} & \underline{42\alpha} & \underline{49\alpha} \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \\ A_7 \end{bmatrix} \quad (10)$$

The corresponding simultaneous equations are:

$$\begin{aligned} S_0 &= A_0 + A_1 + A_2 + A_3 + A_4 + A_5 + A_6 + A_7 \\ S_1 &= A_0 + A_1 \underline{1\alpha} + A_2 \underline{2\alpha} + A_3 \underline{3\alpha} + A_4 \underline{4\alpha} + A_5 \underline{5\alpha} + A_6 \underline{6\alpha} + A_7 \underline{7\alpha} \\ S_2 &= A_0 + A_1 \underline{2\alpha} + A_2 \underline{4\alpha} + A_3 \underline{6\alpha} + A_4 \underline{8\alpha} + A_5 \underline{10\alpha} + A_6 \underline{12\alpha} + A_7 \underline{14\alpha} \\ S_3 &= A_0 + A_1 \underline{3\alpha} + A_2 \underline{6\alpha} + A_3 \underline{9\alpha} + A_4 \underline{12\alpha} + A_5 \underline{15\alpha} + A_6 \underline{18\alpha} + A_7 \underline{21\alpha} \\ S_4 &= A_0 + A_1 \underline{4\alpha} + A_2 \underline{8\alpha} + A_3 \underline{12\alpha} + A_4 \underline{16\alpha} + A_5 \underline{20\alpha} + A_6 \underline{24\alpha} + A_7 \underline{28\alpha} \\ S_5 &= A_0 + A_1 \underline{5\alpha} + A_2 \underline{10\alpha} + A_3 \underline{15\alpha} + A_4 \underline{20\alpha} + A_5 \underline{25\alpha} + A_6 \underline{30\alpha} + A_7 \underline{35\alpha} \\ S_6 &= A_0 + A_1 \underline{6\alpha} + A_2 \underline{12\alpha} + A_3 \underline{18\alpha} + A_4 \underline{24\alpha} + A_5 \underline{30\alpha} + A_6 \underline{36\alpha} + A_7 \underline{42\alpha} \\ S_7 &= A_0 + A_1 \underline{7\alpha} + A_2 \underline{14\alpha} + A_3 \underline{21\alpha} + A_4 \underline{28\alpha} + A_5 \underline{35\alpha} + A_6 \underline{42\alpha} + A_7 \underline{49\alpha} \end{aligned} \quad (11)$$

We re-group these equations by combining the even terms and the odd terms respectively.

$$\begin{aligned}
S_0 &= A_0 + A_2 + A_4 + A_6 + A_1 + A_3 + A_5 + A_7 \\
S_1 &= A_0 + A_2 \underline{2\alpha} + A_4 \underline{4\alpha} + A_6 \underline{6\alpha} + A_1 \underline{1\alpha} + A_3 \underline{3\alpha} + A_5 \underline{5\alpha} + A_7 \underline{7\alpha} \\
S_2 &= A_0 + A_2 \underline{1\alpha} + A_4 \underline{3\alpha} + A_6 \underline{12\alpha} + A_1 \underline{2\alpha} + A_3 \underline{6\alpha} + A_5 \underline{10\alpha} + A_7 \underline{14\alpha} \\
S_3 &= A_0 + A_2 \underline{6\alpha} + A_4 \underline{12\alpha} + A_6 \underline{18\alpha} + A_1 \underline{3\alpha} + A_3 \underline{9\alpha} + A_5 \underline{15\alpha} + A_7 \underline{21\alpha} \\
S_4 &= A_0 + A_2 \underline{8\alpha} + A_4 \underline{16\alpha} + A_6 \underline{24\alpha} + A_1 \underline{4\alpha} + A_3 \underline{12\alpha} + A_5 \underline{20\alpha} + A_7 \underline{28\alpha} \\
S_5 &= A_0 + A_2 \underline{10\alpha} + A_4 \underline{20\alpha} + A_6 \underline{30\alpha} + A_1 \underline{5\alpha} + A_3 \underline{15\alpha} + A_5 \underline{25\alpha} + A_7 \underline{35\alpha} \\
S_6 &= A_0 + A_2 \underline{12\alpha} + A_4 \underline{24\alpha} + A_6 \underline{36\alpha} + A_1 \underline{6\alpha} + A_3 \underline{18\alpha} + A_5 \underline{30\alpha} + A_7 \underline{42\alpha} \\
S_7 &= A_0 + A_2 \underline{14\alpha} + A_4 \underline{28\alpha} + A_6 \underline{42\alpha} + A_1 \underline{7\alpha} + A_3 \underline{21\alpha} + A_5 \underline{35\alpha} + A_7 \underline{49\alpha}
\end{aligned} \tag{12}$$

Then writing these equations into the matrix form again, we have

$$\begin{pmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \underline{2\alpha} & \underline{4\alpha} & \underline{6\alpha} & \underline{1\alpha} & \underline{3\alpha} & \underline{5\alpha} & \underline{7\alpha} \\ 1 & \underline{1\alpha} & \underline{3\alpha} & \underline{12\alpha} & \underline{2\alpha} & \underline{6\alpha} & \underline{10\alpha} & \underline{14\alpha} \\ 1 & \underline{6\alpha} & \underline{12\alpha} & \underline{18\alpha} & \underline{3\alpha} & \underline{9\alpha} & \underline{15\alpha} & \underline{21\alpha} \\ 1 & \underline{8\alpha} & \underline{16\alpha} & \underline{24\alpha} & \underline{4\alpha} & \underline{12\alpha} & \underline{20\alpha} & \underline{28\alpha} \\ 1 & \underline{10\alpha} & \underline{20\alpha} & \underline{30\alpha} & \underline{5\alpha} & \underline{15\alpha} & \underline{25\alpha} & \underline{35\alpha} \\ 1 & \underline{12\alpha} & \underline{24\alpha} & \underline{36\alpha} & \underline{6\alpha} & \underline{18\alpha} & \underline{30\alpha} & \underline{42\alpha} \\ 1 & \underline{14\alpha} & \underline{28\alpha} & \underline{42\alpha} & \underline{7\alpha} & \underline{21\alpha} & \underline{35\alpha} & \underline{49\alpha} \end{pmatrix} \begin{pmatrix} A_0 \\ A_2 \\ A_4 \\ A_6 \\ A_1 \\ A_3 \\ A_5 \\ A_7 \end{pmatrix} \tag{13}$$

$$\text{Or in compact notation: } [S] = [W] [A] \tag{13a}$$

It is noted that the frequency vector  $\{S_0, \dots, S_7\}^T$  is in the natural order while the magnitude vector in the time domain

$\{A_0, A_2, A_4, \dots, A_7\}^T$  is not in the natural order. We call this scrambled pattern DECLARATION IN TIME.

The square matrix in (13) can be factored into two square matrices as follows

$$[W] = \begin{bmatrix} 1 & 0 & 0 & 0 & \underline{10\alpha} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \underline{1\alpha} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \underline{2\alpha} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \underline{3\alpha} \\ 1 & 0 & 0 & 0 & -\underline{10\alpha} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -\underline{1\alpha} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -\underline{2\alpha} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -\underline{3\alpha} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & \underline{2\alpha} & \underline{4\alpha} & \underline{6\alpha} & 0 & 0 & 0 & 0 \\ 1 & \underline{4\alpha} & \underline{8\alpha} & \underline{12\alpha} & 0 & 0 & 0 & 0 \\ 1 & \underline{6\alpha} & \underline{12\alpha} & \underline{18\alpha} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & \underline{2\alpha} & \underline{4\alpha} & \underline{6\alpha} \\ 0 & 0 & 0 & 0 & 1 & \underline{4\alpha} & \underline{8\alpha} & \underline{12\alpha} \\ 0 & 0 & 0 & 0 & 1 & \underline{6\alpha} & \underline{12\alpha} & \underline{18\alpha} \end{bmatrix} \quad (14)$$

or

$$[W] = [U_I][V_I] \quad (14a)$$

Substituting (14a) into (13) and putting  $[V][A]$  together, we obtain

$$[V_I][A_I] = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \end{bmatrix} \quad (15)$$

Rewrite (13) with the new notations so far developed:

$$\begin{aligned} [S] &= [W][A_I] \\ &= [U_I][V_I][A_I] \end{aligned}$$

$$= \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \end{bmatrix} \quad (16)$$

Equation (16) is the matrix explanation of the signal flow graph shown in Fig. 17.

#### 5. Repeating Use of the Algorithm

One of the advantages of the Fast Fourier Transform is that its basic algorithm can be used again and again in all stages.

Consider

$$\begin{bmatrix} V_I \end{bmatrix} \begin{bmatrix} A_I \end{bmatrix} = \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \\ Q_0 \\ Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} \quad (16a)$$

for this N=8 example, it means:

$$\begin{pmatrix}
 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
 1 & \underline{2\alpha} & \underline{4\alpha} & \underline{6\alpha} & 0 & 0 & 0 & 0 \\
 1 & \underline{4\alpha} & \underline{8\alpha} & \underline{12\alpha} & 0 & 0 & 0 & 0 \\
 1 & \underline{6\alpha} & \underline{12\alpha} & \underline{18\alpha} & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
 0 & 0 & 0 & 0 & 1 & \underline{2\alpha} & \underline{4\alpha} & \underline{6\alpha} \\
 0 & 0 & 0 & 0 & 1 & \underline{4\alpha} & \underline{8\alpha} & \underline{12\alpha} \\
 0 & 0 & 0 & 0 & 1 & \underline{6\alpha} & \underline{12\alpha} & \underline{18\alpha}
 \end{pmatrix}
 \begin{pmatrix}
 A_0 \\
 A_2 \\
 A_4 \\
 A_6 \\
 A_1 \\
 A_3 \\
 A_5 \\
 A_7
 \end{pmatrix}
 =
 \begin{pmatrix}
 P_0 \\
 P_1 \\
 P_2 \\
 P_3 \\
 Q_0 \\
 Q_1 \\
 Q_2 \\
 Q_3
 \end{pmatrix}$$

(17)

Without changing the results, we can write the left hand side into an alternative form.

$$\begin{bmatrix}
 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
 1 & \underline{14\alpha} & \underline{12\alpha} & \underline{6\alpha} & 0 & 0 & 0 & 0 \\
 1 & \underline{8\alpha} & \underline{12\alpha} & \underline{12\alpha} & 0 & 0 & 0 & 0 \\
 1 & \underline{12\alpha} & \underline{6\alpha} & \underline{18\alpha} & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
 0 & 0 & 0 & 0 & 1 & \underline{14\alpha} & \underline{12\alpha} & \underline{6\alpha} \\
 0 & 0 & 0 & 0 & 1 & \underline{8\alpha} & \underline{14\alpha} & \underline{12\alpha} \\
 0 & 0 & 0 & 0 & 1 & \underline{12\alpha} & \underline{6\alpha} & \underline{18\alpha}
 \end{bmatrix}
 \begin{bmatrix}
 A_0 \\
 A_4 \\
 A_2 \\
 A_6 \\
 A_1 \\
 A_5 \\
 A_3 \\
 A_7
 \end{bmatrix}
 =
 \begin{bmatrix}
 P_0 \\
 P_1 \\
 P_2 \\
 P_3 \\
 Q_0 \\
 Q_1 \\
 Q_2 \\
 Q_3
 \end{bmatrix}
 \quad (17a)$$

Again, factoring the square matrix of (17a) into two matrices:

$$\begin{bmatrix}
 1 & 0 & \underline{10\alpha} & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & \underline{12\alpha} & 0 & 0 & 0 & 0 \\
 1 & 0 & -\underline{10\alpha} & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & -\underline{12\alpha} & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 & \underline{10\alpha} & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 & \underline{12\alpha} \\
 0 & 0 & 0 & 0 & 1 & 0 & -\underline{10\alpha} & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 & -\underline{12\alpha}
 \end{bmatrix}
 \begin{bmatrix}
 1 & \underline{10\alpha} & 0 & 0 & 0 & 0 & 0 & 0 \\
 1 & -\underline{10\alpha} & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & \underline{10\alpha} & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & -\underline{10\alpha} & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & \underline{10\alpha} & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & -\underline{10\alpha} & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & \underline{10\alpha} \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & -\underline{10\alpha}
 \end{bmatrix}
 \begin{bmatrix}
 A_0 \\
 A_4 \\
 A_2 \\
 A_6 \\
 A_1 \\
 A_5 \\
 A_3 \\
 A_7
 \end{bmatrix}$$

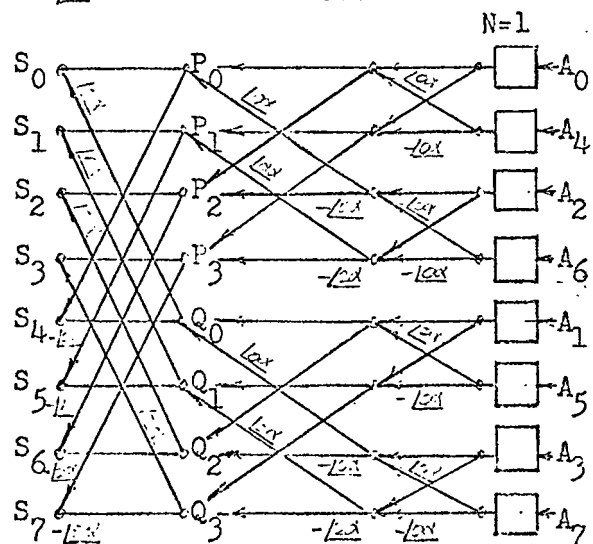
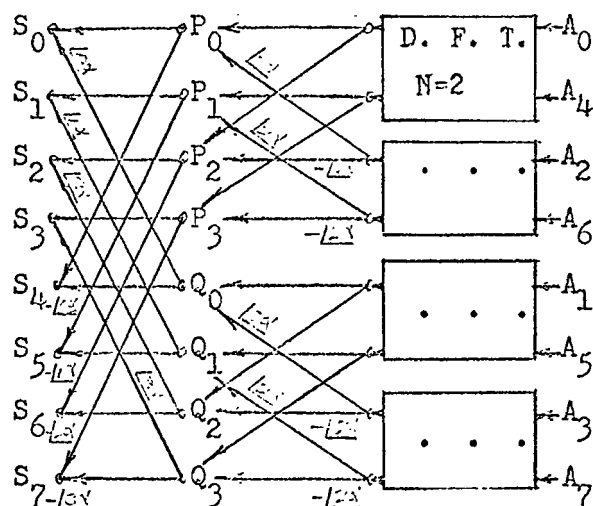
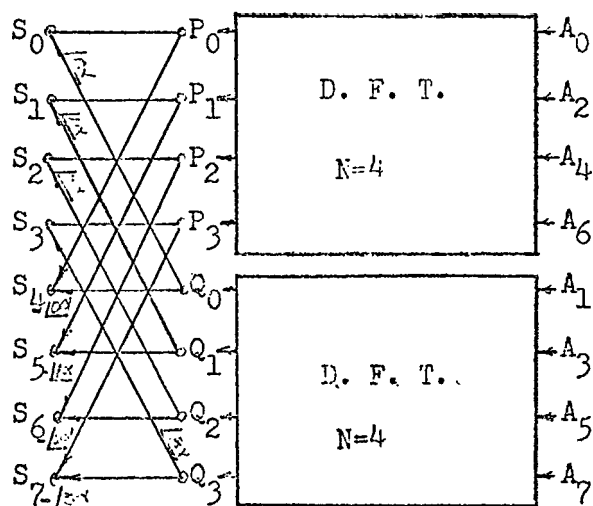
or

$$\begin{bmatrix} U_{II} \end{bmatrix} \begin{bmatrix} U_{III} \end{bmatrix} \begin{bmatrix} A_{II} \end{bmatrix}$$

Therefore, we can write the complete equation as follows:

$$\begin{bmatrix} S \end{bmatrix} = \begin{bmatrix} U_I \end{bmatrix} \begin{bmatrix} U_{II} \end{bmatrix} \begin{bmatrix} U_{III} \end{bmatrix} \begin{bmatrix} A_{II} \end{bmatrix} \quad (19)$$

The corresponding signal flow graphs is shown in Fig. 19



Where

$$[S] = [U_I] [V_I] [A_I]$$

$$[U_I] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \underline{10\alpha} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \underline{1\alpha} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \underline{2\alpha} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \underline{3\alpha} & 0 \\ 1 & 0 & 0 & 0 & -\underline{10\alpha} & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -\underline{1\alpha} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -\underline{2\alpha} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -\underline{3\alpha} & 0 \end{bmatrix}$$

$$[S] = [U_I] [U_{II}] [V_I] [A_{II}]$$

Where

$$[U_{II}] = \begin{bmatrix} 1 & 0 & \underline{10\alpha} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \underline{2\alpha} & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -\underline{10\alpha} & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -\underline{2\alpha} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & \underline{2\alpha} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \underline{2\alpha} & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -\underline{10\alpha} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -\underline{2\alpha} & 0 \end{bmatrix}$$

$$[S] = [U_I] [U_{II}] [U_{III}] [V_{III}] [A_{III}]$$

Where

$$[U_{III}] = \begin{bmatrix} 1 & \underline{10\alpha} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -\underline{10\alpha} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \underline{10\alpha} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -\underline{10\alpha} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \underline{10\alpha} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -\underline{10\alpha} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \underline{10\alpha} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -\underline{10\alpha} & 0 \end{bmatrix}$$

$$[V_{III}] = [I]$$

$$[A_{III}] = [A_{II}]$$

Fig. 19



## 6. Computer Program for Decimation in Time Fast Fourier Transform

Two computer program for the Fast Fourier Transform are written as shown in Program #3 and program #4.

Program #3 is using the analytic time function as an input while program #4 uses numerical information as input.

We still apply the example shown in section II-(3) in testing these two new programs

## 7. Inverse Fast Fourier Transform

If the frequency spectrum is given and the time domain function is required, in the discrete Fourier Transform, we use the formula

$$A_n = \frac{1}{N} \sum_{k=0}^{N-1} S_k e^{j \frac{2\pi nk}{N}} \quad n=0,1,\dots,N-1. \quad (21)$$

as we mentioned before (Section III —1-(2))

Putting N behind the summation sign gives

$$A_n = \sum_{k=0}^{N-1} \frac{S_k}{N} e^{j \frac{2\pi nk}{N}} \quad (22)$$

Expanding and regrouping yield,

$$\begin{aligned} A_n &= \sum_{r=0}^{N/2-1} \frac{S_{2r}}{N} e^{j \frac{2\pi n 2r}{N}} + \sum_{r=0}^{N/2-1} \frac{S_{2r+1}}{N} e^{j \frac{2\pi n (2r+1)}{N}} \\ &= \sum_{r=0}^{N/2-1} \frac{S_{2r}}{N} e^{j \frac{2\pi n r}{N}} + e^{j \frac{2\pi n}{N}} \sum_{r=0}^{N/2-1} \frac{S_{2r+1}}{N} e^{j \frac{2\pi n r}{N}} \end{aligned} \quad (23)$$

Let

$$\begin{aligned} B_n &= \sum_{r=0}^{N/2-1} \frac{S_{2r}}{N} e^{j \frac{2\pi n r}{N}} \quad \text{and} \\ C_n &= \sum_{r=0}^{N/2-1} \frac{S_{2r+1}}{N} e^{j \frac{2\pi n r}{N}} \end{aligned} \quad (24)$$

```

C      PROGRAM #3
C      FAST FOURIER TRANSFORM -DECIMATION IN TIME
C      2**M IS THE TOTAL NUMBER OF SAMPLE POINTS,T IS THE TOTAL TIME
C      FOR INPUT FUNCTION ONLY
C      FUNCTION
      F(X)=10.*(EXP(-X)-EXP(-2.*X))
      COMPLEX A(2500),T1
      READ(5,1) M,T
1     FORMAT(I10,F15.5)
      N=2**M
      DES=1./T
      DET=1/N
      NH=N/2
      NH1=NH+1
C     BIT REVERSAL
      DO 20 I=1,N
      KA=I-1
      NN=0
      DO 30 J=1,M
      UN=N.*2
      NB=NA-NA/2*2
      NA=NA/2
30    NN=NA+NB
20    A(NN+1)=F((I-1)*DET)*DET
C     FAST FOURIER TRANSFORM
      DO 100 I=1,M
      IA=2*(I-1)
      L=2*(I-1)
      DO 200 J=1,IA
      DO 200 K=1,L
      T1=A(2*(J-1)*L+L+K)*CEXP((C./I-1.)*2.*3.1416*(K-1)/(2*I))
      A(2*(J-1)*L+L+K)=A(2*(J-1)*L+K)-T1
200   A(2*(J-1)*L+K)=A(2*(J-1)*L+K)+T1
100   CONTINUE
      WRITE(6,2)
2     FORMAT(15X,'F',15X,'REAL PART',10X,'IMAGINARY PART',10X,'ABSOLUTE
1VALUE')
      DO 300 I=NH1,N
      FN=-(N-I+1)*DES
      AR=REAL(A(I))
      AI=AIMAG(A(I))
      AT=CABS(A(I))
300   WRITE(6,3) FN,AR,AI,AT
3     FORMAT(10X,4F15.5)
      DO 400 I=1,NH
      FN=(I-1)*DES
      AR=REAL(A(I))
      AI=AIMAG(A(I))

      AT=CABS(A(I))
400   WRITE(6,3) FN,AR,AI,AT
      STOP
      END

```

```

C      PROGRAM #4
C      FAST FOURIER TRANSFORM -DECIMATION IN TIME
C      2**M IS THE TOTAL NUMBER OF SAMPLE POINTS,T IS THE TOTAL TIME
C      FOR INPUT DATA ONLY
      COMPLEX A,T1
      DIMENSION A(1100),B(1100)
      READ(5,1) M,T
1     FORMAT(I10,F15.5)
      N=2**M
      DES=1./T
      DET=T/N
      NH=N/2
      NH1=NH+1
      READ(5,5) (B(I),I=1,N)
      DO 10 I=1,N
10    B(I)=B(I)*DET
      5    FORMAT(8F10.5)
C      BIT REVERSAL
      DO 20 I=1,N
      NA=I-1
      NN=0
      DO 30 J=1,M
      NN=NN*2
      NB=NA-NA/2**2
      NA=NA/2
30    NN=NN+NB
      20    A(I)=CMPLX(B(NN+1),0.)
C      FAST FOURIER TRANSFORM
      DO 100 I=1,M
      IA=2***(M-I)
      L=2***(I-1)
      DO 200 J=1,IA
      DO 200 K=1,L
      T1=A(2*(J-1)*L+L+K)*CEXP((0.,-1.)*2.*3.1416*(K-1)/(2**I))
      A(2*(J-1)*L+L+K)=A(2*(J-1)*L+K)-T1
200   A(2*(J-1)*L+K)=A(2*(J-1)*L+K)+T1
100   CONTINUE
      WRITE(6,2)
      2    FORMAT(15X,'F',15X,'REAL PART',10X,'IMAGINARY PART',10X,'ABSOLUTE
1VALUE')
      DO 300 I=NH1,N
      F=-(N-I+1)*DES
      AR=REAL(A(I))
      AI=AIMAG(A(I))
      AT=CABS(A(I))
300   WRITE(6,3) F,AR,AI,AT.
      3    FORMAT(10X,4F15.5)
      DO 400 I=1,NH

```

```

F=(I-1)*DES
AR=REAL(A(I))
AI=AIMAG(A(I))
AT=CABS(A(I))
400 WRITE(6,3) F,AR,AI,AT
STOP
END

```

	REAL PART	IMAGINARY PART		ABSOLUTE VALU
-1.00000	-0.59835	0.00000	0.59835	
-0.75000	-0.65295	0.16064	0.67242	
-0.50000	-0.81425	0.55760	0.98687	
-0.25000	-0.51703	2.01314	2.07848	
0.00000	4.56685	0.00000	4.56685	
0.25000	-0.51705	-2.01314	2.07848	
0.50000	-0.81425	-0.55760	0.98688	
0.75000	-0.65296	-0.16064	0.67243	

Then (23) becomes

$$A_n = B_n + e^{j\frac{2\pi n}{N}} C_n \quad n=0,1,\dots,N-1. \quad (25)$$

Again,  $B_n$  and  $C_n$  have the following periodic nature:

$$\begin{aligned} B_n &= B\left(n + \frac{N}{2}\right) \\ C_n &= C\left(n + \frac{N}{2}\right) \end{aligned} \quad (26)$$

Equation (25) can be written into two equations:

$$A_n = B_n + e^{j\frac{2\pi n}{N}} C_n, \quad \text{when } 0 \leq n < \frac{N}{2}; \quad (25a)$$

$$\text{Let } n = \frac{N}{2} + n', \quad \text{where } \left(\pi + \frac{2\pi n'}{N}\right)$$

$$\begin{aligned} A_{\frac{N}{2}+n'} &= B_{\frac{N}{2}+n'} + e^{j\frac{2\pi(\frac{N}{2}+n')}{N}} C_{\frac{N}{2}+n'} \\ &= B_{n'} - e^{j\frac{2\pi n'}{N}} C_{n'} \end{aligned} \quad (25b)$$

Therefore, we obtain the basic algorithm of the inverse Fast Fourier Transform as follows:

$$\begin{aligned} A_n &= B_n + e^{j\frac{2\pi n}{N}} C_n \\ A_{\frac{N}{2}+n} &= B_n - e^{j\frac{2\pi n}{N}} C_n \quad n=0,1,\dots,\frac{N}{2}-1 \end{aligned} \quad (26)$$

The corresponding signal flow graph of (26) is shown in Fig. 20. ( $n=8$ , for example)

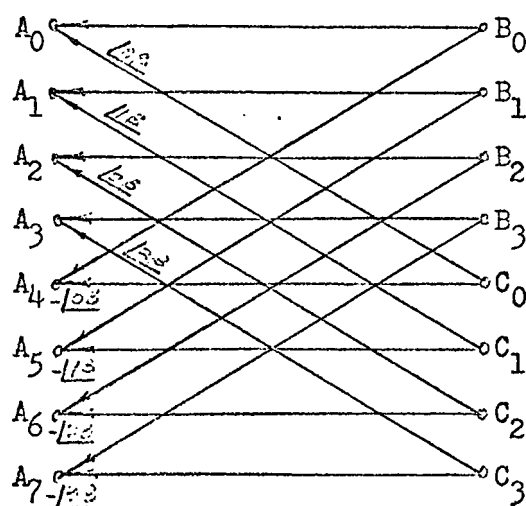


Fig. 20.

where  $\beta = -\frac{\pi}{4}$

It should be noted that while we used  $\alpha = -\frac{\pi}{4}$ , or negative  $\frac{\pi}{4}$ , for the Fast Fourier Transform; we now use  $\beta = \frac{\pi}{4}$ , or positive  $\frac{\pi}{4}$ , for the inverse Fast Fourier Transform. Structurally, Fig. 20 and Fig 16 are the same if we use the following proper notation changes:

S	→	A
P	→	B
Q	→	C
$\alpha$	→	$\beta$

## 8. Computer Programs for Inverse Fast Fourier Transform

```

C      PROGRAM 45
C      INVERSE FFT
C      2**M IS TOTAL SAMPLE POINTS, T IS TOTAL TIME
C      FOR FUNCTION ONLY
      COMPLEX A, T1, DES, CFUN, S
      DIMENSION A(200)
      READ(5,1) M, T
1    FORMAT(11D, F15.5)
      RITL(6,2)
2    FORMAT(20X, 'TIME', 10X, 'X(T)')
      N=2**M
      NH=N/2
      DEL=1/T
      DES=(0., 1.) * 2 * 3.1416/T
      DO 100 I=1, N
        SA=I-1
        AN=0
        DO 200 J=1, M
          CN=J * 2
          SE=SA - SA/2 * 2
          SA=SA/2
200    AN=AN + CB
          IF (I-AN) 250, 250, 250
250    A(CN+1)=CFUN((I-1)*DES)
        DO 300 T=1, 100
300    A(CN+1)=CFUN((I-AN+1)*DES)
100  CONTINUE
      DO 300 I=1, M
        IA=2*(I-1)
        L=2*(I-1)
        DO 400 J=1, IA
          DO 400 K=1, L
            T1=A(2*(J-1)*L+L+K)*DEXP((0., 1.) * 2 * 3.1416*(K-1)/(2**I))
            A(2*(J-1)*L+L+K)=A(2*(J-1)*L+K)-T1
400    A(2*(J-1)*L+K)=A(2*(J-1)*L+K)+T1
300  CONTINUE
      DO 500 I=1, NH
        TN=2*(I-1)*DEL
        AN=REAL(A(2*I-1))/T
500  WRITE(6,501) TN, AN
501  FORMAT(10X, 2F15.5)
      STOP
      END

      COMPLEX FUNCTION CFUN(S)
      COMPLEX S
      CFUN=(10., 0.)/((S+(1., 0.))*(S+(2., 0.)))
      RETURN
      END

```

```

C      PROGRAM #6
C      INVERSE FFT
C      2**M IS TOTAL SAMPLE POINTS, T IS TOTAL TIME
C      FOR INPUT DATA ONLY.
      COMPLEX A,T1,SI
      DIMENSION A(8200),SI(4)
      READ(5,1) M,T
1  FORMAT(I10,F15.5)
      N=2**M
      NH=N/2
      DET=T/N
      N8=N/8
      DO 10 I=1,N8
      READ(5,3) (SI(IA),IA=1,4)
3  FORMAT(8F10.5)
      DO 50 IB=1,4
      NA=NH+4*(I-1)+IB-1
      NN=0
      DO 90 J=1,M
      NN=NN*2
      NB=NA-NA/2*2
      NA=NA/2
90  NN=NN+NB
50  A(NN+1)=SI(IB)
10  CONTINUE
      DO 100 I=1,N8
      READ(5,3) (SI(IA),IA=1,4)
      DO 150 IB=1,4
      NA=4*(I-1)+IB-1
      NN=0
      DO 200 J=1,M
      NN=NN*2
      NB=NA-NA/2*2
      NA=NA/2
200 NN=NN+NB
150 A(NN+1)=SI(IB)
100 CONTINUE
      WRITE(6,2)
2  FORMAT(20X,'TIME',10X,'X(T)')
      DO 300 I=1,M
      IA=2***(M-I)
      L=2***(I-1)
      DO 400 J=1,IA
      DO 400 K=1,L
      T1=A(2*(J-1)*L+L+K)*CEXP((0., 1.)*2.*3.1416*(K-1)/(2**I))
      A(2*(J-1)*L+L+K)=A(2*(J-1)*L+K)-T1
400 A(2*(J-1)*L+K)=A(2*(J-1)*L+K)+T1
300 CONTINUE

```



```

DO 500 I=1,N
  T=(I-1)*DFT
  X=REAL(A(I))/T
500 WRITE(6,201) T,X,AN
201 FORMAT(1X,2F16.6)
STOP
END

```

TIME	X(T)
0.000000	0.000020
0.500000	1.850723
1.000000	1.901199
1.500000	1.754743
2.000000	1.169979
2.500000	0.732273
3.000000	0.835800
3.500000	0.828251

## V. Fast Fourier Transform--Decimation in Frequency

### 1. Basic Algorithm

Again, we starting with the formula of the discrete Fourier Transform (Section II-14 (a))

$$S_k = \sum_{n=0}^{N-1} A_n e^{-j \frac{2\pi nk}{N}}, \quad k=0,1,2,\dots,N-1. \quad (1)$$

We regroup it in the following new way:

$$S_k = \sum_{n=0}^{\frac{N}{2}-1} A_n e^{-j \frac{2\pi nk}{N}} + \sum_{n=\frac{N}{2}}^{N-1} A_n e^{-j \frac{2\pi nk}{N}} \quad (2)$$

The second term of the right handside starts with  $n=\frac{N}{2}$  which is not convenient for later formulation. We can write an equivalent form, consequently, we have

$$\begin{aligned} S_k &= \sum_{n=0}^{\frac{N}{2}-1} A_n e^{-j \frac{2\pi nk}{N}} + \sum_{n=0}^{\frac{N}{2}-1} A_{\frac{N}{2}+n} e^{-j \frac{2\pi (\frac{N}{2}+n)k}{N}} \\ &= \sum_{n=0}^{\frac{N}{2}-1} A_n e^{-j \frac{2\pi nk}{N}} + \sum_{n=0}^{\frac{N}{2}-1} A_{\frac{N}{2}+n} e^{-jk\pi} \cdot e^{-j \frac{2\pi nk}{N}} \\ &= \sum_{n=0}^{\frac{N}{2}-1} (A_n + A_{\frac{N}{2}+n} e^{-jk\pi}) e^{-j \frac{2\pi nk}{N}} \\ &\quad k=0,1,\dots,N-1. \end{aligned} \quad (3)$$

When  $k$  is even, letting  $k=2r$ ,

$$\begin{aligned} S_{2r} &= \sum_{n=0}^{\frac{N}{2}-1} (A_n + A_{\frac{N}{2}+n}) e^{-j \frac{2\pi n 2r}{N}} \\ &= \sum_{n=0}^{\frac{N}{2}-1} (A_n + A_{\frac{N}{2}+n}) e^{-j \frac{2\pi nr}{\frac{N}{2}}} \\ &\quad r=0,1,\dots,\frac{N}{2}-1. \end{aligned} \quad (4)$$

when  $k$  is odd, letting  $k=2r+1$

$$\begin{aligned}
 S_{2r+1} &= \sum_{n=0}^{N/2-1} (A_n - A_{\frac{N}{2}+n}) e^{-j \frac{2\pi n(2r+1)}{N}} \\
 &= \sum_{n=0}^{N/2-1} (A_n - A_{\frac{N}{2}+n}) e^{-j \frac{2\pi n}{N}} e^{-j \frac{2\pi nr}{N/2}} \\
 &\quad r=0,1,\dots,\frac{N}{2}-1. \quad (5)
 \end{aligned}$$

Equations (4) and (5) are the basic algorithm of Fast Fourier Transform — Decimation in Frequency.

## 2. Vector Diagram Interpretation

Decimation in frequency, for our  $n=8$  example, means that we find  $S_0, S_2, S_4$ , and  $S_6$  by using (4); and calculate  $S_1, S_3, S_5$ , and  $S_7$  by using (5), keeping  $A_0, A_1, \dots, A_7$  in the natural order.

The given data are shown in the table below

$A_0$	0
$A_1$	1.199
$A_2$	1.663
$A_3$	0.867
$A_4$	0.585
$A_5$	0.377
$A_6$	0.286
$A_7$	0.146

(6)

Find the corresponding frequency spectrum by the method of decimation in frequency.

When we substitute the data (6) into (4) and (5) respectively,

$$S_{2r} = \sum_{n=0}^3 (A_n + A_{4+n}) e^{-j\frac{\pi n 2r}{4}} \quad r=0,1,2,3 \dots \quad (7)$$

and

$$S_{2r+1} = \sum_{n=0}^3 (A_n - A_{4+n}) e^{-j\frac{\pi}{4} n} e^{-j\frac{\pi}{4} n \cdot 2r} \quad r=0,1,2,3 \quad (8)$$

Their expanded forms are

$$\begin{aligned} S_0 &= (A_0 + A_4) \underline{10\alpha} + (A_1 + A_5) \underline{10\alpha} + (A_2 + A_6) \underline{10\alpha} + (A_3 + A_7) \underline{10\alpha} \\ S_2 &= (A_0 + A_4) \underline{10\alpha} + (A_1 + A_5) \underline{2\alpha} + (A_2 + A_6) \underline{14\alpha} + (A_3 + A_7) \underline{16\alpha} \\ S_4 &= (A_0 + A_4) \underline{10\alpha} + (A_1 + A_5) \underline{14\alpha} + (A_2 + A_6) \underline{8\alpha} + (A_3 + A_7) \underline{12\alpha} \\ S_6 &= (A_0 + A_4) \underline{10\alpha} + (A_1 + A_5) \underline{6\alpha} + (A_2 + A_6) \underline{12\alpha} + (A_3 + A_7) \underline{18\alpha} \end{aligned} \quad (7a)$$

and

$$\begin{aligned} S_1 &= [(A_0 - A_4) \underline{10\alpha}] \underline{10\alpha} + [(A_1 - A_5) \underline{1\alpha}] \underline{10\alpha} + [(A_2 - A_6) \underline{2\alpha}] \underline{10\alpha} + [(A_3 - A_7) \underline{3\alpha}] \underline{10\alpha} \\ S_3 &= [(A_0 - A_4) \underline{10\alpha}] \underline{10\alpha} + [(A_1 - A_5) \underline{1\alpha}] \underline{2\alpha} + [(A_2 - A_6) \underline{2\alpha}] \underline{14\alpha} + [(A_3 - A_7) \underline{3\alpha}] \underline{16\alpha} \\ S_5 &= [(A_0 - A_4) \underline{10\alpha}] \underline{10\alpha} + [(A_1 - A_5) \underline{1\alpha}] \underline{14\alpha} + [(A_2 - A_6) \underline{2\alpha}] \underline{8\alpha} + [(A_3 - A_7) \underline{3\alpha}] \underline{12\alpha} \\ S_7 &= [(A_0 - A_4) \underline{10\alpha}] \underline{10\alpha} + [(A_1 - A_5) \underline{1\alpha}] \underline{6\alpha} + [(A_2 - A_6) \underline{2\alpha}] \underline{12\alpha} + [(A_3 - A_7) \underline{3\alpha}] \underline{18\alpha} \end{aligned} \quad (8a)$$

Each equation can be interpreted as summation of a set of vectors. However, the additional or subtractional orders are different from the orders we used before.

For finding  $S_1$ , we show two diagrams: (a), is for discrete Fourier Transform and (b) is illustrating how to find  $S_1$ , through the method of decimation in frequency. Of course, the two answers should be the same.

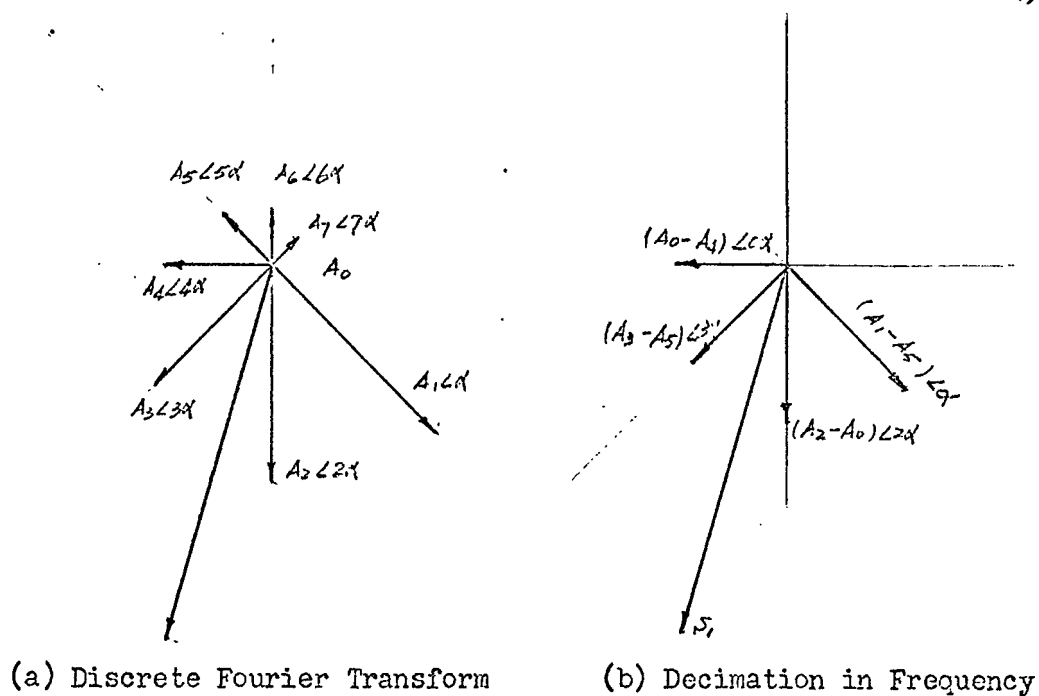
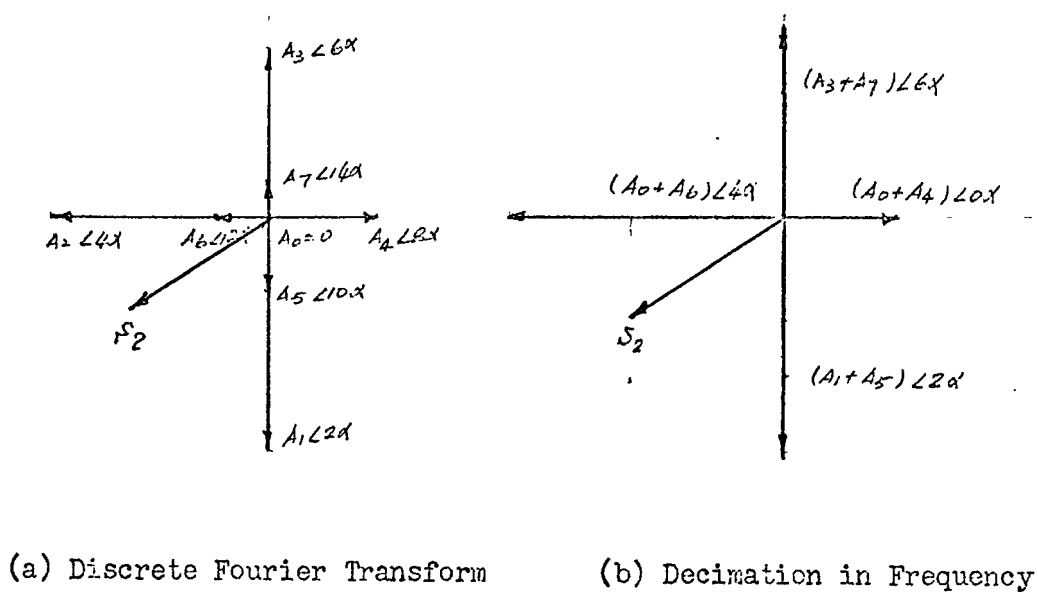
Fig. 22 shows a graphical interpretation for finding  $S_2$ .

### 3. Matrix Explanation

The matrix expression of the algorithm has been mentioned in II—(14), we rewrite here:

$$\begin{pmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \underline{1\alpha} & \underline{2\alpha} & \underline{3\alpha} & \underline{4\alpha} & \underline{5\alpha} & \underline{6\alpha} & \underline{7\alpha} \\ 1 & \underline{2\alpha} & \underline{4\alpha} & \underline{6\alpha} & \underline{8\alpha} & \underline{10\alpha} & \underline{12\alpha} & \underline{14\alpha} \\ 1 & \underline{3\alpha} & \underline{6\alpha} & \underline{9\alpha} & \underline{12\alpha} & \underline{15\alpha} & \underline{18\alpha} & \underline{21\alpha} \\ 1 & \underline{4\alpha} & \underline{8\alpha} & \underline{12\alpha} & \underline{16\alpha} & \underline{20\alpha} & \underline{24\alpha} & \underline{28\alpha} \\ 1 & \underline{5\alpha} & \underline{10\alpha} & \underline{15\alpha} & \underline{20\alpha} & \underline{25\alpha} & \underline{30\alpha} & \underline{35\alpha} \\ 1 & \underline{6\alpha} & \underline{12\alpha} & \underline{18\alpha} & \underline{24\alpha} & \underline{30\alpha} & \underline{36\alpha} & \underline{42\alpha} \\ 1 & \underline{7\alpha} & \underline{14\alpha} & \underline{21\alpha} & \underline{28\alpha} & \underline{35\alpha} & \underline{42\alpha} & \underline{49\alpha} \end{pmatrix} \begin{pmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \\ A_7 \end{pmatrix} \quad (9)$$

Expanding the above matrix equation into eight simultaneous equations and then put the second, fourth, sixth and eighth equations as the first group, and put the first, third, fifth and seventh as the second group. Then rewrite the adjusted equations into a matrix equation again.

Fig. 21. Evaluation  $S_1$ Fig. 22. Evaluating  $S_2$

$$\begin{bmatrix} S_0 \\ S_2 \\ S_4 \\ S_6 \\ S_1 \\ S_3 \\ S_5 \\ S_7 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \underline{12\alpha} & \underline{14\alpha} & \underline{16\alpha} & \underline{18\alpha} & \underline{10\alpha} & \underline{12\alpha} & \underline{14\alpha} \\ 1 & \underline{14\alpha} & \underline{18\alpha} & \underline{12\alpha} & \underline{16\alpha} & \underline{20\alpha} & \underline{24\alpha} & \underline{28\alpha} \\ 1 & \underline{16\alpha} & \underline{12\alpha} & \underline{18\alpha} & \underline{24\alpha} & \underline{30\alpha} & \underline{36\alpha} & \underline{42\alpha} \\ 1 & \underline{18\alpha} & \underline{16\alpha} & \underline{19\alpha} & \underline{12\alpha} & \underline{15\alpha} & \underline{16\alpha} & \underline{17\alpha} \\ 1 & \underline{10\alpha} & \underline{12\alpha} & \underline{19\alpha} & \underline{12\alpha} & \underline{15\alpha} & \underline{18\alpha} & \underline{21\alpha} \\ 1 & \underline{12\alpha} & \underline{20\alpha} & \underline{15\alpha} & \underline{20\alpha} & \underline{25\alpha} & \underline{30\alpha} & \underline{35\alpha} \\ 1 & \underline{14\alpha} & \underline{24\alpha} & \underline{21\alpha} & \underline{28\alpha} & \underline{35\alpha} & \underline{42\alpha} & \underline{49\alpha} \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \\ A_7 \end{bmatrix} \quad (10)$$

or

$$[S_I] = [\Omega] [A] \quad (10a)$$

Factoring the square matrix:

$$\begin{bmatrix} S_0 \\ S_2 \\ S_4 \\ S_6 \\ S_1 \\ S_3 \\ S_5 \\ S_7 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & \underline{12\alpha} & \underline{14\alpha} & \underline{16\alpha} & 0 & 0 & 0 & 0 \\ 1 & \underline{14\alpha} & \underline{18\alpha} & \underline{12\alpha} & 0 & 0 & 0 & 0 \\ 1 & \underline{16\alpha} & \underline{12\alpha} & \underline{18\alpha} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & \underline{12\alpha} & \underline{14\alpha} & \underline{16\alpha} \\ 0 & 0 & 0 & 0 & 1 & \underline{14\alpha} & \underline{18\alpha} & \underline{12\alpha} \\ 0 & 0 & 0 & 0 & 1 & \underline{16\alpha} & \underline{12\alpha} & \underline{18\alpha} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ \underline{10\alpha} & 0 & 0 & 0 & -\underline{10\alpha} & 0 & 0 & 0 \\ 0 & \underline{1\alpha} & 0 & 0 & 0 & -\underline{1\alpha} & 0 & 0 \\ 0 & 0 & \underline{2\alpha} & 0 & 0 & 0 & -\underline{2\alpha} & 0 \\ 0 & 0 & 0 & \underline{3\alpha} & 0 & 0 & 0 & -\underline{3\alpha} \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \\ A_7 \end{bmatrix} \quad (11)$$

or simply write

$$[S_I] = [\Phi_I] [P_I] [A] \quad (11a)$$

Combining the last square matrix and the last vector, we have

$$\begin{pmatrix} S_0 \\ S_2 \\ S_4 \\ S_6 \\ S_1 \\ S_3 \\ S_5 \\ S_7 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & \underline{2\alpha} & \underline{4\alpha} & \underline{6\alpha} & 0 & 0 & 0 & 0 \\ 1 & \underline{4\alpha} & \underline{8\alpha} & \underline{12\alpha} & 0 & 0 & 0 & 0 \\ 1 & \underline{8\alpha} & \underline{16\alpha} & \underline{24\alpha} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & \underline{2\alpha} & \underline{4\alpha} & \underline{6\alpha} \\ 0 & 0 & 0 & 0 & 1 & \underline{4\alpha} & \underline{8\alpha} & \underline{12\alpha} \\ 0 & 0 & 0 & 0 & 1 & \underline{8\alpha} & \underline{16\alpha} & \underline{24\alpha} \end{pmatrix} \begin{pmatrix} A_0 + A_4 \\ A_1 + A_5 \\ A_2 + A_6 \\ A_3 + A_7 \\ (A_0 - A_4) \underline{10\alpha} \\ (A_1 - A_5) \underline{1\alpha} \\ (A_2 - A_6) \underline{2\alpha} \\ (A_3 - A_7) \underline{3\alpha} \end{pmatrix} \quad (12)$$

Partitioning (12) and expanding yield

$$\begin{pmatrix} S_0 \\ S_2 \\ S_4 \\ S_6 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \underline{2\alpha} & \underline{4\alpha} & \underline{6\alpha} \\ 1 & \underline{4\alpha} & \underline{8\alpha} & \underline{12\alpha} \\ 1 & \underline{8\alpha} & \underline{16\alpha} & \underline{24\alpha} \end{pmatrix} \begin{pmatrix} A_0 + A_4 \\ A_1 + A_5 \\ A_2 + A_6 \\ A_3 + A_7 \end{pmatrix} \quad (13)$$

and

$$\begin{pmatrix} S_1 \\ S_3 \\ S_5 \\ S_7 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \underline{2\alpha} & \underline{4\alpha} & \underline{6\alpha} \\ 1 & \underline{4\alpha} & \underline{8\alpha} & \underline{12\alpha} \\ 1 & \underline{8\alpha} & \underline{16\alpha} & \underline{24\alpha} \end{pmatrix} \begin{pmatrix} A_0 - A_4 \underline{10\alpha} \\ A_1 - A_5 \underline{1\alpha} \\ A_2 - A_6 \underline{2\alpha} \\ A_3 - A_7 \underline{3\alpha} \end{pmatrix} \quad (14)$$

Equations (13) and (14) are (7a) and (8a) respectively.



#### 4. Signal Flow Graph Representation

A corresponding signal flow graph is easily drawn as shown in the Fig. 23b

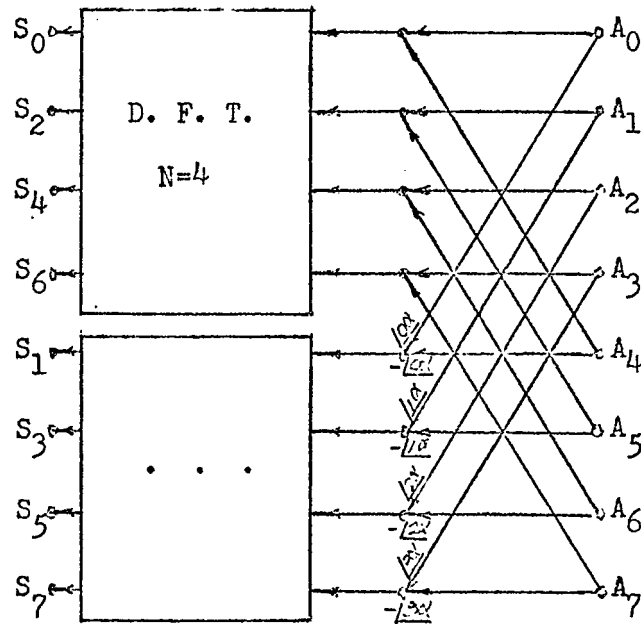


Fig. 23b.

The natural order of  $[A]$  and the disturbed order of  $[S]$  are clearly indicated in Fig. 23.

#### 5. Repeating Use of the Algorithm

Rewrite (11a)

$$[S_1] = [\Phi_1][\Psi_1][A] \quad (11a)$$

$$\begin{pmatrix} s_0 \\ s_2 \\ s_4 \\ s_6 \\ s_1 \\ s_3 \\ s_5 \\ s_7 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & \underline{2\alpha} & \underline{4\alpha} & \underline{6\alpha} & 0 & 0 & 0 & 0 \\ 1 & \underline{4\alpha} & \underline{8\alpha} & \underline{12\alpha} & 0 & 0 & 0 & 0 \\ 1 & \underline{8\alpha} & \underline{16\alpha} & \underline{24\alpha} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & \underline{2\alpha} & \underline{4\alpha} & \underline{6\alpha} \\ 0 & 0 & 0 & 0 & 1 & \underline{4\alpha} & \underline{8\alpha} & \underline{12\alpha} \\ 0 & 0 & 0 & 0 & 1 & \underline{8\alpha} & \underline{16\alpha} & \underline{24\alpha} \end{pmatrix} [\Psi_1] [A] \quad (15)$$

Without changing the result, we rearrange the rows:

$$\begin{pmatrix} s_0 \\ s_4 \\ s_2 \\ s_6 \\ s_1 \\ s_5 \\ s_3 \\ s_7 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & \underline{4\alpha} & \underline{8\alpha} & \underline{12\alpha} & 0 & 0 & 0 & 0 \\ 1 & \underline{2\alpha} & \underline{4\alpha} & \underline{6\alpha} & 0 & 0 & 0 & 0 \\ 1 & \underline{8\alpha} & \underline{16\alpha} & \underline{24\alpha} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & \underline{4\alpha} & \underline{8\alpha} & \underline{12\alpha} \\ 0 & 0 & 0 & 0 & 1 & \underline{2\alpha} & \underline{4\alpha} & \underline{6\alpha} \\ 0 & 0 & 0 & 0 & 1 & \underline{8\alpha} & \underline{16\alpha} & \underline{24\alpha} \end{pmatrix} [\Psi_1] [A] \quad (16)$$

Then factoring the main square matrix of (16):

$$\begin{pmatrix} s_0 \\ s_4 \\ s_2 \\ s_6 \\ s_1 \\ s_5 \\ s_3 \\ s_7 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & \underline{4\alpha} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \underline{4\alpha} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \underline{4\alpha} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \underline{4\alpha} \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ \underline{10\alpha} & 0 & -\underline{10\alpha} & 0 & 0 & 0 & 0 & 0 \\ 0 & \underline{2\alpha} & 0 & -\underline{2\alpha} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & \underline{10\alpha} & 0 & -\underline{10\alpha} & 0 \\ 0 & 0 & 0 & 0 & 0 & \underline{2\alpha} & 0 & -\underline{2\alpha} \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad (17)$$

or

$$[S_{II}] = [\Phi_{II}] [\Psi_{II}] [\Psi_{II}] [A] \quad (17a)$$

A summary of the decomposition processes and the signal flow graphs are shown in Fig. 23 which is self explanatory.

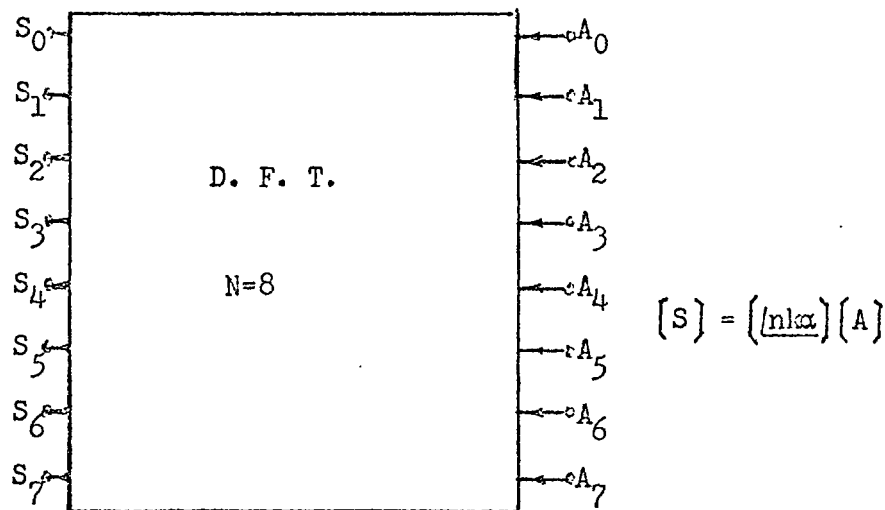


Fig. 23(a)

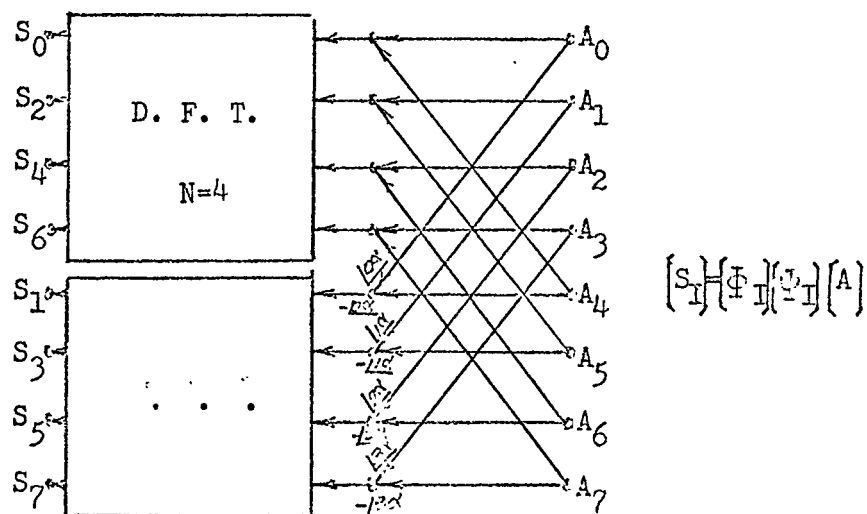


Fig. 23 (b)

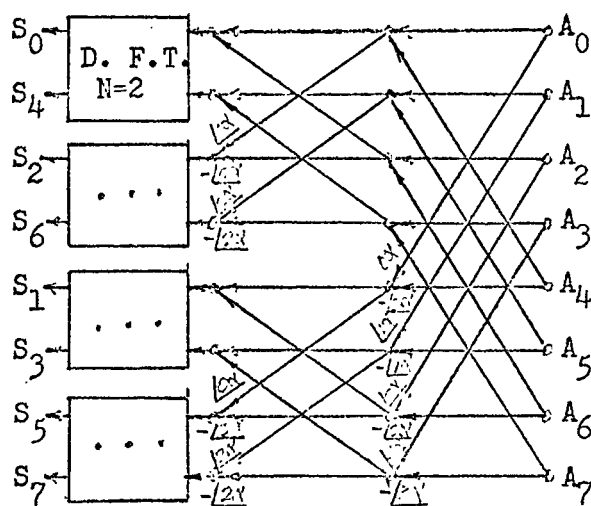


Fig. 23(c)

$$[S_{II}] = [T_{II}] [U_{II}] [V_I] [A]$$

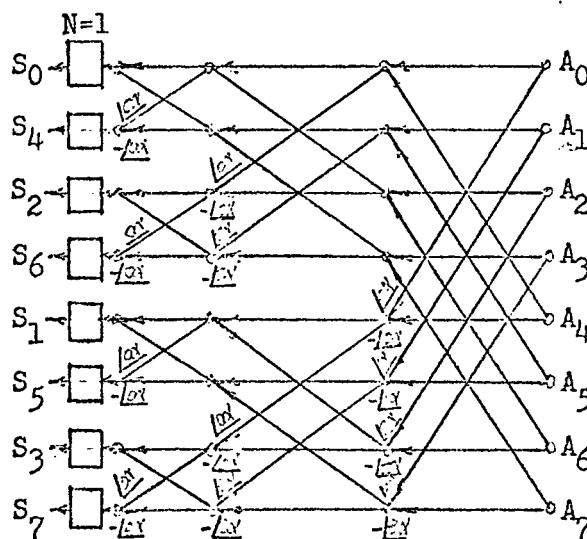


Fig. 23(d)

$$[S_{III}] = [T_{III}] [U_{III}] [V_I] [A]$$

$$[T_{III}] = [I]$$

$$[U_{III}] = [U_{II}]$$

$$[S_{III}] = [S_{II}]$$

## 6. Computer Program for the Fast Fourier Transform—Decimation in Frequency

Computer Program #8 and #7 are for the Fast Fourier Transform--decimation in frequency. The former is using an analytic function as input while the input of the latter is numerical.

```

C      PROGRAM #7
C      FFT DECIMATION IN FREQUENCY
C      T IS THE TOTAL TIME, 2**M IS THE TOTAL SAMPLE POINTS
C      FOR INPUT DATA ONLY
      COMPLEX A,T1
      DIMENSION A(1100),B(1100)
      READ(5,1) M,T
1     FORMAT(110,F15.5)
      N=2**M
      NH=N/2
      DET=1/N
      NH1=NH+1
      READ(5,7) (B(I),I=1,N)
7     FORMAT(8F10.5)
      DO 10 I=1,N
10    A(I)=CMPLX(B(I),0.)*DET
C      FFT
      DO 100 I=1,M
      IA=2** (I-1)
      L=2** (M-I)
      DO 200 J=1,IA
      DO 200 K=1,L
      T1=A(2*(J-1)*L+K)
      A(2*(J-1)*L+K)=T1+A(2*(J-1)*L+L+K)
200  A(2*(J-1)*L+L+K)=(T1-A(2*(J-1)*L+L+K))*CEXP((0.,-1.)*2*3.1416*(K-1)
      1)/(2*L))
100  CONTINUE
      WRITE(6,3)
3     FORMAT(20X,'FREQ.',10X,'REAL PART',10X,'IMAGINARY PART')
      DO 300 I=NH1,N
      NA=I-1
      NN=0
      DO 400 J=1,M
      NN=NN*2
      NB=NA-NA/2*2
      NA=NA/2
400  NN=NN+NB
      FN=-(N-I+1)/T
      AR=REAL(A(NN+1))
      AI=AIMAG(A(NN+1))
      AT=CABS(A(NN+1))
300  WRITE(6,4) FN,AR,AI,AT
4     FORMAT(10X,4F16.6)
      DO 500 I=1,NH
      NA=I-1
      NN=0
      DO 600 J=1,M
      NN=NN*2

```

```

      NB=NA-NA/2*2
      NA=NA/2
600  NN=NV+NB
      FN=(1-1)/T
      AR=REAL(A(NN+1))
      AI=AIMAG(A(NN+1))
      AT=CABS(A(NN+1))
500  WRITE(6,4)FN,AR,AI,AT
      STOP
      END

```

FREQ.	REAL PART	IMAGINARY PART	ABS. VALUE
-1.000000	-0.598351	0.000000	0.598351
-0.750000	-0.652950	0.160641	0.672421
-0.500000	-0.814248	0.557600	0.986872
-0.250000	-0.517035	2.013143	2.078476
0.000000	4.566848	0.000000	4.566848
0.250000	-0.517055	-2.013139	2.078478
0.500000	-0.814251	-0.557600	0.986875
0.750000	-0.652958	-0.160645	0.672429

```

C      PROGRAM #8
C      FFT DECIMATION IN FREQUENCY
C      T IS THE TOTAL TIME, 2**M IS THE TOTAL SAMPLE POINTS
C      FOR FUNCTION ONLY
      FUN(X)=10.*(EXP(-X)-EXP(-2.*X))
      COMPLEX A,T1
      DIMENSION A(1100)
      READ(5,1) M,T
1     FORMAT(110,F15.5)
      N=2**M
      NH=N/2
      DET=T/N
      NH1=NH+1
      DO 10 I=1,N
10    A(I)=(1.,0.)*FUN((I-1)*DET)*DET
C      FFT
      DO 100 I=1,M
      IA=2**I
      L=2**(M-I)
      DO 200 J=1,IA
      DO 200 K=1,L
      T1=A(2*(J-1)*L+K)
      A(2*(J-1)*L+K)=T1+A(2*(J-1)*L+L+K)
200  A(2*(J-1)*L+L+K)=(T1-A(2*(J-1)*L+L+K))*CEXP((0.-1.)*2*3.1416*(K-
1)/L)
100  CONTINUE
      WRITE(6,3)
3     FORMAT(20X,'FREQ.',10X,'REAL PART',10X,'IMAGINARY PART')
      DO 300 I=NH1,N
      NA=I-1
      NN=0
      DO 400 J=1,M
      NN=NN*2
      NB=NA-NA/2*2
      NA=NA/2
400  NN=NN+NB
      FN=-(N-I+1)/T
      AR=REAL(A(NN+1))
      AI=AIMAG(A(NN+1))
      AT=CABS(A(NN+1))
300  WRITE(6,4)FN,AR,AI,AT
4     FORMAT(10X,4F16.6)
      DO 500 I=1,NH
      NA=I-1
      NN=0
      DO 600 J=1,M
      NN=NN*2
      NB=NA-NA/2*2

```

```
      NA=NA/2
600  NN=NN+NB
      FN=(I-1)/T
      AR=REAL(A(NN+1))
      AI=AIMAG(A(NN+1))
      AT=CABS(A(NN+1))
500  WRITE(6,4)FN,AR,AI,AT
      STOP
      END
```



## 7. Inverse Fast Fourier Transform

The basic discrete Fourier Transform:

$$A_n = \frac{1}{N} \sum_{k=0}^{N-1} S_k e^{-j2\pi kn/N} \quad n=0,1,\dots,N-1. \quad (18)$$

is divided into two groups as follows:

$$\begin{aligned} A_n &= \sum_{k=0}^{\frac{N}{2}-1} \left( \frac{S_k}{N} \right) e^{j\frac{2\pi kn}{N}} + \sum_{k=\frac{N}{2}}^{N-1} \left( \frac{S_k}{N} \right) e^{j\frac{2\pi kn}{N}} \\ &= \sum_{k=0}^{\frac{N}{2}-1} \left( \frac{S_k}{N} \right) e^{j\frac{2\pi kn}{N}} + \sum_{k=0}^{\frac{N}{2}-1} \left( \frac{S_{\frac{N}{2}+k}}{N} \right) e^{j\frac{2\pi (\frac{N}{2}+k)n}{N}} \\ &= \sum_{k=0}^{\frac{N}{2}-1} \left( \frac{S_k}{N} + \frac{S_{\frac{N}{2}+k}}{N} e^{jn\pi} \right) e^{j\frac{2\pi kn}{N}} \end{aligned} \quad (19)$$

$n=0,1,\dots,N-1.$

when  $n$  is even, letting  $n=2r$ , we have

$$A_{2r} = \sum_{k=0}^{\frac{N}{2}-1} \left( \frac{S_k}{N} + \frac{S_{\frac{N}{2}+k}}{N} \right) e^{j\frac{2\pi kr}{N/2}} \quad r=0,1,\dots,\frac{N}{2}-1. \quad (20)$$

when  $n$  is odd, letting  $n=2r+1$ , we have

$$A_{2r+1} = \sum_{k=0}^{\frac{N}{2}-1} \left( \frac{S_k}{N} - \frac{S_{\frac{N}{2}+k}}{N} \right) e^{j\frac{2\pi kr}{N/2}} e^{j\frac{2\pi kr}{N}} \quad r=0,1,\dots,\frac{N}{2}-1. \quad (21)$$

Equations (20) and (21) are the another pair of the basic algorithm for the inverse Fast Fourier Transform. For  $n=8$ , the complete signal flow graph is shown in Fig. 24.

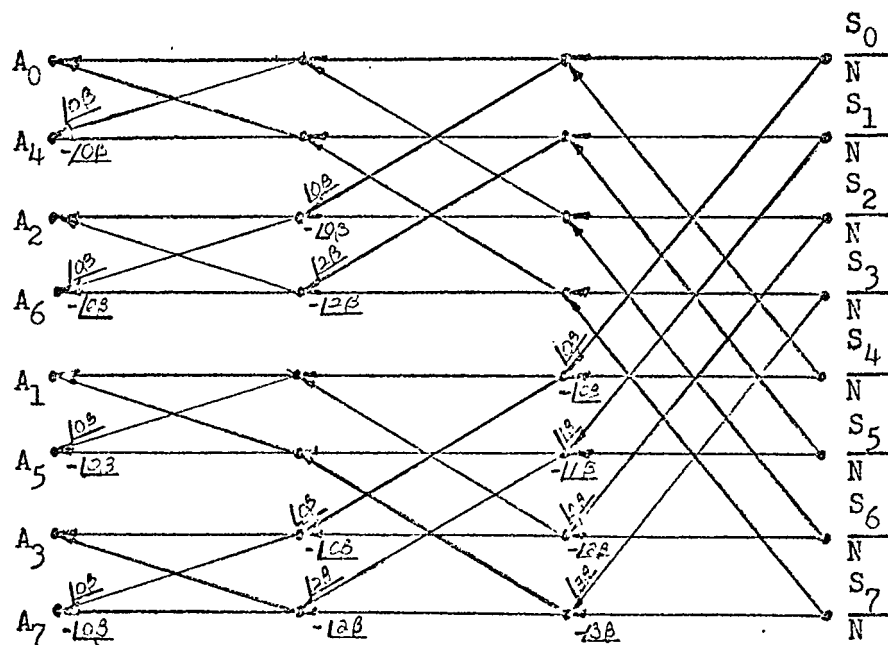


Fig. 24

### 8. Computer Program for Inverse Fast Fourier Transform

```

C      PROGRAM# 9
C      IFFT
C      T IS THE TOTAL TIME,2**M IS THE TOTAL SAMPLE POINTS
C      FOR FUNCTION ONLY
C      COMPLEX A ,T1,DES,S,CFUN
      DIMENSION A(1500)
      READ(5,1) M,T
1    1 FORMAT(110,F15.5)
      N=2**M
      NH=N/2
      DET=T/N
      DES=(0.,1.)*2.*3.1416/T
      DO 10 I=1,NH
        A(I)=CFUN((I-1)*DES)
10   A(NH+I)=CFUN(-(NH-I+1)*DES)
C      INVERSE FAST FOURIER TRANSFORM
      DO 100 I=1,M
        IA=2** (I-1)
        I=2** (M-I)
        DO 200 J=1,IA
          DO 200 K=1,L
            T1=A(2*(J-1)*L+K)
            A(2*(J-1)*L+K)=T1+A(2*(J-1)*L+L+K)
200   A(2*(J-1)*L+L+K)=(T1-A(2*(J-1)*L+L+K))*CEXP((C.,1.)*2.*3.1416*(K-
1) / (2*L))
100  CONTINUE
      WRITE(6,3)
3    FORMAT(20X,'TIME',10X,'X(T)')
      DO 300 I=1,N
        NA=I-1
        NN=0
        DO 400 J=1,M
          NN=NN*2
          NB=NA-NA/2*2
          NA=NA/2
400   NN=NN+NB
          AR=REAL(A(NN+1))/T
          TN=(I-1)*DET
300   WRITE(6,4) TN,AR
4    FORMAT(10X,2F16.6)
      STOP
      END

```

```

      COMPLEX FUNCTION CFUN(S)
      COMPLEX S
      CFUN=(1C.,0.)/((S+(1.,0.))*(S+(2.,0.)))
      RETURN
      END

```

```

C      PROGRAM#10
C      IFFT
C      T IS THE TOTAL TIME,2**M IS THE TOTAL SAMPLE POINTS
C      FOR INPUT DATA ONLY
C      COMPLEX A,T1,DES
C      DIMENSION A(1500)
C      READ(5,1) M,T
1     FORMAT(110,F15.5)
      N=2**M
      NH=N/2
      DET=T/N
      NH1=NH+1
      DES=(0.,1.)*2.*3.1416/T
      READ(5,2) (A(I),I=NH1,N)
      READ(5,2) (A(I),I=1,NH)
2     FORMAT(8F10.5)
C      INVERSE FAST FOURIER TRANSFORM
      DO 100 I=1,M
      IA=2** (I-1)
      L=2** (M-I)
      DO 200 J=1,IA
      DO 200 K=1,L
      T1=A(2*(J-1)*L+K)
      A(2*(J-1)*L+K)=T1+A(2*(J-1)*L+L+K)
200   A(2*(J-1)*L+L+K)=(T1-A(2*(J-1)*L+L+K))*CEXP((0.,1.)*2.*3.1416*(K-
1)/(2*L))
100   CONTINUE
      WRITE(6,3)
3     FORMAT(20X,'TIME',10X,'X(T)')
      DO 300 I=1,N
      NA=I-1
      NN=0
      DO 400 J=1,M
      NN=NN*2
      NB=NA-NA/2*2
      NA=NA/2
400   NN=NN+NB
      AR=REAL(A(NN+1))/T
      TN=(I-1)*DET
300   WRITE(6,4) TN,AR
      4     FORMAT(10X,2F16.6)
      STOP
      END

```

TIME	X(T)
0.000000	0.000020
0.500000	1.850727
1.000000	1.961199
1.500000	1.754748
2.000000	1.169979
2.500000	0.732273
3.000000	0.835800
3.500000	0.828251

## VI. Fourier Series Evaluation

### 1. Coefficients of the Exponential Form

Originally the algorithm was developed by Tukey and Cooley<sup>4</sup> for Fourier series Coefficients evaluation. Later, their work has been expanded into Fast Fourier Transform. We rather based on the FFT algorithm to derive a method for Fourier series Coefficients evaluation.

Consider that there is a periodic function  $f(t)$  with period  $T$  and its Fourier Coefficients are required.

We write, by definition, the expansion from

$$f(t) = \sum_{n=-\infty}^{n=\infty} C_n e^{j \frac{2n\pi t}{T}} \quad (1)$$

where

$$C_n = \frac{1}{T} \int_0^T f(t) e^{-j \frac{2n\pi t}{T}} dt \quad (2)$$

Because  $T$  is a constant, we can write

$$C_n = \int_0^T \frac{1}{T} f(t) e^{-j \frac{2n\pi t}{T}} dt \quad (3)$$

letting

$$\frac{f(t)}{T} = g(t),$$

we have

$$C_n = \int_0^T g(t) e^{-j \frac{2n\pi t}{T}} dt \quad (4)$$

Now assume that there is a new function  $h(t)$  such that

$$h(t) \begin{cases} = g(t), & 0 \leq t < T \\ = 0 & \text{otherwise} \end{cases} \quad (5)$$

The corresponding Fourier Transform  $H(f)$  is found as

$$H(f) = \int_{-\infty}^{\infty} h(t) e^{-j2\pi ft} dt = \int_0^{\tau} g(t) e^{-j2\pi ft} dt \quad (6)$$

Therefore

$$H\left(\frac{n}{T}\right) = \int_0^{\tau} g(t) e^{-j\frac{2\pi nt}{T}} dt \quad (7)$$

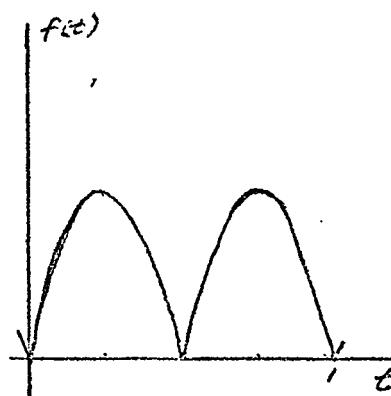
Comparing (7) and (4), we can easily see that  $C_n = H\left(\frac{n}{T}\right)$ . Of course, based on the previous sections, we can evaluate  $H\left(\frac{n}{T}\right)$  immediately by using a method of FFT.

## 2. Computer Program for Finding Coefficients $C_n$

Essentially, this program is similar to program #3; a slight modification is added.

## 3. Fourier Coefficients of Common Wave Forms

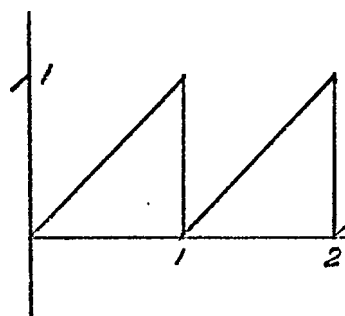
By using the computer program, two examples are evaluated. The summary is shown in the following table:



$$f(t) = |\sin(2\pi t)|$$

N	REAL PART	IMAGINERY PART
-10	-0.001597	0.000000
-9	-0.001972	0.000000
-8	-0.002499	0.000000
-7	-0.003267	0.000000
-6	-0.004454	0.000000
-5	-0.006432	0.000000
-4	-0.010107	0.000000
-3	-0.018191	0.000001
-2	-0.042443	0.000002
-1	-0.212209	0.000007
0	0.636616	0.000000
1	-0.212210	-0.000001
2	-0.042443	-0.000000
3	-0.018192	-0.000000
4	-0.010107	-0.000000
5	-0.006433	-0.000000
6	-0.004454	-0.000000
7	-0.003267	-0.000000
8	-0.002499	-0.000000

9	-0.001973	-0.000000
10	-0.001598	-0.000000



-10	-0.000977	-0.015895
-9	-0.000977	-0.017666
-8	-0.000977	-0.019878
-7	-0.000977	-0.022722
-6	-0.000977	-0.026514
-5	-0.000977	-0.031821
-4	-0.000977	-0.039781
-3	-0.000978	-0.050046
-2	-0.000978	-0.079574
-1	-0.000981	-0.159153
0	0.499923	0.000000
1	-0.000976	0.159152
2	-0.000976	0.079573
3	-0.000976	0.050045
4	-0.000976	0.039781
5	-0.000976	0.031821
6	-0.000976	0.026514
7	-0.000976	0.022722
8	-0.000976	0.019878

9	-0.000976	0.017666
10	-0.000976	0.015895

```

C      PROGRAM #11
C      FIND THE FOURIER COEFFICIENTS BY FFT METHOD
C      IF ND EQUAL 0 FOR INPUT FUNCTION, OTHERWISE FOR DATA
C      2**M TOTAL SAMPLE POINTS, T IS THE PERIOD
C      FUNCTION
C      FUN(X)=X
C      MAIN PROGRAM
C      COMPLEX A,T1
C      DIMENSION A(1100),B(1100)
C      READ(5,101) M,T,ND
C      N=2**M
C      DET=T/N
C      NH=N/2
C      NH1=NH+1
101  FORMAT(I10,F15.5,I5)
C      IF(ND) 5,9,5
C      5 READ(5,102) (B(I),I=1,N)
102  FORMAT(3F10.5)
C      DO 6 I=1,N
C      6 B(I)=B(I)/N
C      GO TO 19
C      9 CONTINUE
C      DO 10 I=1,N
C      10 B(I)=FUN((I-1)*DET)/N
C      19 CONTINUE
C      BIT REVERSAL
C      DO 20 I=1,N
C      NA=I-1
C      NN=0
C      DO 30 J=1,M
C      NN=NN*2
C      NB=NA-NA/2*2
C      NA=NA/2
C      30 NN=NN+NB
C      20 A(I)=CMPLX(B(NN+1),0.)
C      DO 100 I=1,M
C      IA=2**(M-I)
C      L=2**(I-1)
C      DO 200 J=1,IA
C      DO 200 K=1,L
C      T1=A(2*(J-1)*L+L+K)*CEXP((0.-1.)*2.*3.1416*(K-1)/(2**I))
C      A(2*(J-1)*L+L+K)=A(2*(J-1)*L+K)-T1
C      200 A(2*(J-1)*L+K)=A(2*(J-1)*L+K)+T1
C      100 CONTINUE
C      WRITE(6,103)
103  FORMAT(15X,'N',10X,'REAL PART',8X,'IMAGINARY PART')
C      DO 300 I=NH1,N
C      NCC=-(N-I+1)

```



```
CR=REAL(A(I))  
CI=AIMAG(A(I))  
300 WRITE(6,104) NCC,CR,CI  
104 FORMAT(10X,I10,2F16.6)  
DO 400 I=1,NH  
NCC=NCC+1  
CR=REAL(A(I))  
CI=AIMAG(A(I))  
400 WRITE(6,104) NCC,CR,CI  
STOP  
END
```

#### 4. Fourier Coefficients of the Trigonometrical Form

Sometimes the Trigonometrical form is more useful than the exponential form. We usually write the form as follows.

$$f(t) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} a_n \cos \frac{2n\pi t}{T} + \sum_{n=1}^{\infty} b_n \sin \frac{2n\pi t}{T} \quad (8)$$

where

$$a_0 = \frac{2}{T} \int_0^T f(t) dt \quad (9)$$

$$a_n = \frac{2}{T} \int_0^T f(t) \cos \frac{2n\pi t}{T} dt \quad (10)$$

$$b_n = \frac{2}{T} \int_0^T f(t) \sin \frac{2n\pi t}{T} dt \quad (11)$$

Comparing with the exponential form we see that

$$\begin{aligned} a_n &= \frac{2}{T} \int_0^T f(t) \cos \frac{2n\pi t}{T} dt \\ &= \frac{2}{T} \int_0^T f(t) \frac{e^{\frac{j2n\pi t}{T}} + e^{\frac{-j2n\pi t}{T}}}{2} dt \\ &= \frac{1}{T} \int_0^T f(t) e^{\frac{j2n\pi t}{T}} dt + \frac{1}{T} \int_0^T f(t) e^{\frac{-j2n\pi t}{T}} dt \\ &= C_{-n} + C_n \end{aligned} \quad (12)$$

$$\begin{aligned} b_n &= \frac{2}{T} \int_0^T f(t) \sin \frac{2n\pi t}{T} dt \\ &= \frac{2}{T} \int_0^T f(t) \frac{e^{\frac{j2n\pi t}{T}} - e^{\frac{-j2n\pi t}{T}}}{2j} dt \\ &= -j \left[ \frac{1}{T} \int_0^T f(t) e^{\frac{j2n\pi t}{T}} dt - \frac{1}{T} \int_0^T f(t) e^{\frac{-j2n\pi t}{T}} dt \right] \\ &= -j (C_{-n} - C_n) \\ &= j (C_n - C_{-n}) \end{aligned} \quad (13)$$

When  $f(t)$  is a real function, we have

$$\begin{aligned} C_n^* &= \frac{1}{T} \int_0^T f(t) e^{\frac{j2n\pi t}{T}} dt \\ &= C_{-n}, \quad \text{that is} \end{aligned}$$

when  $C_n = x + jy$  (where  $x, y$  are real numbers)

then  $C_{-n} = x - jy$

therefore

$$a_n = C_n + C_{-n} = x + jy + x - jy = 2x = 2 \text{ real } (C_n) \quad (14)$$

and

$$b_n = j(C_n - C_{-n}) = j(x + jy - x + jy) = -2y = -2 \text{ imaginary } (C_n) \quad (15)$$

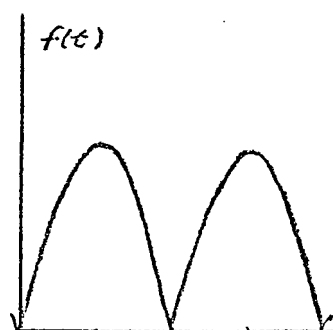
Based on (14) and (15), we write the following computer program for evaluating the coefficients of  $a_n$  and  $b_n$

#### 5. Computers Program to Evaluate Fourier Coefficients of Trigonometrical Form

#### 6. Fourier Coefficients of Common Wave Forms

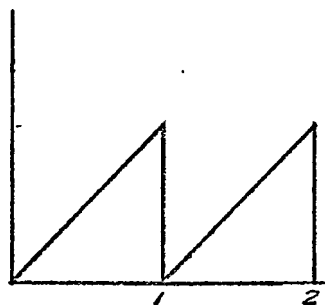
By using the computer program, two examples are evaluated.

The summary is shown in the following table:



$$f(t) = |\sin(2\pi t)|$$

N	COS	SIN
0	1.273232	0.000000
1	-0.424419	0.000001
2	-0.084887	0.000001
3	-0.036384	0.000000
4	-0.020214	0.000000
5	-0.012866	0.000000
6	-0.008908	0.000000
7	-0.006535	0.000000
8	-0.004997	0.000000
9	-0.003946	0.000000
10	-0.003195	0.000000
11	-0.002641	0.000000
12	-0.002218	0.000000
13	-0.001891	0.000000
14	-0.001630	0.000000
15	-0.001421	0.000000
16	-0.001249	0.000000
17	-0.001107	0.000000
18	-0.000987	0.000000
19	-0.000887	0.000000
20	-0.000800	0.000000
21	-0.000726	0.000000



N	COS	SIN
0	0.998347	0.000000
1	-0.001952	-0.318305
2	-0.001953	-0.159146
3	-0.001952	-0.106090
4	-0.001953	-0.079561
5	-0.001953	-0.063642
6	-0.001953	-0.053027
7	-0.001953	-0.045444
8	-0.001953	-0.039757
9	-0.001953	-0.035332
10	-0.001953	-0.031791
11	-0.001953	-0.028893
12	-0.001953	-0.026478
13	-0.001953	-0.024433
14	-0.001953	-0.022650
15	-0.001953	-0.021150
16	-0.001953	-0.019830
17	-0.001953	-0.018656
18	-0.001953	-0.017612
19	-0.001953	-0.016677
20	-0.001953	-0.015835
21	-0.001953	-0.015074

```

C      PROGRAM#12
C      FIND THE FOURIER COEFFICIENTS BY FFT METHOD
C      IF ND EQUAL 0 FOR INPUT FUNCTION, OTHERWISE FOR DATA
C      2**M TOTAL SAMPLE POINTS, T IS THE PERIOD
C      FUNCTION
C      FUN(X)=X
C      MAIN PROGRAM
C      COMPLEX A, T1
C      DIMENSION A(1100), B(1100)
C      READ(5, 101) M, T, ND
C      N=2**M
C      DET=1/N
C      NH=N/2
101  FORMAT(110, F15.5, 15)
C      IF(ND) 5, 9, 5
C      5 READ(5, 102) (B(I), I=1, N)
102  FORMAT(8F10.5)
C      DO 6 I=1, N
C      6 B(I)=B(I)/N
C      GO TO 19
C      9 CONTINUE
C      DO 10 I=1, N
C      10 B(I)=FUN((I-1)*DET)/N
C      19 CONTINUE
C      BIT REVERSAL
C      DO 20 I=1, N
C      NA=I-1
C      NN=0
C      DO 30 J=1, N
C      NN=N * 2
C      NB=NA-NA/2*2
C      NA=NA/2
C      30 NN=NN+NB
C      20 A(I)=CMPLX(B(NN+1), 0.)
C      FAST FOURIER TRANSFORM
C      DO 100 I=1, M
C      IA=2***(M-I)
C      L=2***(I-1)
C      DO 200 J=1, IA
C      DO 200 K=1, L
C      T1=A(2*(J-1)*L+L+K)*CEXP((C., -1.)*2.*3.1416*(K-1)/(2**I))
C      A(2*(J-1)*L+L+K)=A(2*(J-1)*L+K)-T1
C      200 A(2*(J-1)*L+K)=A(2*(J-1)*L+K)+T1
C      100 CONTINUE
C      WRITE(5, 103)
103  FORMAT(15X, 'N', 10X, 'CBS', 10X, 'SIN')
C      DO 300 I=1, NH
C      NA3=I-1

```

```
AN=2.*REAL(A(1))  
BN=-2.*AIMAG(A(1))  
300 WRITE(6,104) NAB,AN,BN  
104 FORMAT(10X,I10,2F16.5)  
STOP  
END
```

## VII. Convolution, Cross-correlation and Auto-correlation

### 1. Convolution Integral

An important application of the Fast Fourier Transform is to evaluate the convolution integral.<sup>9,12</sup>

$$f(t) = \int_{-\infty}^{\infty} f_1(\tau) f_2(t-\tau) d\tau = \int_{-\infty}^{\infty} f_2'(\tau) f_1(t-\tau) d\tau \quad (1)$$

where  $f_1(t)$  and  $f_2(t)$  are two time functions.

A natural approach to this problem is working in the frequency domain, because

$$F(f) = F_1(f) F_2(f) \quad (2)$$

where  $F(f)$ ,  $F_1(f)$  and  $F_2(f)$  are the Fourier Transforms of  $f(t)$ ,  $f_1(t)$  and  $f_2(t)$  respectively. It is seen that both the Fourier Transform and the inverse Fourier Transform are needed.

A simple derivation of (2) is given as follows:

Fourier transforming  $f(t)$  by definition yields

$$F(f) = \int_{-\infty}^{\infty} f(t) e^{-j2\pi ft} dt \quad (3)$$

Substituting (1) into (3)

$$\begin{aligned} F(f) &= \int_{-\infty}^{\infty} \left[ \int_{-\infty}^{\infty} f_1(\tau) f_2(t-\tau) d\tau \right] e^{-j2\pi ft} dt \\ &= \int_{-\infty}^{\infty} f_1(\tau) \left[ \int_{-\infty}^{\infty} f_2(t-\tau) e^{-j2\pi ft} dt \right] d\tau \end{aligned} \quad (4)$$

Letting  $t-\tau=t'$ , then  $t=t'+\tau$  and  $dt=dt'$

Consider the integral

$$\begin{aligned} &\int_{-\infty}^{\infty} f_2(t-\tau) e^{-j2\pi ft} dt \\ &= \int_{-\infty}^{\infty} f_2(t') e^{-j2\pi f(t'+\tau)} dt' \end{aligned}$$

$$\begin{aligned}
 &= e^{-j2\pi f\tau} \int_{-\infty}^{\infty} f_2(t') e^{-j2\pi f t'} dt' \\
 &= e^{-j2\pi f\tau} F_2(f)
 \end{aligned} \tag{5}$$

Substituting (5) into (4), we obtain

$$\begin{aligned}
 F(f) &= \int_{-\infty}^{\infty} f_1(\tau) e^{-j2\pi f\tau} F_2(f) d\tau \\
 &= F_1(f) F_2(f)
 \end{aligned}$$

## 2. Finding the Response Function

If an input time function  $f_1(t)$  is excited to a system which has a weighting function  $f_2(t)$  and the response function is required, we use the Fast Fourier Transform to obtain  $F_1(f)$  and  $F_2(f)$  respectively; Multiply them in the frequency domain to get  $F(f)$ , the inverse Fast Fourier Transform is then applied. Finally the answer,  $f(t)$  is obtained.

The process is illustrated as follows

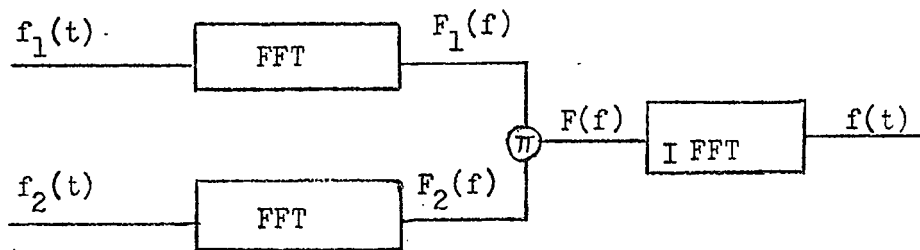


Fig. 25(a)

## 3. Computer Program for Convolution Integral Evaluation

A computer program which consists the Fast Fourier Transform and the Inverse Fast Fourier Transform is written. We use  $f_1(t)=e^{-t}$  and  $f_2(t)=e^{-2t}$ . as a test example; the result is quite satisfactory.



```

C      PROGRAM #13
C      EVALUATION OF CONVOLUTION BY FFT METHOD
C      2**M IS THE TOTAL NUMBER OF SAMPLE POINTS, T IS THE TOTAL TIME
C      INPUT DATA FROM F1 TO F2
C      IF ND IS EQUAL TO 0 FOR INPUT FUNCTION, OTHERWISE FOR DATA
C      FUNCTION
C
C      F1(X)=10.*EXP(-X)
C      F2(X)=10.*EXP(-2.*X)
C
C      COMPLEX A,B,T1
C      DIMENSION A(1500),B(1500),AN(10),BN(10)
C      READ(5,1) M,T,ND
1  FORMAT(I10,F15.5,I5)
C      N=2**M
C      DET=T/N
C      NH=N/2
C      IF( ND .NE. 0) GO TO 19
C      DO 10 I=1,N
C      A(I)=(1.,0.)*F1((I-1)*DET)*DET
10  B(I)=(1.,0.)*F2((I-1)*DET)*DET
C      GO TO 99
19  N8=N/8
C      DO 20 I=1,N8
C      READ(5,3) (AN(J),J=1,8)
C      DO 25 IJ=1,8
25  A(8*(I-1)+IJ)=(1.,0.)*AN(IJ)
20  CONTINUE
C      DO 30 I=1,N8
C      READ(5,3) (BN(J),J=1,8)
3  FORMAT(8F10.5)
C      DO 35 IJ=1,8
35  B(8*(I-1)+IJ)=(1.,0.)*BN(IJ)
30  CONTINUE
99  CONTINUE
C      FAST FOURIER TRANSFORM
C      DO 100 I=1,M
C      IA=2** (I-1)
C      L=2** (M-I)
C      DO 200 J=1,IA
C      DO 200 K=1,L
C      T1=A(2*(J-1)*L+K)
C      A(2*(J-1)*L+K)=T1+A(2*(J-1)*L+L+K)
C      A(2*(J-1)*L+L+K)=(T1-A(2*(J-1)*L+L+K))*CEXP((0.,-1.)*2.*3.1416*(K-
11)/(2*L))
C      T1=B(2*(J-1)*L+K)
C      B(2*(J-1)*L+K)=T1+B(2*(J-1)*L+L+K)
200 B(2*(J-1)*L+L+K)=(T1-B(2*(J-1)*L+L+K))*CEXP((0.,-1.)*2.*3.1416*(K-

```

```

11)/(2*L))
100 CONTINUE
DO 300 I=1,N
300 A(I)=A(I)*B(I)
C INVERSE FAST FOURIER TRANSFORM
DO 400 I=1,M
IA=2** (M-I)
L=2** (I-1)
DO 500 J=1,IA
DO 500 K=1,L
T1=A(2*(J-1)*L+L+K)*CEXP((0.,1.)*2.*3.1416*(K-1)/(2**I))
A(2*(J-1)*L+L+K)=A(2*(J-1)*L+K)-T1
500 A(2*(J-1)*L+K)=A(2*(J-1)*L+K)+T1
400 CONTINUE
WRITE(6,2)
2 FORMAT(20X,'TIME',15X,'F(T)')
DO 600 I=1,N
TA=(I-1)*DET
AR=REAL(A(I))/T
600 WRITE(6,4)TA,AR
4 FORMAT(10X,2F16,6)
STOP
END

```

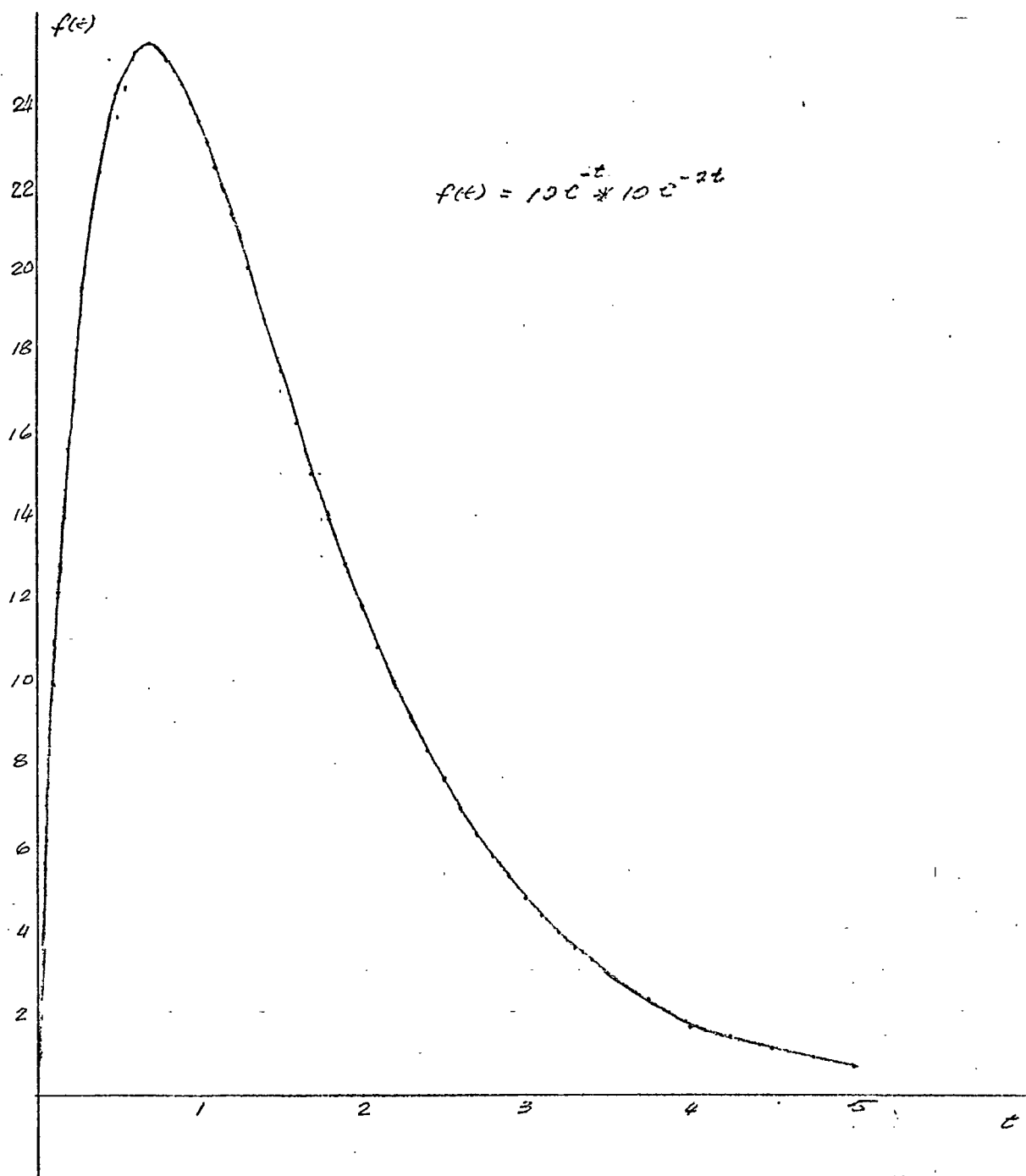


Fig. 25(b)

## 4. Cross-correlation

The definition of Cross-correlation<sup>13,14</sup> is

$$f(t) = \int_{-\infty}^{\infty} f_2(\tau) f_1(t+\tau) d\tau \quad (6)$$

The difference between (6) and (1) is only a sign, although their physical interpretations are quite different. Cross-correlation plays an increasingly important role in communication and statistical control analysis.

Mathematically, however, the evaluation of (6) is not more complicated than that of (1).

For evaluating (6)

$$F(f) = \int_{-\infty}^{\infty} f(t) e^{-j2\pi f t} dt \quad (7)$$

$$\begin{aligned} &= \int_{-\infty}^{\infty} \left[ \int_{-\infty}^{\infty} f_2(\tau) f_1(t+\tau) d\tau \right] e^{-j2\pi f t} dt \\ &= \int_{-\infty}^{\infty} f_2(\tau) \left[ \int_{-\infty}^{\infty} f_1(t+\tau) e^{-j2\pi f t} dt \right] d\tau \end{aligned} \quad (8)$$

We change the independent variable,

$$t + \tau = t'$$

and have

$$\begin{aligned} &\int_{-\infty}^{\infty} f_1(t+\tau) e^{-j2\pi f t} dt \\ &= \int_{-\infty}^{\infty} f_1(t') e^{-j2\pi f (t' - \tau)} dt' \\ &= e^{j2\pi f \tau} F_1(f) \end{aligned} \quad (9)$$

Substituting (9) into (8), we obtain

$$F(f) = \int_{-\infty}^{\infty} f_2(\tau) e^{j2\pi f \tau} F_1(f) d\tau$$

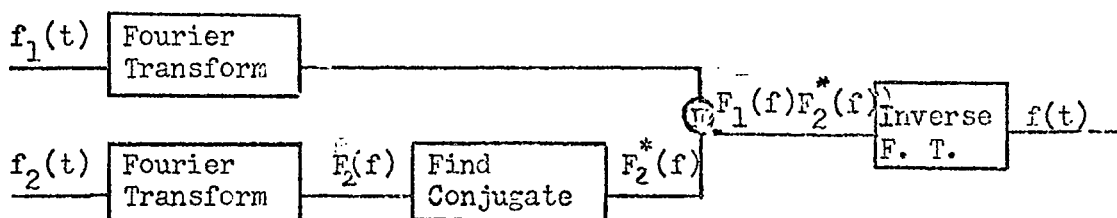
$$\begin{aligned}
 &= F_1(f) \int_{-\infty}^{\infty} f_2(z) e^{j2\pi f z} dz \\
 &= F_1(f) F_2^*(f)
 \end{aligned} \tag{10}$$

Noting that the relation

$$F_2^*(f) = \int_{-\infty}^{\infty} f_2(z) e^{j2\pi f z} dz \tag{11}$$

has been used.

The following diagram illustrates how to use the Fast Fourier Transform to obtain the cross-correlation function.



## 5. Computer Program for Cross-correlation

Program #13 is for finding the cross-correlation function by using FFT.

## 6. Autocorrelation

If  $f_2(t) = f_1(t)$  in equation (6), we have

$$f(t) = \int_{-\infty}^{\infty} f_1(z) f_1(z+t) dz \tag{12}$$

Both cross-correlation and autocorrelation functions are<sup>13</sup> extremely useful in studying statistical theory of communication and control.

## 7. Computer Program and Example

A computer program for the auto-correlation function is shown in program #15.

```

C      PROGRAM #14
C      2**M IS THE TOTAL NUMBER OF SAMPLE POINTS, T IS THE TOTAL TIME
C      INPUT DATA FROM F1 TO F2
C      IF ND IS EQUAL TO 0 FOR INPUT FUNCTION, OTHERWISE FOR DATA
C      FUNCTION
C
C      F1(X)=10.*EXP(-X)
C      F2(X)=10.*EXP(-2.*X)
C
C      COMPLEX A,B,T1
C      DIMENSION A(1500),B(1500),AN(10),BN(10)
C      READ(5,1) M,T,ND
1   FORMAT(I10,F15.5,I5)
C      N=2**M
C      DET=T/N
C      NH=N/2
C      IF(ND.NE.0) GO TO 19
C      DO 10 I=1,N
C      A(I)=(1.,0.)*F1((I-1)*DET)*DET
10  B(I)=(1.,0.)*F2((I-1)*DET)*DET
C      GO TO 99
19  N8=N/8
C      DO 20 I=1,N8
C      READ(5,3) (AN(J),J=1,8)
3   FORMAT(8F10.5)
C      DO 25 IJ=1,8
25  A(8*(I-1)+IJ)=(1.,0.)*AN(IJ)
20  CONTINUE
C      DO 30 I=1,N8
C      READ(5,3) (BN(J),J=1,8)
C      DO 35 IJ=1,8
35  B(8*(I-1)+IJ)=(1.,0.)*BN(IJ)
30  CONTINUE
99  CONTINUE
C
C      FAST FOURIER TRANSFORM
C      DO 100 I=1,M
C      IA=2** (I-1)
C      L=2** (M-I)
C      DO 200 J=1,IA
C      DO 200 K=1,L
C      T1=A(2*(J-1)*L+K)
C      A(2*(J-1)*L+K)=T1+A(2*(J-1)*L+L+K)
C      A(2*(J-1)*L+L+K)=(T1-A(2*(J-1)*L+L+K))*CEXP((0.,-1.)*2.*3.1416*(K-
11)/(2*L))
C      T1=B(2*(J-1)*L+K)
C      B(2*(J-1)*L+K)=T1+B(2*(J-1)*L+L+K)
200 B(2*(J-1)*L+L+K)=(T1-B(2*(J-1)*L+L+K))*CEXP((0.,-1.)*2.*3.1416*(K-
11)/(2*L))

```

```

100 CONTINUE
    DO 300 I=1,N
300  A(I)=A(I)*CONJG(B(I))
C   INVERSE FAST FOURIER TRANSFORM
    DO 400 I=1,M
    IA=2** (M-I)
    L=2** (I-1)
    DO 500 J=1,IA
    DO 500 K=1,L
    T1=A(2*(J-1)*L+L+K)*CEXP((0.,1.)*2.*3.1416*(K-1)/(2**I))
    A(2*(J-1)*L+L+K)=A(2*(J-1)*L+K)+T1
500  A(2*(J-1)*L+K)=A(2*(J-1)*L+K)+T1
400  CONTINUE
    WRITE(6,2)
    2  FORMAT(20X,'TIME',15X,'F(T)')
    DO 600 I=1,N
    TA=(I-1)*DET
    AR=REAL(A(I))/T
600  WRITE(6,4)TA,AR
    4  FORMAT(10X,2F16.6)
    STOP
    END

```

```

C      PROGRAM #15
C      EVALUATION AUTOCORRELATION BY FFT METHOD
C      2**M IS THE TOTAL NUMBER OF SAMPLE POINTS, T IS THE TOTAL TIME
C      IF ND IS EQUAL TO 0 FOR INPUT FUNCTION, OTHERWISE FOR DATA
C      FUNCTION
C
C      F1(X)=10.*EXP(-X)
C
C      COMPLEX A,B,T1
C      DIMENSION A(1500),AN(10)
C      READ(5,1) M,T,ND
1    FORMAT(I10,F15.5,I5)
C      N=2**M
C      DET=T/N
C      NH=N/2
C      IF( ND .NE. 0) GO TO 19
C      DO 10 I=1,N
10   A(I)=(1.,0.)*F1((I-1)*DET)*DET
C      GO TO 99
19   N8=N/8
C      DO 20 I=1,N8
C      READ(5,3) (AN(J),J=1,8)
3    FORMAT(8F10.5)
C      DO 25 IJ=1,8
25   A(8*(I-1)+IJ)=(1.,0.)*AN(IJ)
C      20 CONTINUE
C      99 CONTINUE
C
C      FAST FOURIER TRANSFORM
C      DO 100 I=1,M
C      IA=2** (I-1)
C      L=2** (M-I)
C      DO 200 J=1,IA
C      DO 200 K=1,L
C      T1=A(2*(J-1)*L+K)
C      A(2*(J-1)*L+K)=T1+A(2*(J-1)*L+L+K)
200  A(2*(J-1)*L+L+K)=(T1-A(2*(J-1)*L+L+K))*CEXP((0.,-1.)*2.**3.1416*(K
11)/(2*L))
C      100 CONTINUE
C      DO 300 I=1,N
300  A(I)=A(I)*CONJG(A(I))
C
C      INVERSE FAST FOURIER TRANSFORM
C      DO 400 I=1,M
C      IA=2** (M-I)
C      L=2** (I-1)
C      DO 500 J=1,IA
C      DO 500 K=1,L
C      T1=A(2*(J-1)*L+L+K)*CEXP((0.,1.)*2.**3.1416*(K-1)/(2**I))
C      A(2*(J-1)*L+L+K)=A(2*(J-1)*L+K)-T1

```



```
500 A(2*(J-1)*L+K)=A(2*(J-1)*L+K)+T1
400 CONTINUE
    WRITE(6,2)
    2 FORMAT(20X,'TIME',15X,'F(T)')
    DO 600 I=1,N
        TA=(I-1)*DET
        AR=REAL(A(I))/T
600 WRITE(6,4)TA,AR
    4 FORMAT(10X,2F16.6)
    STOP
    END
```

# VIII. Inverse Laplace Transform of Rational Transfer Functions

## 1. Jordans' s Lemma

Can we perform the inverse Laplace transform by using Fourier transform? The answer is affirmative ; It is also a common practice in the engineering field. The justification of doing this is based on the Jordan Lemma.<sup>16</sup>

Suppose  $Q(s)$  is a function of a complex variable  $s$ . It has the following properties:

- (a) In the upper half plane, it has finite number of singularities.
- (b) When  $0 < \angle s < \pi$  and  $|s| \rightarrow \infty$ ,  $Q(s)$  uniformly approaches zero.

and

- (c)  $m$  is positive.

Then

$$\lim_{R \rightarrow \infty} \int_C e^{jms} Q(s) ds = 0 \quad (1)$$

where  $C$  is the semicircle shown in Fig. 30.

Equation (1) is usually referred as the Jordan Lemma.

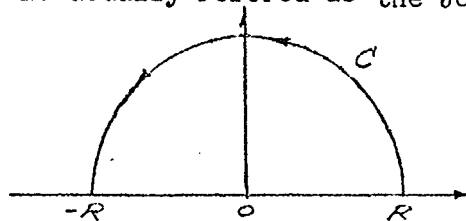


Fig. 30

Laplace transform can be considered as an application of this lemma.

## 2. From Laplace Transform to Fourier Transform

Consider the Laplace transform pair;

$$F(s) = \int_0^{\infty} f(t) e^{-st} dt \quad (2)$$

$$f(t) = \frac{1}{2\pi j} \int_{c-j\infty}^{c+j\infty} F(s) e^{st} ds \quad (3)$$

To perform the inverse Laplace transform by using (3), we usually take the contour integral

$$\begin{aligned} \oint_{\mathcal{T}} F(s) e^{st} ds \\ = \int_{c-j\infty}^{c+j\infty} e^{st} F(s) ds + \int_{\mathcal{T}_1} e^{st} F(s) ds \end{aligned} \quad (4)$$

The constant  $c$  is large enough such that all the singularities are enclosed in the left of the line  $c-j\infty$  to  $c+j\infty$ . This is shown in Fig. 31.

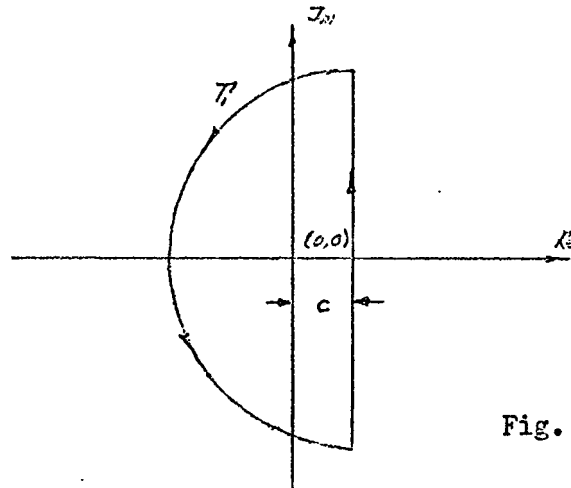


Fig. 31

By using the Jordan Lemma, if the second term of (4) is equal to zero. Then we have

$$2\pi j f(t) = \int_{c-j}^{c+j} e^{st} F(s) ds$$

$$\begin{aligned}
&= \oint F(s) e^{st} ds \\
&= 2\pi j (\sum \text{Res. inside the contour}) \quad (5)
\end{aligned}$$

$$\text{Let } s=c + j\omega, \quad (6)$$

$$\text{then } ds= j d\omega \quad (6a)$$

Substituting (6) into (5), we have

$$\begin{aligned}
f(t) &= \frac{1}{2\pi j} \int_{-\infty}^{\infty} F(c+j\omega) e^{(c+j\omega)t} j d\omega \\
&= \frac{1}{2\pi} \int_{-\infty}^{\infty} F(c+j\omega) e^{(c+j\omega)t} d\omega \\
&= \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{ct} F(c+j\omega) e^{j\omega t} d\omega \quad (7) \\
&= e^{ct} \frac{1}{2\pi} \int_{-\infty}^{\infty} F(c+j\omega) e^{j\omega t} d\omega \\
&= e^{ct} \mathcal{F}^{-1} F(c+j\omega) \quad (8)
\end{aligned}$$

For control engineers, only asymptotically stable systems are interested. In those systems, all singularities are in the left half plane. Therefore, we can choose  $c=0$ . Equation (8) then becomes

$$f(t) = \mathcal{F}^{-1} F(j\omega) \quad (9)$$

Using the Fast Fourier Transform to perform the inverse Laplace transform is thus justified.

### 3. A Special Case

If a transfer function, say,  $M(s) = \frac{G(s)}{s}$ , has a pole in the origin of the  $s$  plane, Fast Fourier Transform can not be directly applied. We can easily overcome this difficulty by removing the pole first, or taking the inverse Fourier transform of  $G(s)$ .

After finding the time function  $g(t)$ , then a numerical integration is performed in order to find  $m(t)$ .

#### 4. Computer Program for Inverse Laplace Transforming — $\frac{G(s)}{s}$

Program #16 is for inverse Laplace transform of the special case shown in the last section.

In the next section, we will use this program many times to inverse transform the irrational transfer function.

#### 5. Illustrative Examples

For the general case, however, we simply change  $s$  to  $j\omega$  and do the Fast Fourier Transform. These examples are tested by using program #5 and #9.

Example 1.

$$L^{-1} \frac{1}{s+1}$$

Example 2.

$$L^{-1} \frac{10}{(s+1)(s+2)}$$

Example 3.

$$L^{-1} \frac{10}{(s+1)^2(s+2)}$$

The results are shown in Fig. 32. It is noted that the repeated root case does not give us particular difficulties of complication.

```

C      PROGRAM #16
C      T IS THE TOTAL SAMPLE TIME, AND 2**N IS TOTAL NUMBER OF SAMPLE POINTS

C      A COMPUTER PROGRAM FOR COMPUTING THE INVERSE LAPLASE TRANSFORM
      COMPLEX A,T1,DES,CFUN,EA,EB,S
      DIMENSION A(200)
      READ(5,1) M,T
      1  FORMAT(I10,F15.5)
      WRITE(6,2)
      2  FORMAT(10X,'TIME',10X,'SQRT(T)',10X,'X(T)')
      N=2**M
      DET=T/4
      DES=(0.,1.)*2.*3.1416/T
      EB=1/2
      IF 100 I=1,N
      LA=I-1
      LN=0
      DO 200 J=1,M
      LB=2**J-1
      LB=LA-LB/2+2
      LA=LB/2
200  LB=LA+LB
      IF(I-1)H 250,250,260
250  A(LB+1)=CFUN((I-1)*DES)
      DO 300 I=1,100
260  A(LB+1)=CFUN((I-1)*DES)
300  CONTINUE
      DO 300 I=1,M
      LA=2***(M-I)
      LB=2***(I-1)
      DO 400 J=1,LA
      IF 400 K=1,LB
      T1=A(2*(J-1)*LB+L+K)*CFXP((0.,1.)*2.*3.1416*(K-1)/(2**I))
      A(2*(J-1)*LB+L+K)=A(2*(J-1)*LB+K)+T1
400  A(2*(J-1)*LB+K)=A(2*(J-1)*LB+K)+T1
500  CONTINUE
      AL=(L-1)/2
      AL=0
      DO 500 I=1,NBH
      T1=2.*I*DET
      TR=SQRT(T1)
      AL=AL+(REAL(A(2*I-1))+4.*REAL(A(2*I))+REAL(A(2*I+1)))/(3.*N)
500  WRITE(6,501) TR,AL
501  FORMAT(10X,3F16.6)
      STOP
      END

```

```

      COMPLEX FUNCTION CFUN(S)
      COMPLEX S,EA,EB
      EA=(0.,1.)*2.*3.1416/360.*70.
      EB=1/2
      CFUN=CSQRT(S)/((CSQRT(S)-CEXP(EA))*(CSQRT(S)-CEXP(EB)))
      RETURN
      END

```

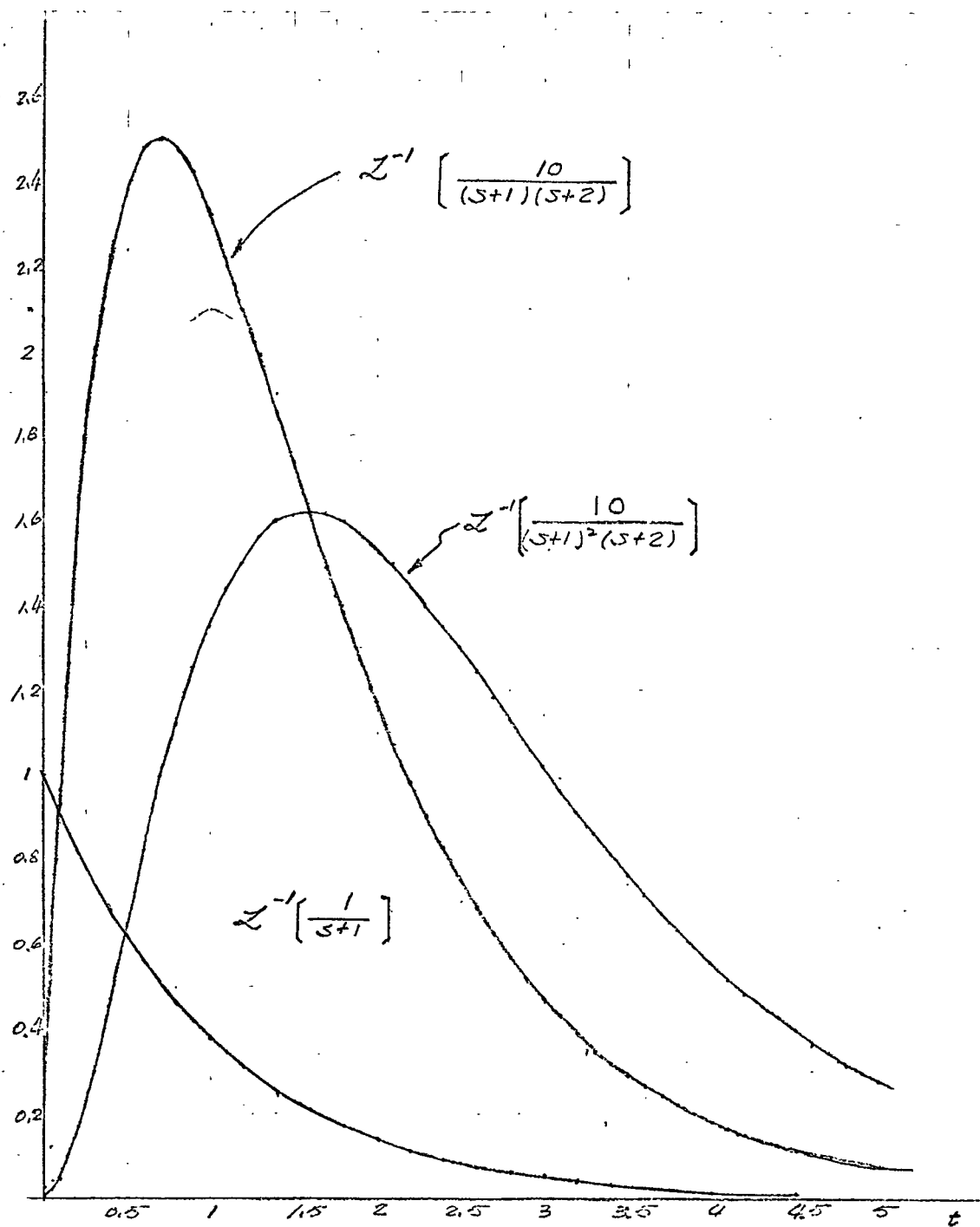


Fig. 32

## IX. Inverse Laplace Transform of Irrational Transfer Functions

### 1. Difficulties in Analytical Approach

A large number of control systems have distributed parameters.<sup>10,11</sup> The mathematical models of those systems contain the operator  $s$  under the radical sign. To find the inverse Laplace transforms of those irrational transfer functions is extremely difficult.

Among the available methods, the most notable one is developed by Kilomeitseva and Netushil. We summarize their theory first and then examine the difficulties involved.

An irrational transfer function is usually given as follows:

$$W(s) = \frac{\alpha_0 s^m + \alpha_1 s^{m-1} + \dots + \alpha_m}{\beta_0 \sqrt{s}^{(2n+1)} + \beta_1 \sqrt{s}^{(2n)} + \beta_2 \sqrt{s}^{(2n-1)} + \dots + \beta_{2n+1}} \quad (1)$$

Let  $\sqrt{s} = z$ ; equation (1) becomes a rational transfer function of  $z$ . or

$$W(z) = \frac{\alpha_0 z^{2m} + \alpha_1 z^{2(m-1)} + \dots + \alpha_m}{\beta_0 z^{2n+1} + \beta_1 z^{2n} + \beta_2 z^{2n-1} + \dots + \beta_{2n+1}} \quad (2)$$

or simply

$$= \frac{\sum_{i=0}^m \alpha_i z^{2(m-i)}}{\sum_{i=0}^{2n+1} \beta_i z^{2n+1-i}} \quad (3)$$

This rational fraction may be factored into a sum of elementary fractions

$$W(z) = \sum_{i=1}^{2n+1} \frac{A_i}{z - z_i} \quad (4)$$

if there is no repeated root in the characteristic equation of (2).



The inverse Laplace transform of the typical term of (4) with a unit step as input can be performed by using the following Laplace transform pair:

$f(t)$	$F(s)$
$e^{a^2 t} \{ b-a \operatorname{erf}(a\sqrt{t}) \} - b e^{b^2 t} \operatorname{erfc}(b\sqrt{t})$	$\frac{b^2 - a^2}{(s - a^2)(b + \sqrt{s})}$

or let

$a=0$ , for special case:

$$b - b e^{b^2 t} \operatorname{erfc}(b\sqrt{t}) \quad \left| \quad \frac{b^2}{s(\sqrt{s}+b)} \right. \quad (5)$$

By using (5), we have the inverse Laplace transform of a typical term of (4), as follows:

$$L^{-1} \frac{A}{s(\sqrt{s}+a)} = \frac{A}{a} \left[ 1 - e^{a^2 t} \operatorname{erfc}(a\sqrt{t}) \right] \quad (6)$$

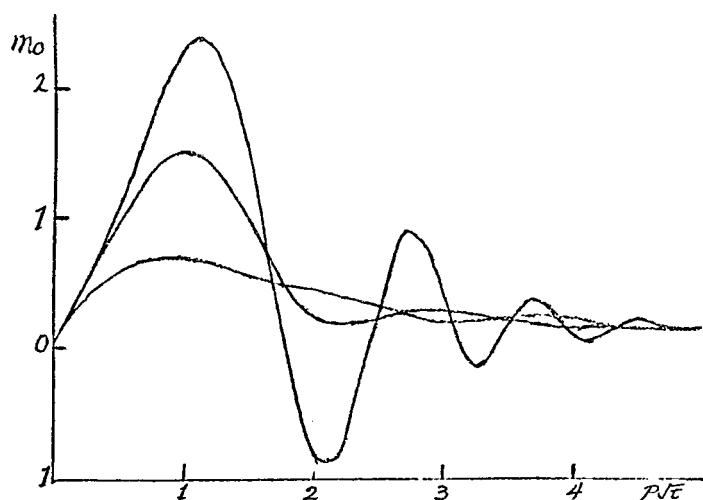
If complex roots are involved, pair (5) can not be directly applied, Kilomeitsiva and Netushil's approach is to reform the pair into a new function. For example, assume the complex conjugate roots are  $z_k$  and  $z_{k+1}$ ,

Consider

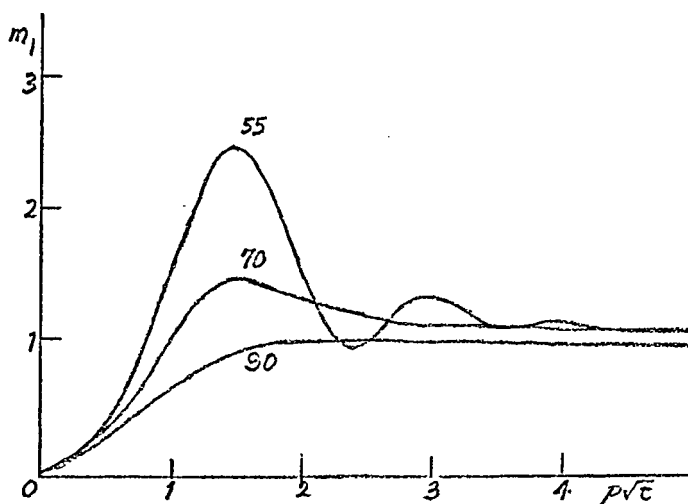
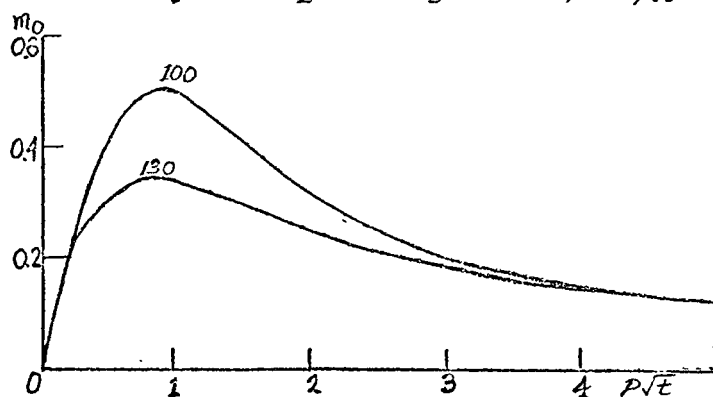
$$W_1(z) = \frac{A_k}{z - z_k} + \frac{A_{k+1}}{z - z_{k+1}} = \frac{E}{(z - z_k)(z - z_{k+1})} + \frac{Dz}{(z - z_k)(z - z_{k+1})} \quad (7)$$

The inverse Laplace transform will be

$$L^{-1} \frac{W_1(z)}{s} = E \left( \frac{1}{\rho^2} m_0(\rho\sqrt{t}, \theta) \right) + D \left( \frac{1}{\rho^2} m_1(\rho\sqrt{t}, \theta) \right) \quad (8)$$



$$m_0 = \frac{1}{z_k + z_{k+1}} \left\{ e^{z_k^2 t} \operatorname{erfc}(-z_k \sqrt{t}) - e^{z_{k+1}^2 t} \operatorname{erfc}(-z_{k+1} \sqrt{t}) \right\}$$



$$m_1 = \frac{1}{z_k - z_{k-1}} \left\{ \frac{1}{z_{k+1}} \left[ 1 + e^{z_{k+1}^2 t} \operatorname{erfc}(-z_{k+1} \sqrt{t}) \right] - \frac{1}{z_k} \left[ 1 - e^{z_k^2 t} \operatorname{erfc}(-z_k \sqrt{t}) \right] \right\}$$

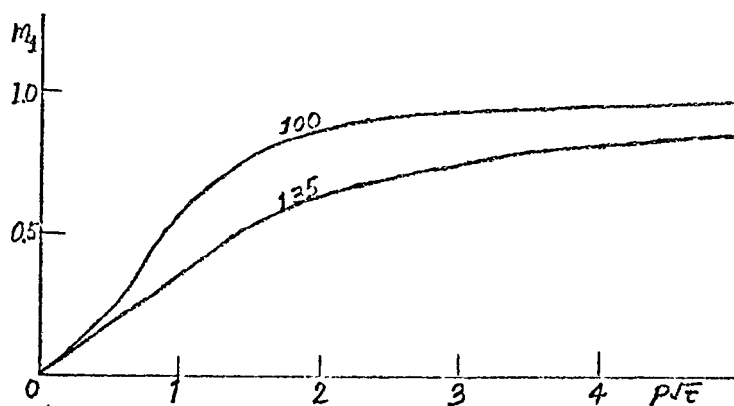


Fig.33.

where  $\rho$  and  $\theta$  are defined by  $z_k = \rho e^{j\theta}$ , and  $m_0$  and  $m_1$  functions are given by graphical forms as shown in Fig. 33.

The difficulties involved in their approach are as follows.

In the first place, if there is a repeated root in the transfer function, they don't have a corresponding formula for the inverse Laplace transformation.

Secondly, this is not a complete analytic method, because they have to use tables or auxiliary figures in Fig. 33 to evaluate the curves.

To remedy the shortcomings of this method is by no means easy task. However, if we look at the problem from a different angle and attack it from a numerical viewpoint in the very beginning, we would appreciate the power and simplicity of the Fast Fourier Transform method.

## 2. Fast Fourier Transform Approach

We use a simple problem which involves function element to explain the procedure.

Consider the feedback system shown in Fig. 34.

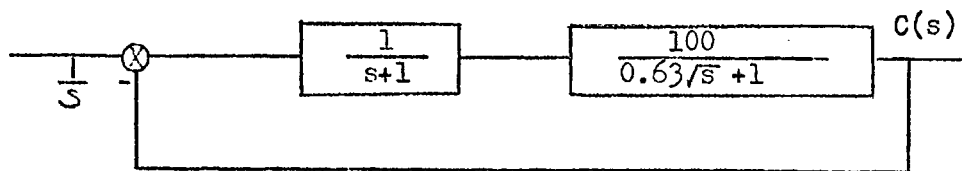


Fig. 34

Find the response function  $c(t)$ , if a unit step is applied.

First, find the close loop transfer function as follows.

$$M(s) = \frac{C(s)}{R(s)} = \frac{100}{(s+1)(0.63\sqrt{s}+1) + 100} \quad (8)$$

Program #16 will give the answer  $c(t)$ .

The sub-step involves in the program can be summarized as follows.

- (1) Substituting  $s$  by  $j\omega$  to evaluate  $M(j\omega)$
- (2) Inverse Fast Fourier Transform to obtain  $m(t)$
- (3) Numerical integration to get  $c(t)$

The input of this program should be written as subroutine. For this example it reads

```
COMPLEX FUNCTION CFUN(S)
COMPLEX S,EA,EB
CFUN=(100.,0.)/((S+(1.,0.))*(0.63*CSQRT(S)+(1.,0.))+(100.,0.))
RETURN
END
```

The output curve  $c(t)$  is plotted as shown in Fig. 35.

### 3. Special Case involving Repeated Root

While there is no available analytical method for evaluating an irrational transfer function with repeated roots, we use an example to demonstrate that there is no difficulties in the Fast Fourier Transform method.

Solve for the response of the following system with a unit step as input

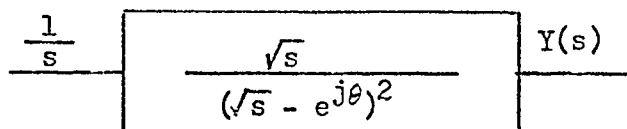


Fig. 36

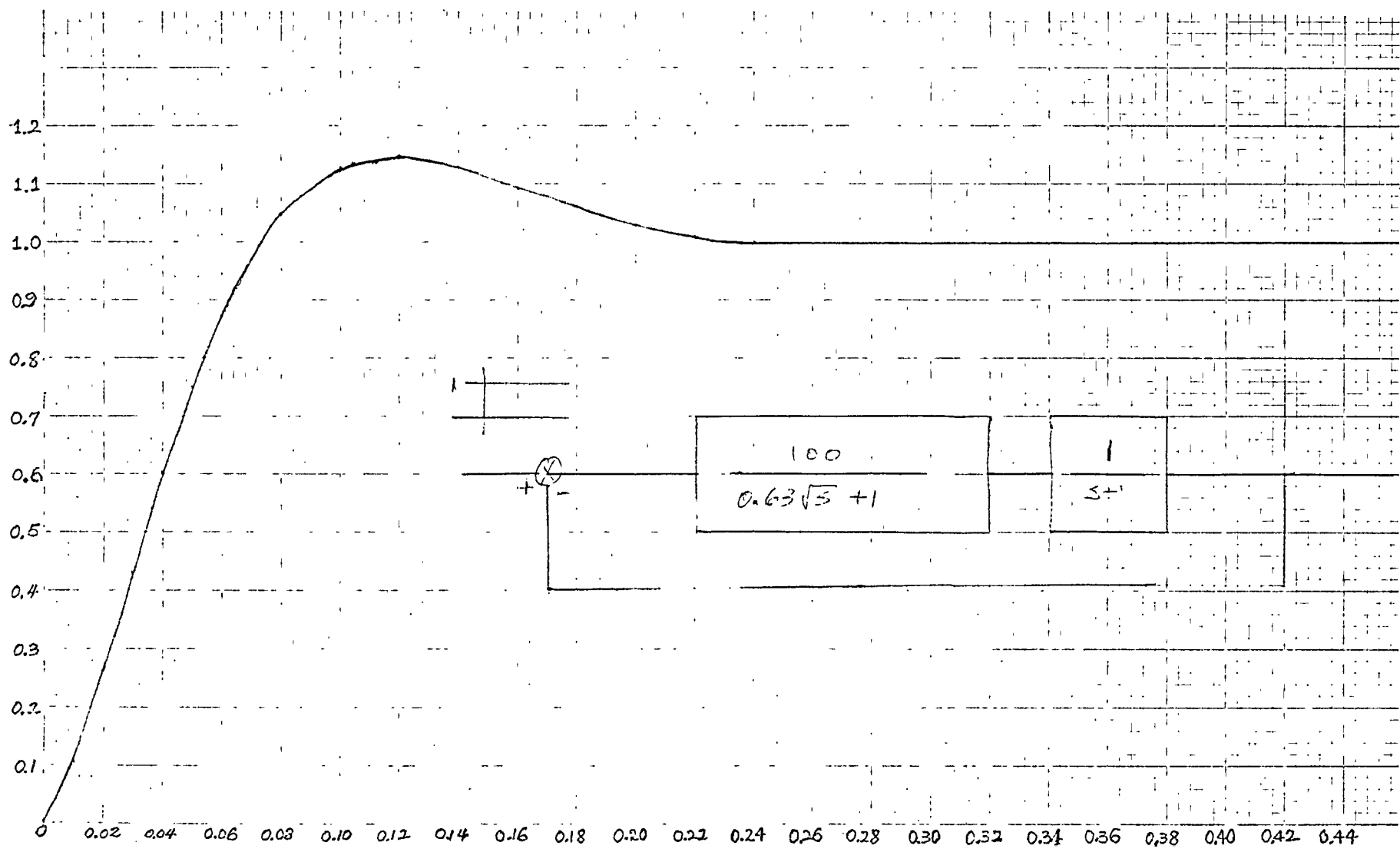


Fig.35

Again, we use program #16, with the following subroutine as computer input

```

COMPLEX FUNCTION CFUN(S)
COMPLEX S,EA,EB
EA=(0.,1.)*2.*3.1416/360.*55.
EB=EA
CFUN=CSQRT(S)/((CSQRT(S)*CLXP(EA))*(CSQRT(S)*CEXP(EB)))
RETURN
END

```

The computer output  $y(t)$  is plotted as shown in Fig. 37.

#### 4. Verification of $m_0$

The curves shown in Fig. 33 were given by Kiloneitseva and Netushil. We would use the Fast Fourier Transform method to verify their results. Thus we have  $m_0$  calculated and plotted as shown in Fig. 38. There is no recognizable difference between their results with ours.

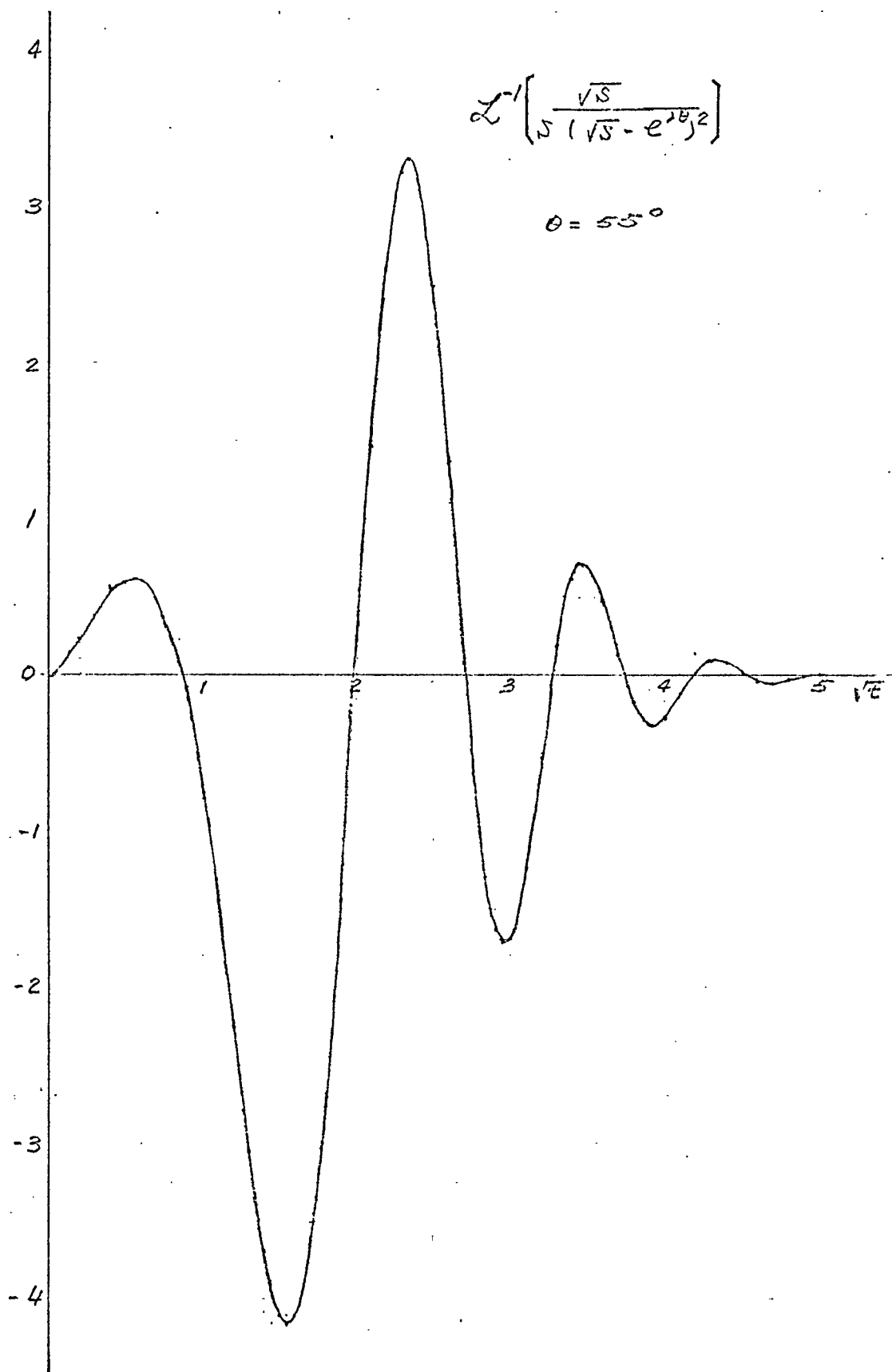


Fig. 37.

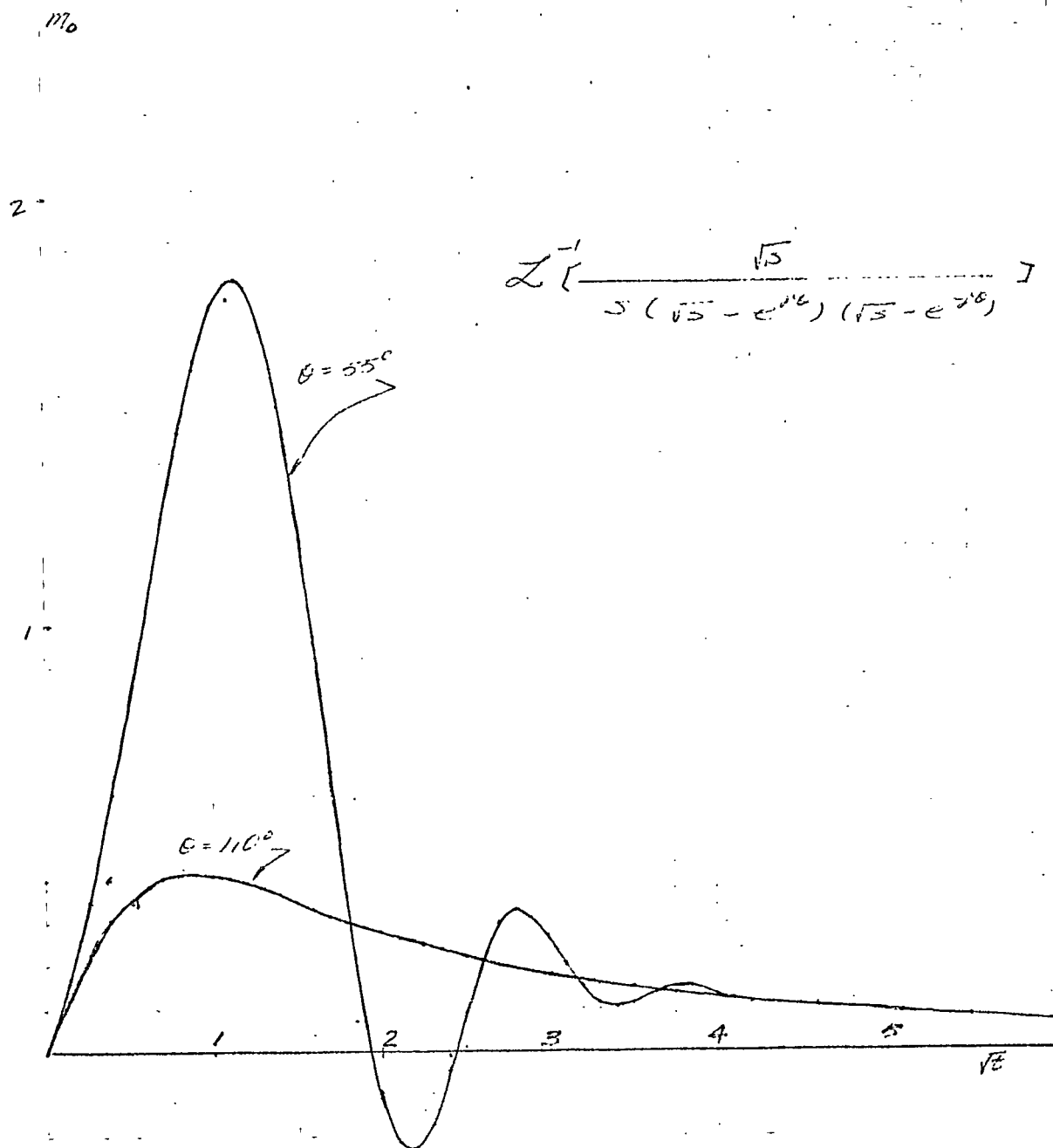


Fig. 38



## X. Inverse Laplace Transform of Transcendental Transfer Function

### 1. Time Delay Functions

If a control system involves a time delay element, its transfer function would involve an element of  $e^{-s}$  which makes the analysis of the system very difficult.<sup>10,15</sup>

One of the approaches to the problem is to replace the exponential function of  $s$  by a ratio of two polynomials. The method is known as Pade's approximation.<sup>10</sup> In fact, the approximate technique is the only practical method so far available.

When we have the powerful tool- Fast Fourier Transform, the solution of the problem becomes routine.

### 2. Delay Function in the Over All Transfer Function

We first use the following simple example: Find the weighting function of the system:

$$W_1(s) = \frac{e^{-s}}{s + 1} \quad (1)$$

By using Program # 5 and write the subroutine as follows

```

COMPLEX FUNCTION CFUN(S)
COMPLEX S
CFUN=CEXP(-S)/(S+(1.,0.))
RETURN
END

```

The output curve  $w(t)$  is shown in Fig. 39. Inspecting this curve, we note that our result is so accurate that it is much better than that obtained by Pade's formula.

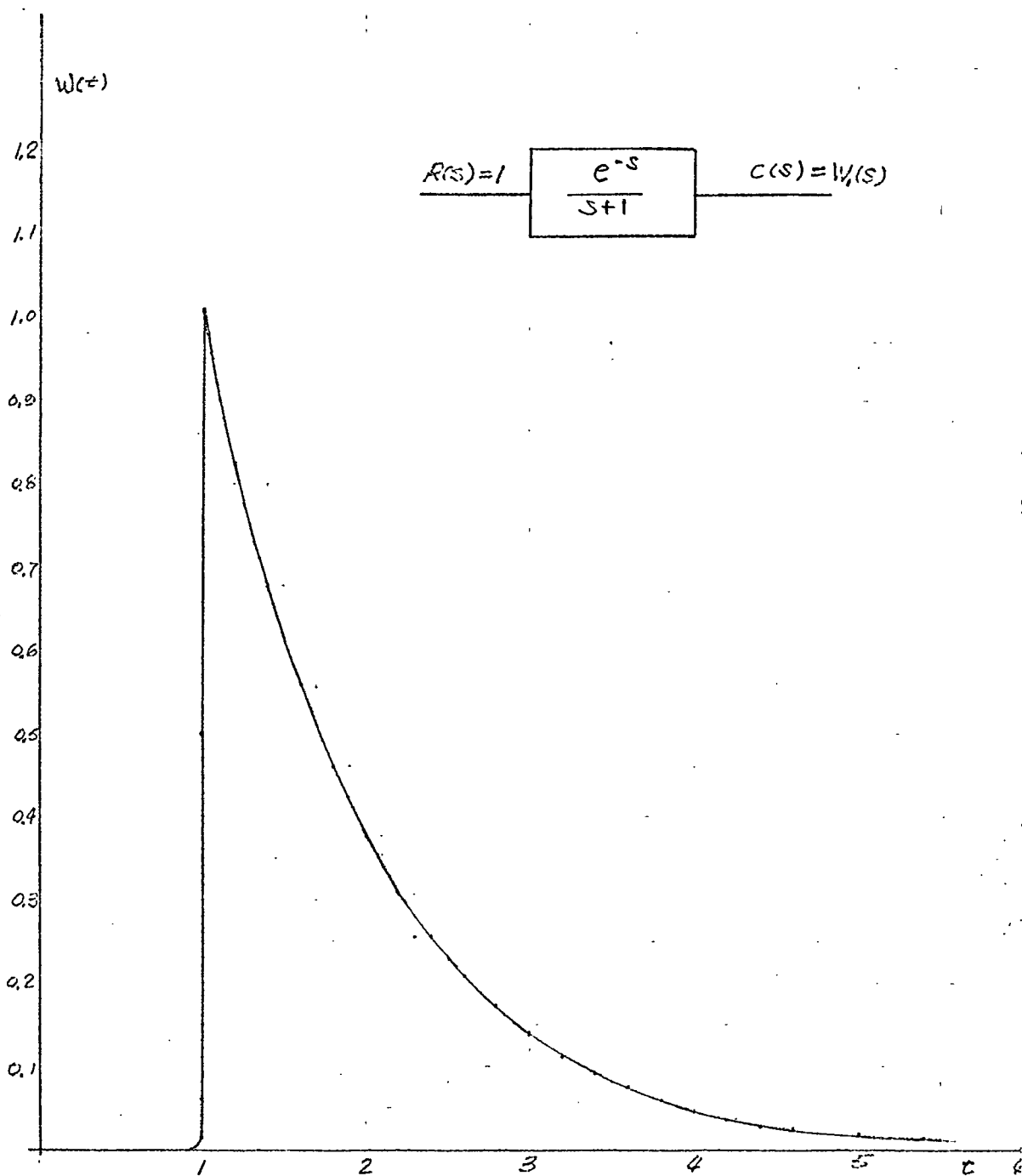


Fig. 39

### 3. Delay Function in the Loop

If the delay element is either in the feedforward path (Fig. 40a) or in the feedback link of a feedback system (Fig. 41a), the problem becomes complicated. However, with our approach, the solution is still a routine. After using program #16, we find the step responses of the system of Fig. 40(a) and 41(a) are the curves shown in Fig. 40(b) and 41(b) respectively.

### 4. Two Delay Functions

When two delay functions are applied in a feedback system as shown in Fig. 42, the solution problem is more complicated. Again, we solve it by using program #16 and find the step response as shown in Fig. 43.

### 5. General Transcendental Transfer Functions

If a system consists a general transcendental function, which could be both of  $s$  and  $e^{\sin(s)}$  etc. There is no any particular difficulty if we use the Fast Fourier Transform method to solve it.

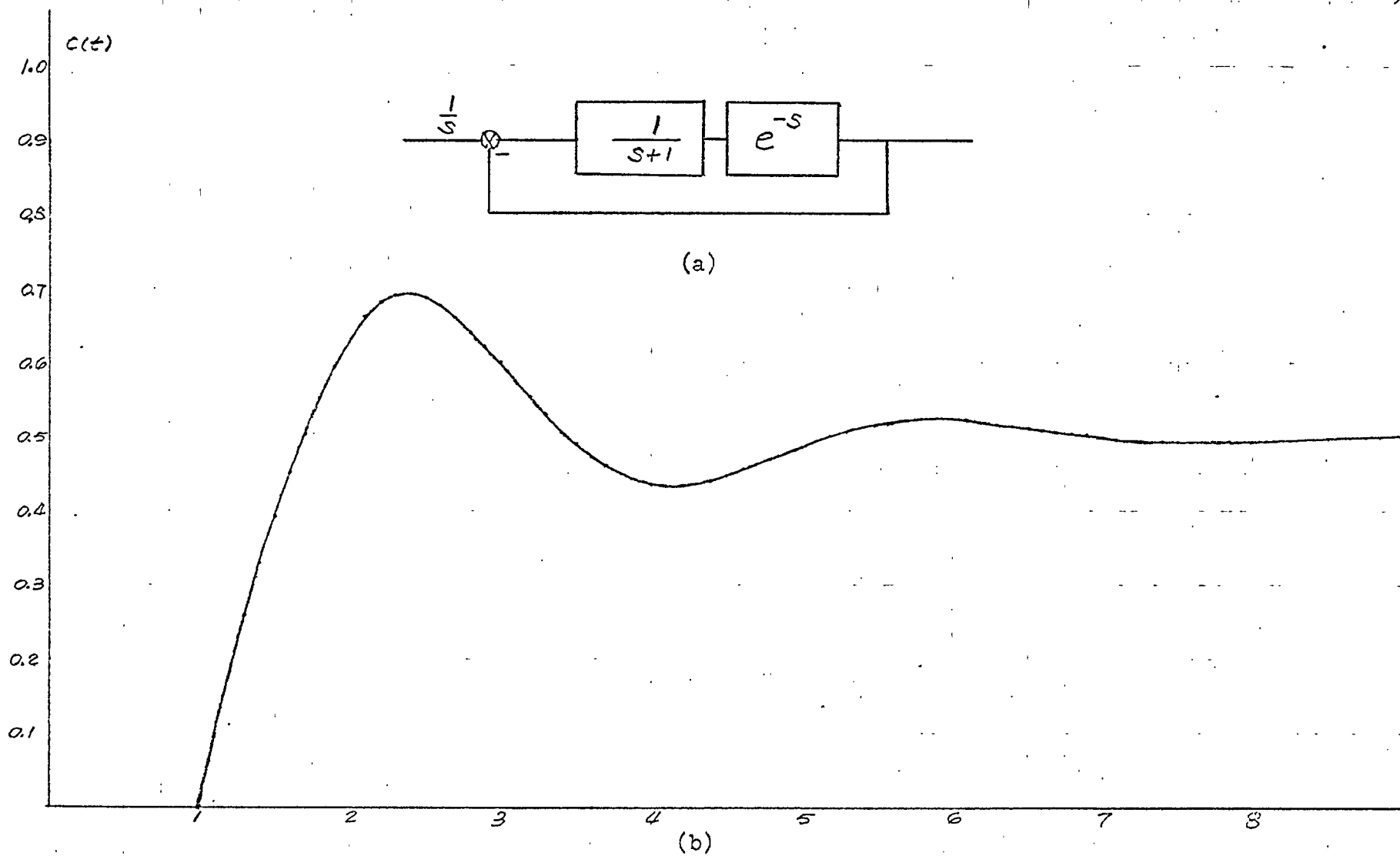
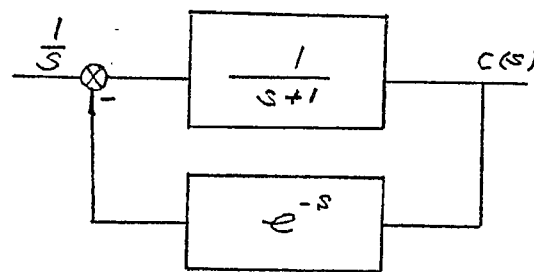
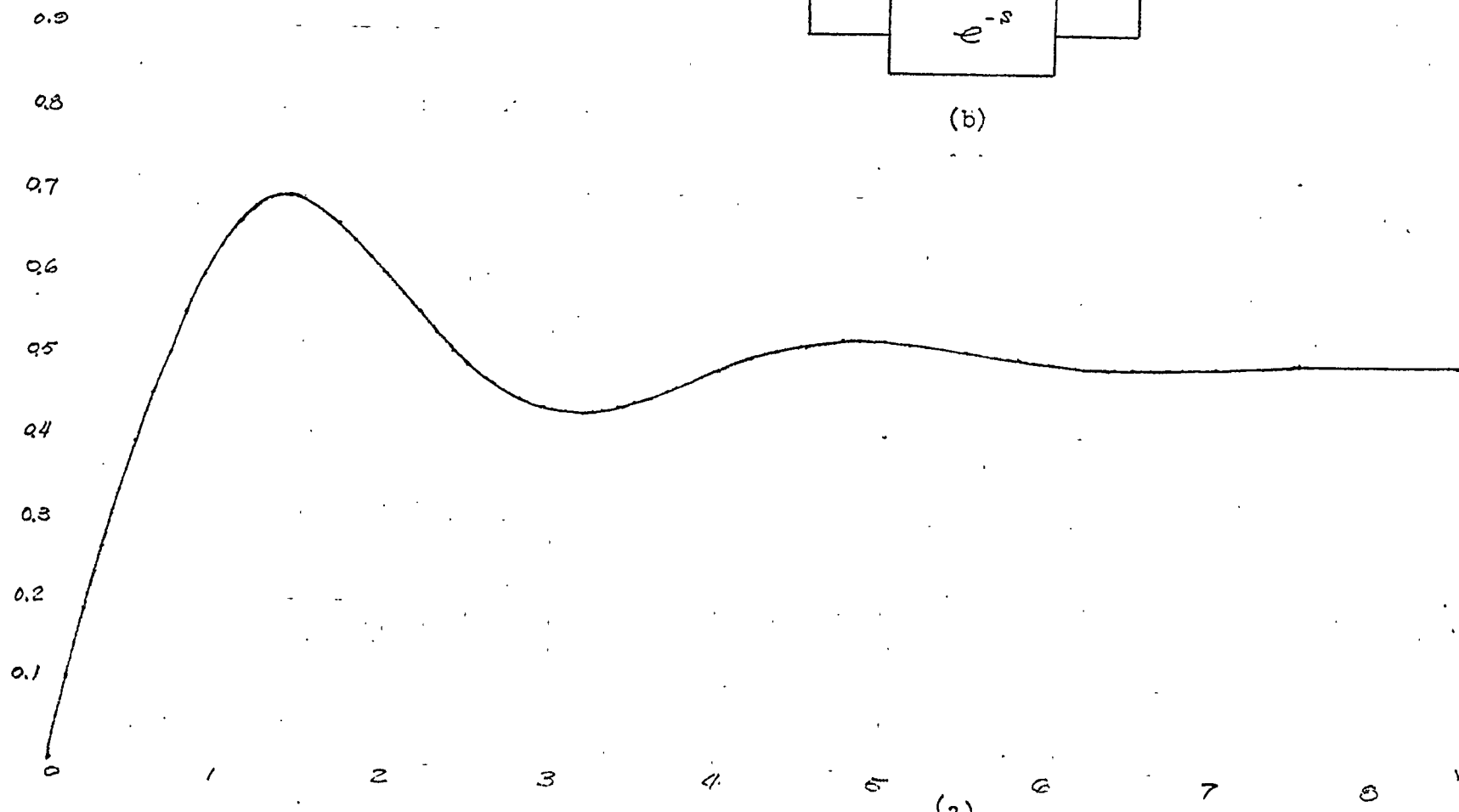


Fig. 40



(b)

Fig.41

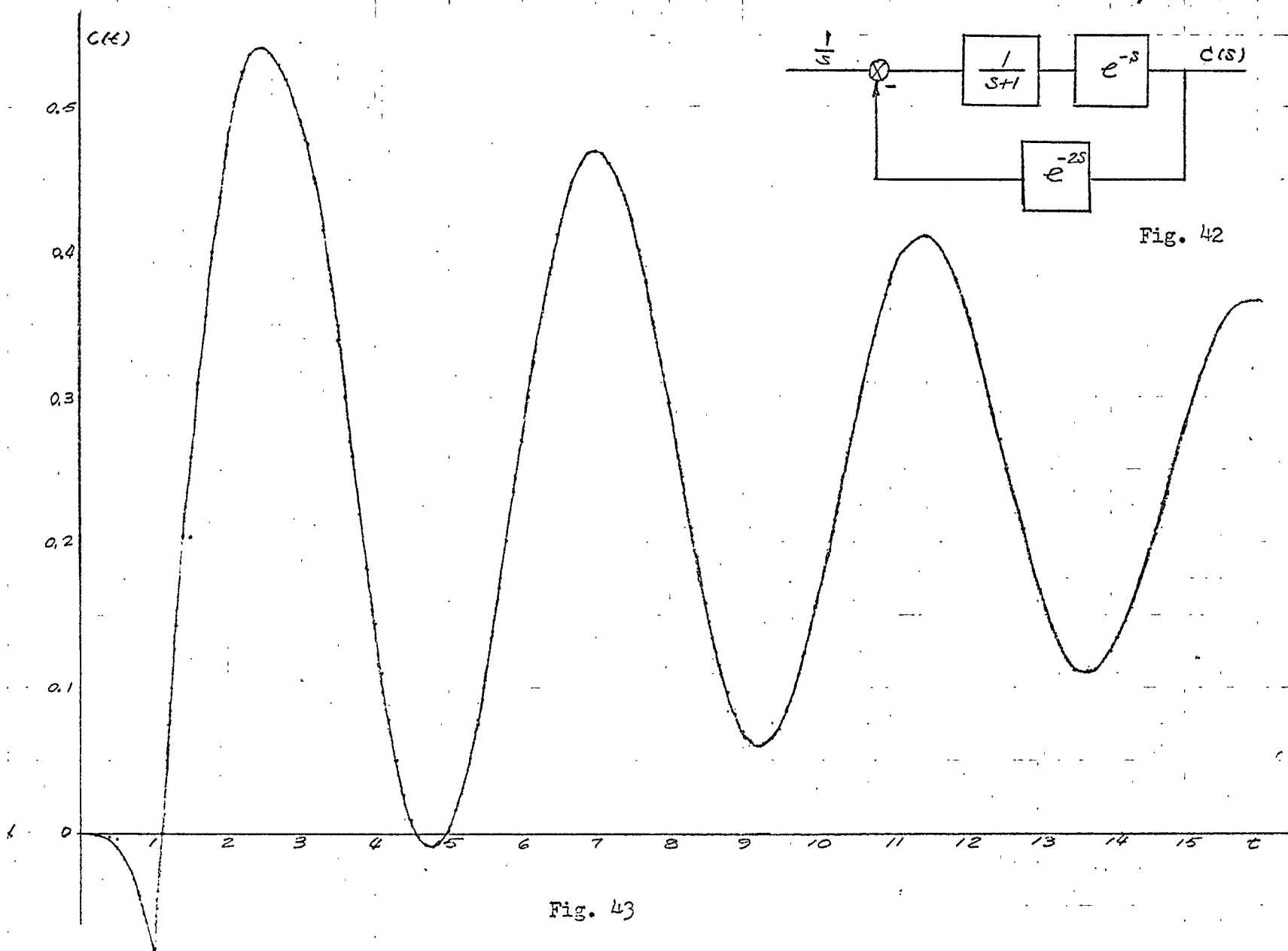


Fig. 43

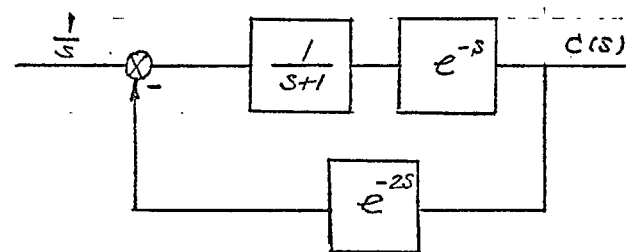


Fig. 42

## XI. Conclusion

The Fast Fourier Transform has demonstrated its power in the field of communication, optics etc. It has never been applied to control system analysis. This research first, states the difficulties of performing the inverse Fourier Transform by complex variable approach; in the meantime, we establish several useful transform pairs. Secondly, a new interpretation of Fast Fourier Transform is presented. Third, Decimation in time and Decimation in frequency are explained and derived. Fourth, the inverse Fourier Transform formulas have been established. Fifth, application to Fourier series evaluation is demonstrated. Sixth, applications to convolution, autocorrelation and crosscorrelation are explained. Seventh, this powerful tool, when applied to irrational transfer functions, and transcendental transfer functions, the inverse Laplace transform become routine.

Sixteen detailed computer programs are included. All of them are tested at the Sigma 7 digital computer.

It is believed that this research is the first one to use Fast Fourier Transform to perform the inverse Laplace Transformation of irrational transfer functions and transcendental transfer functions in literature.

## REFERENCES

1. Athanasios Papoulis, "The Fourier Integral and its Application", McGraw-Hill Book Company, New York, 1962.
2. J. T. Ton, "Digital and Sampled Data Control System", McGraw-Hill Book Company, New York, 1959.
3. C. F. Chen and I. J. Haas, "Elements of Control System Analysis", Prentice Hall, 1968, pp. 170-179.
4. J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series", Math of Comput., Vol. 19, pp. 297-301, April, 1965.
5. E. O. Brigham and R. E. Morrow, "The Fast Fourier Transform", IEEE Spectrum, Vol. 4, pp. 63-70, Dec., 1967.
6. W. T. Cochran, et al, "What is the Fast Fourier Transform?", IEEE Proc., Vol. 55, p. 1667, Oct., 1967.
7. W. M. Gentleman and G. Sande, "Fast Fourier Transforms-for Fun and Profit", 1966 Fall Joint Computer Conf. AFIPS Proc. Vol. 29, Washington, D. C. Spartan, 1966, pp. 563-578.
8. W. T. Cochran, J. J. Downing, D. L. Lang, and D. W. Nelson, "Burst Measurements in the Frequency Domain", Proc. IEEE, Vol. 54, pp. 830-841, June, 1966.
9. T. G. Stockham, "High Speed Convolution and Correlation", 1966 Spring Joint Computer Conference, AFIPS Proc., Vol. 28, Washington, D. C. : Sparten, 1966, pp. 229-233.
10. J. L. Stewart, "Generalized Pade Approximation", Proceedings of IRE, Dec. 1960, pp. 2003-2008.
11. M. B. Kilomeitseva, and A. V. Netushil, "Transients in Automatic Control Systems with Irrational Transfer Functions", English Translation of Avtomatika i Telemekhanika, Vol. 26, NO. 2, Feb. 1965, pp. 335-342.



12. V. V. Solodovnikov, Statistical Dynamics of Linear Automatic Control System, New York: D. Van Nostrand Company, second edition, 1965, pp. 335-342.
13. Y. W. Lee "Statistical Theory of Communication", John Wiley & Sons Inc. 1960.
14. Athanasios Papoulis "Probability, Random Variables, and Stochastic Processes", McGraw-Hill Book Company.
15. R. H. Macmillan "Progress in Control Engineering", Vol. 1, London Heywood & Company. 1962.
16. Pipe, "Applied Mathematics for Engineers and Physicist", McGraw-Hill Book Company, New York, 1958.