TRANSFER FUNCTION ESTIMATION USING THE FAST FOURIER TRANSFORM

Arthur Kislack

United States
Naval Postgraduate School

THESIS

TRANSFER FUNCTION ESTIMATION USING
THE FAST FOURIER TRANSFORM

by

Arthur Kislack

December 1969

This document has been approved for public release and sale; its distribution is unlimited.

Transfer Function Estimation Using the Fast Fourier Transform

by

Arthur Kislack
Lieutenant, United States Navy
B.S.E.E., Purdue University, 1962

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1969

ABSTRACT

A method is proposed for the estimation of the transfer function of a linear, time-invariant system with no numerator dynamics, from random input and output data. The method employed utilizes the Fast Fourier Transform Algorithm and Least Squares Estimation to obtain the coefficients of the system's transfer function.

The procedure has been modeled in FORTRAN IV on an IBM-360 computer. The results of simulation show the feasibility of estimating the order of the transfer function and its coefficients.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

I. INTRODUCTION

The problem of identifying a system (or plant) has been the object of much thought and effort as evidenced by the Identification in Automatic Control Systems Symposium [1]. The effort has been primarily expended on methods which are suitable to a time domain analysis. Identification in the frequency domain has in general involved lengthy computations to obtain a Fourier Transform if the computations are done digitally, or a tradeoff between many parallel narrow band filters or repetitive playback of the system input and output if the analysis is done by analog methods. The method to be used in this investigation has been constrained to make use of the Fast Fourier Transform algorithm. Therefore digital computations and techniques logically follow. The Fast Fourier Transform (FFT) algorithm reduces the number of complex multiplications required in the evaluation of the Fourier Transform and thus may provide a means of performing system identification on-line or approximately in real time. The FFT is discussed in Appendix A.

II. PROBLEM STATEMENT

The mathematical model of a linear, time-invariant system may be expressed as a differential equation or as a transfer function. The determination of the model parameters from observations on the system which is to be modeled is generally called an identification problem. This investigation considers the modeling of a system, which is characterized by a vector differential equation, such that the coefficients of the equivalent transfer function can be estimated.

Let a linear system transfer function be written as

$$\frac{X(f)}{V(f)} = \frac{1}{D(f)} \;; \quad D(f) = \sum_{n=0}^{N} d_n (j\omega)^n \;. \tag{1}$$

The equivalent N^{th} order differential equation is written

$$\sum_{n=0}^{N} d_n \frac{d^n x(t)}{dt^n} = v(t) \;. \tag{2}$$

$$d_N = 1.0$$

From (2) the equivalent N-vector differential equation (3) can be obtained.

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \cdot \\ \cdot \\ \dot{x}_{N-1}(t) \\ \dot{x}_N(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdot & \cdot & \cdot & & \\ \cdot & \cdot & \cdot & & \\ 0 & 0 & 0 & \cdots & 1 \\ -d_0 & -d_1 & -d_2 & \cdots & -d_{N-1} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \\ \\ x_{N-1}(t) \\ x_N(t) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \\ 1 \end{bmatrix} v(t) \tag{3}$$

<u>OR</u>

$$\dot{\underline{x}}(t) = A\underline{x}(t) + Bv(t)$$

The solution of (3) is

$$\underline{x}(t) = \Phi(t,t_0)\underline{x}(t_0) + \int_{t_0}^{t} \Phi(t,\tau) \cdot B \cdot v(\tau)d\tau, \qquad (4)$$

where $\underline{x}(t)$ is the system state vector and the element $x_1(t)$ is the system output $x(t)$. The discrete version of (4), which is derived in Appendix B, is used as the equivalent representation of (1) and the coefficients d_0, d_1, d_2.....d_N are estimated. The estimation scheme is represented by Figure 1.

The investigation is to be constrained so that $v(t)$ is broad-band white noise and the Fast Fourier Transform algorithm must be used in the identification scheme.

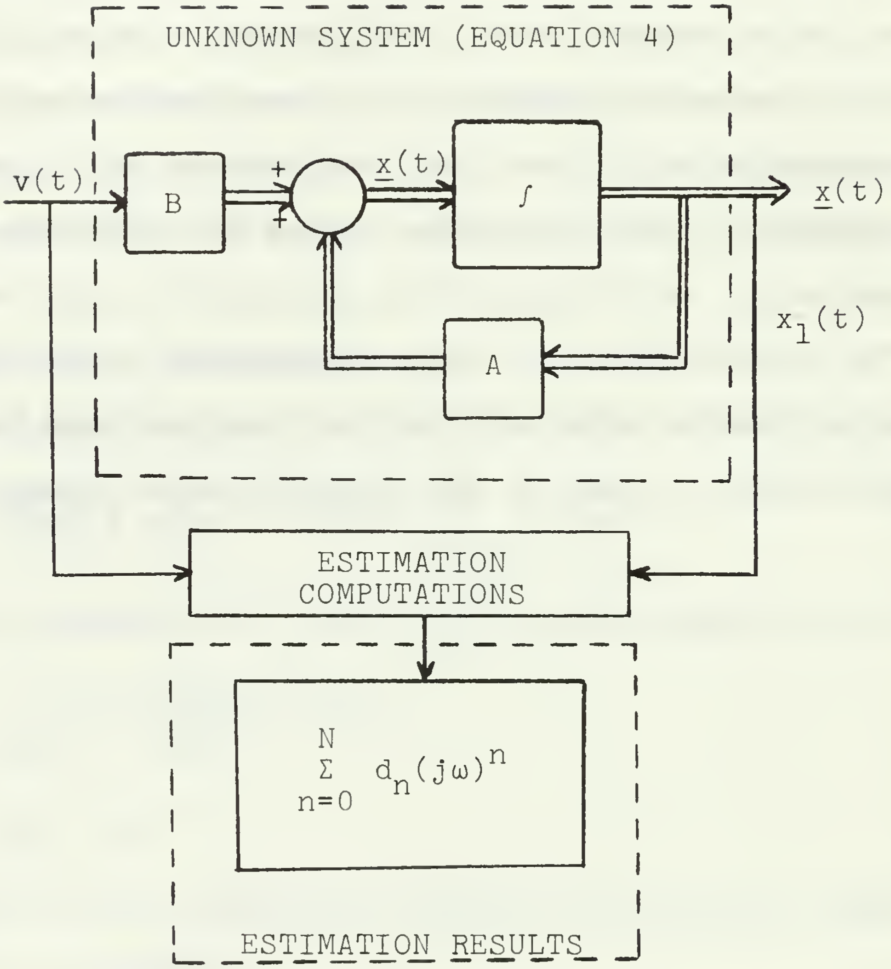


Figure 1. Estimation Scheme Representation

III. SYSTEM MODEL FOR A DIGITAL COMPUTER

A linear, time-invariant system represented by the vector differential equation (5) has equation (6) as its solution.

$$\dot{\underline{x}}(t) = A\underline{x}(t) + Bv(t) \tag{5}$$

$$\underline{x}(t) = \varepsilon^{A(t-t_0)}\underline{x}(t_0) + \int_{t_0}^{t} \varepsilon^{A(t-\tau)}Bv(\tau)d\tau \tag{6}$$

If $v(\tau)$ is deterministic and the convolution integral can be solved, then a discrete solution of (5) can be modeled on a digital computer. $\varepsilon^{A(t-t_0)}$ can be converted to a square matrix by using the Laplace transform method [2, pg. 316]. However, since $v(\tau)$ is not deterministic the convolution integral cannot be solved, so an approximation to (6) is required.

Considering only the unforced response of (6),

$$\underline{x}(t) = \varepsilon^{A(t-t_0)}\underline{x}(t_0) \ . \tag{7}$$

At some instant of time $t = t_M = t_0 + MT$,

$$\underline{x}(t_M) = \varepsilon^{AMT}\ \underline{x}(t_0) \tag{8}$$

which is identical to the unforced response of equation (3) of Appendix B where $\underline{x}(MT) = \underline{x}(t_0 + MT) = \underline{x}(t_M)$,

$$x((m+1)T) = \varepsilon^{AT}\underline{x}(mT) \qquad (9)$$

$$\underline{x}(T) = \varepsilon^{AT}\underline{x}(0) = \varepsilon^{AT}\underline{x}(t_0)$$

$$\underline{x}(2T) = \varepsilon^{AT}\underline{x}(T) = \varepsilon^{A2T}\underline{x}(t_0)$$

$$\vdots \qquad \vdots$$

$$\underline{x}(MT) = \varepsilon^{AMT}\underline{x}(t_0) \qquad (10)$$

Therefore, since (10) is equivalent to (8) which is the discrete solution of (7), then (9) is the discrete solution for the unforced response of (6) for $t = t_{(m+1)} = t_0 + (m+1)T$ or in the notation of Appendix B, $t = (m+1)T$.

The convolution integral from (6),

$$g(t) = \int_{t_0}^{t} \varepsilon^{A(t-\tau)}Bv(\tau)d\tau, \qquad (11)$$

cannot be forced to yield an exact discrete solution without performing the indicated integration. Therefore, the sampled data approach in Appendix B is necessary. Normally this approach yields

$$g((m+1)T) = \int_{0}^{T} \varepsilon^{A(T-\tau)}Bv(mT)d\tau, \qquad (12)$$

where $v(mT)$ is assumed to be constant for $mT \leq t < (m+1)T$ [2, pg. 340]. This approach presupposes that the bandwidth of $v(t)$ is small compared to $\frac{1}{T}$. Since $v(t)$ is assumed to be broadband noise, T should be small. However, as T decreases, the number of iterations required in the simulation for a given time span of $x(t)$ increases. Each iteration requires that both the unforced and forced responses of equation (6) be computed. T should then be large to minimize computation and small to allow the maximum transfer of information to the system concerning the variations of v(t). Equation (12) of Appendix B was chosen as a compromise. By using this equation, v(t) can be sampled at a rate greater than $\frac{1}{T}$ and the unforced response of equation (6) need only be computed once every T time interval.

IV. SYSTEM IDENTIFICATION SCHEME

A continuous, time-invariant system is to be identified by estimating the coefficients of its transfer function. The input to the system is broad-band white noise of known mean and variance. The identification scheme is depicted by Figure 2, where F denotes the Fourier Transform, i.e.,

$V(f) = F[v(t)]$

$X(f) = F[x(t)]$

$v(t)$ - broadband white noise

$x(t)$ - system output

The object of this investigation is to make use of the Fast Fourier Transform (FFT, Appendix A) in place of F, and Least Squares Estimation in the computations for the transfer function coefficients.

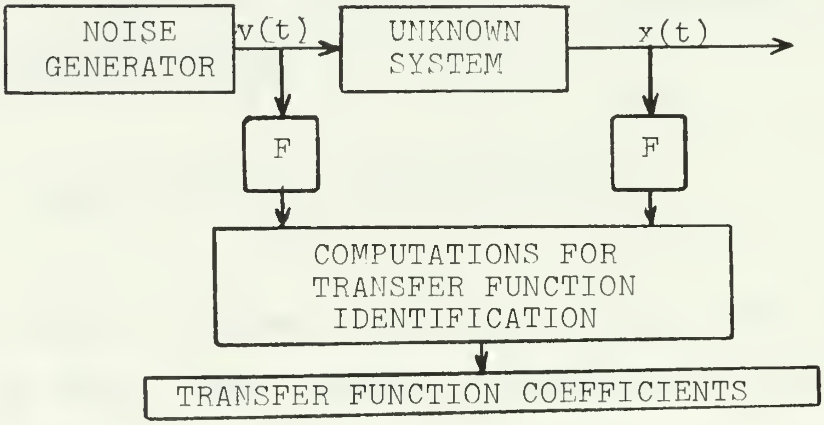

NOISE GENERATOR
v(t)
UNKNOWN SYSTEM
x(t)
F
F
COMPUTATIONS FOR TRANSFER FUNCTION IDENTIFICATION
TRANSFER FUNCTION COEFFICIENTS

Figure 2. Identification Block Diagram

In Figure 2, v(t) is the output of a random process. In particular it is the output of a gaussian random process. Since v(t) is random, then x(t), V(f) and X(f) are random processes. V(f) and X(f) are not themselves deterministic; however, from Goldman, [3, pg. 279]

$$H(f) = X(f)/V(f) \,, \qquad (13)$$

where H(f) is the frequency response of the unknown system. Since H(f) is assumed to be of the form

$$H(f) = \frac{1}{\sum\limits_{n=0}^{N} d_n (j2\pi f)^n} = \frac{1}{D(f)}$$

$$d_N = 1.0,$$

then (13) can be written as [Appendix C, equation (2)]

$$X(f) \cdot D(f) = V(f). \qquad (14)$$

From (14) a residue r(f) is formed,

$$r(f) = X(f) \cdot D(f) - V(f). \qquad (15)$$

Through the Fourier Transform, measurements on x(t) and v(t) yield X(f) and V(f). A modified least squares estimation procedure then uses r(f) from (15) and its complex conjugate to form

$$Q = \int_0^F r(f) \cdot r^*(f) df \qquad (16)$$

where $r^*(f)$ is the complex conjugate of $r(f)$. The object is then to minimize Q with respect to the d_m's, the coefficients of $D(f)$.

Since the FFT [Appendix A] is used in place of the Fourier Transform of Figure 2 and the unknown system is modeled in accordance with Appendix B, the identification scheme takes on the form of Figure 3.

Since $u(t)$ is broadband noise, the low-pass filter is a necessary addition to the identification scheme to prevent foldover of the alias frequencies of V_i. The low-pass filter used is a discrete version of an R-C filter with its break point on a Bode plot being greater than the expected bandwidth of the system under investigation. The sampling rate of the inputs to the FFT's was then chosen to be at least twice as great as the low-pass filter cut-off frequency, that frequency at which the filter output drops by 3db.

X_i and V_i are the results of the Discrete Fourier Transforms of $x(t)$ and $u(t)$ which are observed through a time window of τ seconds, i.e.

$$X_i = X(f_i) = X(\frac{i}{\tau})$$

$$V_i = V(f_i) = V(\frac{i}{\tau}) \quad ,$$

where τ is the time interval over which $u(t)$ and $x(t)$ are observed. As the index i goes from $-\tau W$ to τW, X_i and V_i represent the complex values of the Fourier Transforms of $x(t)$ and $u(t)$ at the discrete frequencies $-W$ to $+W$.

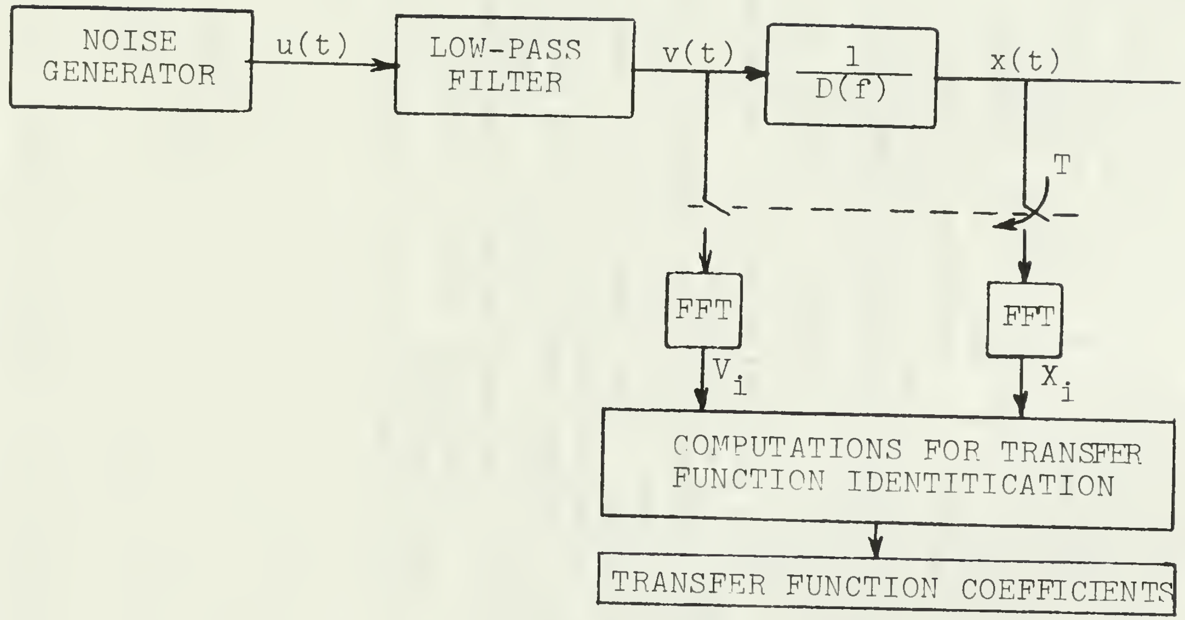

NOISE GENERATOR
u(t)
LOW-PASS FILTER
v(t)
$\frac{1}{D(f)}$
x(t)
T
FFT
FFT
V_i
X_i
COMPUTATIONS FOR TRANSFER FUNCTION IDENTITICATION
TRANSFER FUNCTION COEFFICIENTS

Figure 3. Identification Block Diagram with Signal Sampling

The X_i's and V_i's exist only at frequencies which are multiples of $\frac{1}{\tau}$. If x(t) and v(t) contain only frequencies which are exact multiples of $\frac{1}{\tau}$ then the X_i's and V_i's will be the exact Fourier coefficients of x(t) and v(t)[6]. However, since it is unlikely that the only frequencies present are multiples of $\frac{1}{\tau}$, these other frequency components will cause the X_i's and V_i's to be altered. It would be advisable to weight the frequency spectra of x(t) and v(t) with the hamming or hanning functions [4, 5, 6, Appendix A] to minimize the effects of these other frequency components. This weighting was not performed, although it is admittedly advisable, since computer computation time would be increased. The trade-off of less computation time for increased accuracy was felt to be justified since much of the computer time used in this investigation had to be charged to other projects.

V. RESULTS

Many of the variables in the computer simulation of the "unknown" plant and the identification scheme [Table D1, Appendix D] are peculiar to the approach taken in this investigation. For an actual application of the identification scheme the "unknown" plant is not simulated and only the sampling rate and number of samples to be taken must be determined. The expected or known signal bandwidth dictates the sampling rate, where the minimum rate is the Nyquist sampling rate. An increase in the number of samples taken increases the frequency resolution of the FFT output and the accuracy of the estimated transfer function coefficients. However, it also increases computation time and computer memory required.

A tenth order system such as equation (17) was assumed for each trial.

$$\frac{X(S)}{V(S)} = \frac{1.0}{\sum\limits_{n=0}^{10} \hat{d}_n S^n} \tag{17}$$

The low-pass filter shown in Figure 3 is represented by equation (18).

$$\frac{V(S)}{U(S)} = \frac{2\pi \cdot 100}{S+2\pi \cdot 100} \tag{18}$$

Table 1 presents the most encouraging of the results of this investigation. Trials 1 and 2 represent first-order systems with real S-plane poles. Trials 3 and 4 represent second-order systems with real and complex poles respectively. Trial 5 represents a second-order system with complex poles and trial 6 is a third-order system with real poles of -10, -20 and -30. Only the five lowest-order coefficients of the estimates are given in Table 1 since the higher-order estimates become negligibly small. The results given are the averages of fifty consecutive estimates made for each system's coefficients.

TABLE 1

ESTIMATION RESULTS

Actual System coefficients = d_n

Estimated coefficients = $\hat{d}_n$

d_1 through d_6 = 0

TRIAL	#SAMPLE POINTS	SAMPLING RATE(HERTZ)	n	d_n	$\hat{d}_n$ LOW PASS FILTER IN	OUT
1	256	166	7	0.0	$9.6\cdot10^{-9}$	$-2.6\cdot10-8$
			8	0.0	$4.2\cdot10^{-6}$	$1.0\cdot10^{-5}$
			9	0.0	$-1.8\cdot10^{-3}$	$-3.5\cdot10^{-3}$
			10	1.0	$9.1\cdot10^{-1}$	$9.3\cdot10^{-1}$
			11	$6.0\cdot10^{1}$	$6.2\cdot10^{1}$	$6.2\cdot10^{1}$
2	512	416	7	0.0	$-1.1\cdot10^{-10}$	$-9.2\cdot10^{-9}$
			8	0.0	$2.0\cdot10^{-7}$	$1.2\cdot10^{-6}$
			9	0.0	$-4.2\cdot10^{-4}$	$-2.6\cdot10^{-3}$
			10	1.0	$7.9\cdot10^{-1}$	$5.6\cdot10^{-1}$
			11	$6.0\cdot10^{2}$	$6.1\cdot10^{2}$	$6.8\cdot10^{2}$
3	512	250	7	0.0	$4.7\cdot10^{-6}$	$5.3\cdot10^{-6}$
			8	0.0	$-9.3\cdot10^{-4}$	$-8.7\cdot10^{-4}$
			9	1.0	$8.6\cdot10^{-1}$	$8.6\cdot10^{-1}$
			10	$8.0\cdot10^{1}$	$7.5\cdot10^{1}$	$7.6\cdot10^{1}$
			11	$1.5\cdot10^{3}$	$1.5\cdot10^{3}$	$1.5\cdot10^{3}$
4	512	83	7	0.0	$5.3\cdot10^{-6}$	$2.3\cdot10^{-5}$
			8	0.0	$-1.3\cdot10^{-4}$	$-1.5\cdot10^{-3}$
			9	1.0	$9.3\cdot10^{-1}$	1.0
			10	$6.0\cdot10^{1}$	$5.0\cdot10^{1}$	$5.1\cdot10^{1}$
			11	$3.6\cdot10^{3}$	$3.6\cdot10^{3}$	$3.7\cdot10^{3}$

TRIAL	#SAMPLE POINTS	SAMPLING RATE(HERTZ)	n	d_n	$\hat{d}_n$ LOW PASS FILTER	
					IN	OUT
5	512	416	7	0.0	$2.1\cdot10^{-6}$	$1.3\cdot10^{-6}$
			8	0.0	$-5.3\cdot10^{-5}$	$-1.5\cdot10^{-4}$
			9	1.0	$6.2\cdot10^{-1}$	$5.4\cdot10^{-1}$
			10	$1.2\cdot10^{1}$	$1.1\cdot10^{1}$	8.2
			11	$3.6\cdot10^{3}$	$2.4\cdot10^{3}$	$2.1\cdot10^{3}$
6	1024	83	7	0.0	$2.5\cdot10^{-3}$	$2.0\cdot10^{-3}$
			8	1.0	$7.8\cdot10^{-1}$	$7.8\cdot10^{-1}$
			9	$6.0\cdot10^{1}$	$5.8\cdot10^{1}$	$5.6\cdot10^{1}$
			10	$1.1\cdot10^{3}$	$1.0\cdot10^{3}$	$1.0\cdot10^{3}$
			11	$6.0\cdot10^{3}$	$6.3\cdot10^{3}$	$6.1\cdot10^{3}$

VI. CONCLUSIONS

The identification scheme presented in this investigation does provide good estimates of the transfer function coefficients for an unknown system with no numerator dynamics. The estimated coefficients in Table 1 are seen to be within 50% of the actual coefficients for all cases except when the actual coefficient equals zero. For these coefficients the estimates rapidly approach zero since the estimates are for higher order coefficients than those that exist in the actual system. The identification scheme therefore provides an indication of the order of the system.

The need for a low pass filter on the input to the "unknown" system has not been substantiated. Trials 1 through 6 of Table 1 indicate that the filter can be deleted with no disastrous effects.

The feasibility of using the Fast Fourier Transform as a tool for system (plant) identification has been shown.

VII. RECOMMENDATIONS

This investigation has been unduly long and involved due to the requirement that all the modeling be done on a digital computer. Many of the parameters listed in Table D1 of Appendix D required a large number of trial runs to determine a best set of values for the overall simulation. These trials were necessary to aid in finding the minimum iteration rate and input signal sampling rate required to adequately simulate the continuous "unknown" plant since these rates are fundamental to the total simulation time (not to be confused with identification time). It is suggested that any further investigation in this area be done on a Hybrid computer so that the continuous plant may be run on the analog portion of the computer and the identification scheme on the digital portion.

As a starting point the identification scheme requires that a guess be made as to the maximum number of coefficients in the unknown plant's transfer function. A major effect of this guess is that it determines the dimensions of matrix W of equation (15), Appendix C. Since the inverse of W is required it would appear that its dimensions should be kept as small as possible. However, it has been observed that the dimensions of W that produce the most nearly correct estimates of the transfer function coefficients are not the smallest (consistent with the number of coefficients known to exist) nor the largest (which depend on the amount of computer memory reserved for W). The best dimensions for W are different even for different plants of the same order. It is suggested that any future investigation utilizing this identification scheme should involve a method of avoiding the inverse of W or attempt an analytic solution of W^{-1}.

APPENDIX A

The Fast Fourier Transform (FFT)

The FFT is used to evaluate the complex Fourier transform of a sequence of complex numbers. It is a finitie discrete Fourier transform (DFT) with the restriction that the number of samples in the complex sequence is constrained to be an exact power of two. The power-of-two constraint on the number of samples allows implementation of an algorithm on a digital computer for rapid computation of the finite DFT.

The mechanics of the FFT and its characteristics has been widely discussed in the literature [7, 8, 9, 10, 11, 12, 13]. A few of the characteristics will be presented here.

From Goldman [3], a function g(t) which exists only for $\frac{-T}{2} < t < \frac{T}{2}$ and which has a Fourier transform G(f) which exists only for $-W < f < W$ can be expressed by the DFT pair

$$g\left(\frac{m}{2W}\right) = \frac{1}{T} \sum_{n=-TW}^{TW} G\left(\frac{n}{T}\right) \varepsilon^{j \frac{2\ mn}{2TW}} \qquad (1)$$

$$G\left(\frac{n}{T}\right) = \frac{1}{2W} \sum_{m=-TW}^{TW} g\left(\frac{m}{2W}\right) \varepsilon^{-j \frac{2\ mn}{2TW}} \qquad (2)$$

where $g\left(\frac{m}{2W}\right) = g(t)$, for $t = \frac{m}{2W}$

$G\left(\frac{n}{T}\right) = G(f)$, for $f = \frac{n}{T}$.

$g(t)$ and $G(f)$ are completely determined by their 2TW+1 sample values $g(\frac{m}{2W})$ and $G(\frac{n}{T})$ respectively.

The assumptions leading to (1) and (2), that $g(t)$ exist only for $\frac{-T}{2} < t < \frac{T}{2}$ and $G(f)$ exists only for $-W < f < W$, are physically unrealizable. For example, consider the case where $g(t)$ is a section of a continuous function $x(t)$ such that

$$x(t) \text{ is continuous for } -\infty < t < \infty \tag{3}$$

$$g(t) = x(t) \cdot y(t)$$

$$y(t) = u(t+\tfrac{T}{2}) - u(t-\tfrac{T}{2})$$

$$u(t) \text{ is the unit step function .}$$

Then $$G(f) = \int_{-\infty}^{\infty} X(q)Y(f-q)dq \tag{4}$$

and $$Y(f) = T\,\frac{\sin\,(\pi fT)}{\pi fT}\ . \tag{5}$$

$Y(f)$ and therefore $G(f)$ are continuous for $-\infty < f < \infty$.

By observing (sampling) a section of $x(t)$ such that $g(t) = x(t)$ for $\frac{-T}{2} < t < \frac{T}{2}$ the assumption leading to (1) and (2) are violated.

If a $y(t)$ in (3) is chosen so that $g(t) = x(t) \cdot y(t)$ has negligible value outside of the range $\frac{-T}{2} < t < \frac{T}{2}$ and $G(f)$ is negligible outside the range $-W < f < W$, then (1) and (2) can be made to be approximately exact. $y(t)$'s to satisfy the above include the Hanning and Hamming functions [4, 5] and the Parzen spectral window [14].

For the FFT, (1) and (2) are normally expressed as

$$g(m) = \sum_{n=0}^{N-1} G(n)\varepsilon^{j\frac{2\pi mn}{N}} \qquad (6)$$

$$G(n) = \frac{1}{N}\sum_{m=0}^{N-1} g(m)\ \varepsilon^{-j\frac{2\pi mn}{N}}\ , \qquad (7)$$

where $g(m) = g(\frac{m}{2W})$

$G(n) = G(\frac{n}{T})$

$n,m = 0$ to $N-1$

$N = 2^k$.

Some authors put the weighting $\frac{1}{N}$ on (6) rather than (7). However, the FFT used in this investigation is from the IBM/360 Scientific Subroutine package (360A-CM-03X) and in this algorithm the $\frac{1}{N}$ weighting is placed on (7).

APPENDIX B

Discrete Solution for $\underline{x}(t)$

Given the vector differential equation (1) it is desired to obtain a discrete solution for $\underline{x}(t)$.

$$\dot{\underline{x}}(t) = A\underline{x}(t) + Bv(t), \qquad (1)$$

where $\underline{x}(t) = [x_1(t)\ x_2(t)\ \text{----------}x_N(t)]^T$

$$\dot{\underline{x}}(t) = [\dot{x}_1(t)\ \dot{x}_2(t)\ \text{----------}\dot{x}_N(t)]^T$$

A is an (N,N) constant matrix

B is an (N,1) constant vector

u(t) is a scalar .

The superscript T denotes the transpose of the vector. From Ogata [2] the continuous time solution of (1) is

$$\underline{x}(t) = \varepsilon^{A(t-t_0)}\,\underline{x}(t_0) + \int_{t_0}^{t} \varepsilon^{A(t-\tau)}\, Bv(\tau)\, d\tau, \qquad (2)$$

where $\varepsilon^{A(t-t_0)}$ is the transition matrix $\Phi(t,t_0)$

For a discrete solution of (2), where x(t) is sampled at intervals of time T, let

$$t_0 = nT$$

$$t = (n+1)T\ .$$

Then

$$\underline{x}\ [(n+1)T] = \varepsilon^{AT}\underline{x}(nT) + \int_{nT}^{(n+1)T} \varepsilon^{A[(n+1)T-\tau]}Bv(\tau)d\tau \qquad (3)$$

The usual procedure is to assume that $v(\tau) = v(nT)$ for, $nT \leq \tau < (n+1)T$. The implementation of $v(\tau)$ in this investigation possesses characteristics which make it advisable to sample this forcing function at as high a rate as possible. Since $v(\tau)$ is assumed to be broadband noise, it is varying very rapidly with respect to time. Therefore T should be very small if $v(\tau)$ is to be fairly constant over the sampling interval. Also it must be noted that in the simulation of the identification problem $v(\tau)$ is represented by a sequence of numbers from a random number generator. The magnitude of the numbers in this sequence are independent of T in that if five numbers are produced they will have the same magnitudes whether they are produced at intervals of T or T/2. The implication then is that for any two consecutive numbers from the generator, such as $v(i)$ and $v(i+1)$, those which are produced at T/2 second intervals represent a $v(\tau)$ that has a higher rate of change with respect to time (and a higher frequency) than those that are produced at T second intervals, i.e.,

$$\left(\frac{v(i+1)-v(i)}{T/2} > \frac{v(i+1)-v(i)}{T} \right)$$

Therefore if the v's in (3) can be generated at intervals smaller than T, then the effective bandwidth of the noise can be made wider as the sampling interval of $v(\tau)$ becomes smaller. Some averaging of the rapid fluctuations of $v(\tau)$, weighted by the system parameters in the matrix A, will be made by the integral in (3). The result is that v(τ) can be sampled at a high rate (higher than $\frac{1}{T}$) and the unforced response in (3) need only be computed every T seconds. To implement this approach let,

$$\underline{x}[(n+1)T] = \varepsilon^{AT}x(nT) + \sum_{m=0}^{M-1} \int_{(\frac{m}{M}+n)T}^{\frac{(m+1)}{M}+n)T} \varepsilon^{A}[(n+1)T-\tau]Bv(\tau)d\tau \tag{4}$$

$$v(\tau) = v[(\tfrac{m}{M}+n)T], \ (\tfrac{m}{M}+n)T \leq \tau < (\tfrac{m+1}{M}+n)T \ .$$

Since B is a constant vector and v(τ) will be a constant over the interval of integration,

$$\underline{x}[(n+1)T] = \varepsilon^{AT}\underline{x}(nT) + \sum_{m=0}^{M-1} \int_{(\frac{m}{M}+n)T}^{(\frac{m+1}{M}+n)T} \varepsilon^{A}[(n+1)T-\tau]d\tau \cdot B \cdot v[(\tfrac{m}{M}+n)T]. \tag{5}$$

To remove the parameter n from the integral in (5) let

$$\tau = nT-\lambda$$

$$\lambda = nT-\tau$$

Then

$$\underline{x}[(n+1)T]=\varepsilon^{AT}\underline{x}(nT)+\varepsilon^{AT} \sum_{m=0}^{M-1} - \int_{\frac{-mT}{M}}^{-(\frac{m+1}{M})T} \varepsilon^{A\lambda}d\lambda \cdot B \cdot v[(\tfrac{m}{M}+n)T]. \tag{6}$$

To remove the negative limits on the integral in (6) let

$$\lambda = -\tau - \frac{mT}{M}$$

$$\tau = -\lambda - \frac{mT}{M}$$

Then (6) becomes

$$\underline{x}[(n+1)T] = \varepsilon^{AT}\underline{x}(nT)+\varepsilon^{AT}\sum_{m=0}^{M-1}\int_{0}^{\frac{T}{M}}\varepsilon^{-A(\tau+\frac{mT}{M})}d\tau\cdot B\cdot v[(\tfrac{m}{M}+n)T]. \quad (7)$$

Since the integral in (7) is not a function of m, then

$$\underline{x}[(n+1)T] = \varepsilon^{AT}\underline{x}(nT)+\varepsilon^{AT}\int_{0}^{\frac{T}{M}}\varepsilon^{-A\tau}d\tau\sum_{m=0}^{M-1}\varepsilon^{-A\frac{mT}{M}}Bv[(\tfrac{m}{M}+N)T]. \quad (8)$$

To evaluate the integral in (8) on a digital computer, $\varepsilon^{-A\tau}$ is expanded in an infinite series.

$$\int_{0}^{\frac{T}{M}}\varepsilon^{-A\tau}d\tau = \int_{0}^{\frac{T}{M}}\sum_{i=0}^{\infty}\frac{(-A\tau)^{i}}{i!}d\tau$$

$$= \sum_{i=0}^{\infty}\frac{(-A)^{i}}{i!}\cdot\int_{0}^{\frac{T}{M}}\tau^{i}d\tau$$

$$= \sum_{i=0}^{\infty}\frac{(-A)^{i}}{i!}\cdot\frac{\frac{T}{M}^{(i+1)}}{(i+1)}$$

$$= \frac{T}{M}\sum_{i=0}^{\infty}\frac{(\frac{-AT}{M})^{i}}{(i+1)!} \quad (9)$$

Equation (9) is the solution of $\int_{0}^{\frac{T}{M}}\varepsilon^{-A\tau}d\tau$ to be used. However, if A is known to be nonsingular, then

$$\int_{0}^{\frac{T}{M}}\varepsilon^{-A\tau}d\tau = A^{-1}[I-\varepsilon^{\frac{-AT}{M}}] \quad (10)$$

where I is the identity matrix of the same dimensions as A.

A^{-1} should exist except for those cases where the system characterized by (1) has a pole at S=0. However, to avoid compounding errors by taking an unnecessary matrix inverse on the computer, (9) is the solution of the integral to be used.

From (8) and (9) and letting $\lambda = \frac{T}{M}$,

$$\underline{x}[(n+1)T] = \varepsilon^{AT}\underline{x}(nT)+\varepsilon^{AT}\cdot\lambda\cdot\sum_{i=0}^{\infty}\cdot\frac{(-A\lambda)^{i}}{(i+1)!}\cdot\sum_{m=0}^{M-1}\varepsilon^{-Am\lambda}\cdot B\cdot v(nT+m\lambda) \tag{11}$$

OR

$$x[(n+1)T] = \Phi(T)\underline{x}(nT) + \Gamma(T,M)\underline{v}(nT,M) \tag{12}$$

where

$\underline{x}(nT)$ is the (N,1) state vector of the plant at time t=nT

$\Phi(T)$ is the (N,N) transition matrix ε^{AT}

T is the transition time between the system's states

v(nT,M) is the (M,1) control vector $\begin{bmatrix} v(nT) \\ v(nT+\lambda) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ v(nT+(M-1)\lambda) \end{bmatrix}$.

$\Gamma(T,M)$ is an (N,M) transfer matrix such that

$$\Gamma(T,M) = \Phi(T)\left[\lambda\sum_{i=0}^{\infty}\frac{(-A\lambda)^{i}}{(i+1)1}\right]\cdot\left[\begin{array}{c:c:c:c:c} & & & & \\ I & \varepsilon^{-A\lambda} & \varepsilon^{-2A\lambda} & \cdots\cdots & \varepsilon^{-(M-1)A\lambda} \cdot B \\ & & & & \end{array}\right].$$

APPENDIX C

Implementation of the System Identification Solution

A given system is represented by Figure C1, where $D(\omega) = \sum_{k=0}^{K} d_k(j\omega)^k$. It is desired that the coefficients of $D(\omega)$ be identified through measurements on $v(t)$ and $x(t)$.

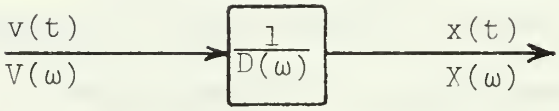


Figure C1. System Representation in the Frequency Domain

Since

$$\frac{1}{D(\omega)} = \frac{X(\omega)}{V(\omega)} , \tag{1}$$

then

$$X(\omega)D(\omega) = V(\omega) \tag{2}$$

If $X(\omega)$ and $V(\omega)$ are known for discrete values of ω where $\omega \rightarrow \omega_i$ then

$$X(\omega_i)D(\omega_i) = V(\omega_i) . \tag{3}$$

$X(\omega_i)$ and $V(\omega_i)$ result from application of the Fast Fourier Transform algorithm [Appendix A] to measurements of x(t) and v(t). Therefore $X(\omega_i)$ and $V(\omega_i)$ are assumed to be known. If $D(\omega_i)$ in (3) is replaced by $\hat{D}(\omega_i)$, the estimate of $D(\omega_i)$, then (3) becomes

$$X(\omega_i)\hat{D}(\omega_i) = V(\omega_i) \tag{4}$$

Let $r(\omega_i)$ equal the residue, or error, due to the estimate $\hat{D}(\omega_i)$.

$$r(\omega_i) = X(\omega_i)\hat{D}(\omega_i) - V(\omega_i) \tag{5}$$

Then applying a least squares estimation procedure to

$$Q = \sum_{i=0}^{I} r(\omega_i) \cdot r^*(\omega_i), \tag{6}$$

$*$ denotes the complex conjugate

Q is to be minimized.

From (5) and (6),

$$Q = \sum_{i=0}^{I} [X(\omega_i)\hat{D}(\omega_i)-V(\omega_i)] \cdot [X(\omega_i)\hat{D}(\omega_i)-V(\omega_i)]^* \tag{7}$$

Let

$$X(\omega_i) = X_i$$

$$V(\omega_i) = V_i$$

$$\hat{D}(\omega_i) = \sum_{k=0}^{K} \hat{d}_k (j\omega_i)^k = \hat{D}_i \quad .$$

Then, setting the partial of Q with respect to a coefficient $\hat{d}_m$ equal to zero,

$$\frac{\partial Q}{\partial \hat{d}_m} = \sum_{i=0}^{I} (X_i\hat{D}_i - V_i)X^*_i \frac{\partial \hat{D}^*_i}{\partial \hat{d}_m} + X_i \frac{\partial \hat{D}_i}{\partial \hat{d}_m} (X_i\hat{D}_i - V_i)^* = 0, \qquad (8)$$

where

$$\frac{\partial \hat{D}_i}{\partial \hat{d}_m} = (j\omega_i)^m \qquad (9)$$

$$\frac{\partial \hat{D}^*_i}{\partial \hat{d}_m} = \frac{\partial}{\partial \hat{d}_m} \sum_{k=0}^{K} \hat{d}_k(-j\omega_i)^k = (-1)^m(j\omega_i)^m \quad . \qquad (10)$$

From (8), (9) and (10),

$$\sum_{i=0}^{I} X_iX^*_i \; [\hat{D}_i(-1)^m(j\omega_i)^m + \hat{D}^*_i(j\omega_i)^m]$$

$$\sum_{i=0}^{I} V_iX^*_i(-1)^m(j\omega_i)^m + X_iV^*_i(j\omega_i)^m \quad . \qquad (11)$$

Since $(j)^m$ is common to both sides of (11) and is not a function of i,

$$\sum_{i=0}^{I} X_iX^*_i \; [\sum_{k=0}^{K} \; (-1)^m\hat{d}_k(j\omega_i)^k + (-1)^k\hat{d}_k(j\omega_i)^k] \; \omega_i{}^m$$

$$= \sum_{i=0}^{I} \; [(-1)^m \cdot X^*_iV_i + V^*_iX_i] \; \omega_i{}^m \; , \qquad (12)$$

where $\hat{D}_i$ has been replaced by its summation. The $\hat{d}_k$'s are not functions of i or m, and

$$
\begin{aligned}
(-1)^m+(-1)^k &= 0, \text{ if m and k are not both even or odd} \\
&= 2, \text{ if m and k are both even} \\
&= -2 \text{ if m and k are both odd}
\end{aligned}
$$

Therefore, for m = 0 to M and k = 0 to K the left side of (12) can be expressed as

$$
\left[2 \sum_{i=0}^{I} X_i \cdot X^*_i \begin{bmatrix}
1 & 0 & \text{-----} & w(0,K) \\
0 & w(1,1) & \text{-----} & w(1,K) \\
w(2,0) & 0 & \text{-----} & - \\
0 & w(3,1) & \text{-----} & - \\
- & - & \text{-----} & - \\
- & - & \text{-----} & - \\
w(M,0) & w(M,1) & \text{-----} & w(M,K)
\end{bmatrix} \right] \cdot \underline{\hat{d}} \qquad (13)
$$

where

$$
w(m,k) = j^k \, [(-1)^m+(-1)^k] \cdot \omega_i^{(m+k)}
$$

$$
\underline{\hat{d}} = \begin{bmatrix} \hat{d}_0 \\ \hat{d}_1 \\ \hat{d}_2 \\ \cdot \\ \cdot \\ \hat{d}_K \end{bmatrix} .
$$

The right hand side of (12) can be expressed as

$$2\left[\sum_{i=0}^{I}\begin{bmatrix} \mathrm{Re}\ \ (X_i{*}V_i) \\ -j\,\mathrm{Im}(X_i{*}V_i)\ \ {}_i \\ \mathrm{Re}\ \ (X_i{*}V_i)\ \ {}_i^{2} \\ \vdots \\ (-1)^M\cdot X_i{*}V_i+V_i+V_i{*}X_i\ \ \cdot\ \ {}_i^{M}\end{bmatrix}\right], \qquad (14)$$

where $\mathrm{Re}(X_i{*}V_i)$ = real part of $X_i{*}V_i$

$\mathrm{Im}(X_i{*}V_i)$ = imaginary part of $X_i{*}V_i$

$\omega_i = 2\pi f_i = \frac{2\pi i}{\tau}$

τ = the time span over which v(t) and x(t) are observed.

Now to go from the foregoing general derivation to the particular case where the order of the system is N, let M = K = N. Then from (12), (13) and (14) and after simplification, the solution for $\underline{\hat{d}}$ becomes

$$\underline{\hat{d}} = G^{-1}W^{-1}Z \qquad (15)$$

where $$\underline{d} = \begin{bmatrix} \hat{d}_0 \\ \hat{d}_1 \\ \hat{d}_2 \\ \vdots \\ \hat{d}_N \end{bmatrix}$$

$$G^{-1} = \begin{bmatrix} g_0 & & & & \\ & g_1 & & \bigcirc & \\ & & g_2 & & \\ & & & \ddots & \\ & \bigcirc & & & g_N \end{bmatrix}$$

$$g_0 = 1$$

$$g_1 = \frac{-\tau}{2\pi}$$

$$g_n = -g_{n-2}\left(\frac{\tau}{2\pi}\right)^2$$

$$w^{-1} = \left[\sum_{i=0}^{I} X_i X_i^* \begin{bmatrix} 1 & 0 & \text{-----} & w(0,N) \\ 0 & w(1,1) & \text{-----} & w(1,N) \\ w(2,0) & 0 & \text{-----} & - \\ 0 & w(3,1) & \text{-----} & - \\ w(4,0) & 0 & \text{-----} & - \\ - & - & \text{-----} & - \\ - & - & \text{-----} & - \\ w(N,0) & w(N,1) & \text{-----} & w(N,N) \end{bmatrix} \right]^{-1}$$

$w(m,k) = 0$, if $m+k$ is odd

i^{m+k}, otherwise

$m,k = 0$ to N

$$Z = \left[\sum_{i=0}^{I} \begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ \cdot \\ \cdot \\ \cdot \\ z_N \end{bmatrix} \right]$$

$$z_n = \begin{cases} \mathrm{Re}(X_i{*}V_i)i^n & \text{for n even} \\ -\mathrm{Im}(X_i{*}V_i)i^n & \text{for n odd} \end{cases}$$

$n = 0$ to N .

The implementation of the identification scheme then takes on the form of Figure C2.

For example, if the order of the system N equals 2 and v(t) and x(t) are observed through a time window of τ seconds then,

$$\begin{bmatrix} \hat{d}_0 \\ \hat{d}_1 \\ \hat{d}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -\frac{\tau}{2\pi} & 0 \\ 0 & 0 & -(\frac{\tau}{2\pi})^2 \end{bmatrix} \left[\sum_{i=0}^{I} X_i X_i{*} \begin{bmatrix} 1 & 0. & i^2 \\ 0 & i^2 & 0 \\ i^2 & 0 & i^4 \end{bmatrix} \right]^{-1} \left[\sum_{i=0}^{I} \begin{bmatrix} \mathrm{Re}(X_i{*}V_i) \\ -\mathrm{Im}(X_i{*}V_i)i \\ \mathrm{Re}(X_i{*}V_i)i^2 \end{bmatrix} \right]$$

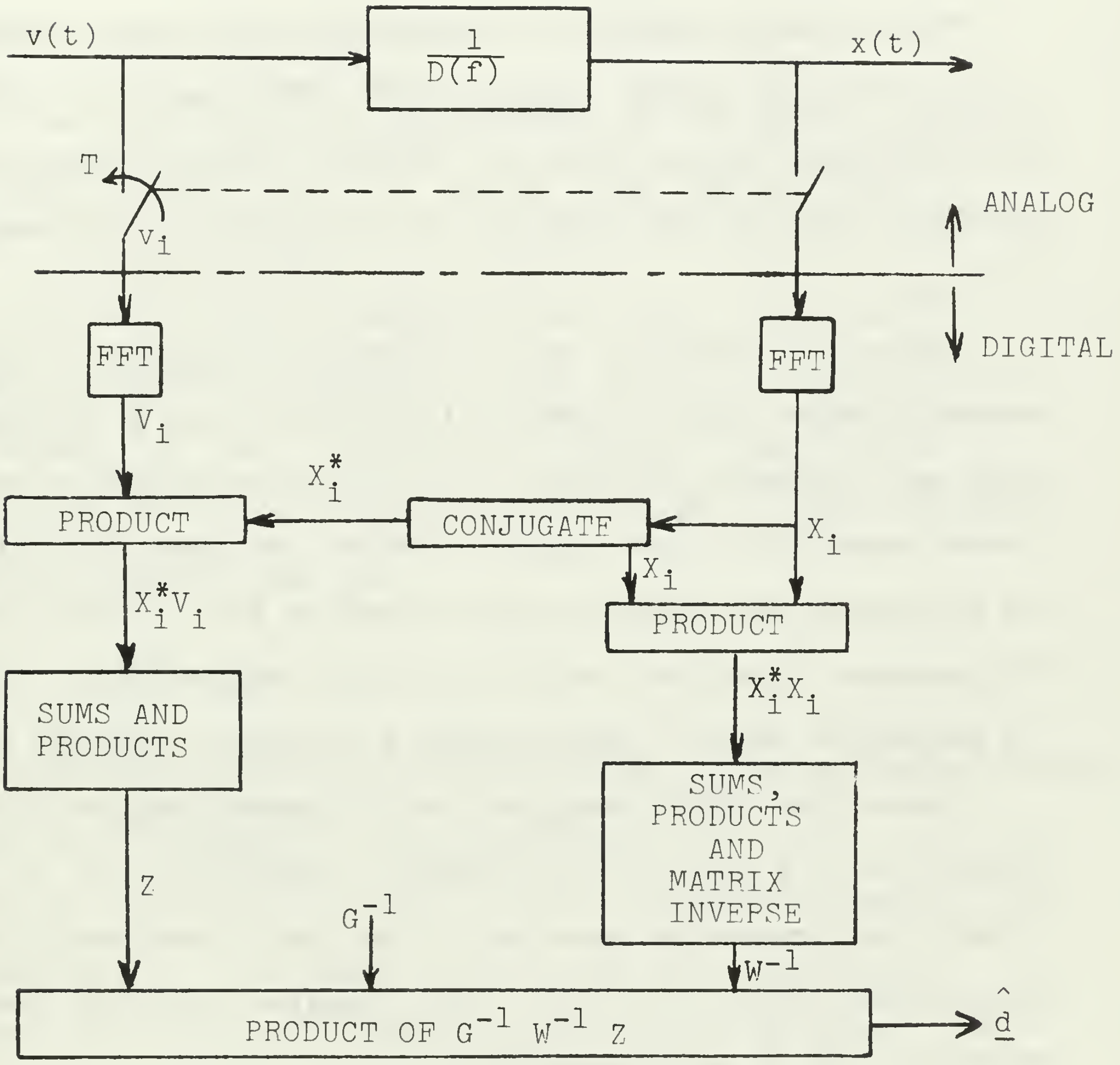


Figure C2. Implementation of the Identification Scheme

APPENDIX D

Computer Program PLEST (PLANT ESTIMATION)

The computer program is composed of the main program and subroutines GAUSS, DHARM, PLANT, RECIP and EA. GAUSS and DHARM are called from the IBM/360 Scientific Subroutine package (360A-CM-03X) and are not included in the program listing.

Subroutine GAUSS is used to obtain a sequence of real random numbers with a normal distribution and a selectable mean and standard deviation. The expected value of the power spectrum of the output of GAUSS has been investigated by utilizing the Fast Fourier transform and has been found to approach a constant value over all frequencies. GAUSS is therefore used to approximate a source of "white" noise.

Subroutine DHARM computes the Discrete Complex Fourier Transform of a sequence of numbers. DHARM is used to obtain the frequency spectrum of the input and output of the system (plant) for which the transfer function coefficients are to be determined.

Subroutine RECIP is used to obtain the inverse of a square matrix with dimensions of from (1,1) to(12,12).

Subroutine EA computes either,

$$\sum_{n=0}^{\infty} \frac{A^n}{n!} \qquad (1)$$

or

$$\sum_{n=0}^{\infty} \frac{A^n}{(n+1)!} \tag{2}$$

where A is a square matrix

When the sum of the magnitudes of all of the elements in the N^{th} term of (1) or (2) is less than or equal to 10^{-20}, the computation is stopped.

Subroutine PLANT computes the Φ and Γ matrices of equation (12) in Appendix B. The inputs to PLANT consist of either the coefficients of a transfer function for a system or the A matrix and B vector from a vector differential equation such as;

$$\dot{\underline{x}}(t) = A\underline{x}(t)+Bv(t) \tag{3}$$

The MAIN program (PLEST) performs the normal bookkeeping functions required of any comprehensive program. It also utilizes subroutine PLANT so that a continuous system such as Figure D1 is converted to the discrete system in Figure D2. The MAIN program, subroutines DHARM and RECIP perform the operations required by Appendix C to identify the system under test.

The primary program parameters are identified in the table of Computer Program Data Cards (Table D1).

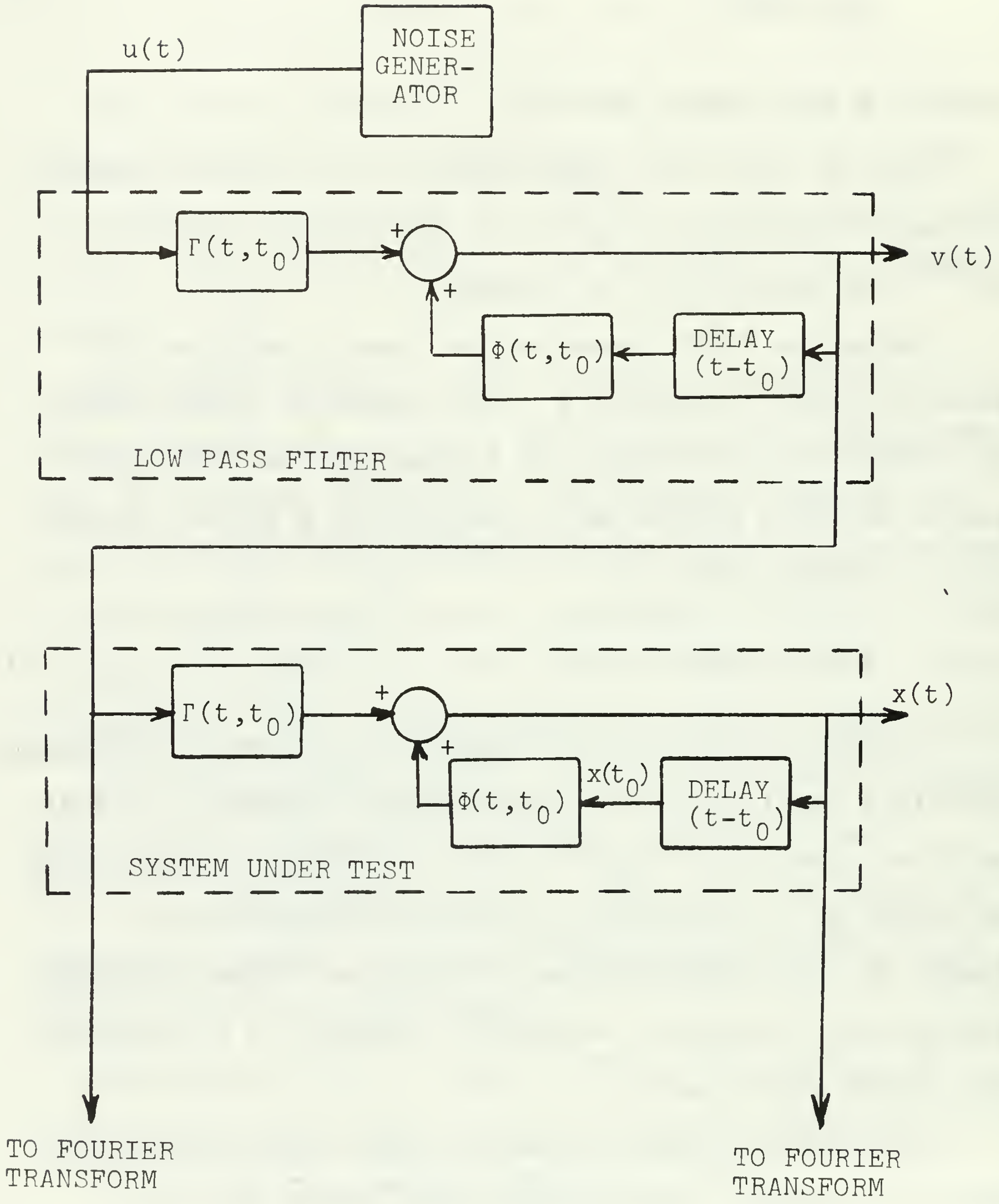
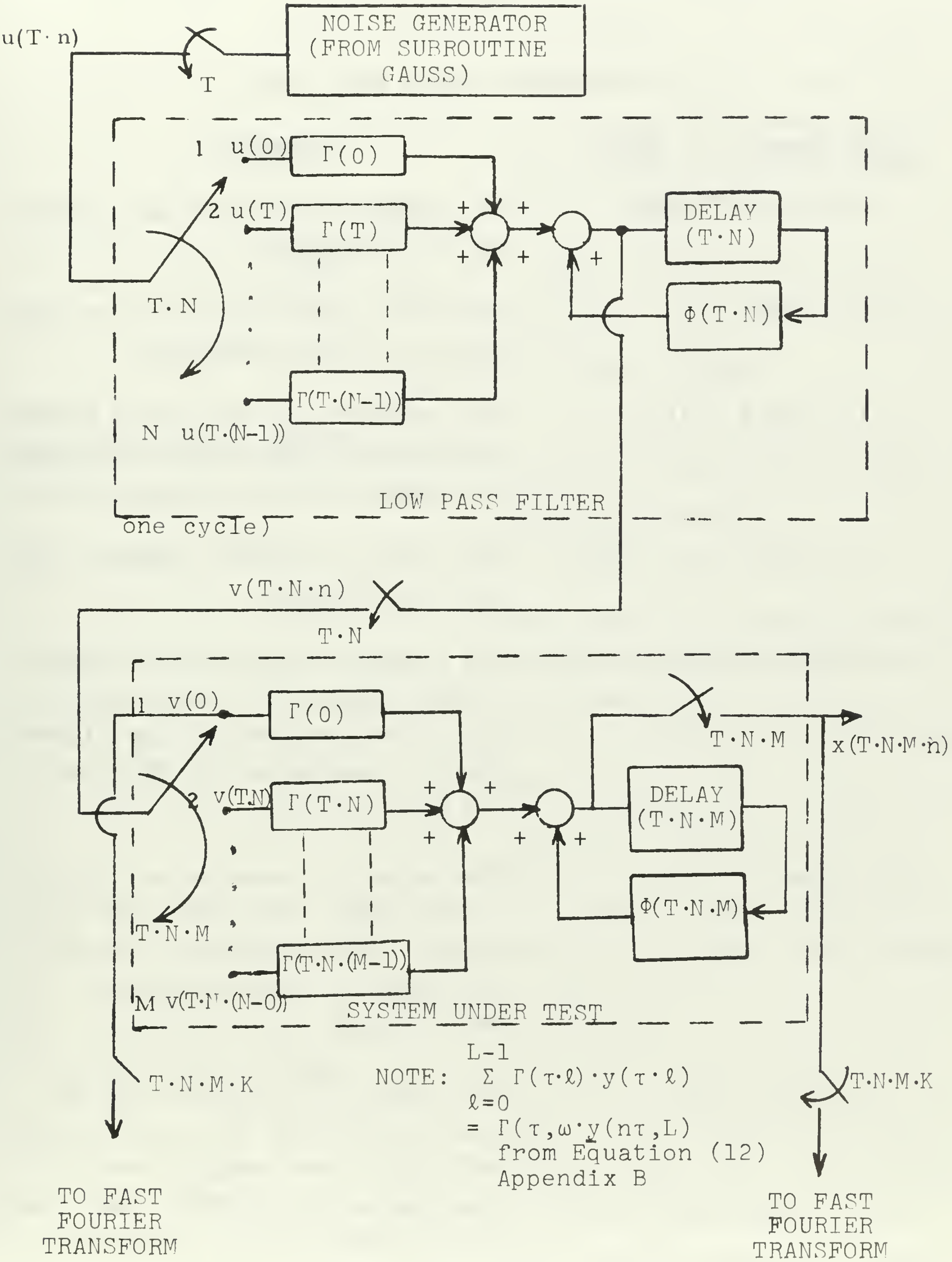


Figure D1. Continuous Version of the System Under Test



NOTE: $\sum_{\ell=0}^{L-1} \Gamma(\tau\cdot\ell)\cdot y(\tau\cdot\ell)$
$= \Gamma(\tau,\omega\cdot \underline{y}(n\tau,L)$
from Equation (12)
Appendix B

Figure D2. Discrete Version of the System under Test

TABLE D1

COMPUTER PROGRAM DATA CARDS

Card	Format	Name	Comments
1	I4	NJOBS	The number of sets of data cards to be operated on.
2	10A8		Arbitrary comments to be read and written on the print-out.
3	D10.4	VARN	The variance of the random signal generated by the subroutine gauss (the mean has been preset to 0).
4	D10.4	T	The time (T) between samples for the filter input.
5	I4	N	The number of successive inputs to the filter for one output. The iteration time of the filter (TN) is determined by TN = T.N. $0 \leq N \leq 12$
6	I4	M	The number of successive inputs to the plant under test for one output. The iteration time of the plant is determined by TMN = T·M·N. $0 \leq M \leq 12$
7	I4	K	For every K^{th} iteration of the plant under test the plant input and output are sampled once for the inputs to the FFT. The inputs to the FFT occur at intervals of TKMN = T·K·M·N. $0 \leq K \leq 12$

Card	Format	Name	Comments
8	I4	NRUNS	The number of times the plant coefficients will be determined. The resulting coefficients are the average of the results from NRUNS estimates of the coefficients.
9	I4	LTRMS	A lower-limit guess on the number of terms in the plant transfer function. $1 \leq \text{LTRMS} \leq \text{NTRMS}$
10	I4	NTRMS	An upper-limit guess on the number of terms in the plant transfer function. $1 \leq \text{NTRMS} \leq 12$
11	I4	ISTST	The number of iterations the filter-plant combination will go through to allow it to reach a steady state condition.
12	I4	M(1)	$2^{M(1)}$ samples of the plant input and output are to be used on each run (each NRUNS from card 8) to estimate the transfer function coefficients. $3 \leq M(1) \leq 10$
13	I2	IFLT	The method by which the filter will be characterized. If IFLT = 1 the transfer function coefficients are to be entered. If IFLT = 2 the bottom row of the A matrix and each element of the B vector is to be entered.

Card	Format	Name	Comments
14	I2	NDPFLT	The exponent of highest power of S in the denominator of the filter transfer function. $0 \leq \text{NDPFLT} \leq 11$
15	I2	NPFLT	The exponent of highest power of S in the numerator of the filter transfer function. $0 \leq \text{NPFLT} \leq \text{NDPFLT}$

NOTE: if IFLT = 2, go to (2) 16A

(1) 16A	D10.4	DE(1)	The coefficient of the highest power of S in the denominator.
(1) 16B	D10.4	DE(2)	
⋮	⋮	⋮	
(1) 16-	D10.4	DE(NDPFLT+1)	The coefficient of S^0 in the transfer function denominator.
(1) 17A	D10.4	UN(1)	The coefficient of the highest power of S in the transfer function numerator.
(1) 17B	D10.4	UN(2)	
⋮	⋮	⋮	
(1) 17-	D10.4	UN(NPFLT+1)	The coefficient of S^0 in the transfer function numerator.

NOTE: if IFLT = 1, go to 18

(2) 16A	D10.4	A(NDPFLT,1)	The element A(NDPFLT,1) in the filter A matrix.

Card	Format	Name	Comments
(2) 16B	D10.4	A(NDPFLT,2)	
⋮	⋮	⋮	
(2) 16-	D10.4	A(NDPFLT,NDPFLT)	The element A(NDPFLT,NDPFLT) in the filter A matrix.
(2) 17A	D10.4	B(1)	The element B(1) of the filter B vector
(2) 17B	D10.4	B(2)	
⋮	⋮	⋮	
(2) 17-	D10.4	B(NDPFLT)	The element B(NDPFLT) of the filter B vector.
18	I2	IPLT	The method by which the plant will be characterized. Same comments as for card 13.

Cards 19 through 22 (for the plant) are similar to cards 14 through 17 (for the filter).

NOTE: to omit the filter enter the following data

Card	Data
5	0001
13	01
14	00
15	00
(1) 16A	1.0000D+00
(1) 17A	1.0000D+00

Repeat Data Cards 2 through 22 NJOBS (from Card 1) times.

```
//PLEST JOB (900010,005,D),'KISLACK 2263',MSGLEVEL=1,CLASS=G
//          EXEC FORTHCLG,PARM.FORT='MAP,XREF,ID,EBCD'
//SYSUT2  DD  UNIT=SYSDA,SPACE=(TRK,(1,1))
//FORT.SYSIN DD *
      DIMENSION X(1024), RX(2048), V(1024), RV(2048), SFFT(256),
     1MFFT(3), INVFF(256), UF1(12), VF1(12), XP(12),
     1UP(12), PHIF(12,12), PHIPL(12,12), GAMF(12,12), GAMPL(12,12),
     1DH(12), COM(1), RICOM(2),
     1ALFA(12),T3(12),T4(12),DMEAN(12,12), DSTD(12,12)
      COMMON/REC/T2(12,12),T1(12,12)
C
      COMPLEX*16 X,V,COM
      REAL*8 RX,RV,SFFT,VF1,UF1,XP,PHIF,PHIPL,GAMF,GAMPL,DH,
     1RICOM,DA,ALFA,UP,T,Z,ZZ,ST,PI,DFREQ,TN,TMN,TKMN,
     1HIFRQ,T1,T2,T3,T4,DMEAN,DSTD,DTCT
      EQUIVALENCE(X(1),RX(1)),(V(1),RV(1)),(COM(1),RICOM(1))
C     SUPPRESS THE UNDERFLOW DIAGNOSTIC
      CALL ERRSET(208,0,-1,1)
C
      PI = 6.28318D+00
      MFFT(2) = 0
      MFFT(3) = 0
      READ(5,5) IJOBS
    5 FORMAT(I4)
      DO 330 NJOBS = 1, IJOBS
    6 FORMAT('1','  JOB NUMBER  ',I2)
      WRITE(6,6) NJOBS
C     INITIALIZE VARIABLES TO 0
      Z = 0.0D0
      DO 10 I = 1, 2048
      RX(I) = Z
   10 RV(I) = Z
      DO 20 I = 1, 12
      VF1(I) = Z
      UF1(I) = Z
      XP(I) = Z
      DH(I) = Z
      T3(I) = Z
      T4(I) = Z
      DO 20 J = 1, 12
      DMEAN(I,J) = Z
      DSTD(I,J) = Z
      T1(I,J) = Z
   20 T2(I,J) = Z
C     READ AND WRITE COMMENTS FOR THIS JOB
      READ(5,30)(T3(I), I = 1, 10)
   30 FORMAT(10A8)
      WRITE(6,30)(T3(I), I = 1, 10)
```

```
C
C
      READ(5,35) VARN
   35 FORMAT(D10.4)
      WRITE(6,36) VARN
   36 FORMAT(X,'VARIANCE OF NOISE SOURCE = ',PD11.4)
      STDV = SQRT(VARN)
C
      READ(5,40)T,N,M,K,NRUNS,LTRMS,NTRMS,ISTST,MFFT(1)
   40 FORMAT(D10.4, 8(/,I4))
C
C
      NUMSP = 2**MFFT(1)
      MUMSP = 2*NUMSP
      LUMSP = NUMSP/2
      TN = DBLE(FLOAT(N))*T
      TMN = DBLE(FLOAT(M))*TN
      TKMN = DBLE(FLOAT(K))*TMN
      ST = TKMN*DBLE(FLOAT(NUMSP))
      DFREQ = 1.0D0/ST
      HIFRQ = DFREQ*DBLE(FLOAT(LUMSP))
      DTOT = DBLE(FLOAT(NRUNS))
C
C
      TSTST = SNGL(TKMN)*FLOAT(ISTST)
      WRITE(6,50)ST,NUMSP,TKMN,NRUNS,TSTST,T,TN,TMN,DFREQ,HIFRQ
   50 FORMAT(X,'IN ONE ',PD11.4,' SECOND RUN OF THE PLANT AND FILTER ',
     1I4,' SAMPLES WERE TAKEN FOR',/,X,'THE INPUTS TO THE FFT',/
     1 X,'THESE SAMPLES WERE TAKEN EVERY ',PD11.4, 'SECONDS',/,
     1 X,'THE TOTAL NUMBER OF RUNS MADE = ', I4,/,
     1 X,'THE FILTER AND PLANT WERE RUN FOR ',PE11.4,' SECONDS TO REACH
     1STEADY STATE',/,
     1X,'THE INPUT TO THE FILTER WAS SAMPLED EVERY ',PD11.4,' SECONDS',
     1/,X,'THE FILTER WENT THROUGH ONE ITERATION IN ',PD11.4, ' SECONDS'
     ',/,X,'THE PLANT WENT THROUGH ONE ITERATION IN  ',PD11.4,' SECONDS'
     1,/,X,'THE FREQUENCY RESOLUTION OF THE RESULTANT SPECTRUM IS ',
     1PD11.4,' HERTZ',/,X,'THE HIGHEST FREQUENCY USED IS ',PD11.4,
     1'HERTZ')
      ISTST = ISTST + 1
C
C     COMPUTE THE PHI AND GAMMA MATRICES FOR THE FILTER AND PLANT
C
      WRITE(6,90)
   90 FORMAT(////,X,'FILTER CHARACTERISTICS')
      CALL PLANT(PHIF,GAMF,TN,NNFLT,N)
      WRITE(6,100)
  100 FORMAT(///,X,'PLANT CHARACTERISTICS')
      CALL PLANT(PHIPL,GAMPL,TMN,NNPLT,M)
      NACTL = NNPLT + 1
C
C
      IFSET = -1
      SIGN = -1.0
      JS = 0
      ALFA(1) = 1.0D0
      DO 110 IS =2, 12
      JS = JS + 1
      IF(JS .NE. 3) GOTO 110
      JS = 1
      SIGN = -SIGN
  110 ALFA(IS) = DBLE(SIGN)*((ST/PI)**(IS-1))
      IX = 65549
C
C
      DO 300 IA = 1, NRUNS
      DO 210 IAA = 1, NUMSP, 2
  120 DO 200 IAB = 1, K
      DO 190 IAC = 1, M
      DO 150 IB = 1, N
      CALL GAUSS(IX,STDV,0.0,GV)
  150 UF1(IB) = DBLE(GV)
      DO 154 IR = 1, NNFLT
      T3(IR) = Z
      DO 153 JR = 1, NNFLT
  153 T3(IR) = T3(IR) + PHIF(IR,JR)*VF1(JR)
      DO 154 LR = 1, N
  154 T3(IR) = T3(IR) + GAMF(IR,LR)*UF1(LR)
      DO 155 IR = 1, NNFLT
  155 VF1(IR) = T3(IR)
      T1(IAC,1) = VF1(1)
  190 UP(IAC) = VF1(1)
      DO 196 IR = 1, NNPLT
      T3(IR) = Z
      DO 195 JR = 1, NNPLT
  195 T3(IR) = T3(IR) + PHIPL(IR,JR)*XP(JR)
      DO 196 LR = 1, M
  196 T3(IR) = T3(IR) + GAMPL(IR,LR)*UP(LR)
      DO 197 IR = 1, NNPLT
  197 XP(IR) = T3(IR)
  200 CONTINUE
      ISTST = ISTST - 1
      IF(ISTST .GT. 0) GOTO 120
      ISTST = -2
      RV(IAA) = T1(1,1)
      RV(IAA+1) = Z
      RX(IAA) = XP(1)
  210 RX(IAA+1) = Z
      CALL DHARM(X,MFFT,INVFF,SFFT,IFSFT,IFX)
      IFSFT = -2
      CALL DHARM(V,MFFT,INVFF,SFFT,IFSFT,IFV)
C
C
      DO 220 IAA = 1, LUMSP
      COM(1) = DCONJG(X(IAA))
      RX(IAA) = RICOM(1)**2 + RICOM(2)**2
  220 V(IAA) = V(IAA)*COM(1)
C
      DO 230 IB = 1, NTRMS
      DH(IB) = Z
      T3(IB) = Z
      DO 230 JB = 1, NTRMS
  230 T1(IB,JB) = Z
      T1(1,1) = RX(1)
      T3(1) = RV(1)
      KC = 1
      ZZ = Z
      DO 245 KB = 2, LUMSP
      KC = 2*KB - 1
      ZZ = ZZ + 1.0D0
      KJ = 1
      DO 235 IB = 1, NTRMS, 2
      T3(IB) = T3(IB) + RV(KC)*(ZZ**(IB-1))
  235 T3(IB+1) = T3(IB+1) - RV(KC+1)*(ZZ**IB)
      DO 245 IB = 1, NTRMS
      DO 240 JB = KJ, NTRMS, 2
  240 T1(IB,JB) = T1(IB,JB) + RX(KB)*(ZZ**(IB+JB-2))
      KJ = KJ + 1
      IF(KJ .GE. 3) KJ = 1
  245 CONTINUE
CCCC
CCCC
      DO 300 IQQ = LTRMS, NTRMS
      CALL RECIP(IQQ, KER, 0)
      DO 260 IB = 1, IQQ
      DH(IB) = Z
      DO 250 JB = 1, IQQ
  250 DH(IB) = DH(IB) + T2(IB,JB)*T3(JB)
  260 DH(IB) = ALFA(IB)*DH(IB)
CCCC
      DO 290 JB = 1, 12
      T = DH(JB)
      DMEAN(JB,IQQ)  = DMEAN(JB,IQQ) + T/DTOT
  290 DSTD(JB,IQQ) = DSTD(JB,IQQ) + (T**2)/DTOT
  300 CONTINUE
```

```
      DO 330 L = LTRMS, NTRMS
      WRITE(6,310)
  310 FORMAT(/////)
      DO 330 I = 1, 12
      J = 13 - I
      K = NACTL - 12 + I
      DSTD(J,L) = DSQRT(DABS(DSTD(J,L) - DMEAN(J,L)**2))
      WRITE(6,320) K, DMEAN(J,L), DSTD(J,L)
  320 FORMAT(3X,'DMEAN(',I4,') =',1PD11.4,'    STD DEV = ', 1PD11.4)
  330 CONTINUE
      RETURN
      END


      SUBROUTINE PLANT(PHI, GAM, DT, NDP, KKK)
      DIMENSION PHI(12,12), GAM(12,12), A(12,12), B(12,12), UN(12),
     1DE(12), AN(12,12)
      REAL*8 PHI, GAM, A, B, UN, DE, DT,      LAMDA,ZD,TEMP,TAMP,TT, AN
      COMMON/REC/TAMP(12,12),TEMP(12,12)
C     THIS ROUTINE COMPUTES THE PHI AND GAMMA MATRICES FROM
C     A TRANSFER FUNCTION SUCH AS
C     (UN(1)*S(NP) + UN(2)*S(NP-1) +....+UN(NP+1))/
C     (DE(1)*S(NDP) + DE(2)*S(NDP-1)+....+DE(NDP+1))
C     NP IS THE HIGHEST POWER OF S IN THE NUMERATOR
C     NDP IS THE HIGHEST POWER OF S IN THE DENOMINATOR
C     S(M) = S**M
C     NDP .LT. 11, NP .LT. NDP
C     THE METHOD USED IS FROM 'STATE SPACE ANALYSIS OF CONTROL
C     SYSTEMS', BY OGATA, PAGE 188
C     WHERE, X(NT+T) = PHI(T)*X(NT) + GAM(T,TAU)*U(NT,TAU)
      DO 10 I = 1, 12
      UN(I) = 0.0D0
      DE(I) = 0.0D0
      DO 10 J = 1, 12
      PHI(I,J) = 0.0D0
      GAM(I,J) = 0.0D0
      A(I,J) = 0.0D0
      B(I,J) = 0.0D0
      AN(I,J) = 0.0D0
      TEMP(I,J) = 0.0D0
   10 TAMP(I,J) = 0.0D0
      READ(5,20) IAB
   20 FORMAT(I2)
C      READ THE DENOMINATOR POWER
      READ(5, 20) NDP
C     READ THE NUMERATOR POWER
      READ(5,20) NP
      NDT = NDP + 1
      NT   = NP + 1
      LAMDA = DT/(DBLE(FLOAT(KKK)))
C
      WRITE(6,40) DT, LAMDA, NDP, NP
   40 FORMAT(10X, 'PLANT SAMPLING INCREMENT= ',1PE11.3,/,10X,
     1'INPUT SIGNAL SAMPLING INCREMENT = ', 1PE11.3,/, 10X,
     1'DENOMINATOR POWER = ', I2,/, 10X, 'NUMERATOR POWER = ', I2)
      GOTO(45, 145),IAB
C     READ THE DENOMINATOR COEFFICIENTS
   45 READ(5,30) (DE(I), I = 1, NDT)
   30 FORMAT(D10.4)
C      READ THE NUMERATOR COEFFICIENTS
      READ(5,30) (UN(I), I = 1, NT )
      WRITE(6,50) ( I, UN(I), I = 1, NT )
   50 FORMAT(10X,'NUMERATOR TERMS',/(10X,'UN(',I2,') = ', 1PE11.3))
      WRITE (6,60) ( I, DE(I), I = 1, NDT)
   60 FORMAT(10X,'DENOMINATOR TERMS',/(10X,'DE(',I2,') = ', 1PE11.3))
      IF(NDP .GT. 0) GOTO 65
      GAM(1,1) = UN(1)/DE(1)
      NDP = 1
      GOTO 208
C     FIND THE A AND B MATRICES
C     **********
   65 ZD = DE(1)
      DO 80 I = 1, NT
   80 UN(I) = UN(I)/ZD
      DO 90 I = 1, NDT
   90 DE(I) = DE(I)/ZD
      NN = NDT - NP
      LL = 0
      DO 100 I = NN, NDT
      LL = LL + 1
  100 AN(I,1) = UN(LL)
      IF(NDP .LE. 1) GOTO 115
      NDM = NDP - 1
      DO 110 I = 1, NDM
  110 A(I, I+1) = 1.0D0
  115 DO 120 J = 1, NDP
  120 A(NDP,J) = -DE(NDT + 1 - J)
      DO 130 I = 1, NDT
      DO 130 J = 1, I
  130 TEMP(I,J) = DE(I-J+1)
      CALL RECIP(NDT, KER, 0)
      DO 135 L = 1, NDT
      TEMP(L,1) = 0.0D0
      DO 135 M = 1, NDT
  135 TEMP(L,1) = TEMP(L,1) + TAMP(L,M)*AN(M,1)
      DO 140 I = 1, NDP
  140 B(I,1) = TEMP(I+1,1)
      GOTO 149
  145 NDM = NDP - 1
      DO 146 I = 1, NDM
  146 A(I,I+1) = 1.0D0
      READ(5,30) (A(NDP,I), I = 1, NDP)
      READ(5,30) (B(I,1), I = 1, NDP)
  149 WRITE(6,150)
  150 FORMAT(5(/), 10X, 'A MATRIX')
      CALL RITE(A, NDP, NDP)
      WRITE(6,160)
  160 FORMAT(5(/), 10X, 'B MATRIX')
      CALL RITE(B,NDP, 1)
C     **********
      DO 170 I = 1, 12
      DO 170 J = 1, 12
      TEMP(I,J) = 0.0D0
  170 TAMP(I,J) = 0.0D0
      DO 180 I = 1, NDP
      B(I,1) = B(I,1)*LAMDA
      DO 180 J = 1, NDP
  180 TEMP(I,J) = A(I,J)*DT
      CALL EA(TEMP,PHI, NDP,1.0D0)
      DO 200 K = 1, KKK
      TT = LAMDA*DBLE(FLOAT(K-1))
      DO 190 I = 1, NDP
      DO 190 J = 1, NDP
  190 TEMP(I,J) = -A(I,J)*TT
      CALL EA(TEMP,TAMP,NDP,1.0D0)
      DO 195 L = 1, NDP
      TEMP(L,1) = 0.0D0
      DO 195 M = 1, NDP
  195 TEMP(L,1) = TEMP(L,1) + TAMP(L,M)*B(M,1)
      DO 200 I = 1, NDP
  200 AN(I,K) = TEMP(I,1)
      DO 205 I = 1, NDP
      DO 205 J = 1, NDP
  205 GAM(I,J) = -A(I,J)*LAMDA
      CALL EA(GAM, TEMP, NDP, 2.0D0)
      DO 206 L = 1, NDP
      DO 206 M = 1, NDP
      TAMP(L,M) = 0.0D0
      DO 206 N = 1, NDP
  206 TAMP(L,M) = TAMP(L,M) + PHI(L,N)*TEMP(N,M)
      DO 207 L = 1, NDP
      DO 207 M = 1, KKK
      GAM(L,M) = 0.0D0
      DO 207 N = 1, NDP
  207 GAM(L,M) = GAM(L,M) + TAMP(L,N)*AN(N,M)
C     PRINT THE PHI AND GAMMA MATRICES
  208 WRITE(6,210)
  210 FORMAT(5(/), 10X, 'PHI MATRIX')
      CALL RITE(PHI,NDP, NDP)
      WRITE(6,220)
  220 FORMAT(5(/), 10X, 'GAMMA MATRIX')
      CALL RITE(GAM, NDP,KKK)
      RETURN
      END


      SUBROUTINE RECIP(N, KER, NPRINT)
      DIMENSION A(12,12)
      COMMON/REC/X(12,12),B(12,12)
      DOUBLE PRECISION A, B, X, RATIO, S, ZE, Z
C     X(N,N) = RECIP(B(N,N))
C      KER =
C      0, IF A SATISFACTORY INVERSE IS FOUND
C     1, IF THE MATRIX IS SINGULAR
C     IF NPRINT .NE. 0 THEN THE ORIGINAL MATRIX(B), THE INVERSE(X)
C     AND THE PRODUCT B*X IS PRINTED
      IF(NPRINT .EQ. 0) GOTO 2
      WRITE(6,1)
    1 FORMAT(5(/), 10X, 'ORIGINAL MATRIX')
      CALL RITE(B, N,N)
    2 CONTINUE
      ZE = 0.0D0
      DO 3 I = 1, 12
      DO 3 J = 1, 12
      A(I,J) = ZE
    3 X(I,J) = ZE
      IF(N .GT. 1) GOTO 4
      IF(B(1,1) .EQ. 0.0D0) GOTO 110
      X(1,1) = 1.0D0/B(1,1)
      KER = 0
      GOTO 130
    4 DO 5 I = 1, N
      X(I,I) = 1.0D0
      DO 5 J = 1, N
    5 A(I,J) = B(I,J)
      DO 80 L = 1, N
      KP = 0
      Z = ZE
      DO 20 K = L, N
      IF(Z .GE. DABS(A(K,L))) GOTO 20
      Z = DABS(A(K,L))
      KP = K
```

```
 20 CONTINUE
    IF(L .GE. KP) GOTO 50
    DO 30 J = L,N
    Z = A(L,J)
    A(L,J) = A(KP,J)
 30 A(KP,J) = Z
    DO 40 J = 1, N
    Z = X(L,J)
    X(L,J) = X(KP,J)
 40 X(KP,J) = Z
 50 IF(DABS(A(L,L)) .LE. 1.0D-10) GOTO 110
    IF(L .GE. N) GOTO 80
    LP1 = L + 1
    DO 80 K = LP1, N
    IF(A(K,L) .EQ. ZE ) GOTO 80
    RATIO = A(K,L)/A(L,L)
    DO 60 J = LP1, N
 60 A(K,J) = A(K,J) - RATIO*A(L,J)
    DO 70 J = 1, N
 70 X(K,J) = X(K,J) - RATIO*X(L,J)
 80 CONTINUE
    DO 100 I = 1, N
    I1 = N + 1 - I
    DO 100 J = 1, N
    S = ZE
    IF(I1 .GE. N) GOTO 100
    I1P1 = I1 + 1
    DO 90 K = I1P1, N
 90 S = S + A(I1,K)*X(K,J)
100 X(I1,J) = (X(I1,J) - S)/A(I1,I1)
    KER = 0
    GOTO 130
110 KER = 1
    WRITE(6,120)
120 FORMAT(//, 10X,'MATRIX IS SINGULAR')
    RETURN
130 CONTINUE
    IF(NPRINT .EQ. 0) GOTO 180
    WRITE(6, 140)
140 FORMAT(5(/), 10X, ' INVERSE',//)
    CALL RITE(X,  N, N)
    DO 150 I = 1, N
    DO 150 J = 1, N
    A(I,J) = 0.0D0
    DO 150 K = 1, N
150 A(I,J) = A(I,J) + B(I,K)*X(K,J)
    WRITE(6, 160)
160 FORMAT(5(/), 10X, 'TEST RESULTS=ORIGINAL MATRIX X INVERSE',//)
```

```
      CALL RITE(A,  N, N)
      WRITE(6, 170) KER
  170 FORMAT(//, 10X, 'KER = ', I3,
     15X, '0 = OK, 1 = SINGULAR MATRIX')
  180 CONTINUE
      RETURN
      END


      SUBROUTINE RITE(A, IR, IC)
      DIMENSION A(12,12)
      DOUBLE PRECISION A
      REAL FM(4)/'(5X,     D10.2)   '/
      REAL FT(12)/'000100020003000400050006000700080009001000110012'/
      FM(2) = FT(IC)
      WRITE(6, FM)  ((A(I,J), J = 1, IC), I = 1, IR)
      RETURN
      END


      SUBROUTINE FA(A,B,N,CASE)
      DIMENSION A(12,12), B(12,12), C(12,12), D(12,12)
      REAL*8 A,B,C,D,AL,Z,SUM,CASE
C     TO SOLVE FOR B WHERE, B=EXP(A), FOR CASE =1.0D0
C                J=INFINITY
C     AND B = SUM((A**J)/(FACTORIAL(J+1))) FOR CASE = 2.0D0
C                J=0
C     B AND A ARE DIMENSIONED (N,N)
C     THE METHOD USED IS FROM'STATE SPACE ANALYSIS OF CONTROL SYSTEMS',
C     BY OGATA, PAGE 316
      Z = 1.0D-20
      DO 10 I = 1, N
      DO 20 J = 1, N
      C(I,J) = A(I,J)/CASE
   20 B(I,J) = C(I,J)
   10 B(I,I) = 1.0D0 + B(I,I)
      AL = CASE
   30 AL = AL + 1.0D0
      SUM = 0.0D0
      DO 45 I = 1, N
      DO 45 J = 1, N
      D(I,J) = 0.0D0
      DO 40 K = 1, N
   40 D(I,J) = D(I,J) + (A(I,K)*C(K,J))/AL
      SUM = SUM + DABS(D(I,J))
      B(I,J) = B(I,J) + D(I,J)
   45 C(I,J) = D(I,J)
      IF(SUM-Z) 50, 50, 30
```

```
60 RETURN
   END
```

BIBLIOGRAPHY

1. "Identification in Automatic Control Systems" IFAC Symposium, Prague, Czechoslovakia, June 1967.

2. K. Ogata, "State Space Analysis of Control Systems," Prentice-Hall, Inc.

3. S. Goldman, "Information Theory," Dover Publications, Inc.

4. C. Bingham, M. D. Godfrey and J. W. Tukey, "Modern Techniques of Power Spectrum Estimation," IEEE Transactions on Audio and Electroacoustics, vol. AU-15, no. 2, June 1967.

5. R. B. Blackman and J. W. Tukey, "The Measurement of Power Spectra," Dover Publications, Inc.

6. G. C. Maling, W. T. Morrey and W. W. Lang, "Digital Determination of Third-Octave and Full Octave Spectra of Acoustical Noise," IEEE Transactions on Audio and Electroacoustics, vol. AU-15, No. 2, June 1967.

7. J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," Mathematics of Computations, vol. 19, April 1965.

8. E. O. Brigham and R. E. Morrow, "The Fast Fourier Transform," IEEE Spectrum, December 1967.

9. W. T. Cochran and others, "What is the Fast Fourier Transform?", IEEE Transactions on Audio and Electroacoustics, vol. AU-15, no. 2, June 1967.

10. W. M. Gentleman and G. Sande, "Fast Fourier Transforms - For Fun and Profit," Proceedings - Fall Joint Computer Conference, 1966.

11. R. C. Singleton, "A Method for Computing the Fast Fourier Transform with Auxiliary Memory and Limited High-Speed Storage," IEEE Transactions on Audio and Electroacoustics, vol. AU-15, no. 2, June 1967.

12. T. G. Stockham, "High-Speed Convolution and Correlation," Proceedings - Spring Joint Computer Conference, 1966.

13. H. D. Helms, "Fast Fourier Transform Method of Computing Difference Equations and Simulating Filters," IEEE Transactions on Audio and Electroacoustics, vol. AU-15, no. 2, June 1967.

14. P. D. Welch, "The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short Modified Periodograms," IEEE Transactions on Audio and Electroacoustics, vol. AU-15, no.2, June 1967.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	20
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Prof. J. S. Demetry (Code 52DE) Dept. of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
4.	LT Arthur Kislack AETD, NAVAIR DEVCEN Johnsville, Warminster, Pennsylvania 18974	1

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*

Naval Postgraduate School
Monterey, California 93940

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3. REPORT TITLE

TRANSFER FUNCTION ESTIMATION USING THE FAST FOURIER TRANSFORM

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Master's Thesis; December 1969

5. AUTHOR(S) *(First name, middle initial, last name)*

Arthur Kislack, Lieutenant, United States Navy

6. REPORT DATE December 1969	7a. TOTAL NO. OF PAGES 64	7b. NO. OF REFS 14

8a. CONTRACT OR GRANT NO.

b. PROJECT NO.

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

Naval Postgraduate School
Monterey, California 93940

13. ABSTRACT

A method is proposed for the estimation of the transfer function of a linear, time-invariant system with no numerator dynamics, from random input and output data. The method employed utilizes the Fast Fourier Transform Algorithm and Least Squares Estimation to obtain the coefficients of the system's transfer function.

The procedure has been modeled in FORTRAN IV on an IBM-360 computer. The results of simulation show the feasibility of estimating the order of the transfer function and its coefficients.

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
ESTIMATION						
IDENTIFICATION						
FAST-FOURIER						
TRANSFORMS						
LEAST-SQUARES						