D. P. PASSERI
R. M. WILLETT

DECEMBER 1970

Technical Report ERI–93500

VARIABLE SAMPLING RATE SAMPLED DATA ADAPTIVE CONTROL OF FLEXIBLE AEROSPACE VEHICLES

Project Themis
Automatic Navigation and Control

CONTRACT NO. N000-14-68-A-0162
OFFICE OF NAVAL RESEARCH
DEPARTMENT OF THE NAVY
WASHINGTON, D. C. 20360

Project 712–S

120

ENGINEERING RESEARCH INSTITUTE
IOWA STATE UNIVERSITY
AMES, IOWA 50010 USA

ENGINEERING RESEARCH

ENGINEERING RESEARCH
ENGINEERING RESEARCH
ENGINEERING RESEARCH
ENGINEERING RESEARCH

TECHNICAL REPORT

VARIABLE SAMPLING RATE SAMPLED DATA ADAPTIVE CONTROL OF FLEXIBLE AEROSPACE VEHICLES

D. P. Passeri and R. M. Willett
December 1970

Submitted to:
Office of Naval Research
Department of the Navy
Themis Contract N00014-68-A-0162

ISU–ERI–AMES–93500
Project 712–S

ENGINEERING RESEARCH INSTITUTE
IOWA STATE UNIVERSITY AMES

BLANK PAGE

Unclassified

Security Classification

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)	2a. REPORT SECURITY CLASSIFICATION
Engineering Research Institute Iowa State University Ames, Iowa 50010	Unclassified
	2b. GROUP

3. REPORT TITLE

VARIABLE SAMPLING RATE SAMPLED DATA ADAPTIVE CONTROL OF FLEXIBLE AEROSPACE VEHICLES

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

Technical Report December 1970

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

D. P. Passeri and R. M. Willett

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
December 1970		9

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
N00014-68-A-0162	ISU-ERI-AMES-93500
b. PROJECT NO. ERI 712-S	
c. d.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

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

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Office of Naval Research Department of the Navy Washington, D.C. 20360

13. ABSTRACT

Adaptive sample-data control of time-varying bending modes of a flexible airframe is analyzed. Adaptive control is achieved by variation of the sampling rate in conjunction with a fixed digital compensator. Means for the identification of the bending modes are conceived and analyzed. Simulation results of the adaptive control scheme are shown to verify the adaptive control technique.

DD FORM 1 NOV 68 1473 (PAGE 1)

S/N 0101-807-6811

Unclassified

Security Classification

A-31408

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
flexible airframe bending modes adaptive sample data control time-varying sampling rate process identification						

ABSTRACT

Adaptive sample-data control of time-varying bending modes of a flexible airframe is analyzed. Adaptive control is achieved by variation of the sampling rate in conjunction with a fixed digital compensator. Means for the identification of the bending modes are conceived and analyzed. Simulation results of the adaptive control scheme are shown to verify the adaptive control technique.

CONTENTS

PART I. THE USE OF A VARIABLE SAMPLING RATE TO ACHIEVE ADAPTIVE CONTROL OF FLEXIBLE AIRFRAMES

Introduction

Need for Adaptive Control of Flexible Airframes

With the advent of high performance flexible aircraft and missiles the dynamic response of these vehicles created a need for adaptive control systems that can provide positive control of the vehicle in the presence of command inputs and atmospheric disturbances such as wind gusts or continuous turbulence. The transfer function of such vehicles is dependent upon the rigid body dynamics as well as the structural dynamic modes (body bending modes) of the vehicle; furthermore, the frequency and amplitude of the body bending modes varies with flight condition and flight time. A control system designed for a particular flight condition could give completely unsatisfactory behavior at other flight conditions even to the point of instability.

Many of the problems associated with the proper compensation of flexible airframes are solved to a large extent through the use of adaptive control techniques. Adaptive control differs from ordinary gain changing in that adjustments are made on the basis of operating system performance and not on the basis of measurements made on some external phenomena. An adaptive control system identifies the process and provides proper compensation to the system dynamics.

Several self adaptive systems are described by Bongiorno[1] and by Willett[2]. All of these differ in the process identification but all end up using a direct action on the gain when the system gets close to instability.

This paper will discuss the use of an adaptive sampled data control system for maintaining the stability of the airframe over various flight profiles. The method of achieving adaptation, described by Willett, is based on changing the sampling rate of the process compensator based on bending mode information received from the process identification scheme[2,3].

Problem Description

In order to facilitate the description of the operation of the adaptive control technique a simplified model of the longitudinal dynamics of a flexible airframe was considered. The airframe transfer function, Eq. (1), was considered to consist of second-order rigid body dynamics, ω_{nr}, and the airframe's first bending mode, ω_{n1}.

$$\frac{\alpha}{\eta}(s) = G(s) = \frac{k(s + a)}{(s^2 + 2\xi_r\omega_{nr}s + \omega_{nr}^2)(s^2 + 2\xi_1\omega_{n1}s + \omega_{n1}^2)} \qquad (1)$$

where α is the angle of attack of the airframe and η is the elevator control surface as shown in Fig. 1.

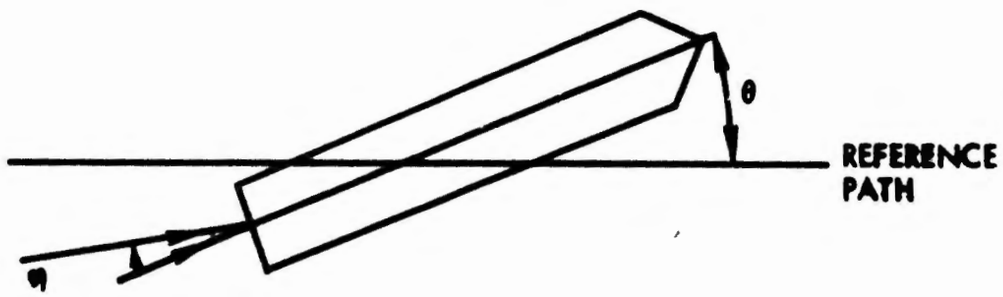


Fig. 1. Input and output of airframe transfer function.

The flight profile connected with the airframe details three possible transfer functions for various flight conditions. These transfer functions are given in Eqs. (2) through (4).

Flight Condition 1 – Sea level, heavy weight, Mach 0.4:

$$G_1(s) = \frac{K(s + 0.15)}{(s + 0.2 \pm j2.0)(s + 0.077 \pm j7.69)} . \qquad (2)$$

Flight Condition 2 – 15,000 ft, medium weight, Mach 0.4:

$$G_2(s) = \frac{K(s + 0.15)}{(s + 0.2 \pm j2.0)(s + 0.1 \pm j10.0)} . \qquad (3)$$

Flight Condition 3 – 25,000 ft, light weight, Mach 0.4:

$$G_3(s) = \frac{K(s + 0.15)}{(s + 0.2 \pm j2.0)(s + 0.13 \pm j13.0)} . \qquad (4)$$

It can be seen from the various transfer functions that the first bending mode is time varying; furthermore, it will be assumed that the first bending mode can assume any number of values between $\omega_{n1} = 7.69$ to $\omega_{n1} = 13.0$ rad/sec during the flight profile.

Consider the problem of providing proper compensation for the airframe for the various flight profiles. A block diagram of the control loop of a sampled data control system is shown in Fig. 2. The external disturbance to the airframe will be considered continuous turbulence having a power spectral density characteristic as shown in Fig. 3[4].

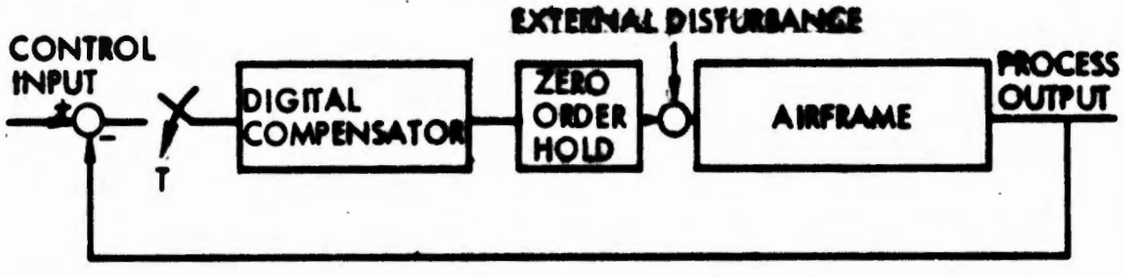


Fig. 2. Open loop spectral characteristics of airframe.

In designing a compensator for the flexible airframe it becomes necessary to obtain the z-transfer function HG(z) of the airframe

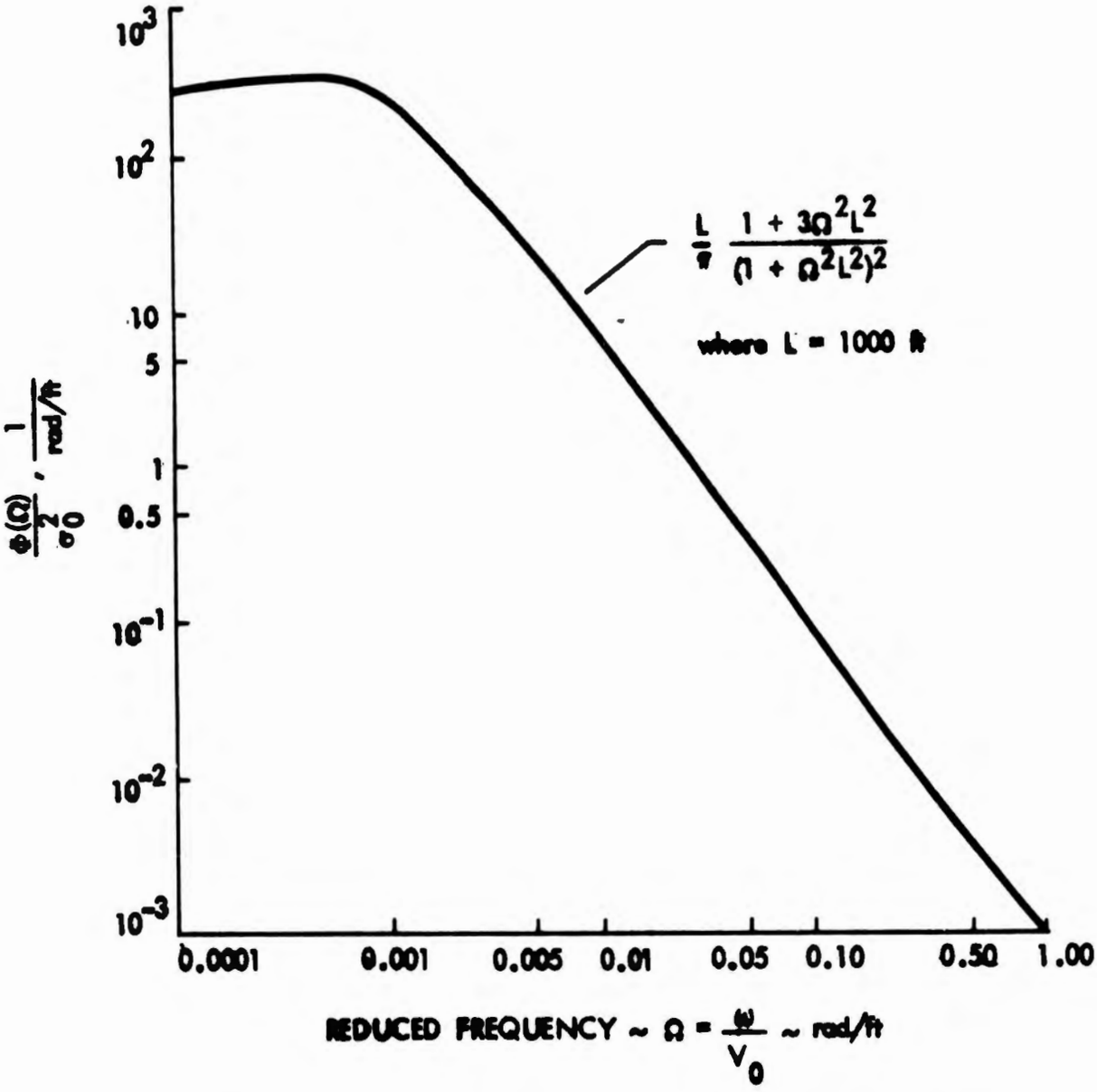


Fig. 3. Typical turbulence power spectral density characteristics.

preceded by a zero order hold. The z-transfer functions for the three flight conditions are given in Table 1 for various values of sampling time T. Note that the z-transfer functions can be made approximately identical for all three flight conditions by judicious selection of the sampling rate T.

Table 1. z-transforms for various flight conditions.

G(s)	T	HG(z)
Flight Condition 1		
$\frac{(s + 0.15)}{(s+0.2\pm j2.0)(s+0.077\pm j7.69)}$	0.0693	$\frac{(z + 0.27)(z - 0.9896)(z + 3.64)}{(z - 0.857\pm j0.506)(z - 0.977\pm j0.136)}$
	0.05199	$\frac{(z + 0.269)(z - 0.992)(z + 3.676)}{(z - 0.917\pm j0.388)(z - 0.984\pm j0.103)}$
	0.08667	$\frac{(z + 2.73)(z - 0.987)(z + 3.60)}{(z - 0.781\pm j0.614)(z - 0.968\pm j0.170)}$
Flight Condition 2		
$\frac{(s + 0.15)}{(s+0.2\pm j2.0)(s+0.1\pm j10.0)}$	0.053	$\frac{(z + 0.27)(z - 0.992)(z + 3.64)}{(z - 0.857\pm j0.506)(z - 0.984\pm j0.105)}$
	0.040	$\frac{(z + 0.27)(z - 0.994)(z + 3.68)}{(z - 0.917\pm j0.388)(z - 0.989\pm j0.079)}$
	0.667	$\frac{(z + 0.27)(z - 0.99)(z + 3.605)}{(z - 0.781\pm j0.614)(z - 0.978\pm j0.131)}$
Flight Condition 3		
$\frac{(s + 0.15)}{(s+0.2\pm j2.0)(s+0.13\pm j13.0)}$	0.041	$\frac{(z + 0.27)(z - 0.994)(z + 3.65)}{(z - 0.857\pm j0.506)(z - 0.988\pm j0.0812)}$
	0.03077	$\frac{(z + 0.269)(z - 0.995)(z + 3.68)}{(z - 0.917\pm j0.387)(z - 0.992\pm j0.0611)}$
	0.0513	$\frac{(z + 0.27)(z - 0.99)(z + 3.61)}{(z - 0.78\pm j0.614)(z - 0.985\pm j0.101)}$

Given a fixed sampling rate T = 0.0533 sec and the digital compensator,

$$D(z) = \frac{(z - 0.9)(z - 0.8859 \pm j0.4536)}{(z - 1.0)(z + 0.4)(z + 0.6)}, \qquad (5)$$

root locus plots for the compensated airframe system can be made to evaluate the stability of the system and the closed loop system time

response. Figure 4 is a set of root locus plots for the three flight conditions. For flight condition 1 the locus of the bending mode moves outside the unit circle and the system is unstable. For flight condition 2 the system is stable and has a minimum amount of body bending frequency in the output. For flight condition 3 the system is again stable; however, there is a substantial amount of lightly damped high frequency oscillation present due to the closed loop compensation poles approaching the unit circle. The time domain closed loop response of the airframe and compensation were obtained through the use of an analog computer. The analog computer simulation diagram of the airframe and the compensation is shown in Fig. 5. See Appendix A for a detailed explanation on how digital compensators can be simulated on an analog computer. Figure 6 shows the results of exciting the compensated airframe with a simulated step change in the elevator control surface.

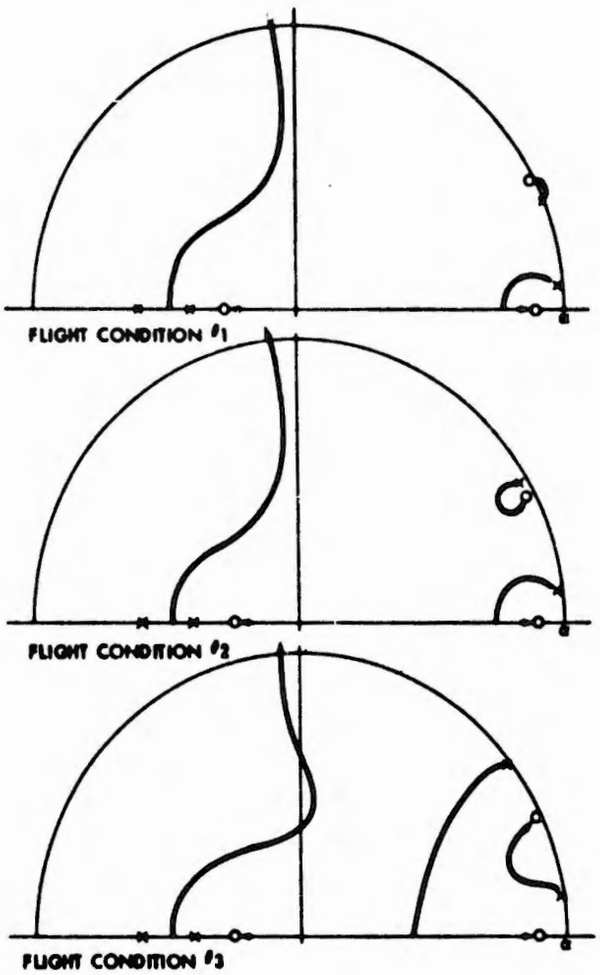


Fig. 4. Root locus plots for various flight conditions.

This is the crux of the problem. Given any fixed compensation the compensator becomes inadequate for any other flight condition different than the one it was specifically designed for; consequently, the need

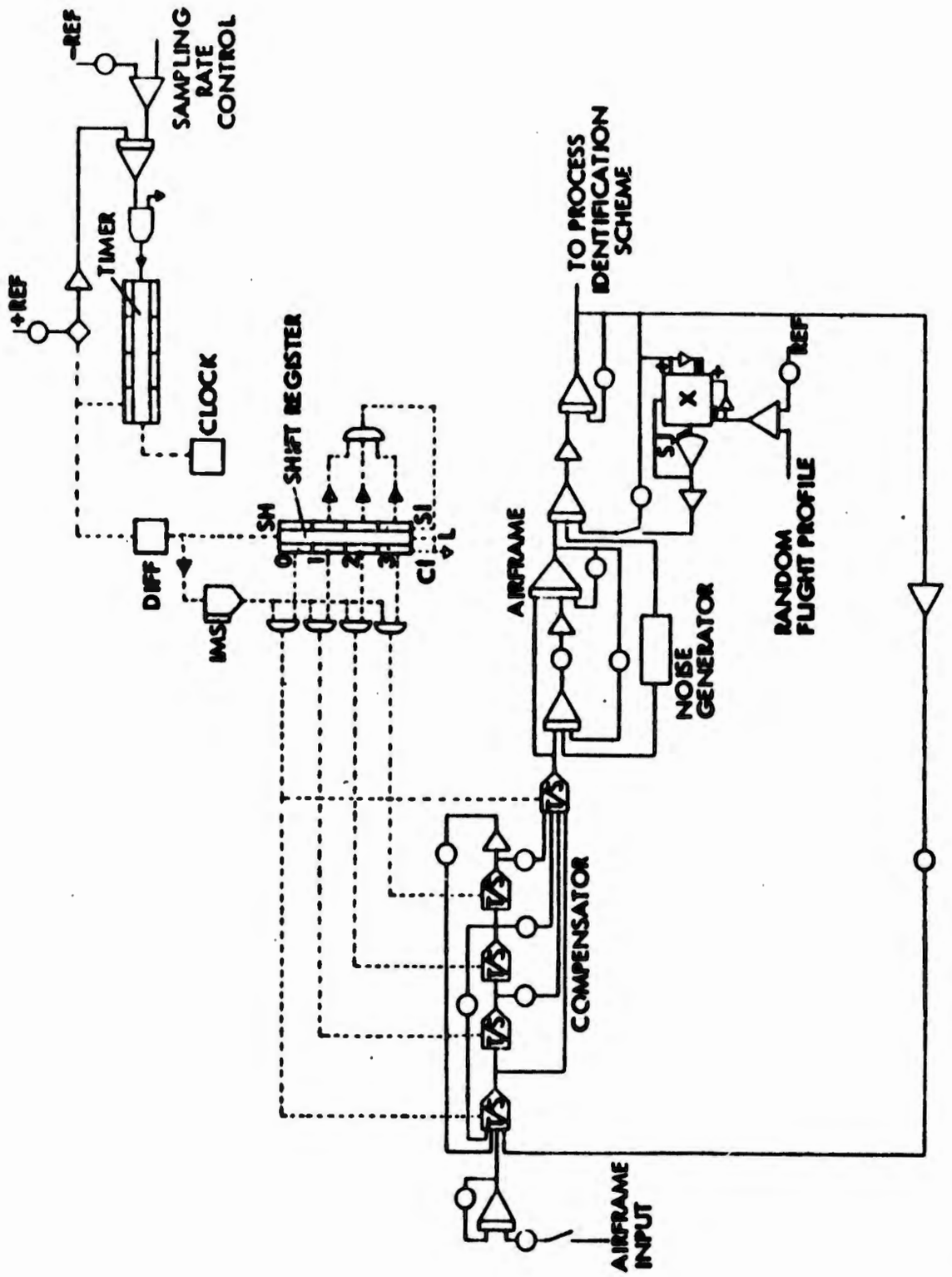


Fig. 5. Analog computer simulation diagram flexible airframe.

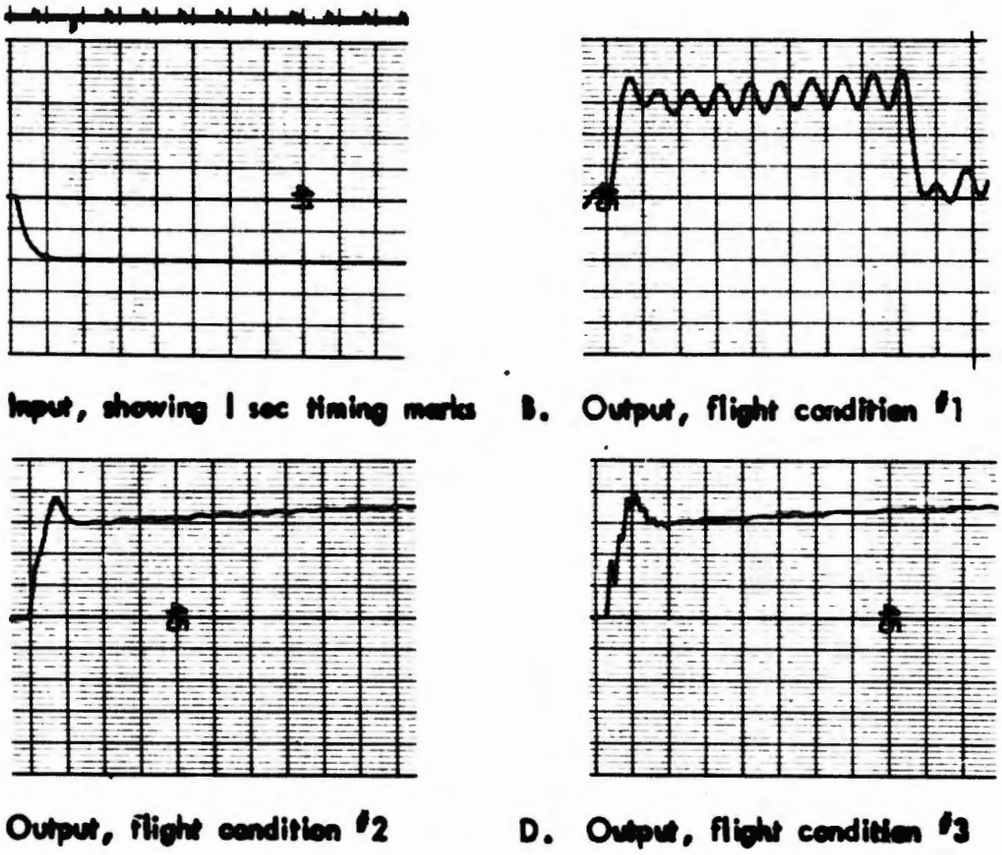


Fig. 6. Analog computer simulation results for fixed sample rate.

for an adaptive compensator. Table 1 shows that by proper selection of the sampling time T the z-transfer functions for various flight conditions can be made approximately identical. This suggests that if the sampling rate T is changed based on the pole position of the bending mode, adequate compensation of the airframe may be obtained. This is the basis of the adaptive scheme to be discussed.

Basis for an Adaptive Digital Compensator

Theoretical Background

A model for a fixed linear pulse filter constructed to both receive and deliver continuous signals is shown in Fig. 7. The continuous signal is sampled, processed by the digital filter, and

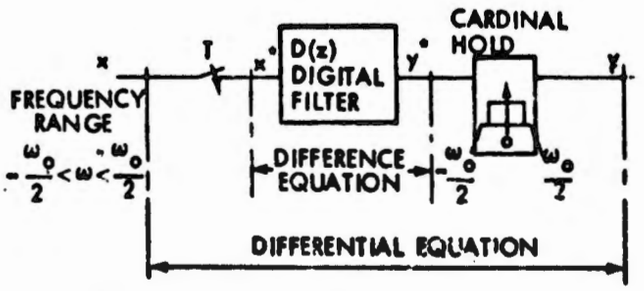


Fig. 7. Adjustable continuous filter implemented by a sampled data system.

reconstituted through an ideal hold. Provided that the sampling period is small enough to satisfy the sampling theorem, the difference equation describing the pulse filter and the differential equation describing the continuous filter can be related. For $t = nT$, where T is the sampling period and n an integer, a general difference equation describing the pulsed filter is of the form,

$$y(nT) + b_1 y[(n-1)T] + \cdots + b_m y[(n-m)T] = a_0 x(nT) + a_1 x[(n-1)T] + \cdots + a_k x[(n-k)T], \quad (6)$$

Using a shifting operator $E^{-1}x(nT) = x[(n-1)T]$ Eq. (6) can be expressed as

$$y(nT)[1 + b_1 E^{-1} + \cdots + b_m E^{-m}] = x(nT)[a_0 + a_1 E^{-1} + \cdots + a_k E^{-k}]. \quad (7)$$

The shifting operator E^{-1} can be related to the differentiation operator $Dx(t) = x'(t)$ by use of Taylor's formula, the operator equivalence being $E^{-1} = e^{-TD}$. That is,

$$E^{-1}x(t) = x(t - T)$$

$$= x(t) - T\frac{dx(t)}{dt} + \frac{T^2}{2!}\frac{d^2x(t)}{dt^2} - \cdots \frac{T^n}{n!}\frac{d^nx(t)}{dt^n} + \cdots$$

$$E^{-1}x(t) = 1 - TD + \frac{T^2D^2}{2!} - \cdots \frac{T^n}{n!}D^n + \cdots x(t)$$

$$E^{-1}x(t) = e^{-TD}x(t).$$

The operator equivalence is then

$$E^{-1} = e^{-TD}.$$

Replacing E^{-1} in Eq. (7) by a series expansion in terms of TD a differential equation can be formed that relates the continuous input and output signals. That is,

$$y(t)[1 + c_0 + c_1D^1 + \cdots + c_nD^n + \cdots] =$$

$$x(t)[a_0 + d_0 + d_1D^1 + \cdots + d_kD^k + \cdots] \qquad (8)$$

where

$$c_j = \frac{(-T)^j}{j!}\left(\sum_{i=1}^{m} (i)^j b_i\right) \text{ and}$$

$$d_j = \frac{(-T)^j}{j!}\left(\sum_{i=1}^{k} (i)^j a_i\right).$$

Retaining only the lower order derivatives results in a differential equation whose coefficients are dependent on the value of the sampling period T. For a fixed digital filter it is then possible to vary the sampling period T in order to approximate different continuous filters.

It is then theoretically possible to choose T so that the differential equation is such that the overall system (Fig. 2) is stable according to the Routh criterion for instance. The idea of changing the properties of a filter from the point of view of the continuous domain without changing it from the digital point of view is the basis of the adaptation by variation of sampling rate.

Achievement of System Stability by Variation of Sampling Rate

The simplest way to describe the achievement of system stability by variation of sampling rate is to consider the pole-zero locations of the airframe and digital compensator in the z-plane. The digital compensator's pulse transfer function remains constant (the coefficients are not varied, only the sampling rate) during system operation. A zero-order hold converts the impulsive output of the compensator into a continuous signal that drives the airframe. Now the stability of the system can be studied in the z-plane by mapping the poles of the continuous airframe transfer function in the s-plane by the transformation,

$$z = e^{sT} , \tag{9}$$

where T is the sampling rate.

Considering only the bending mode pole of HG(s) with coordinates $s = \sigma \pm j\omega$ in the s-plane, its mapping in the z-plane is

$$z = e^{(\sigma \pm j\omega)T} = e^{\sigma T} e^{\pm j\omega T} , \tag{10}$$

where σ is the real part of the bending mode poles in the s-plane, ω the imaginary part of those poles, and T the sampling period. Since the bending modes are lightly damped, σ will be a small negative number and

$e^{\sigma T}$ will approach unity; that is, the z-plane pole corresponding to the s-plane bending mode will lie near the unit circle. The angle with respect to the positive real axis will be ωT. These parameters are shown in Fig. 8. If ω varies, the angle ωT changes accordingly; but if T changes in an inverse manner, the product ωT is held constant.

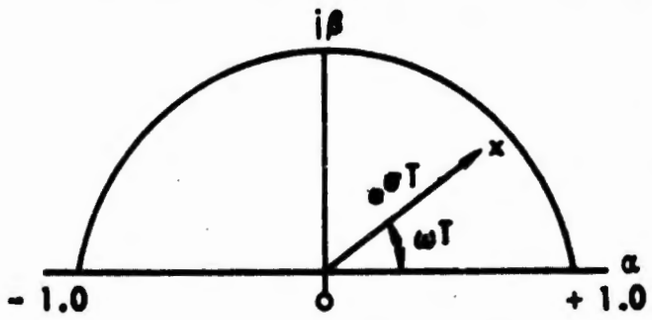


Fig. 8. Bending mode pole location in the z-plane.

The magnitude of the pole, $e^{\sigma T}$ also changes with T, but since σ is small, this change is relatively insignificant. Obviously, other bending mode poles and zeros also change their positions as do the rigid body poles and zeros. The higher frequency bending modes must therefore be given careful attention when using this method of adaptation. The rigid body poles and zeros must also be studied carefully to be certain that their motion in the z-plane due to changes in sampling rate is not troublesome. However, since ω_{nr} for the rigid body is usually somewhat smaller than the first bending mode frequency, the change in pole position due to changes in $\omega_{nr}T$ will be smaller than the corresponding change in the bending mode pole position.

To adequately damp the first bending mode then, the digital controller D(z) can place a pair of complex zeros near the complex poles. If the zeros are properly placed with respect to the poles (and other poles and zeros are properly chosen), the root locus shows that the bending poles always stay within the unit circle as shown in Fig. 9a. If the bending frequency changes so that the pole moves to the position shown in Fig. 9b, then the locus moves outside of the

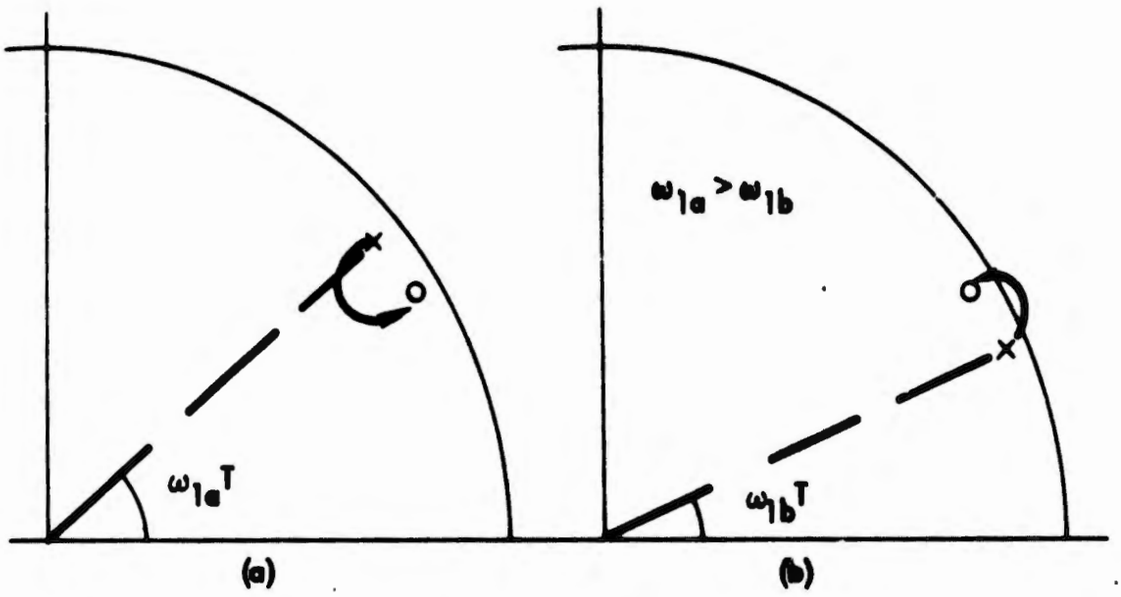


Fig. 9. Effect of bending frequency changes on stability.

unit circle, indicating the possibility of an instability. The pole and zero locations of D(z) depend only upon the coefficients in the difference equations in D(z) and do not depend on the sampling rate. Therefore their positions remain fixed in the z-plane when T changes. If ω changes so that the pole zero configuration of Fig. 9b represents the system, the sample interval T can be increased until the product ωT increases enough to again give the configuration of Fig. 9a. This is the crux of the adaptive feature of the system. Obviously it is necessary to determine ω during flight to be able to adjust T so that ωT is held at the proper value. The problem of bending mode identification will be discussed in Part II of this report.

Through proper selection of the sampling rate T the airframe can be made stable for any flight condition. The effect of changing the sampling rate on the stability of the system is shown in Fig. 10. The root locus plots of all three flight conditions have been superimposed

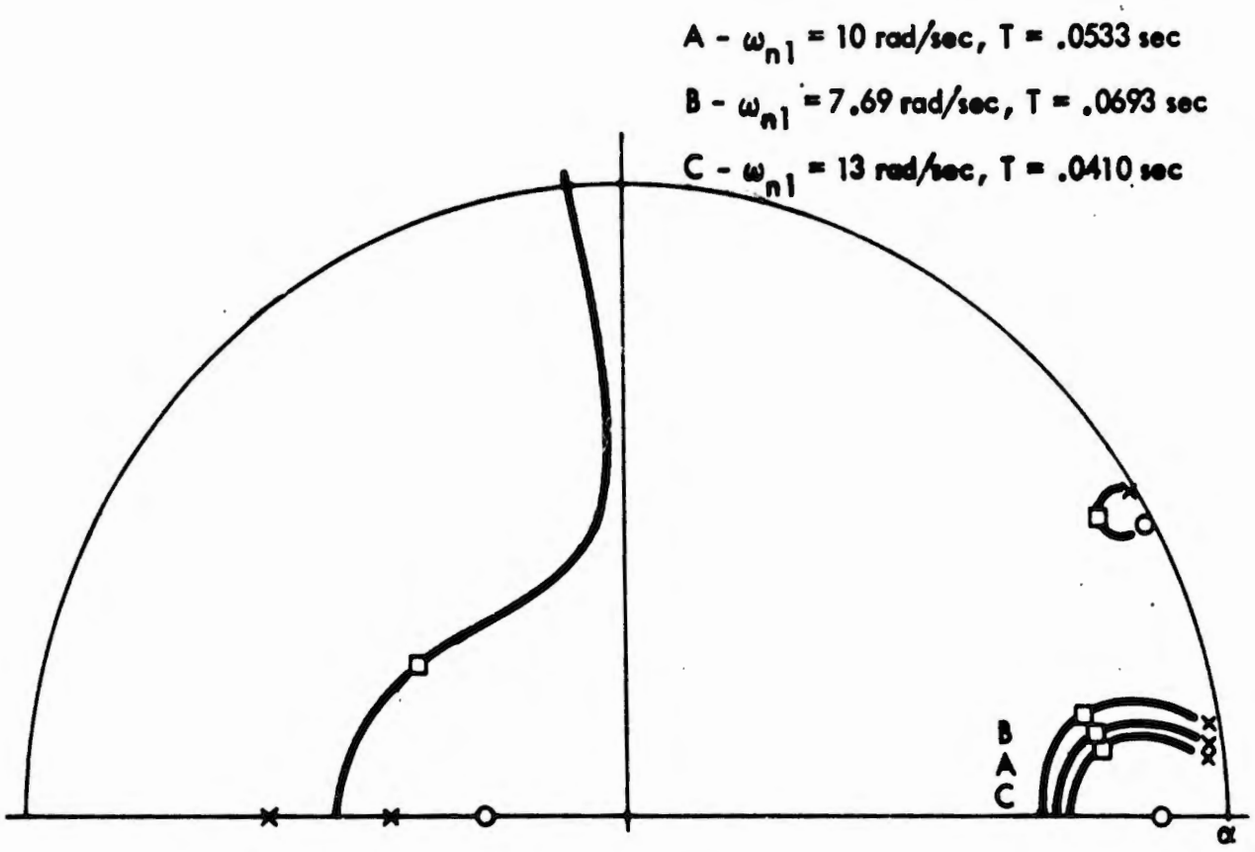


Fig. 10. Root locus plot showing adaptive effect of changes in T.

so that a comparison can be easily made. The bending mode's frequency has been changed but in this case T has also been changed so that the product T remains constant. In addition the gain has been changed in proportion to T to compensate for the effective gain change due to the sample rate change. The closed loop bending mode poles for the three flight conditions lie in almost exactly the same position and cannot be resolved on the plot but the closed loop rigid body poles have moved due to changes in the sampling time.

In each case the acceptable sampling rate is one which places the pole somewhat farther around the unit circle, in a counterclockwise sense, than the compensation zero. In effect instead of tracking the pole position with the compensation zero, changes in sampling rate force the pole to stay in the proper position with respect to the zero.

This is the unique feature of this adaptive scheme. The roots of D(z) do not change in the z-plane when the sampling rate is changed, but the pole characteristics of the airframe dynamics do change and in the case of the body bending modes can be placed in a favorable position.

A digital computer simulation (Appendix B) of the airframe with the adaptive compensator was made, and the results of the simulation are shown in Fig. 11. The digital computer simulation results show essentially the same form of output response for a step change in the elevator control surface. One can conclude from the simulation that the control system now gives an adequate response for all flight conditions.

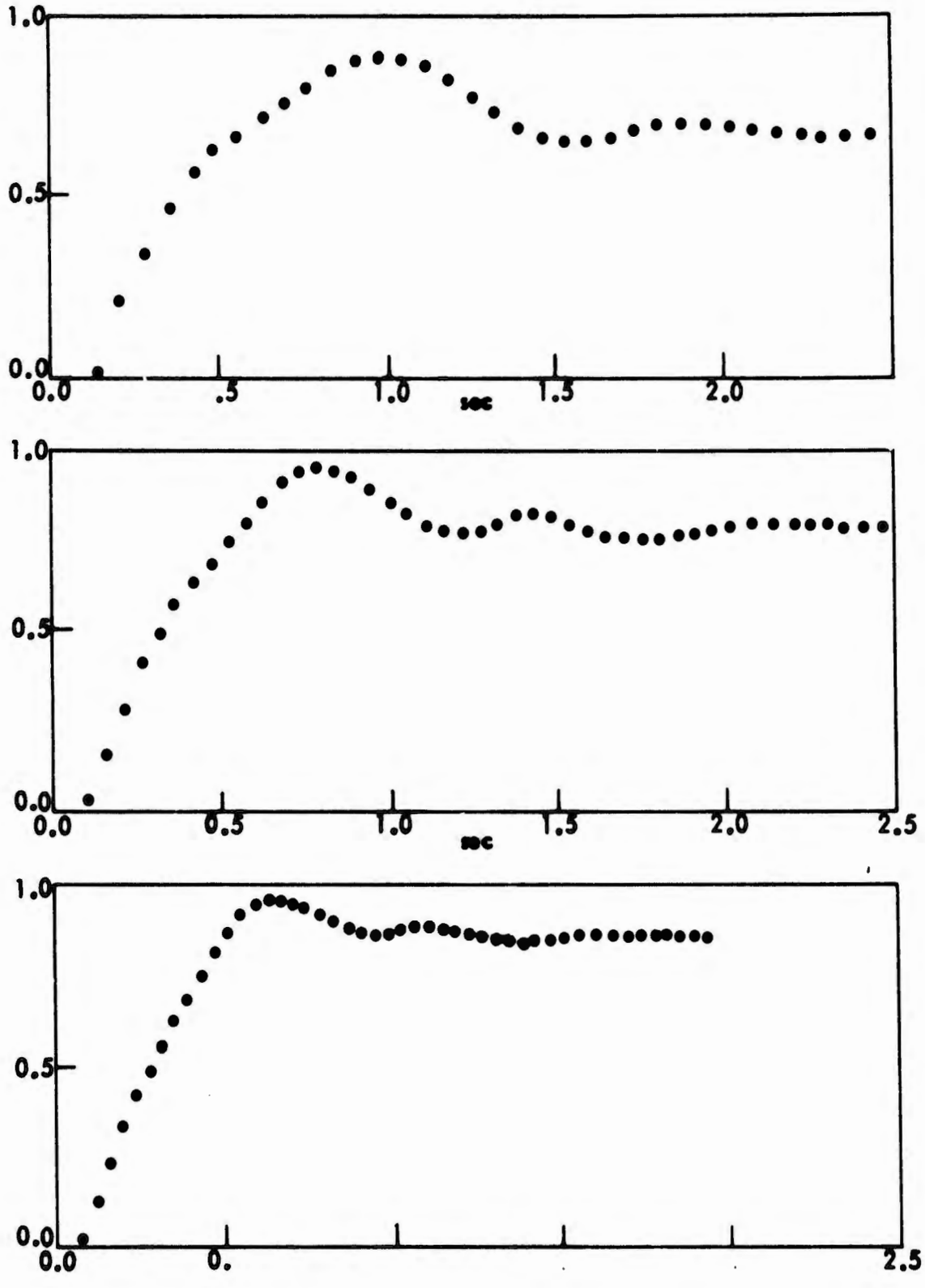


Fig. 11. Digital simulation results of adaptive systems.

Conclusions

The adaptive control technique investigated gives excellent results provided the bending mode pole positions can be observed. That is, changes in the sampling rate will move the pole into the correct position if the bending mode's position is known *a priori* or can be identified by a process identification scheme.

The dynamics of the airframe have been simplified in this study by ignoring higher order modes and the dynamic characteristics of the gyros. These effects should be included in future studies.

The unique feature of this adaptive system is that the roots of D(z), the compensation, do not change in the z-plane, but the pole characteristics of the airframe dynamics do change and in the case of the body bending modes can be placed in a favorable position. The advantage of changing the sampling rate is that the frequency characteristics of the compensator are shifted rather than merely changing the amplitude characteristics. Thus the method is particularly well suited to the control of systems whose frequency characteristics either vary with time or are not well known initially. Examples of such systems are flexible launch vehicles, the SST, and some of the satellites which have been launched.

PART II. IDENTIFICATION OF FLEXIBLE AIRFRAME BENDING MODES

Introduction

In order to properly adjust the value of T, the sampling interval, there must be some means of determining the airframe bending poles in the z-plane or s-plane. The identification of the pole positions is usually referred to as process identification. Figure 12 shows a block diagram of the process and process identifier.

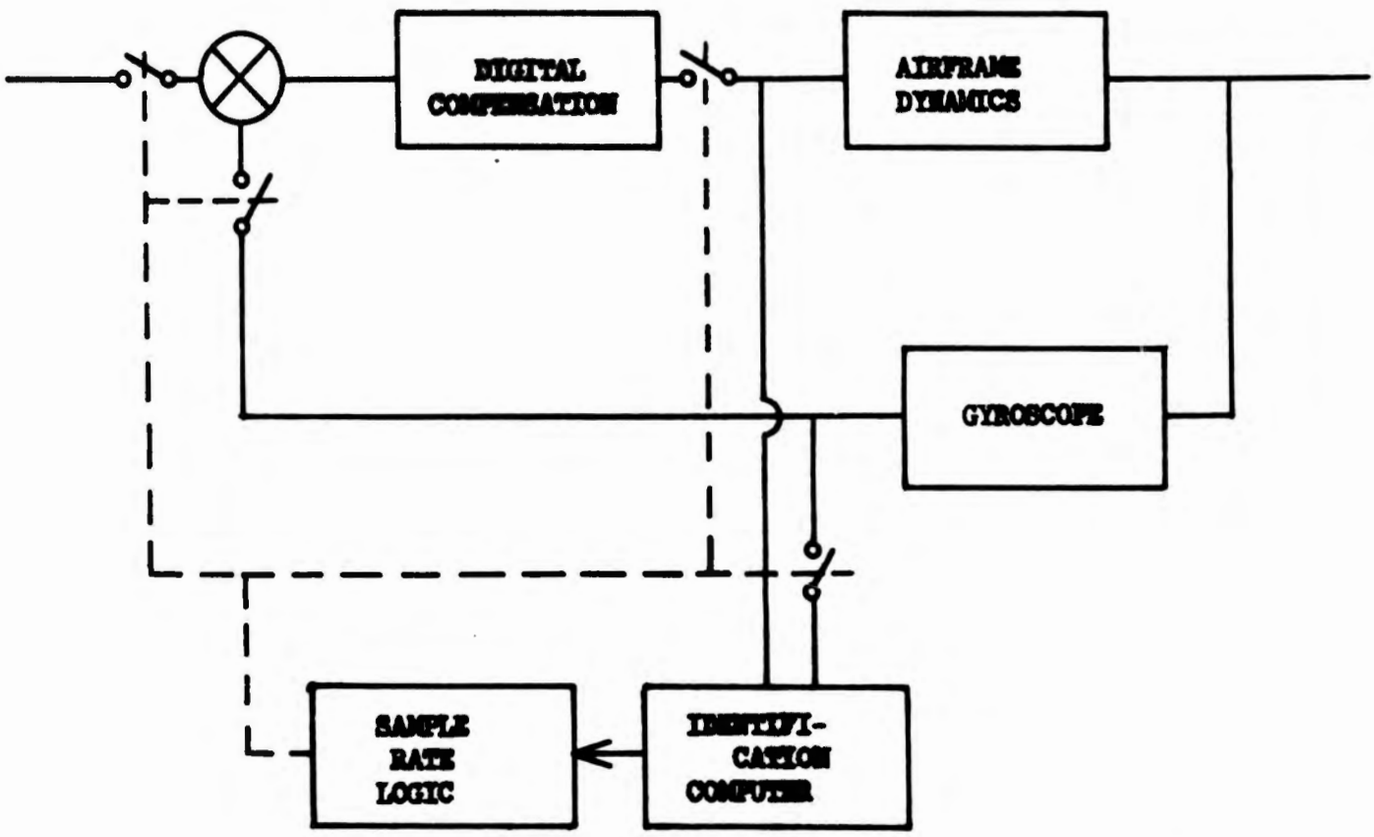


Fig. 12. Block diagram of digital adaptive controller with variable sampling rate.

The investigated techniques used to determine the bending modes of the flexible airframe were: (a) estimation of the bending mode pole position by means of an adaptive digital discriminator; and (b) estimation of the spectral density function of the process output. The

spectral density of the process output was determined by one of three methods: (a) direct Fourier analysis of the process output; (b) utilization of the Fast Fourier Transform; and (c) use of the spectral filter method.

Estimation of the Bending Modes through Use of an Adaptive Digital Discriminator

Tunable Digital Filters through the Use of Variable Sampling Rates

Basis for a Tunable Digital Filter

A digital filter or pulse filter can be described as a functional operator that acts on a set of discrete numbers (input) producing a second set of discrete numbers (output). A fixed linear pulse filter whose input, r(t), and output, c(t), are defined only at discrete instants in time t = nT can be described by the difference equation,

$$c(nT) = \sum_{j=0}^{n} a_{j+1} r[(n - j)T] - \sum_{j=1}^{n} b_{j+1} c[(n - j)T] \qquad (11)$$

where the a_j's and the b_j's are constants.

The difference equation provides a recursive relationship for computing the output of the filter at discrete time intervals in terms of past input samples and previous values of the output. A simulation diagram for an nth order difference equation is shown in Fig. 13.

The system dynamics for the nth order system can be represented as a set of first order difference equations through the use of state variables[5]. The vector-matrix difference equations can be written as,

$$X(n + 1)T = AX(nT) + B\underline{r}(nT), \qquad (12)$$

$$c(nT) = CX(nT) + D\underline{r}(nT) \qquad (13)$$

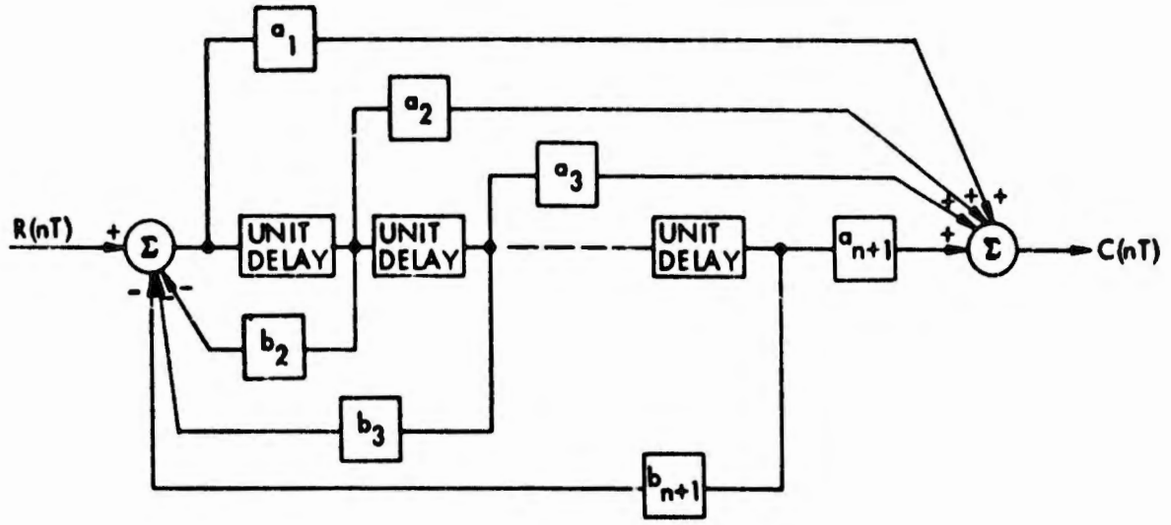


Fig. 13. Simulation diagram nth order difference equation.

$$c(nT) = CX(nT) + D\underline{r}(nT). \tag{14}$$

If the matrices, A, B, C, D, are nontime-varying, the time domain solution relating the discrete input variable and the discrete output variable is given by,

$$c(nT) = C\Phi[(n - n_o)T]X(n_oT) + \sum_{m=n_o}^{n-1} C\Phi[(n - m - 1)T]B\underline{r}(mT) + D\underline{r}(nT), \tag{15}$$

where Φ is defined as the state transition matrix and n_o represents the initial iteration. If the characteristic matrix A is constant, the state transition matrix can be written as,

$$\Phi[(n, n_o)T] = A^{(n-n_o)}, \tag{16}$$

and the solution of the difference equation is dependent only upon time differences. That is, since the characteristic matrix is constant the number of iterations, $n - n_o$, provides a measure of the number of samples it takes to go from some initial state in state space to another state. In real time, $t = nT$; therefore the physical time rate of transition in state space is a function of the sampling time T.

This is the basis for the adaptive digital frequency discriminator herein described.

As an example consider a first-order (two-pole) Butterworth bandpass filter whose difference equation can be written as,

$$c(nT) - 2\alpha c(n-1)T + \gamma^2 c(n-2)T = r(n-1)T - r(n-2)T \quad (17)$$

The signal flow graph of the difference equation representing the digital filter is shown in Fig. 14. The state equations are

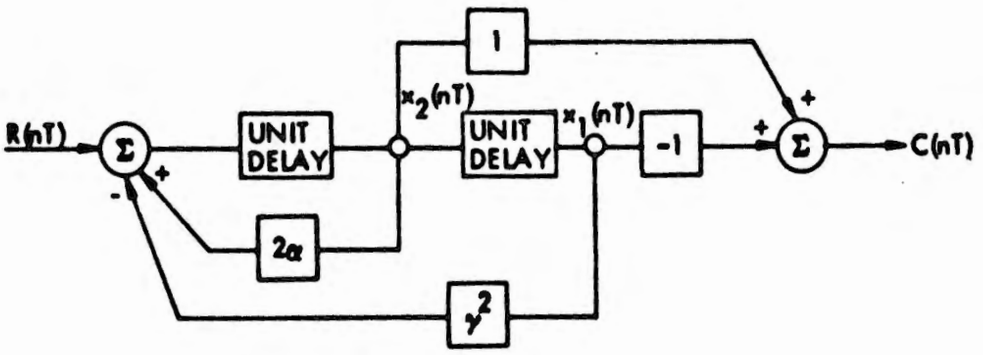


Fig. 14. First-order Butterworth bandpass filter.

$$\begin{bmatrix} x_1[(n+1)T] \\ x_2[(n+1)T] \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \gamma^2 & 2\alpha \end{bmatrix} \begin{bmatrix} x_1(nT) \\ x_2(nT) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(nT), \quad (18)$$

$$c(nT) = \begin{bmatrix} -1 & 1 \end{bmatrix} \begin{matrix} x_1(nT) \\ x_2(nT) \end{matrix} . \quad (19)$$

Since distinct eigenvalues exist, a similarity transformation on the matrix equations can be performed and Eqs. (18) and (19) can be put into normal form:

$$\begin{bmatrix} q_1[(n+1)T] \\ q_2[(n+1)T] \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} q_1(nT) \\ q_2(nT) \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} r(nT). \quad (20)$$

$$c(nT) = \begin{bmatrix} c_1 & c_2 \end{bmatrix} \begin{bmatrix} q_1(nT) \\ q_2(nT) \end{bmatrix} . \quad (21)$$

The state transition matrix is given by,

$$\Phi[(n, n_o)T] = \Lambda^{(n-n_o)} = \begin{bmatrix} \lambda_1^{(n-n_o)} & 0 \\ 0 & \lambda_2^{(n-n_o)} \end{bmatrix} . \tag{22}$$

It is clear that the rate of motion of the state vector Q(nT) is strictly a function of the number of iterations of the difference equations:

$$Q(nT) = \Lambda^{(n-n_o)} Q(n_oT) + \sum_{k=n_o}^{n-1} \Lambda^{(n-1-k)} Br(nT), \tag{23}$$

$$c(nT) = CQ(nT). \tag{24}$$

This can be illustrated by considering the solution to difference Eq. (17) for a unit step input. The solution is obtained through the use of a computer program (Appendix B) and the solution is plotted in Fig. 15 for various values of sampling time. The real time interval is t = nT so the actual response time is a function of the sampling time. Figure 15 shows that by changing the sampling time the frequency response of the digital filter is changed.

Frequency Response of a Tunable Digital Filter

The solution of the difference equations characterizing the digital filter in the frequency domain can be obtained through the use of the z-transform. Just as the Laplace transform transforms a differential equation into an algebraic equation, the z-transform maps a difference equation into an algebraic equation.

The standard one-sided z-transform is defined through the concept of an ideal sampler, where the input is f(t) and the output is f*(t); that is,

$$f^*(t) = \sum_{n=0}^{\infty} f(nT)\,\delta(t - nT), \tag{25}$$

where T is the sampling interval and $\delta(t - nT)$ is the Dirac delta function. At t = nT, the sampler output is $f(nT)\,\delta(t - nT)$. Taking the Laplace transform of Eq. (25),

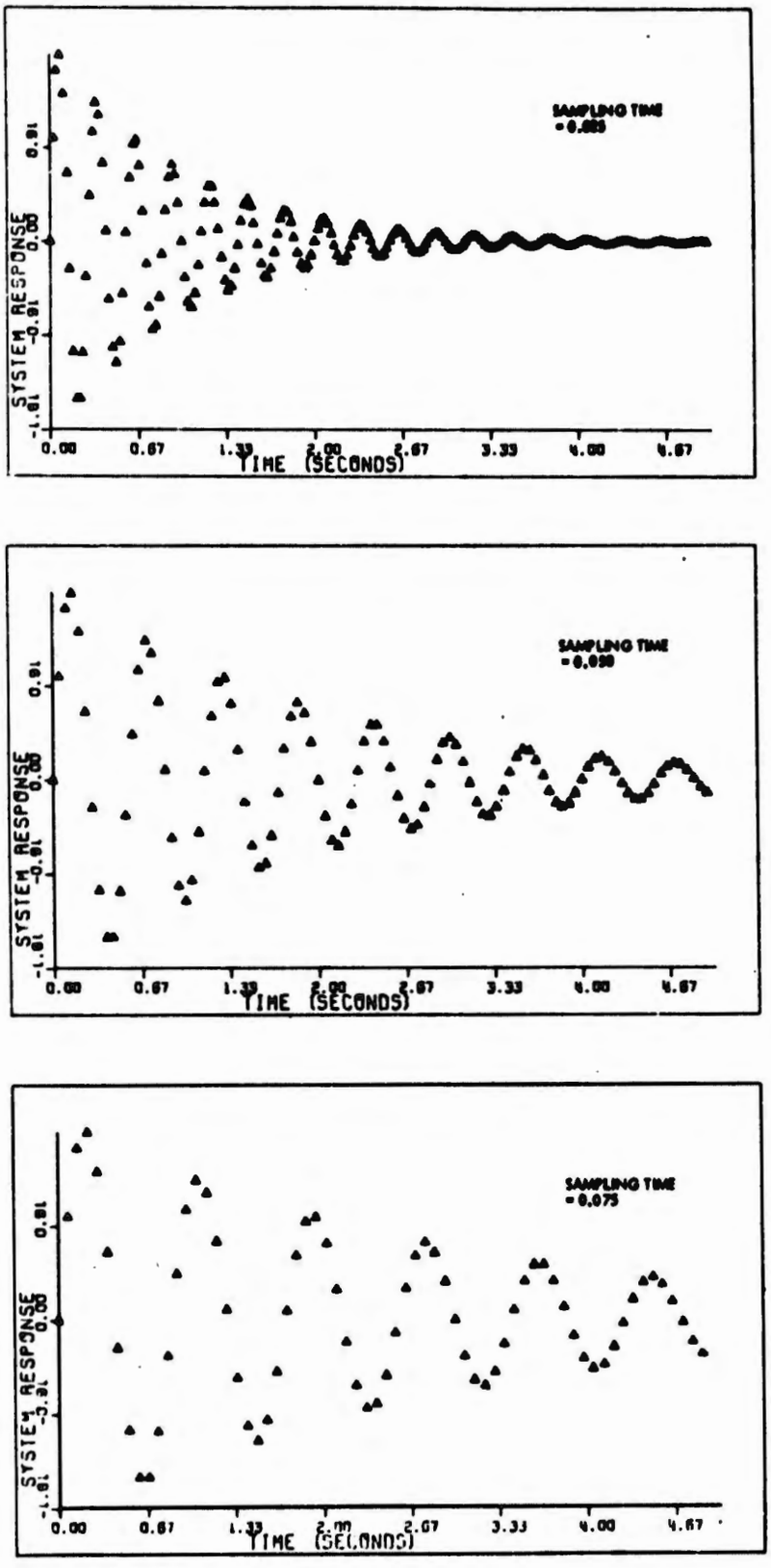


Fig. 15. Unit step response of bandpass filter.

$$\mathcal{L}[f^*(t)] = \sum_{n=0}^{\infty} f(nT)e^{-snT} = F^*(s), \tag{26}$$

and making the substitution $z = e^{-sT}$, Eq. (26) can be transformed into an algebraic equation:

$$F(z) = F^*(s) = \sum_{n=0}^{\infty} f(nT)z^{-n} = Z[f(t)]. \tag{27}$$

Applying the z-transform to the general form of the pulse filter difference equation, the difference equation can be written in the z-domain as,

$$C(z) = R(z) \sum_{i=0}^{n} a_{i+1}z^{-i} - C(z) \sum_{i=1}^{n} b_{i+1}z^{-i} . \tag{28}$$

The z-transfer function D(z) of the digital filter can be expressed as the ratio of two polynomials,

$$\frac{C(z)}{R(z)} = D(z) = \frac{\sum_{i=0}^{n} a_{i+1}z^{-i}}{1 + \sum_{i=1}^{n} b_{i+1}z^{-i}} . \tag{29}$$

It is always possible to manipulate D(z) into this closed form if a physically realizable filter is being modeled. The standard z-transform gives excellent results in modeling continuous filters into sampled data filters when applied to all pole low-pass and bandpass filters such as Butterworth, Bessel and Chebyshev I designs, provided the filters are preceded with zero-order holds. The frequency response of the pulse transfer function can be evaluated by making the substitution,

$$z^n = e^{j\omega nT} . \tag{30}$$

The relationship,

$$D(e^{j\omega T}) = \frac{\sum_{i=0}^{n} a_{i+1}e^{-j(i\omega T)}}{1 + \sum_{i=1}^{n} b_{i+1}e^{-j(i\omega T)}} \tag{31}$$

provides the frequency characteristics of the pulsed filter.

The tunable features of the digital filter (through changing the sampling rate) can be shown in the frequency domain. The z-transfer function of a first-order Butterworth bandpass filter preceded by a zero-order hold is

$$G(z) = \frac{(z - 1)e^{-aT}\sin(\omega T)}{z^2 - 2ze^{-aT}\cos(\omega T) - e^{-2aT}}, \tag{32}$$

where the s-plane transfer function is

$$G(s) = \frac{s}{(s - a)^2 - \omega^2}. \tag{33}$$

The frequency response for the digital filter is shown in Fig. 16. Changing the sampling rate essentially changes the center frequency of the bandpass filter; therefore, the digital filter can be made to adapt or track a time varying input frequency.

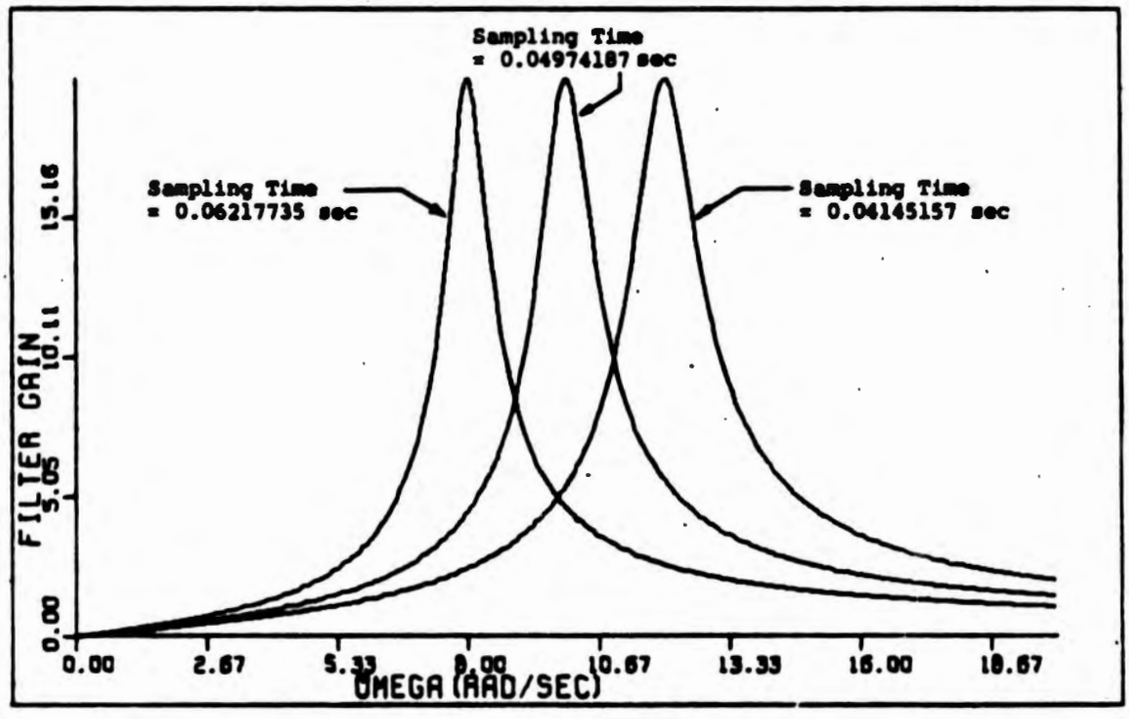


Fig. 16. Frequency response of first-order Butterworth bandpass filter.

An Adaptive Digital Frequency Discriminator

Description of the Digital Frequency Discriminator

Utilizing the fact that by changing the sampling rate of a digital filter the center frequency of the filter will change a corresponding proportion, an adaptive digital frequency discriminator can can be implemented as shown in Fig. 17. The adaptive digital frequency

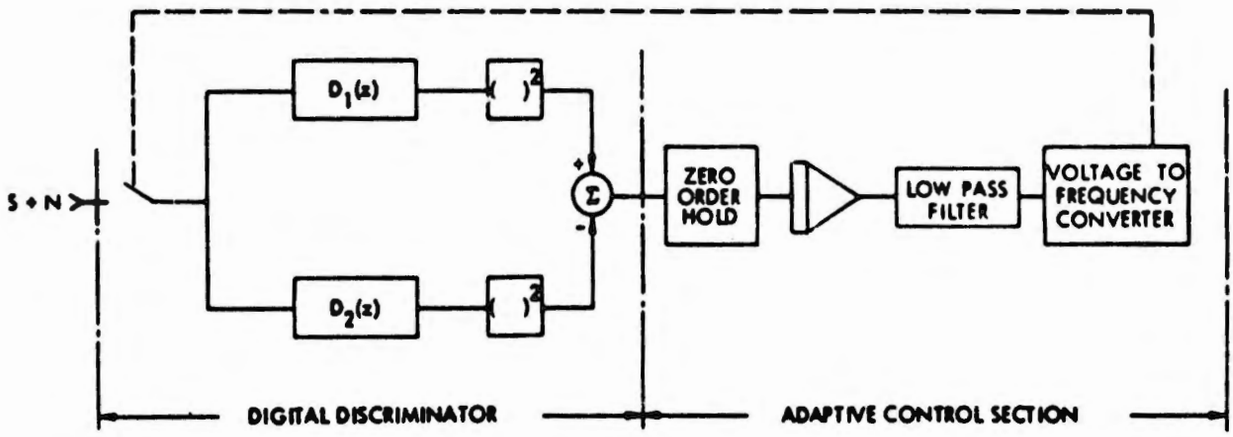


Fig. 17. Adaptive digital frequency discriminator.

discriminator consists of two portions: the digital discriminator and the adaptive control section. The digital discriminator consists of two Butterworth bandpass digital filters whose outputs are squared and differenced. The adaptive control section consists of an integrator, low-pass filter, and a voltage-to-frequency converter that provides a variable sampling rate to the digital discriminator.

In operation the discriminator action is achieved by passing the signal plus noise through the two bandpass digital filters whose center frequencies are slightly offset and forming the difference (error) between the squared outputs of the filters. The resultant error signal is integrated and passed through a low-pass filter changing the sampling rate via the voltage-to-frequency converter until the

error (difference in output between the two filters) is driven to zero. In this manner the discriminator can track the input signal and the output of the integrator can be used as an estimate of the input frequency.

The coefficients for the difference equations describing the pulsed filters used in the discriminator are derived by applying the z-transform to the s-transform of a Butterworth bandpass filter preceded by a zero order hold. The coefficients are chosen so that the half power points of each filter coincide as shown in Fig. 18.

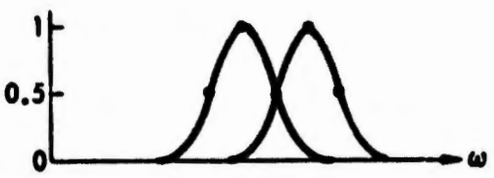
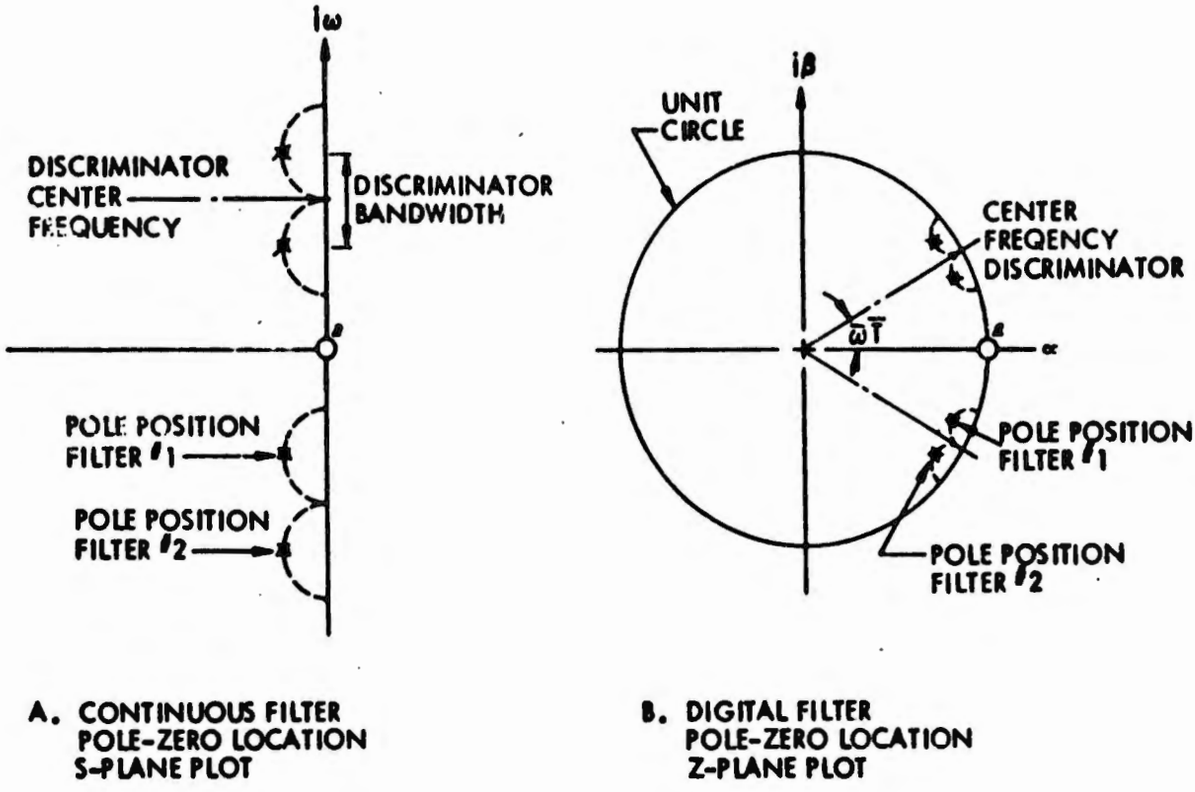


Fig. 18. Stagger-tuned filters.

Since the pole-zero locations of the pulse filter in the z-plane are dependent on the sampling time T, a priori knowledge of the frequency distribution of the input signal to be tracked is assumed. For a bandlimited signal with mean input frequency $\bar{\omega}$ the mean sampling interval $\bar{T}$ must be chosen so that the sampling theorem is not violated for the highest frequency component present in the frequency distribution. This will insure that the discriminator will have sufficiently large capture aperture when initially beginning to track the input signal. Figure 19 shows typical pole-zero locations for two continuous first-order Butterworth bandpass filters and the corresponding pole-zero locations of the digital filters. If the product $\bar{\omega}\bar{T} \ll 1$, the half circles describing the bandwidth of the continuous filters are very nearly mapped into half circles in the z-plane[6].



A. CONTINUOUS FILTER POLE-ZERO LOCATION S-PLANE PLOT

B. DIGITAL FILTER POLE-ZERO LOCATION Z-PLANE PLOT

Fig. 19. Pole-zero locations of continuous and digital filters.

In steady-state operation of the discriminator the product ωT remains constant; that is, as the input frequency increases the sampling interval decreases as shown in Fig. 20. The frequency response characteristics for a digital frequency discriminator made up of first-order Butterworth band-pass filters is shown in Fig. 21. Again as the sampling interval T is changed the center frequency of the discriminator translates on the frequency axis.

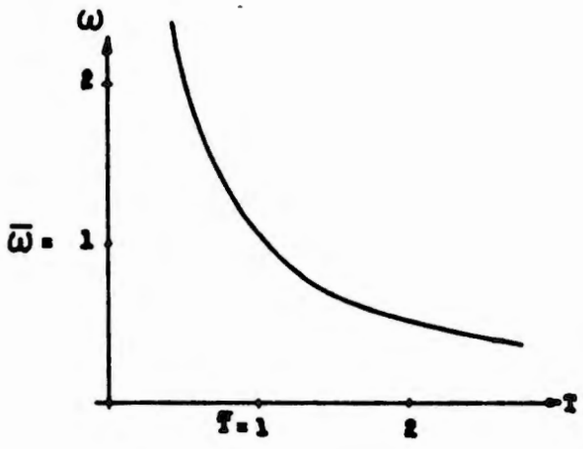


Fig. 20. Normalized relationship between input frequency and sampling interval.

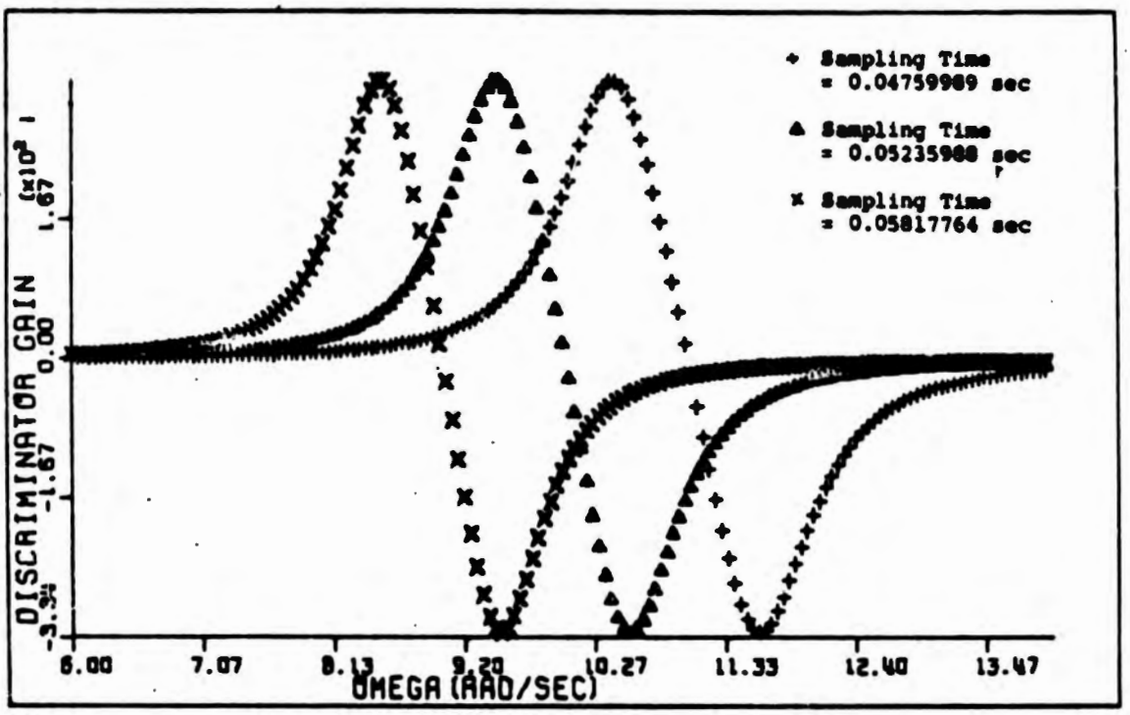


Fig. 21. Frequency response digital frequency discriminator.

Tracking a Sinusoidal Input-Analog Computer Simulation

The performance of the digital discriminator in tracking a sinusoidal input was evaluated through an analog computer simulation of the digital frequency discriminator. Figure 22 is an analog computer simulation diagram of the adaptive digital frequency discriminator. The digital filters $D_1(z)$ and $D_2(z)$, are simulated through the use of track/store amplifiers. The analog simulation diagram and the associated logic timing diagram are shown in Fig. 23 for a first-order digital Butterworth bandpass filter where the z-transfer function for the filter is

$$D(z) = \frac{z - 1}{z^2 - 2\alpha z + \gamma^2} \quad . \tag{34}$$

In simulating a pulse filter using an analog computer the accuracy in setting the feedback coefficients sets a lower limit on the bandwidth

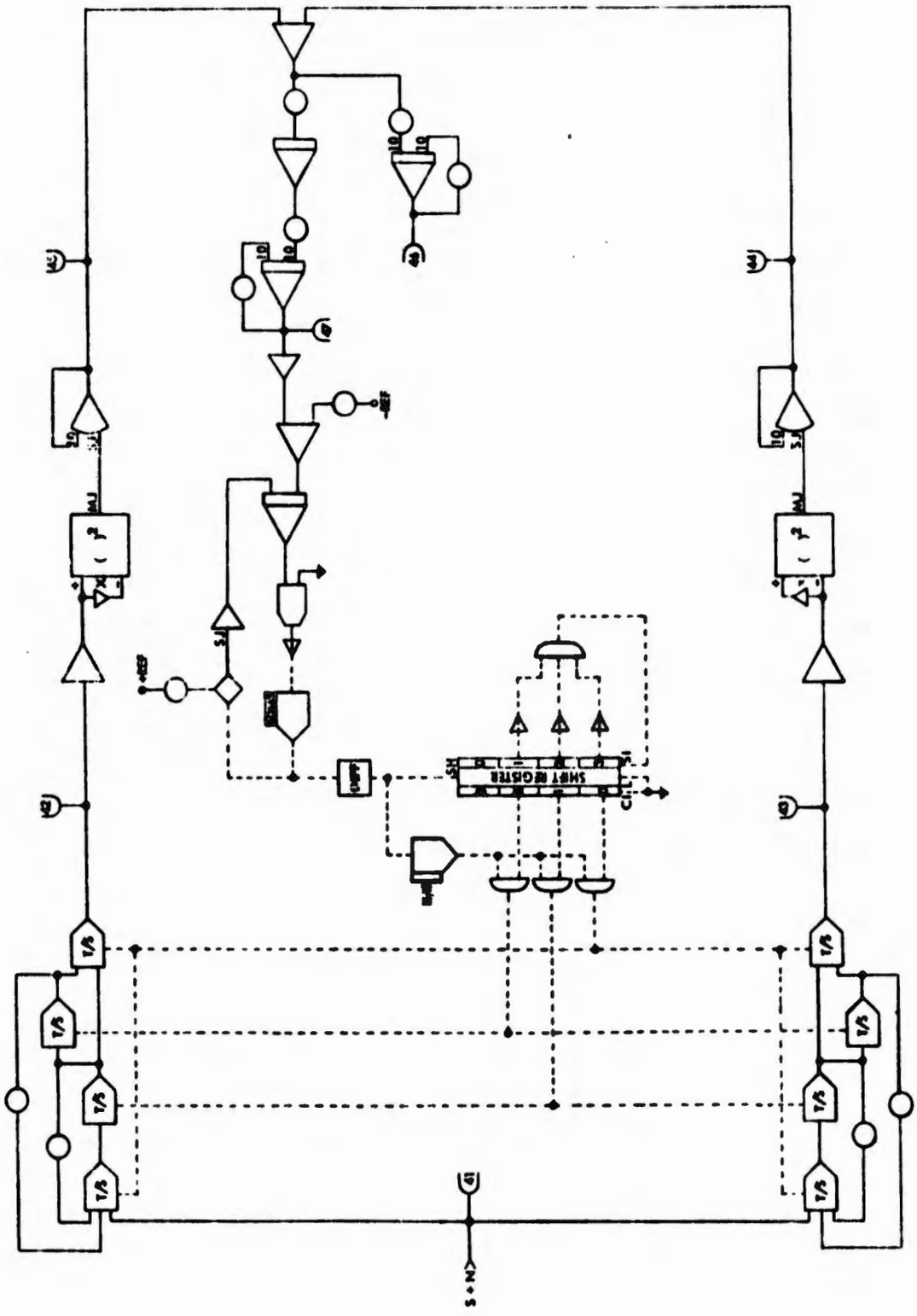


Fig. 22. Analog computer simulation diagram adaptive digital frequency discriminator.

of the pulse filter being simulated. Consequently, narrow band filters are best simulated through the use of a digital computer.

The digital discriminator was made up of two pulse filters having a 0.1592 Hz bandwidth with z-transfer functions:

$$D_1(z) = \frac{z - 1}{z^2 - 1.661214z + 0.9489872}, \tag{35}$$

$$D_2(z) = \frac{z - 1}{z^2 - 1.712216z + 0.9489872}. \tag{36}$$

The performance of the adaptive frequency discriminator in tracking a sinusoidal input having a given signal to noise ratio, S/N, is shown in Figs. 24 through 28. The input, a frequency-modulated sine wave, has a mean frequency of 10 rad/sec with an imposed frequency drift of 0.1 rad/sec^2. The mean sampling interval T is set at 0.05235988 sec so there will be about 19 samples/sec. The noise is bandlimited Gaussian noise with a noise bandwidth of 5 Hz. These simulation results show that the digital discriminator does indeed track the input frequency and provides a good estimate of the input frequency even for noise corrupted signals with S/N ratios as small as - 1 db.

Use of the Adaptive Frequency Discriminator in Tracking a Bending Mode of a Flexible Aircraft

Description of the Problem

In Part I it was shown that by varying the sampling rate the body bending poles of a flexible airframe are constrained to stay in a favorable position with respect to the compensation zeros. This section will investigate the use of a variable sampling rate in conjugation with the digital frequency discriminator as a process

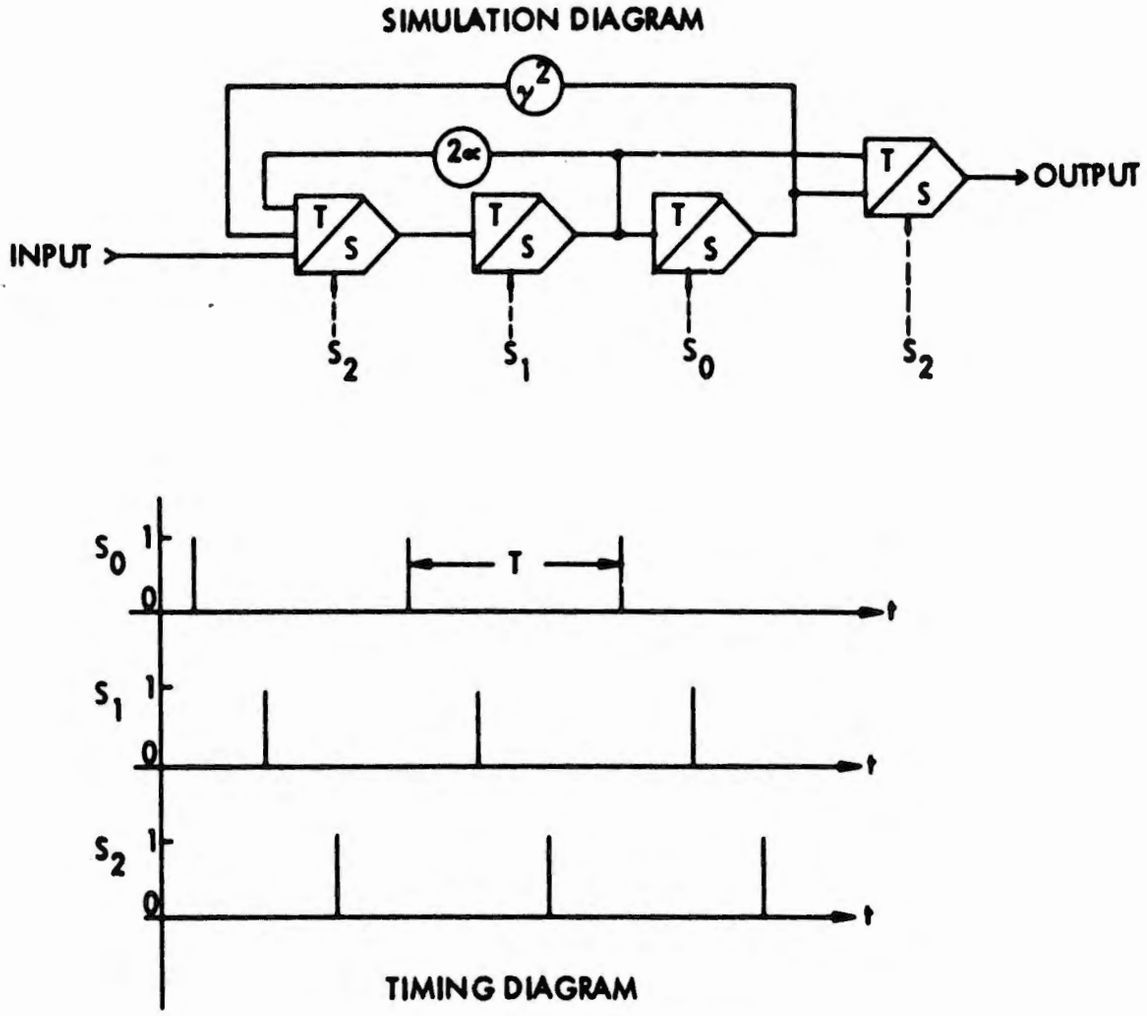


Fig. 23. Analog computer simulation diagram first-order Butterworth bandpass digital filter.

identification technique for such an adaptive control system. That is, the digital frequency discriminator will determine the bending mode frequency so that the sampling rate can be properly selected.

From the longitudinal equation of motion for a flexible aircraft a simplified transfer function relating the angle of attack of the airframe, α, and the elevator angle, η, is given in Eq. (37),

$$\frac{\alpha(s)}{\eta(s)} = \frac{s + 0.15}{(s^2 + 0.7s + 4.04)(s^2 + 0.2s + 100\omega)} . \qquad (37)$$

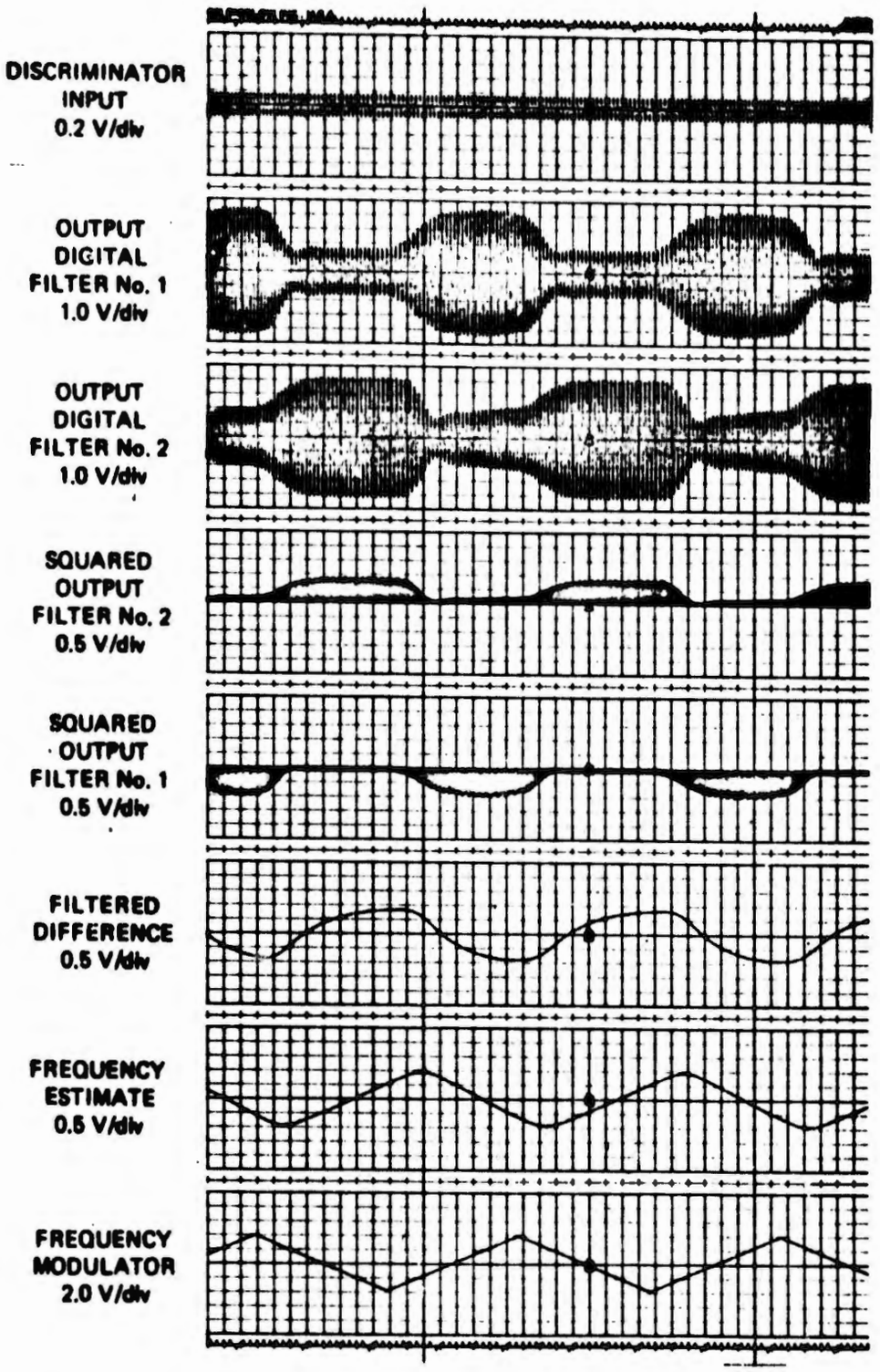


Fig. 24. Simulation results S/N = ∞ db.

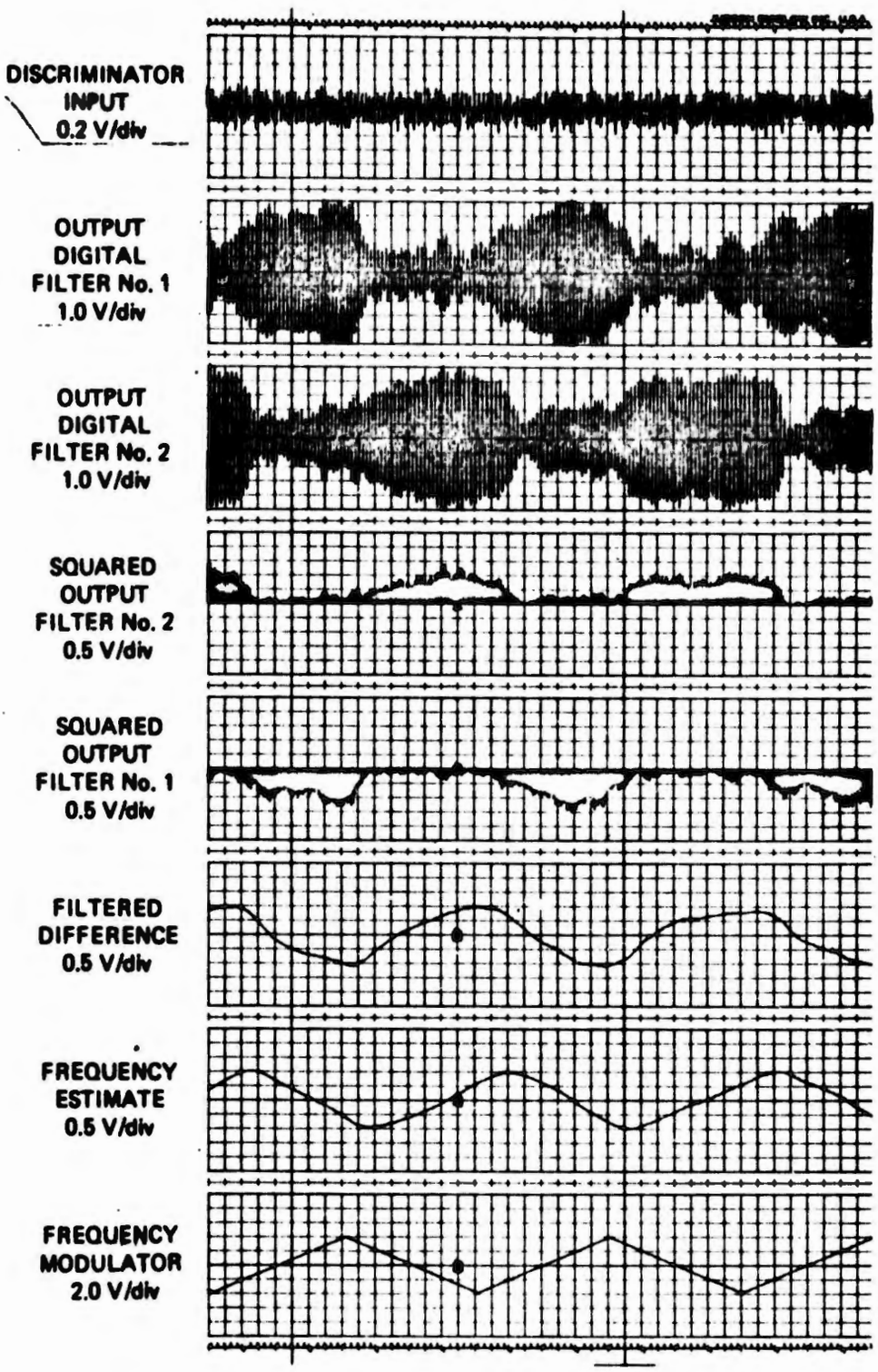


Fig. 25. Simulation results S/N = 3 db.

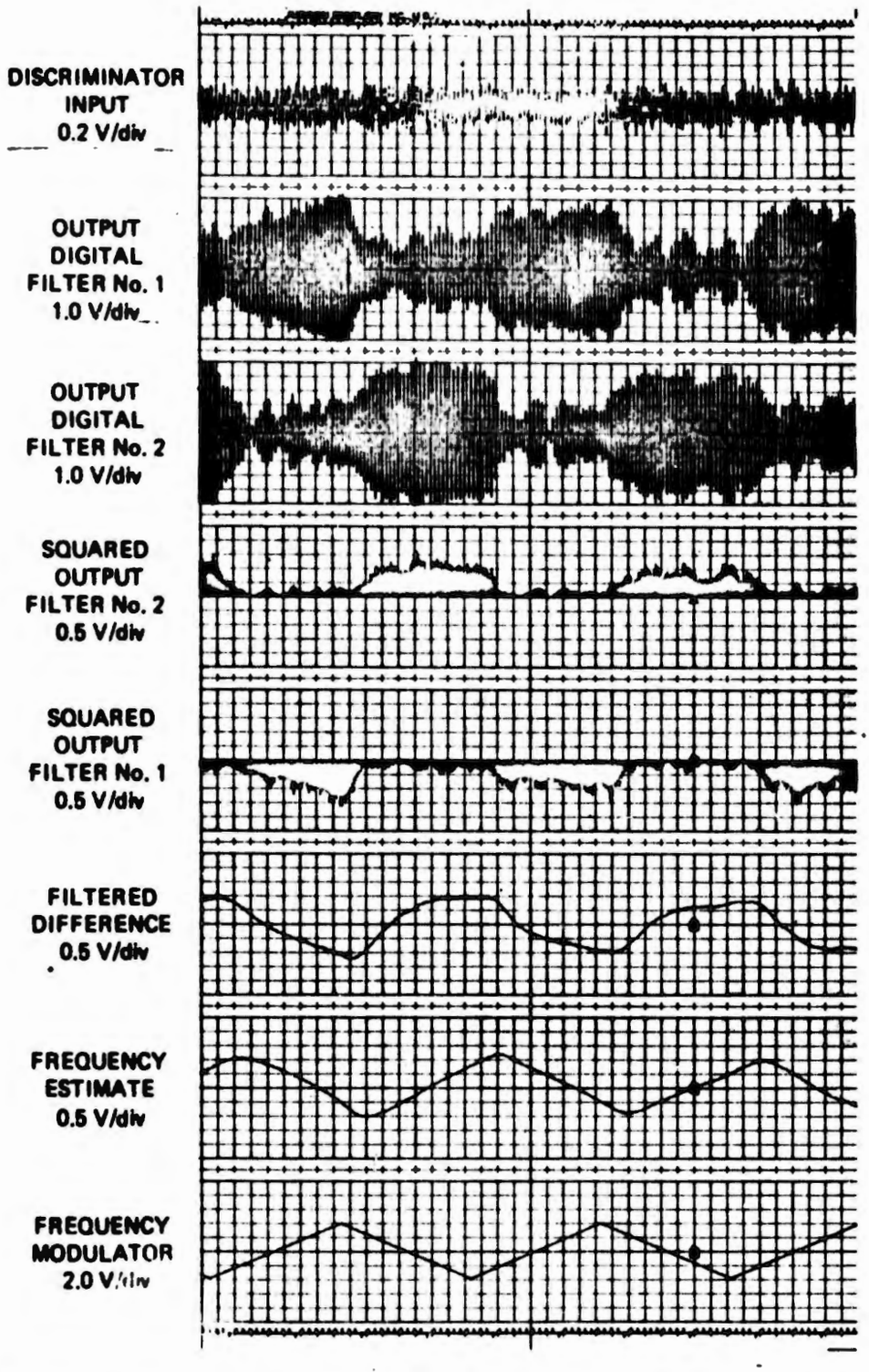


Fig. 26. Simulation results S/N = 1 db.

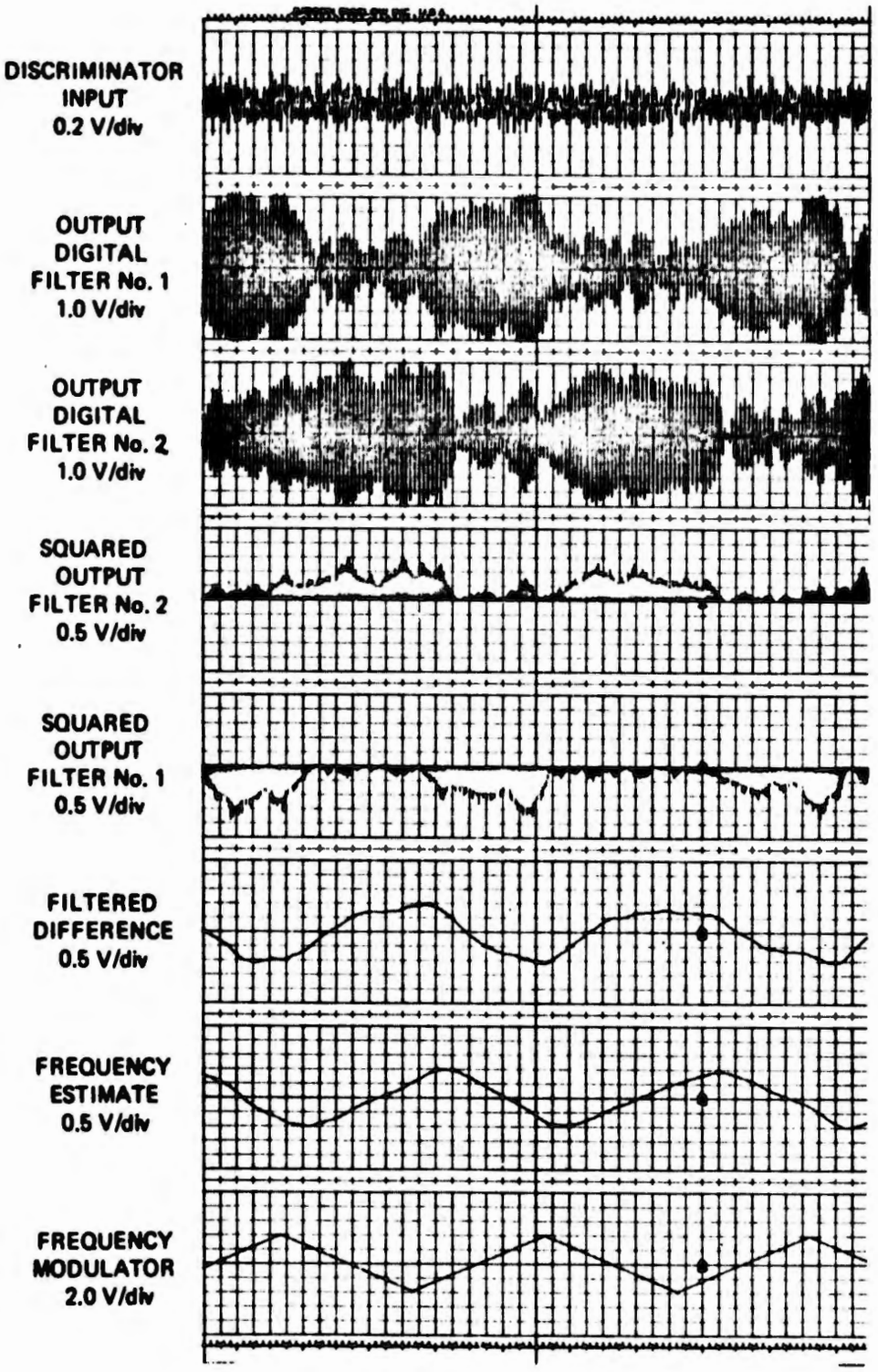


Fig. 27. Simulation results S/N = 0 db.

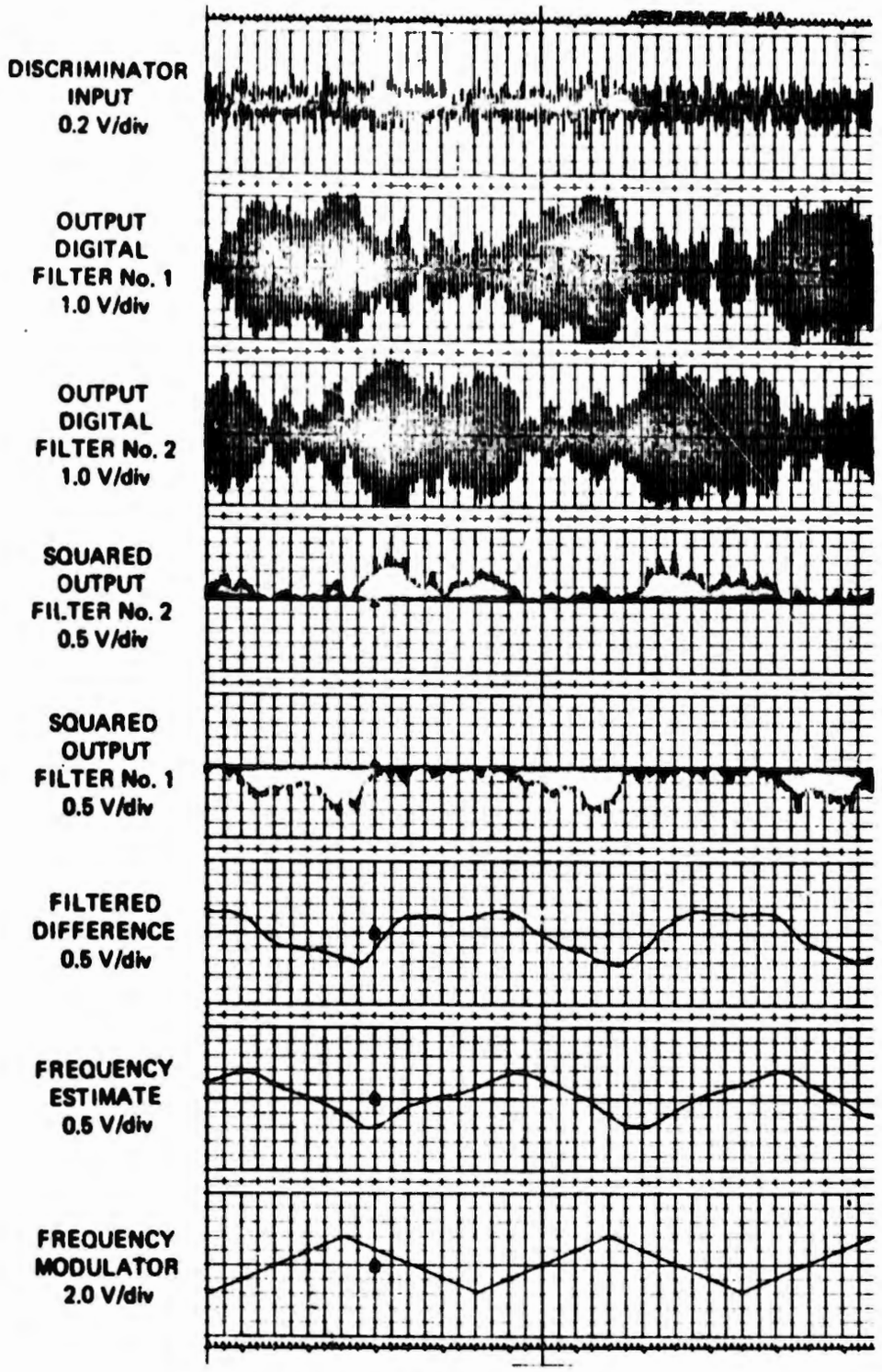


Fig. 28. Simulation results S/N = - 1 db.

The first bending mode has eigenvalues that are time dependent (ω is a time-varying quantity) and must be identified in order to provide proper aircraft stability. A possible means of identification of the bending mode will be through use of the adaptive digital discriminator. The block diagram of such a scheme is shown in Fig. 29.

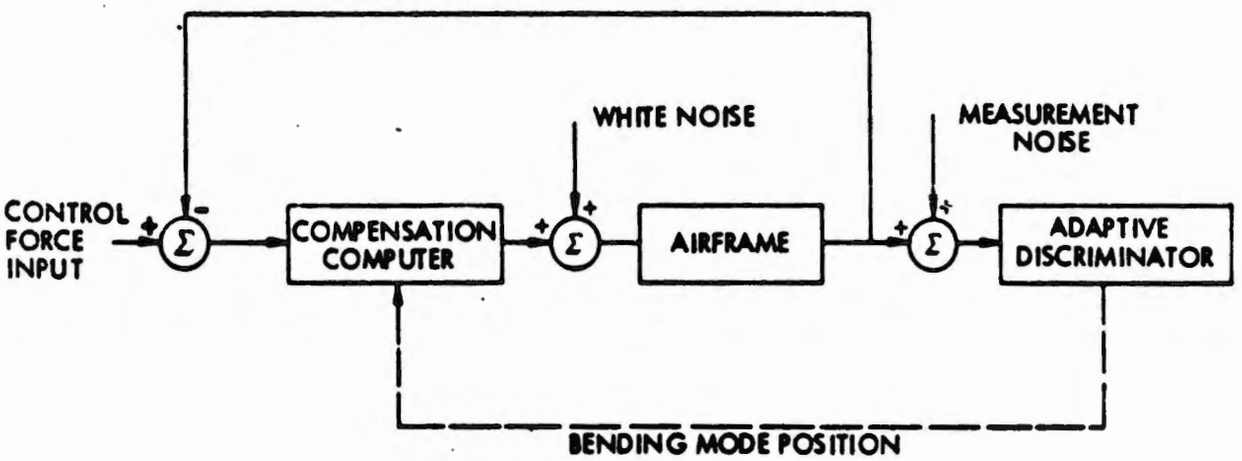


Fig. 29. Use of adaptive digital discriminator for bending mode identification.

In this adaptive system the output of the adaptive digital discriminator provides a feedback signal to the compensation computer. The feedback signal is used as an indication of the first bending mode's pole position and the compensation computer then provides the proper system compensation to maintain stability. If the compensation computer is a digital compensator it can be designed so that the sampling rate of the adaptive discriminator is the correct sampling rate for the digital compensator.

Analog Computer Simulation

The airframe described by Eq. 6 as well as the adaptive digital discriminator was simulated on the EAI 8812 analog computer. The

simulation diagram for the airframe is given in Fig. 30. The first bending mode is made time varying by use of a multiplier in a feedback loop. The time varying poles have a mean frequency of 10 rad/sec and the poles were varied in a random manner by means of a 255 bit pseudo

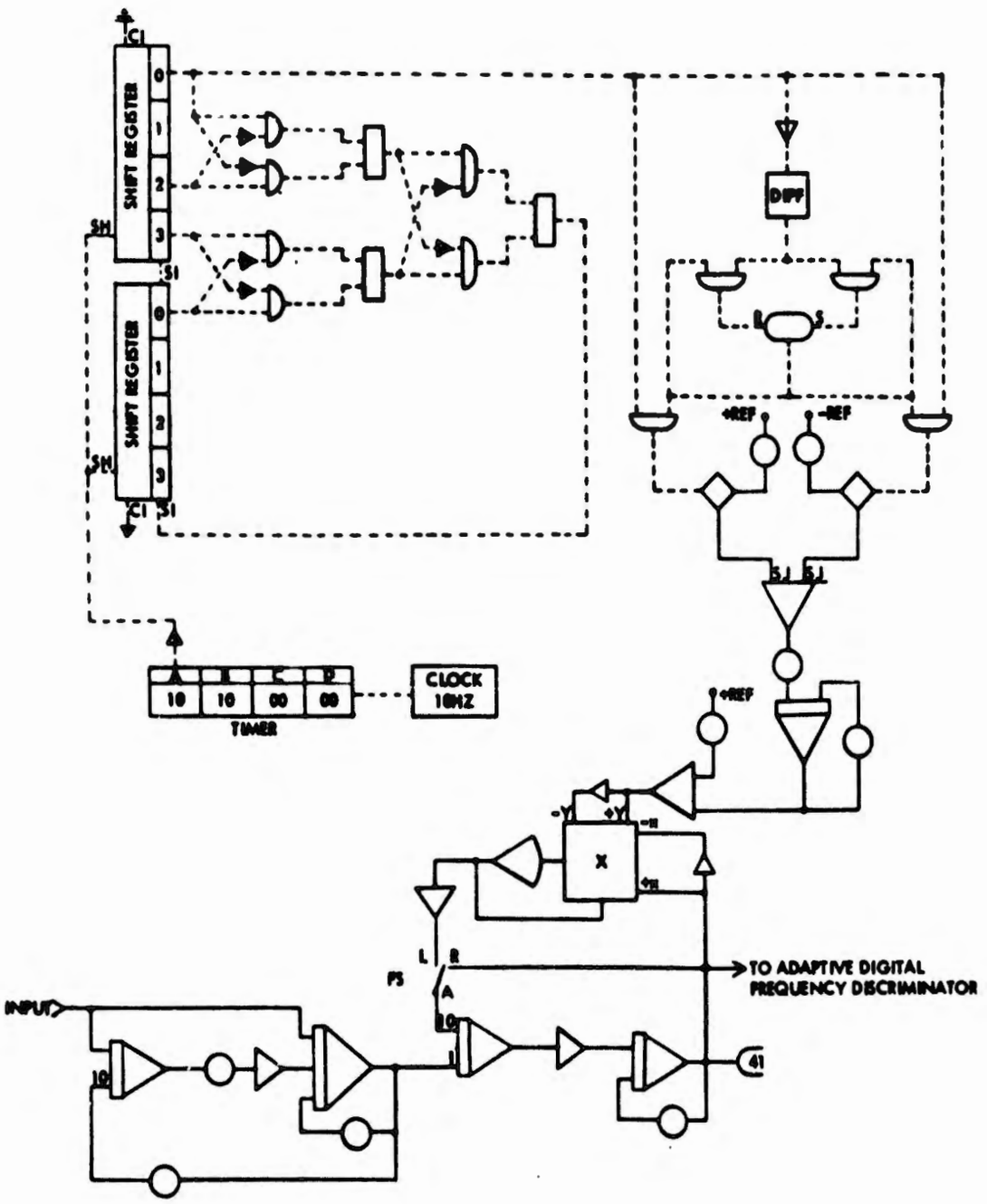


Fig. 30. Analog computer simulation diagram flexible airframe, open loop.

random sequence generator. The maximum excursion of the pole positions is $\pm$ 2 rad/sec.

Evaluation of the tracking ability of the adaptive discriminator in following a random frequency shift was first attempted with a frequency modulated sine wave, the modulator being a random voltage. The results of this simulation, shown in Fig. 31, reveal that the adaptive discriminator has no problem tracking a random modulated sine wave.

The performance of the adaptive discriminator in identifying the first bending mode of the airframe is shown in Fig. 32. In this simulation the bending mode's pole position remains fixed while the airframe is excited with bandlimited white noise having a bandwidth of 15 Hz. One interesting facet was noted. If the discriminator is exactly at null; that is, the input frequency has been precisely identified and the discriminator is locked on it, changes in amplitude of the input signal have no effect on the performance of the discriminator. However, if there is a finite error, changes in input signal amplitude do tend to shift the sampling rate of the discriminator.

Additive white noise with a bandwidth of 50 Hz was added to the output of the airframe and the result of adding measurement noise to the observation is shown in Fig. 33. The adaptive digital discriminator seemed to track the input frequency with less error when the measurement noise was present. A possible explanation of this fact is that the measurement noise perturbs both of the digital filters exciting them at their natural frequency so that if the signal momentarily disappears the error tends to remain constant.

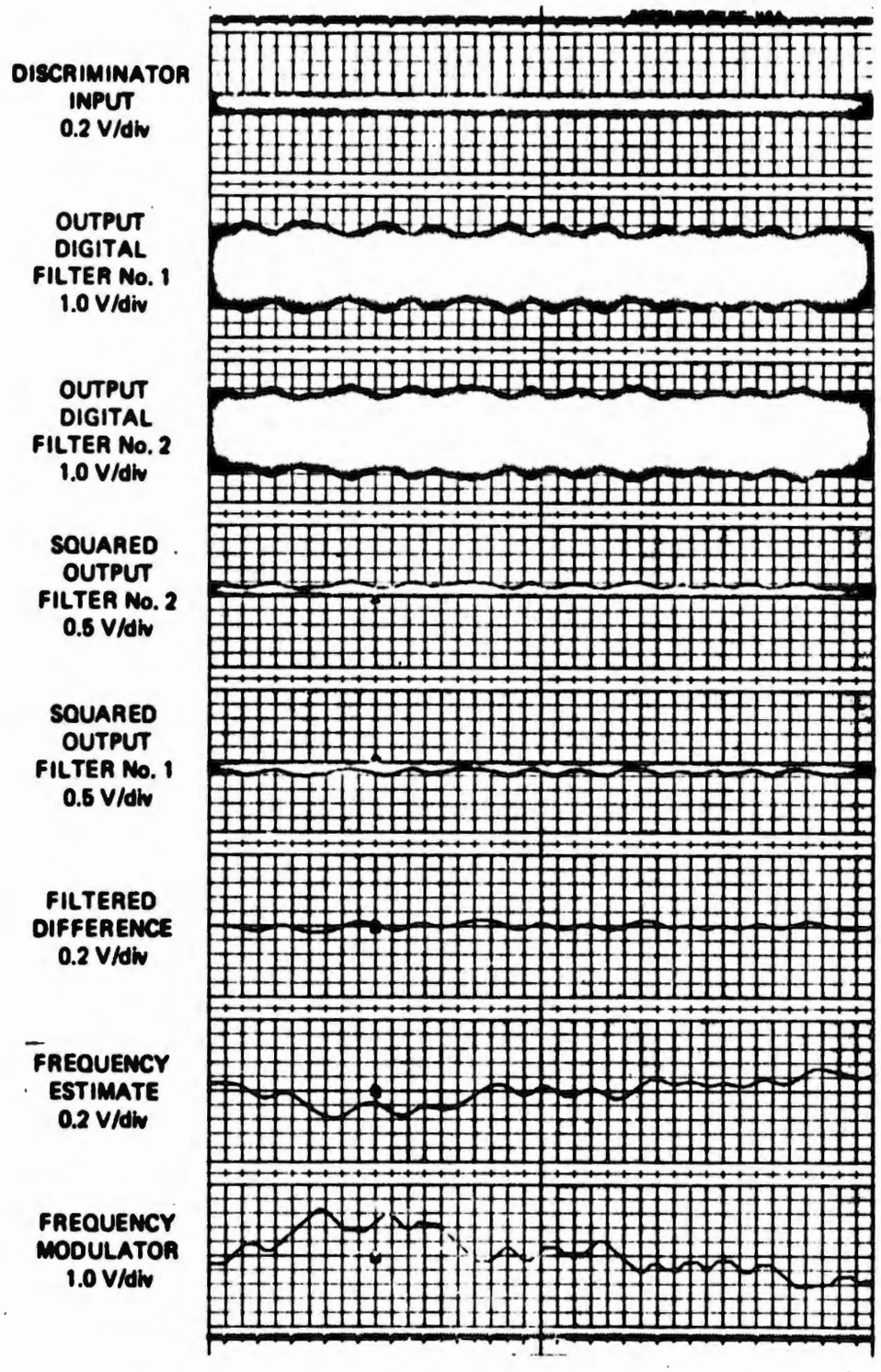


Fig. 31. Simulation results, identification of random modulated sinusoidal.

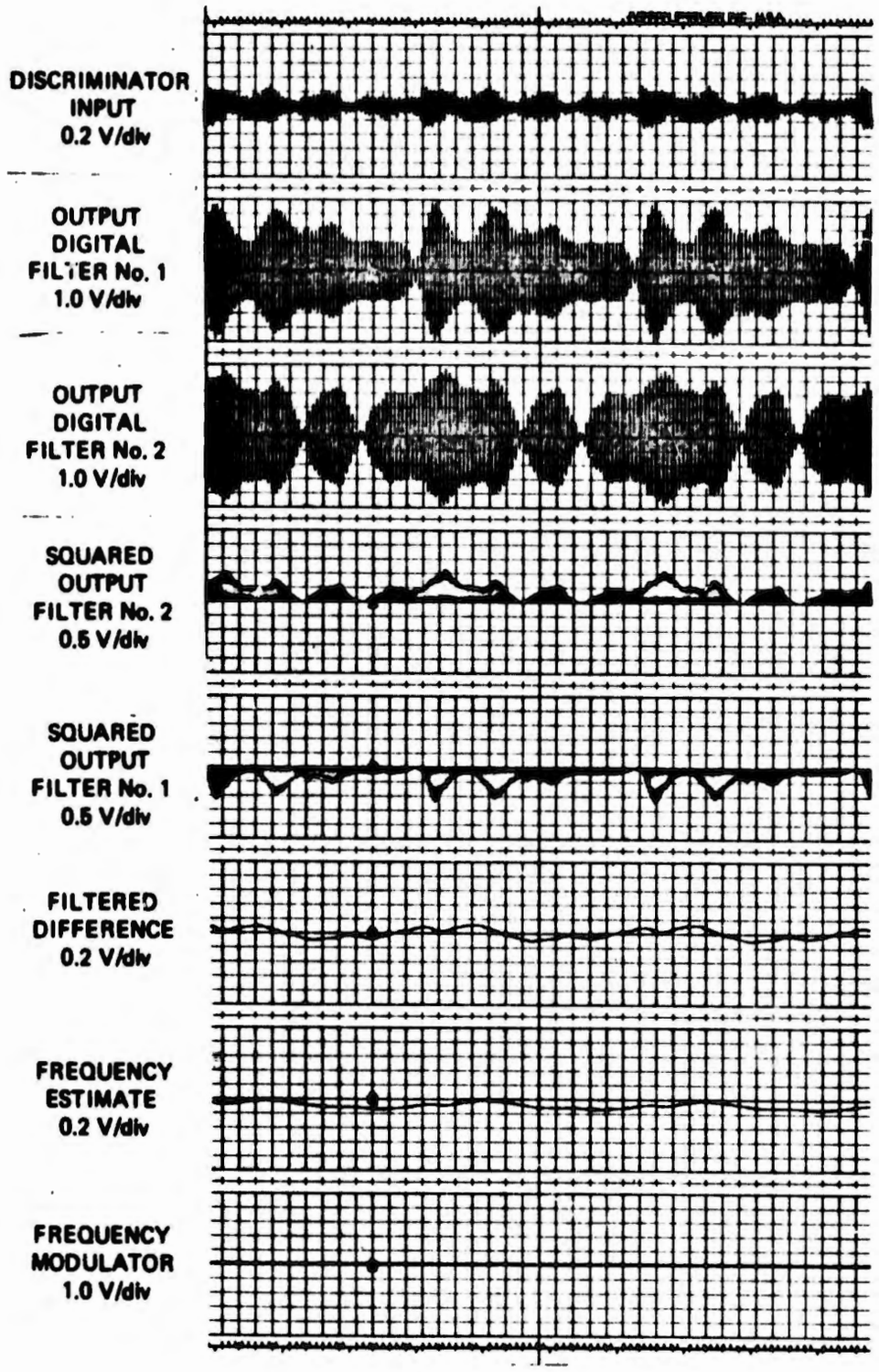


Fig. 32. Simulation results, identification of airframe bending mode.

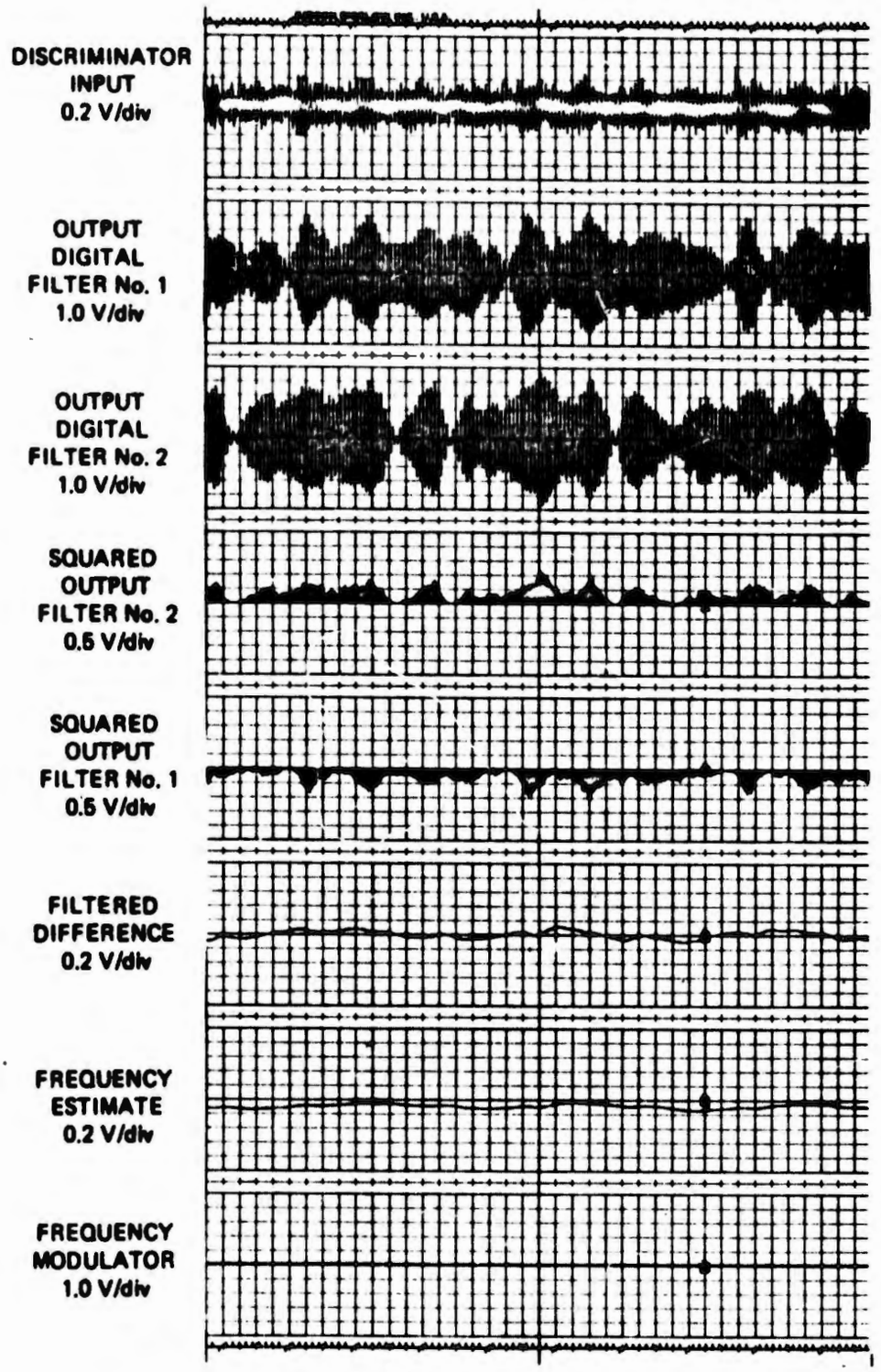


Fig. 33. Simulation results, identification of airframe bending mode, noise corrupted measurement.

In order to more correctly simulate actual flight conditions the pole positions of the first bending mode were changed in a random manner. The measurement noise added to the airframe output was again white noise with a bandwidth of 50 Hz. The signal to noise ratio was about 1 db. The results of this simulation, shown in Fig. 34, show that the tracking filters performance is quite acceptable for this simulated flight condition and its use as a bending mode frequency identifier is indeed practical.

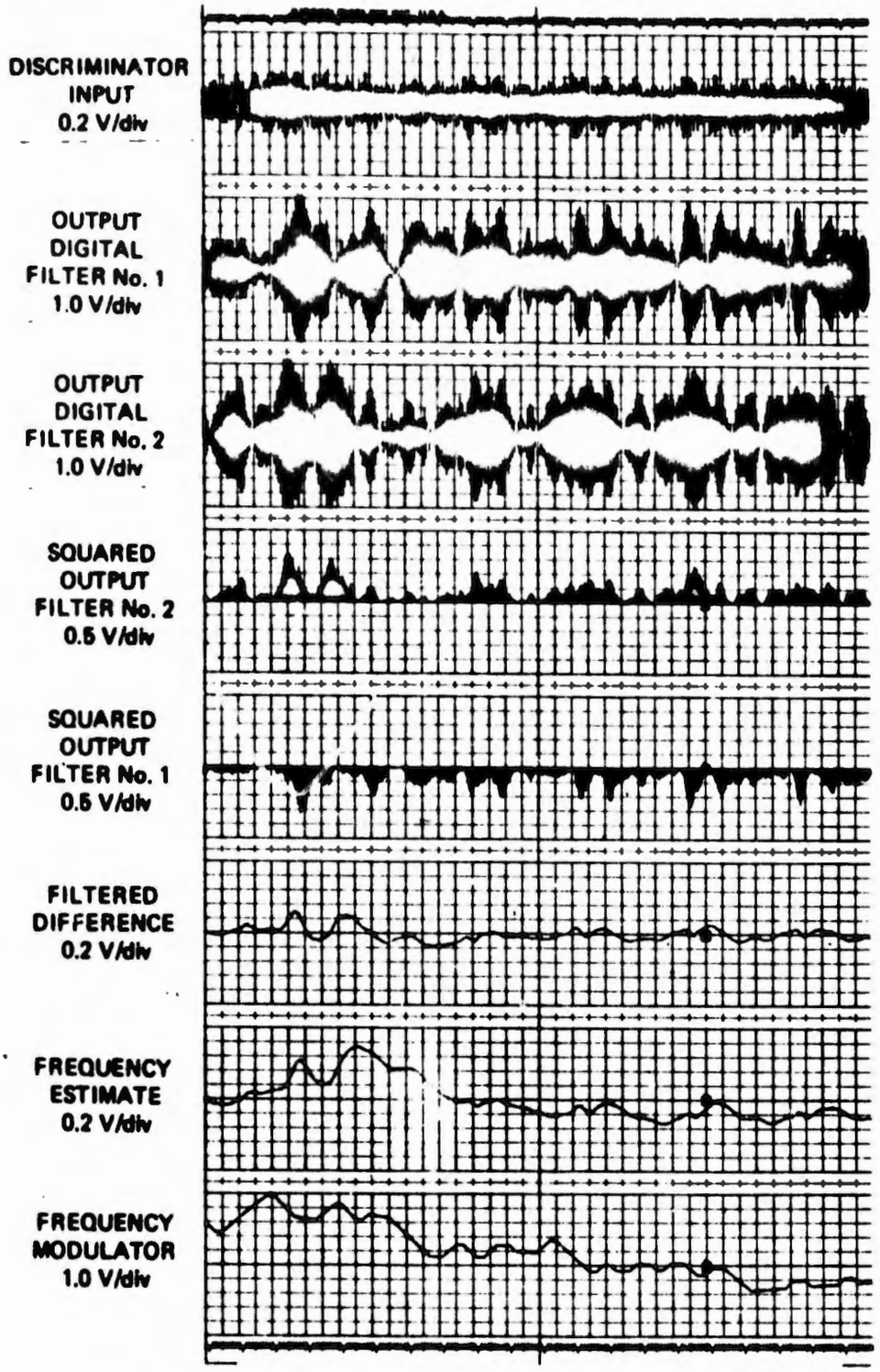


Fig. 34. Simulation results, identification of time varying airframe bending mode.

Estimation of the Bending Modes of a Flexible Airframe through Determination of Spectral Density Process Output

Introduction

The techniques incorporated here utilize the principle of the spectral density function; namely, the spectral density function of an output random process is the square of the frequency response function of the process (system) to be identified multiplied by the spectral density function of an input random process[7]. It follows that the system response function can be determined from the spectral density function of the output process and the spectral density of the input process. The technique employed in this section estimate the output spectral density function from measurements made on the output of the airframe and identify the bending modes of the airframe system[8].

Generation of the Airframe Process

The relationship between the spectral density function and the system response function can be expressed as

$$G_x(\omega) = |H(j\omega)|^2 G_f(\omega) \tag{38}$$

where $G_x(\omega)$ is the spectral density function of the output process and $G_f(\omega)$ is the spectral density function of the input process. Consider the cases of an open loop airframe system and a closed loop airframe system (no compensation) shown in Fig. 35.

In both system configurations additive white noise was added to the output of the airframe process in order to simulate measurement noise.

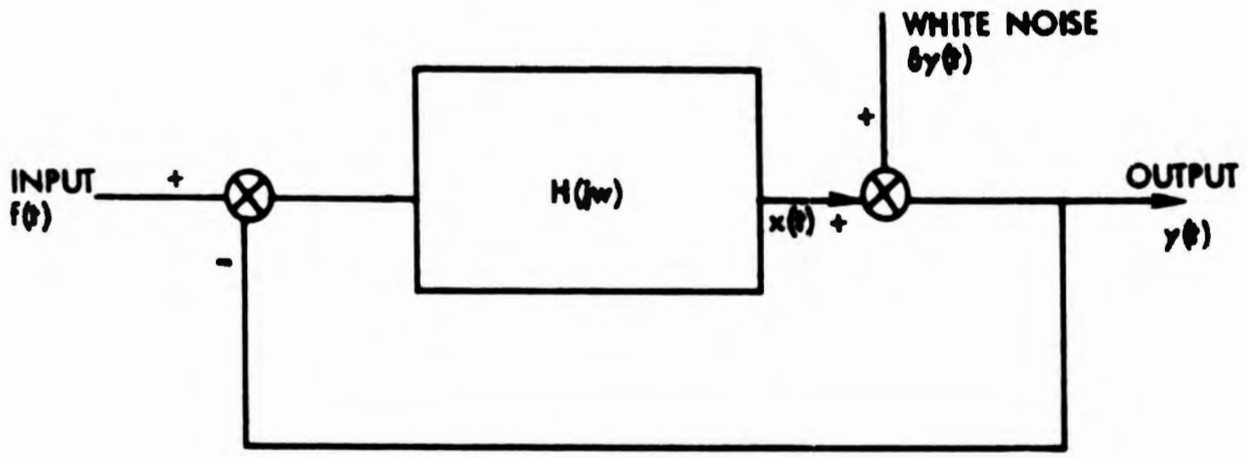


Fig. 35. System used in process identification.

As in the case of the digital discriminator, white noise (continuous turbulence) was used as an input to excite the bending modes of the airframe.

Prior to explanation of system identification schemes, the output spectral density function of the airframe has to be generated. The spectral density function corresponding to the output of the flexible airframe is a fourth-order process of the form

$$G_x(\omega) = \frac{B_3\omega^6 + B_2\omega^4 + B_1\omega^2 + B_0}{A_4\omega^8 + A_3\omega^6 + A_2\omega^4 + A_1\omega^2 + A_0} . \tag{39}$$

This can be shown by considering the general equation of the flexible airframe:

$$H(s) = \frac{K(s + \gamma)}{(s + \alpha_1 \pm j\beta_1)(s + \alpha_2 \pm j\beta_2)} \tag{40}$$

where

$$\left(\alpha_1^2 + \beta_1^2\right)^{1/2} = \omega_{nr} = \omega_1 \text{ (rigid body mode)}$$

$$\left(\alpha_2^2 + \beta_2^2\right)^{1/2} = \omega_{n1} = \omega_2 \text{ (first bending mode)} \tag{41}$$

Using Eq. (38) gives

$$G_x(\omega) = \left| \frac{K(j\omega + \gamma)}{(j\omega + \alpha_1 \pm j\beta_1)(j\omega + \alpha_2 \pm j\beta_2)} \right|^2 G_f(\omega), \tag{42}$$

and the necessary equivalence between Eqs. (39) and (42) can be found by defining

$$\begin{aligned}
&G_f(\omega) = 1.0 \text{ (i.e., white noise)},\\
&A_0 = (\alpha_1^2 + \beta_1^2)^2(\alpha_2^2 + \beta_2^2)^2 ,\\
&A_1 = -2(\alpha_1^2 + \beta_1^2 + \alpha_2^2 + \beta_2^2 + 4\alpha_1\alpha_2)(\alpha_1^2 + \beta_1^2)(\alpha_2^2 + \beta_2^2),\\
&A_2 = (\alpha_1^2 + \beta_1^2 + \alpha_2^2 + \beta_2^2 + 4\alpha_1\alpha_2)^2 + 2(\alpha_1^2 + \beta_1^2)(\alpha_2^2 + \beta_2^2),\\
&A_3 = -2(\alpha_1^2 + \beta_1^2 + \alpha_2^2 + \beta_2^2 + 4\alpha_1\alpha_2)^2 ,\\
&A_4 = 1,\\
&B_0 = K^2\gamma^2 ,\\
&B_1 = K^2 ,\\
&B_2 = 0,\\
&B_3 = 0.
\end{aligned} \tag{43}$$

Generation of an Open Loop Fourth-Order Process

Given the open loop system of Fig. 36 the differential equation which describes the system can be written

$$[D^4 + aD^3 + bD^2 + cD + d]x = K(D + \gamma)f \tag{44}$$

with output equation,

$$y(t) = x(t) + \delta y(t) \tag{45}$$

where

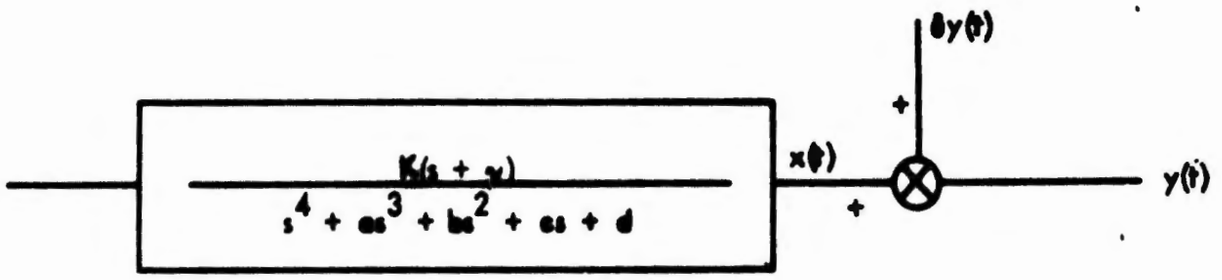


Fig. 36. Open loop fourth-order process generation.

$$
\begin{aligned}
\omega_1 &= \left(\alpha_1^2 + \beta_1^2\right)^{1/2}, \\
\omega_2 &= \left(\alpha_2^2 + \beta_2^2\right)^{1/2}, \\
a &= 2(\alpha_1 + \alpha_2), \\
b &= \omega_1^2 + \omega_2^2, \\
c &= 2\alpha_1\omega_2^2 + 2\alpha_2\omega_1^2, \\
d &= \omega_1^2\omega_2^2 .
\end{aligned}
\tag{46}
$$

The spectral density function $G_x(\omega)$ of the open loop airframe can be expressed as

$$
G_x(\omega) = \frac{K^2(\omega^2 + \gamma^2)}{[(\omega^2 + \omega_1^2)^2 - 4\beta_1\omega^2][(\omega^2 + \omega_2^2)^2 - 4\beta_2\omega^2]} . \tag{47}
$$

In order to achieve a system output suitable for digital computer processing we must discretize x(t) when the airframe system is excited with white noise. The state equations governing the system can be expressed as

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -d & -c & -b & -a \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ \gamma - a \end{bmatrix} Kf(t), \quad (48)$$

$$y(t) = [1 \quad 0 \quad 0 \quad 0] \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \delta y(t). \quad (49)$$

The state transition matrix of the above system can be written as

$$\Phi(t) = \begin{bmatrix} \phi_{11}(t) & \phi_{12}(t) & \phi_{13}(t) & \phi_{14}(t) \\ \phi_{21}(t) & \phi_{22}(t) & \phi_{23}(t) & \phi_{24}(t) \\ \phi_{31}(t) & \phi_{32}(t) & \phi_{33}(t) & \phi_{34}(t) \\ \phi_{41}(t) & \phi_{42}(t) & \phi_{43}(t) & \phi_{44}(t) \end{bmatrix} \quad (50)$$

where

$$\phi_{i4}(t) = \mathcal{L}^{-1} \frac{s^{i-1}}{(s + \alpha_1 \pm j\beta_1)(s + \alpha_2 \pm j\beta_2)}$$

$$= A_i e^{-\alpha_1 t} \cos \beta_1 t + \frac{-\alpha_1 A_i + \beta_i}{\beta_1} e^{-\alpha_1 t} \sin \beta_1 t$$

$$+ C_i e^{-\alpha_2 t} \cos \beta_2 t + \frac{-\alpha_2 C_i + D_i}{\beta_2} e^{-\alpha_2 t} \sin \beta_2 t$$

and

$$A_1 = 2(\alpha_1 - \alpha_2)/\Delta$$

$$B_1 = [4\alpha_1(\alpha_1 - \alpha_2) - \omega_1^2 + \omega_2^2]/\Delta$$

$$C_1 = 2(\alpha_2 - \alpha_1)/\Delta$$

$$D_1 = [4\alpha_2(\alpha_1 - \alpha_2) + \omega_1^2 - \omega_2^2]/\Delta$$

$$A_2 = (\omega_2^2 - \omega_1^2)/\Delta$$

$$B_2 = 2(\alpha_2 - \alpha_1)\omega_1^2/\Delta$$

$$C_2 = (\omega_1^2 - \omega_2^2)/\Delta$$

$$D_2 = 2(\alpha_1 - \alpha_2)\omega_2^2/\Delta$$

$$A_3 = 2(\alpha_2\omega_1^2 - \alpha_1\omega_2^2)/\Delta$$

$$B_3 = \omega_1^2(\omega_1^2 - \omega_2^2)/\Delta$$

$$C_3 = 2(\alpha_1\omega_2^2 - \alpha_2\omega_1^2)/\Delta$$

$$D_3 = \omega_2^2(\omega_2^2 - \omega_1^2)/\Delta$$

$$A_4 = [4\alpha_1(\alpha_2 - \alpha_1) - \omega_1^4 - \omega_1^2\omega_2^2]/\Delta$$

$$B_4 = 2\omega_1^2(\alpha_1\omega_2^2 - \alpha_2\omega_1^2)/\Delta$$

$$C_4 = (\omega_1^2 + \omega_2^2)/\Delta$$

$$D_4 = 2\omega_2^2(\alpha_2\omega_1^2 - \alpha_1\omega_2^2)/\Delta$$

$$\Delta = (\omega_1^2 - \omega_2^2)^2 + 4(\alpha_1 - \alpha_2)(\alpha_1\omega_2^2 - \alpha_2\omega_1^2) \qquad (51)$$

$$\phi_{11} = \phi_{44} + a\phi_{34} + b\phi_{24} + c\phi_{14}$$

$$\phi_{12} = \phi_{34} + a\phi_{24} + b\phi_{14}$$

$$\phi_{13} = \phi_{24} + a\phi_{14}$$

$$\phi_{21} = - d\phi_{14}$$

$$\phi_{22} = \phi_{44} + a\phi_{34} + b\phi_{24}$$

$$\phi_{23} = \phi_{34} + a\phi_{24}$$

$$\phi_{31} = - d\phi_{24}$$

$$\phi_{32} = - c\phi_{24} - d\phi_{14}$$

$$\phi_{33} = \phi_{44} + a\phi_{34}$$

$$\phi_{41} = - d\phi_{34}$$

$$\phi_{42} = - c\phi_{34} - d\phi_{24}$$

$$\phi_{43} = - b\phi_{34} - c\phi_{24} - d\phi_{14}$$

Define g_n to be the discrete state response due to the system driving function f(t) (white noise) that occurs in the time interval Δt between t_n and t_{n+1}. Then,

$$g_n = \int_0^{\Delta t} \Phi(\tau) \begin{bmatrix} 0 \\ 0 \\ 1 \\ \gamma - a \end{bmatrix} Kf(t - \tau)d\tau \tag{52}$$

$$= K \int_0^{\Delta t} \begin{bmatrix} \phi_{13}(\tau) + (\gamma - a)\phi_{14}(\tau) \\ \phi_{23}(\tau) + (\gamma - a)\phi_{24}(\tau) \\ \phi_{33}(\tau) + (\gamma - a)\phi_{34}(\tau) \\ \phi_{43}(\tau) + (\gamma - a)\phi_{44}(\tau) \end{bmatrix} f(t - \tau)d\tau \tag{53}$$

The recursive equations of the airframe output suitable for computer processing can now be expressed as

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}_{n+1} = \Phi(\Delta t) \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}_n + g_n \, , \tag{54}$$

$$y_n = [1 \quad 0 \quad 0 \quad 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \delta y_n \, . \tag{55}$$

Since the input function f(t) is assumed to be white noise the generated sequence, g_n, is independent and can be thought of as a constant multiplied by a random number; that is,

$$g_n = K \begin{bmatrix} \phi_{13}(\frac{\Delta t}{2}) + (\gamma - a)\phi_{14}(\frac{\Delta t}{2}) \\ \phi_{23}(\frac{\Delta t}{2}) + (\gamma - a)\phi_{24}(\frac{\Delta t}{2}) \\ \phi_{33}(\frac{\Delta t}{2}) + (\gamma - a)\phi_{34}(\frac{\Delta t}{2}) \\ \phi_{43}(\frac{\Delta t}{2}) + (\gamma - a)\phi_{44}(\frac{\Delta t}{2}) \end{bmatrix} (\text{sequence of random numbers}). \quad (56)$$

The measurement noise $\delta y(t)$ can be discretized as δy_n using another sequence of random numbers independent of the first. We can now generate a fourth-order process from the recursive equations by substituting the above form for g_n.

Generation of Closed Loop Fourth-Order Process

Suppose the airframe is simulated as a closed loop system shown in Fig. 37. The differential equation describing the system is

$$[D^4 + aD^3 + bD^2 + (e+1)D + d + \gamma]x(t) = K[D + \gamma]f(t) - [D + \gamma]y(t) \quad (57)$$

with output equation

$$y(t) = x(t) + \delta y(t). \quad (58)$$

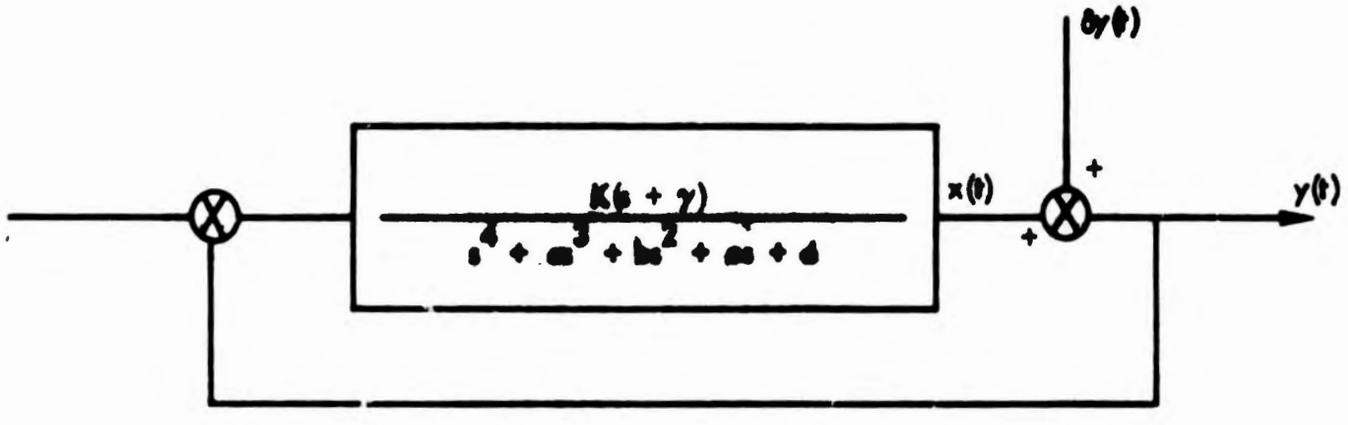


Fig. 37. Closed loop fourth-order process generation.

The system bending modes are defined to be

$$\begin{aligned}
\omega_3 &= \left(\alpha_3^2 + \beta_3^2\right)^{1/2} \text{ (closed loop rigid body mode)},\\
\omega_4 &= \left(\alpha_4^2 + \beta_4^2\right)^{1/2} \text{ (closed loop first bending mode)},\\
a &= 2(\alpha_3 + \alpha_4),\\
b &= \omega_3^2 + \omega_4^2 ,\\
c &= K + 2\alpha_3\omega_4^2 + 2\alpha_4\omega_3^2 ,\\
d &= K\gamma + \omega_3^2\omega_4^2 .
\end{aligned} \tag{59}$$

The spectral density function $G_x(\omega)$ can be expressed as

$$G_x(\omega) = \frac{K^2(\omega^2 + \gamma^2)}{[(\omega^2 + \omega_3^2)^2 - 4\beta_3^2\omega^2][(\omega^2 + \omega_4^2) - 4\beta_4^2\omega^2]} . \tag{60}$$

Again the system output must be discretized. Proceeding as before, the state equations can be written

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -(d+K\gamma) & -(c+k) & -b & -a \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ \gamma - a \end{bmatrix} K[f(t) - \delta y(t)], \tag{61}$$

$$y(t) = [1 \quad 0 \quad 0 \quad 0] \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \nu y(t). \tag{62}$$

In this case the state transition matrix $\Phi(t)$ can be obtained from the open loop state transition matrix by substituting α_3, β_3, α_4 and β_4 for α_1, β_1, α_2 and β_2 respectively and by replacing $c + k$ and $d + K\gamma$ for c and d in Eq. (51).

For this state transition matrix, the discrete output g_n due to a white noise driving function is

$$g_n = \int_0^{\Delta t} \Phi(\tau) \begin{bmatrix} 0 \\ 0 \\ 1 \\ \gamma - a \end{bmatrix} k[f(t-\tau) - \delta y(t-\tau)]d\tau \qquad (63)$$

$$= K \int_0^{\Delta t} = \begin{bmatrix} \phi_{13}(\tau) + (\gamma - a)\phi_{14}(\tau) \\ \phi_{23}(\tau) + (\gamma - a)\phi_{24}(\tau) \\ \phi_{33}(\tau) + (\gamma - a)\phi_{34}(\tau) \\ \phi_{43}(\tau) + (\gamma - a)\phi_{44}(\tau) \end{bmatrix} = [f(t-\tau) - \delta y(t-\tau)]d\tau. \qquad (64)$$

$$g_n = k \begin{bmatrix} \phi_{13}(\frac{\Delta t}{2}) + (\gamma - a)\phi_{14}(\frac{\Delta t}{2}) \\ \phi_{23}(\frac{\Delta t}{2}) + (\gamma - a)\phi_{24}(\frac{\Delta t}{2}) \\ \phi_{33}(\frac{\Delta t}{2}) + (\gamma - a)\phi_{34}(\frac{\Delta t}{2}) \\ \phi_{43}(\frac{\Delta t}{2}) + (\gamma - a)\phi_{44}(\frac{\Delta t}{2}) \end{bmatrix} \text{(input sequence of random numbers)} - \text{(measurement noise random numbers)}. \qquad (65)$$

Therefore the recursive relationships that result in a fourth-order closed loop process are

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}_{n+1} = \Phi(\Delta t) \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}_n + g_n \, . \qquad (66)$$

As a basis for investigation of the three spectral identification techniques, the airframe transfer function will be considered to be in flight condition 2. The sampling interval ΔT is selected to be 0.01 sec and the process is generated over a 55.0 sec time period allowing 5.0 sec for the transient response to die out. The generated input

process has 5000 samples and will be used as an input for each of the spectral identification methods.

Ideal identification of the spectral density of the airframe for flight condition 2 should result in spectral density functions of the form

$$G_x(\omega) = \frac{100(\omega^2 + 0.0225)}{[(\omega^2 + 100.01)^2 - 400\ \omega^2][(\omega^2 + 4.04)^2 - 16\omega^2]} \tag{67}$$

for the open loop process and

$$G_x(\omega) = \frac{100(\omega^2 + 0.0225)}{[(\omega^2 + 100.01)^2 - 400\ \omega^2][(\omega^2 + 4.035)^2 - 15.972\ \omega^2]} \tag{68}$$

for the closed loop process.

Identification of the Flexible Airframe Bending Modes

Direct Fourier Analysis

A method that can be used to determine the spectral density function of a process is the Fourier expansion of the discretized measurement of the process. The spectral density function of a random process is characterized by the average power contained in the measurement [y(t)] over a finite interval. This can be stated as

$$G(f) = \lim_{T\to\infty} \frac{1}{2T} < \left| \int_{-T}^{T} y(\tau) e^{-j2\pi f(\tau)} d\tau \right|^2 > \tag{69}$$

where the symbol <·> denotes an ensemble average.

Introduction of a data window which is the sum of N Dirac delta functions shifted in time Δt gives a concise estimate of the spectral density function. Let the data window be defined as

$$B(t) = \Delta t\ \frac{1}{2}\delta(t - \Delta t) + \sum_{n=2}^{N-1} \delta(t - n\Delta t) + \frac{1}{2}\delta(\tau - N\Delta t)\ . \tag{70}$$

The spectral density function G(f) can be put into integral form by utilization of the data window function; that is,

$$G(f) = \frac{1}{2(N-1)\Delta t} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} B(u)B(v)y(u)y(v)e^{-j2\pi f(u-v)}dudv. \quad (71)$$

It follows directly from Eq. (71) that G(f) can be expressed in the form,

$$G(f) = \frac{1}{2(N-1)\Delta t}(\alpha^2 + \beta^2)$$

where

$$\begin{aligned}\alpha &= \Delta t \left[\frac{1}{2}y_1 \sin(2\pi f\Delta t) + \sum_{n=2}^{N-1} y_n \sin(2\pi fn\Delta t) + \frac{1}{2}y_n \sin(2\pi fn\Delta t)\right] \\ \beta &= \Delta t \left[\frac{1}{2}y_1 \cos(2\pi f\Delta t) + \sum_{n=2}^{N-1} y_n \cos(2\pi fn\Delta t) + \frac{1}{2}y_n \cos(2\pi fn\Delta t)\right] .\end{aligned} \quad (72)$$

The estimate of the spectral density function G(f) by use of Eq. (71) was obtained through use of a digital computer. See Appendix C. Tables 2 and 3 and Figs. 38 and 39 show the results of this identification procedure.

Spectral Filters Method

The spectral filters method for estimating the spectral density function is based on the fact that a spectral density function can be expressed as a Fourier sum of sine and cosine functions. Instead of using the cosine and sine functions in Eq. (72), the spectral density function could be determined by replacing the sine and cosine functions by the kernel functions

$$sqs(2\pi fn\Delta t) = \frac{\sin(2\pi fn\Delta t)}{|\sin(2\pi fn\Delta t)|}, \quad (73)$$

$$sqc(2\pi fn\Delta t) = \frac{\cos(2\pi fn\Delta t)}{|\cos(2\pi fn\Delta t)|}. \quad (74)$$

The spectral filters method assumes that the time slice is chosen to be the period of a spectral filter tuned at frequency f. If $m\Delta t$ is

Table 2. Spectral density function estimates by direct Fourier analysis for an open loop fourth-order process example.

f	G(f)	f	G(f)	f	G(f)
0.0	0.006372	0.300	0.080481	0.600	0.003105
0.010	0.000129	0.310	0.019456	0.610	0.001890
0.020	0.009928	0.320	0.040109	0.620	0.004452
0.030	0.009025	0.330	0.090194	0.630	0.004188
0.040	0.008636	0.340	0.043039	0.640	0.005473
0.050	0.005244	0.350	0.010659	0.650	0.000105
0.060	0.001798	0.360	0.005387	0.660	0.005871
0.070	0.000350	0.370	0.023630	0.670	0.010070
0.080	0.007896	0.380	0.004928	0.680	0.010929
0.090	0.011167	0.390	0.017174	0.690	0.017354
0.100	0.000114	0.400	0.024667	0.700	0.009356
0.110	0.006328	0.410	0.011858	0.710	0.001096
0.120	0.001779	0.420	0.005860	0.720	0.000802
0.130	0.000652	0.430	0.004782	0.730	0.000606
0.140	0.000200	0.440	0.008346	0.740	0.000420
0.150	0.004412	0.450	0.009708	0.750	0.000106
0.160	0.002908	0.460	0.005524	0.760	0.000280
0.170	0.000056	0.470	0.001217	0.770	0.000431
0.180	0.000359	0.480	0.002296	0.780	0.000192
0.190	0.005099	0.490	0.006443	0.790	0.000746
0.200	0.005774	0.500	0.002084	0.800	0.002890
0.210	0.004056	0.510	0.001371	0.810	0.005549
0.220	0.001191	0.520	0.011275	0.820	0.006504
0.230	0.006223	0.530	0.013779	0.830	0.002198
0.240	0.009709	0.540	0.006379	0.840	0.001397
0.250	0.003834	0.550	0.005335	0.850	0.006567
0.260	0.000109	0.560	0.006782	0.860	0.000488
0.270	0.019312	0.570	0.003107	0.870	0.012770
0.280	0.017032	0.580	0.007552	0.880	0.020359
0.290	0.017760	0.590	0.013343	0.890	0.010280

Table 2. Continued.

f	G(f)	f	G(f)	f	G(f)
0.900	0.013120	1.210	0.004110	1.520	0.012064
0.910	0.023397	1.220	0.004849	1.530	0.015015
0.920	0.021952	1.230	0.001227	1.540	0.008875
0.930	0.009867	1.240	0.000472	1.550	0.013914
0.940	0.007427	1.250	0.000505	1.560	0.011155
0.950	0.003157	1.260	0.001801	1.570	0.032841
0.960	0.005959	1.270	0.002489	1.580	0.130266
0.970	0.008043	1.280	0.003683	1.590	0.132914
0.980	0.010449	1.290	0.000985	1.600	0.071187
0.990	0.008126	1.300	0.003717	1.610	0.009458
1.000	0.004652	1.310	0.001759	1.620	0.020229
1.010	0.004914	1.320	0.006705	1.630	0.053647
1.020	0.001979	1.330	0.017927	1.640	0.059426
1.030	0.002517	1.340	0.021287	1.650	0.053232
1.040	0.006209	1.350	0.008513	1.660	0.014391
1.050	0.000346	1.360	0.006165	1.670	0.007839
1.060	0.002715	1.370	0.002841	1.680	0.011402
1.070	0.005644	1.380	0.000091	1.690	0.011125
1.080	0.017333	1.390	0.008612	1.700	0.001008
1.090	0.014347	1.400	0.046884	1.710	0.020773
1.100	0.002195	1.410	0.029689	1.720	0.011143
1.110	0.000163	1.420	0.001302	1.730	0.000993
1.120	0.000542	1.430	0.025521	1.740	0.003817
1.130	0.001979	1.440	0.005454	1.750	0.000500
1.140	0.001369	1.450	0.005295	1.760	0.004467
1.150	0.005898	1.460	0.027199	1.770	0.003194
1.160	0.018501	1.470	0.046050	1.780	0.006274
1.170	0.016713	1.480	0.016021	1.790	0.009891
1.180	0.006231	1.490	0.007085	1.800	0.006324
1.190	0.001834	1.500	0.051012	1.810	0.001133
1.200	0.001878	1.510	0.044624	1.820	0.001678

Table 2. Continued.

f	G(f)	f	G(f)	f	G(f)
1.830	0.000336	1.890	0.004389	1.950	0.015789
1.840	0.000232	1.900	0.001342	1.960	0.005868
1.850	0.002555	1.910	0.002830	1.970	0.001788
1.860	0.010126	1.920	0.001622	1.980	0.004740
1.870	0.005988	1.930	0.008226	1.990	0.005461
1.880	0.002369	1.940	0.014704	2.000	0.000066

the time slice, m an even integer, then Eq. (72) becomes a spectral filter tuned at frequency f; that is,

$$f = \frac{1}{2m\Delta t}$$

$$G(f) = \frac{1}{2}\,(\alpha_m^2 + \beta_m^2) \tag{75}$$

where

$$\alpha_m(2pm\Delta t) = \frac{1}{m(p+1)} \sum_{k=1}^{p} \left[(-1)^{k-1} \sum_{i=1}^{km} y_i + (-1)^k \sum_{i=km+1}^{2\,km} y_i \right]$$

$$\beta_m(2pm\Delta t) = \frac{1}{m(p+1)} \sum_{k=1}^{p} \left[(-1)^{k-1} \sum_{i=1}^{km/2} y_i + (-1)^k \sum_{i=1/2(m+1)}^{3/2\,km} y_i \right. \tag{76}$$

$$\left. + (-1)^{k-1} \sum_{i=3/2(km+1)}^{2\,km} y_i \right] .$$

Now α_m is the average value over p periods of the sine kernel function multiplied by the input measurement signal and β_m is the average value over p periods of the cosine kernel function multiplied by the input measurement signal. The estimates of G(f) using the spectral filter method are given in Tables 4 and 5 and Figs. 40 and 41 for the open and closed loop process respectively.

Table 3. Spectral density function estimates by direct Fourier analysis for a closed loop fourth-order process example.

f	G(f)	f	G(f)	f	G(f)
0.0	0.000254	0.300	0.052767	0.600	0.002065
0.010	0.006527	0.310	0.019801	0.610	0.025675
0.020	0.010786	0.320	0.104973	0.620	0.023848
0.030	0.011285	0.330	0.246680	0.630	0.009646
0.040	0.007271	0.340	0.134029	0.640	0.003510
0.050	0.000140	0.350	0.018081	0.650	0.009010
0.060	0.003437	0.360	0.011619	0.660	0.003824
0.070	0.002517	0.370	0.044651	0.670	0.003547
0.080	0.000369	0.380	0.015612	0.680	0.008620
0.090	0.002926	0.390	0.007813	0.690	0.000476
0.100	0.007086	0.400	0.011644	0.700	0.004310
0.110	0.011667	0.410	0.008810	0.710	0.008657
0.120	0.001538	0.420	0.014124	0.720	0.009865
0.130	0.004243	0.430	0.006017	0.730	0.000987
0.140	0.000992	0.440	0.014174	0.740	0.000796
0.150	0.002033	0.450	0.014339	0.750	0.001016
0.160	0.000388	0.460	0.017480	0.760	0.011595
0.170	0.010944	0.470	0.017947	0.770	0.002383
0.180	0.021375	0.480	0.008820	0.780	0.015064
0.190	0.004208	0.490	0.006234	0.790	0.051926
0.200	0.005345	0.500	0.003704	0.800	0.032079
0.210	0.020127	0.510	0.001690	0.810	0.004615
0.220	0.009271	0.520	0.001008	0.820	0.001296
0.230	0.000735	0.530	0.009822	0.830	0.000031
0.240	0.001291	0.540	0.009485	0.840	0.004240
0.250	0.014386	0.550	0.005908	0.850	0.012507
0.260	0.009942	0.560	0.011077	0.860	0.011734
0.270	0.034209	0.570	0.005778	0.870	0.005048
0.280	0.054182	0.580	0.014375	0.880	0.007120
0.290	0.001879	0.590	0.009570	0.890	0.016082

Table 3. Continued.

f	G(f)	f	G(f)	f	G(f)
0.900	0.005001	1.210	0.004319	1.520	0.006679
0.910	0.001796	1.220	0.001777	1.530	0.010107
0.920	0.001516	1.230	0.000667	1.540	0.011347
0.930	0.005228	1.240	0.002197	1.550	0.001720
0.940	0.019362	1.250	0.001272	1.560	0.005501
0.950	0.007178	1.260	0.001640	1.570	0.067164
0.960	0.003842	1.270	0.001284	1.580	0.129358
0.970	0.005273	1.280	0.002131	1.590	0.053342
0.980	0.008418	1.290	0.018722	1.600	0.097500
0.990	0.012785	1.300	0.022006	1.610	0.071238
1.000	0.004119	1.310	0.005523	1.620	0.007476
1.010	0.000297	1.320	0.000442	1.630	0.036212
1.020	0.001479	1.330	0.008904	1.640	0.047229
1.030	0.001347	1.340	0.024108	1.650	0.049769
1.040	0.001121	1.350	0.012787	1.660	0.028784
1.050	0.001102	1.360	0.007919	1.670	0.032477
1.060	0.005654	1.370	0.001334	1.680	0.003345
1.070	0.020327	1.380	0.004331	1.690	0.021028
1.080	0.017919	1.390	0.011784	1.700	0.031734
1.090	0.002003	1.400	0.009909	1.710	0.013636
1.100	0.005246	1.410	0.004205	1.720	0.001967
1.110	0.002707	1.420	0.001706	1.730	0.010678
1.120	0.000346	1.430	0.018870	1.740	0.017420
1.130	0.000563	1.440	0.013002	1.750	0.015908
1.140	0.001065	1.450	0.005183	1.760	0.001379
1.150	0.006271	1.460	0.011029	1.770	0.006494
1.160	0.007900	1.470	0.002657	1.780	0.006807
1.170	0.005298	1.480	0.000249	1.790	0.004832
1.180	0.002049	1.490	0.007962	1.800	0.019418
1.190	0.000935	1.500	0.032541	1.810	0.016722
1.200	0.000664	1.510	0.028467	1.820	0.007063

Table 3. Continued.

f	G(f)	f	G(f)	f	G(f)
1.830	0.007758	1.890	0.005985	1.950	0.009393
1.840	0.007906	1.900	0.002057	1.960	0.010535
1.850	0.002656	1.910	0.005783	1.970	0.014672
1.860	0.001837	1.920	0.003917	1.980	0.000980
1.870	0.003372	1.930	0.005482	1.990	0.008666
1.880	0.009003	1.940	0.010999	2.000	0.017657

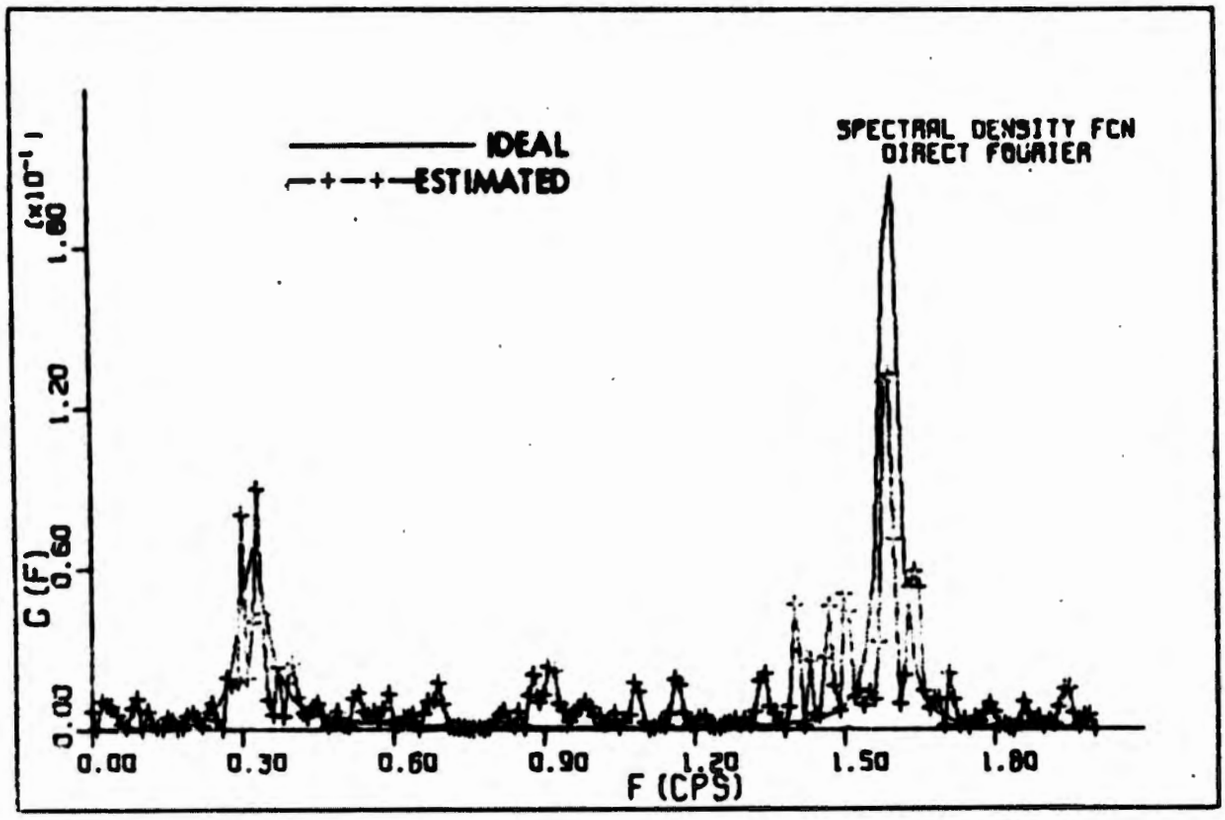


Fig. 38. Spectral density function estimates by direct Fourier analysis for an open loop fourth-order process example.

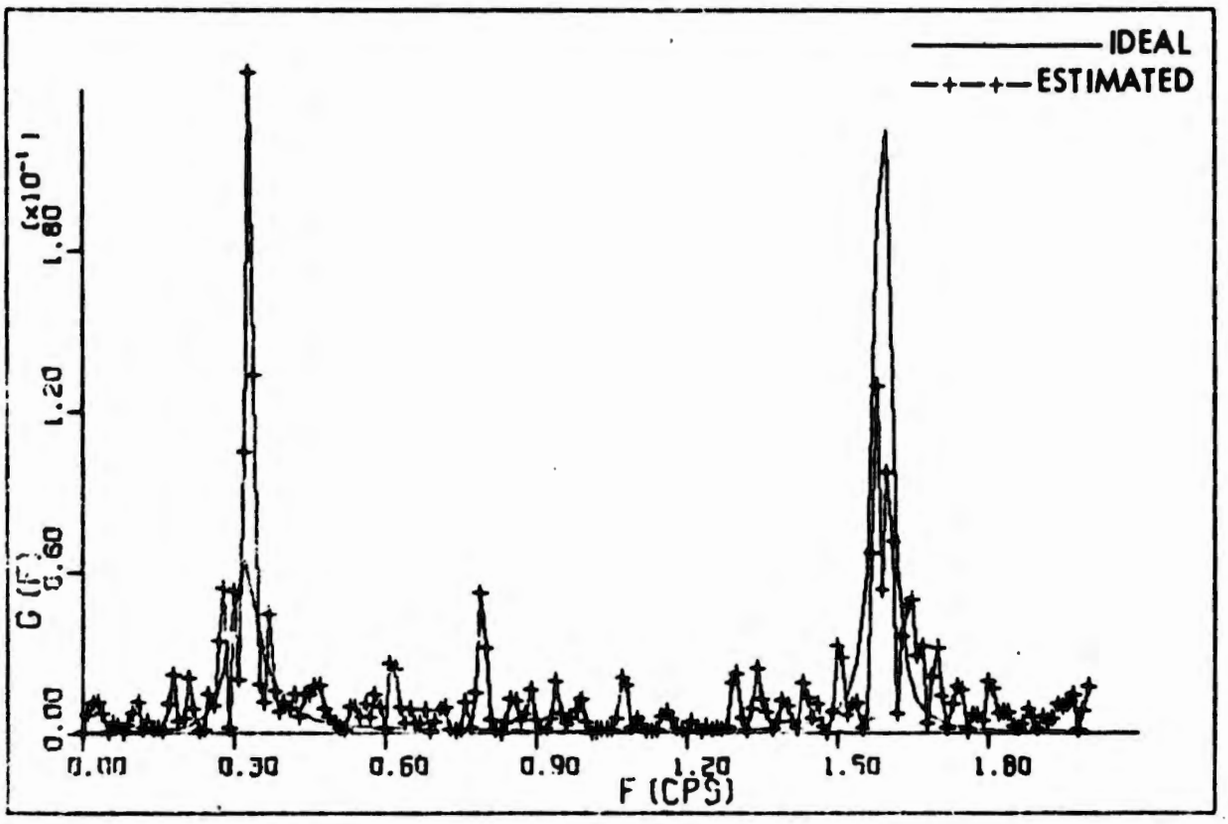


Fig. 39. Spectral density function estimates by direct Fourier analysis for a closed loop fourth-order process example.

Fast Fourier Transform

The Fast Fourier Transform is subject to the computer calculation of the equation

$$S(f) = \int_{-T}^{T} y(\tau) e^{-j2\pi f\tau} d\tau. \tag{77}$$

Consider the data window B(t) such that

$$B(t) = \Delta t \sum_{n=0}^{N-1} \delta(t - n\Delta t). \tag{78}$$

Then Eq. (77) can be written in discrete form as

$$S(f_n) = \Delta t \sum_{k=0}^{N-1} y(t_k) e^{-j \frac{2\pi nk}{N}} \tag{79}$$

where

Table 4. Spectral density function estimates by spectral filters for an open loop fourth-order process example.

f	G(f)	f	G(f)	f	G(f)
25.000	0.056040	0.806	0.004180	0.410	0.012664
12.500	0.131163	0.781	0.000359	0.403	0.017649
8.333	0.008610	0.758	0.001815	0.397	0.021272
6.250	0.001916	0.735	0.001303	0.391	0.022952
5.000	0.034319	0.714	0.000104	0.385	0.030987
4.167	0.010316	0.694	0.004876	0.379	0.031017
3.571	0.002172	0.676	0.002709	0.373	0.031909
3.125	0.002463	0.658	0.001643	0.368	0.030939
2.778	0.003969	0.641	0.008641	0.362	0.034999
2.500	0.072150	0.625	0.005371	0.357	0.035130
2.273	0.049052	0.610	0.004706	0.352	0.035372
2.083	0.010795	0.595	0.002277	0.347	0.038053
1.923	0.029464	0.581	0.005989	0.342	0.042178
1.786	0.119947	0.568	0.003207	0.338	0.042202
1.667	0.212800	0.556	0.005276	0.333	0.036230
1.563	0.197653	0.543	0.017171	0.329	0.034657
1.471	0.206010	0.532	0.021943	0.325	0.038239
1.389	0.081293	0.521	0.024990	0.321	0 035538
1.316	0.016396	0.510	0.007493	0.316	0.036909
1.250	0.000320	0.500	0.000524	0.313	0.040057
1.190	0.007605	0.490	0.004145	0.309	0.035377
1.136	0.000116	0.481	0.002006	0.305	0.036195
1.087	0.006239	0.472	0.001145	0.301	0.034199
1.042	0.013370	0.463	0.005220	0.275	0.005878
1.000	0.005745	0.455	0.007763	0.248	0.000909
0.962	0.004474	0.446	0.008025	0.225	0.002393
0.926	0.002919	0.439	0.004532	0.200	0.003008
0.893	0.003915	0.431	0.006368	0.175	0.001066
0.862	0.000249	0.424	0.007858	0.150	0.000868
0.833	0.003106	0.417	0.008940	0.125	0.001206

Table 5. Spectral density function estimates by spectral filters for a closed loop fourth-order process example.

f	G(f)	f	G(f)	f	G(f)
25.000	0.378459	0.806	0.002610	0.410	0.002009
12.500	0.133900	0.781	0.010443	0.403	0.003623
8.333	0.098150	0.758	0.007969	0.397	0.007745
6.250	0.051548	0.735	0.006265	0.391	0.014443
5.000	0.002587	0.714	0.014110	0.385	0.011847
4.167	0.022504	0.694	0.007921	0.379	0.016938
3.571	0.017990	0.676	0.017758	0.373	0.017540
3.125	0.020258	0.658	0.024124	0.368	0.018006
2.778	0.028018	0.641	0.013930	0.362	0.016324
2.500	0.045823	0.625	0.010421	0.357	0.024066
2.273	0.065849	0.610	0.016618	0.352	0.038036
2.083	0.022670	0.595	0.009520	0.347	0.044380
1.923	0.018173	0.581	0.003567	0.342	0.036048
1.786	0.072285	0.568	0.008415	0.338	0.035911
1.667	0.122789	0.556	0.001891	0.333	0.036888
1.563	0.128722	0.543	0.003259	0.329	0.038242
1.471	0.106168	0.532	0.0033269	0.325	0.046358
1.389	0.045389	0.521	0.032165	0.321	0.055591
1.316	0.000110	0.510	0.016215	0.316	0.061580
1.250	0.003556	0.500	0.007113	0.313	0.060819
1.190	0.026122	0.490	0.006994	0.309	0.056615
1.136	0.002853	0.481	0.003490	0.305	0.053357
1.087	0.005547	0.472	0.002728	0.301	0.039755
1.042	0.014275	0.463	0.003065	0.275	0.015529
1.000	0.004626	0.455	0.003941	0.248	0.007320
0.962	0.010822	0.446	0.002915	0.225	0.006822
0.926	0.003393	0.439	0.003972	0.200	0.000151
0.893	0.002790	0.431	0.004622	0.175	0.002745
0.862	0.000047	0.424	0.002095	0.150	0.002334
0.833	0.005362	0.417	0.005223	0.125	0.000906

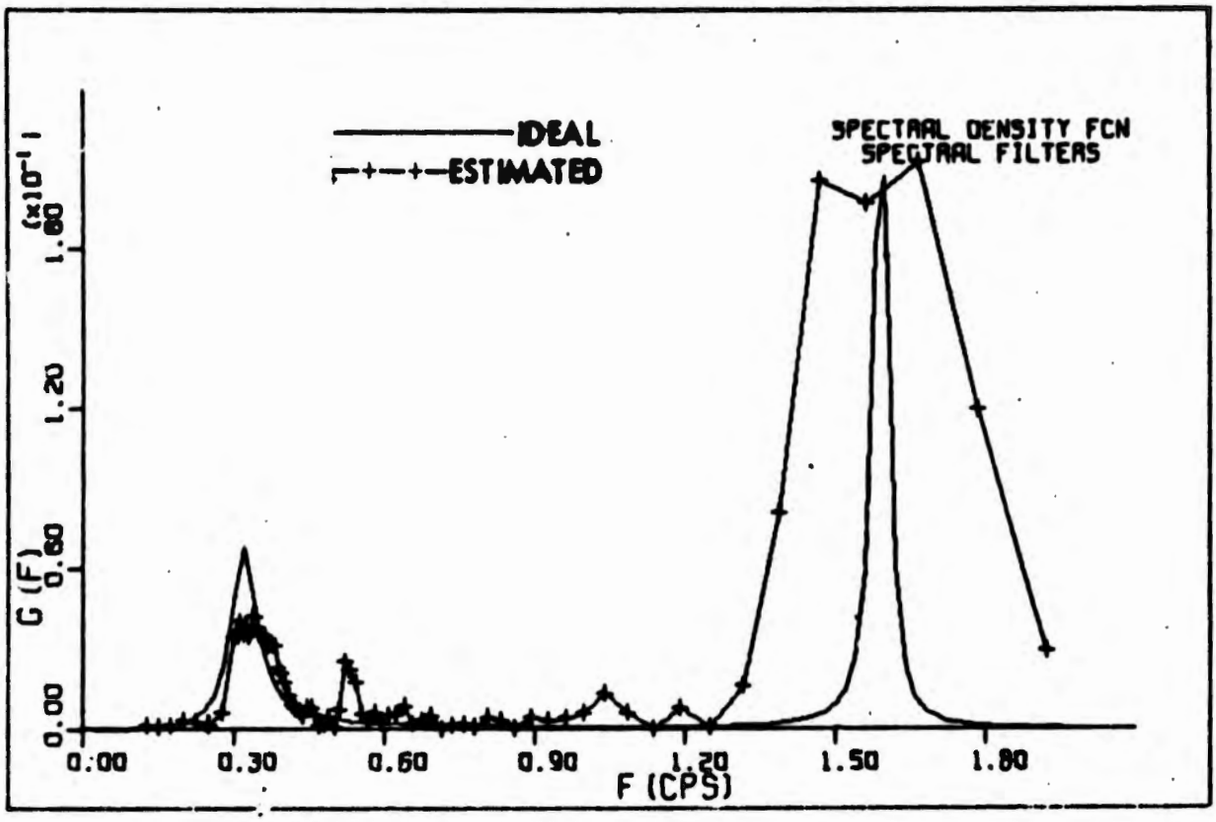


Fig. 40. Spectral density function estimates by spectral filters for an open loop fourth-order process example.

$$t_k = k\Delta t$$

$$f_n = n\Delta f \qquad n = 0, 1, \ldots, N-1 \tag{80}$$

$$\Delta t = T/N$$

$$\Delta f = 1/T$$

In matrix notation, we get

$$[S(N)]_{N\times 1} = [W^{nk}]_{N\times N}\,[Y(k)]_{N\times 1} \tag{81}$$

where

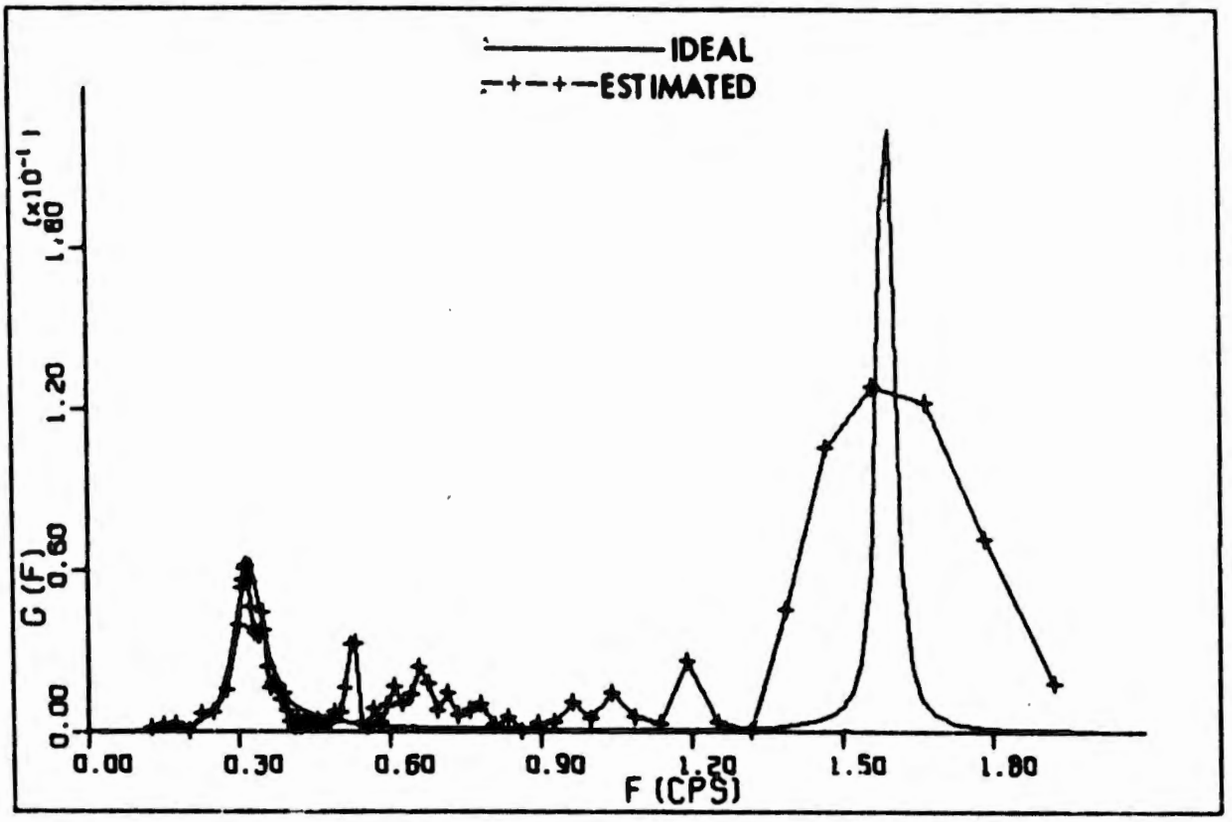


Fig. 41. Spectral density function estimates by spectral filters for a closed loop fourth-order process example.

$$[W^{nk}] = \begin{bmatrix} W^0 & W^0 & W^0 & \cdots & W^0 & \cdots & W^0 \\ W^0 & W^{1\cdot 1} & W^{1\cdot 2} & \cdots & W^{1\cdot k} & \cdots & W^{1\cdot (N-1)} \\ \cdot & \cdot & \cdot & & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot & & \cdot \\ W^0 & W^{(N-1)\cdot 1} & W^{(N-1)2} & \cdots & W^{(N-1)k} & \cdots & W^{(N-1)(N-1)} \end{bmatrix} \tag{82}$$

and

$$W = e^{-\frac{j2\pi}{N}} .$$

In general the execution of Eq. (81) requires N^2 complex multiplication and additions. The fast fourier transform owes its success to the reduction of the number of arithmetic operations. The Fast Fourier Transform requires only $N \log_2 N$ computations to perform the above operation[9]. Using the Fast Fourier Transform to estimate the spectral

density function of the output process gives the results shown in Tables 6 and 7 and Figs. 42 and 43 for the open and closed loop airframes respectively.

Table 6. Spectral density function estimates by Fast Fourier Transform for an open loop fourth-order process example.

f	G(f)	f	G(f)	f	G(f)
0.0	0.002929	0.684	0.020695	1.367	0.008309
0.024	0.011453	0.708	0.001327	1.392	0.026919
0.049	0.002462	0.732	0.000323	1.416	0.003775
0.073	0.001321	0.757	0.000505	1.440	0.006203
0.098	0.002250	0.781	0.000301	1.465	0.030998
0.122	0.004421	0.806	0.005761	1.489	0.002044
0.146	0.004629	0.830	0.001515	1.514	0.041187
0.171	0.000448	0.854	0.004036	1.538	0.015657
0.195	0.009696	0.879	0.018853	1.563	0.013470
0.220	0.003809	0.903	0.032208	1.587	0.202620
0.244	0.001600	0.928	0.007140	1.611	0.002833
0.269	0.014396	0.952	0.007588	1.636	0.098040
0.293	0.023404	0.977	0.009984	1.660	0.023676
0.317	0.021845	1.001	0.005462	1.685	0.009273
0.342	0.069221	1.025	0.002990	1.709	0.018522
0.366	0.027854	1.050	0.000339	1.733	0.002537
0.391	0.023577	1.074	0.006584	1.758	0.004833
0.415	0.007276	1.099	0.007043	1.782	0.009753
0.439	0.009635	1.123	0.001853	1.807	0.001214
0.464	0.003138	1.147	0.003549	1.831	0.000151
0.488	0.005050	1.172	0.016744	1.855	0.005266
0.513	0.003162	1.196	0.003573	1.880	0.004059
0.537	0.013481	1.221	0.003525	1.904	0.004131
0.562	0.004989	1.245	0.000246	1.929	0.005052
0.586	0.009998	1.270	0.004331	1.953	0.016249
0.610	0.000954	1.294	0.001308	1.978	0.011717
0.635	0.007247	1.318	0.006113	2.002	0.003148
0.659	0.003706	1.343	0.017419		

Table 7. Spectral density function estimates by Fast Fourier Transform for a closed loop fourth-order process example.

f	G(f)	f	G(f)	f	G(f)
0.0	0.001596	0.684	0.000158	1.367	0.001617
0.024	0.000774	0.708	0.006698	1.392	0.009129
0.049	0.005897	0.732	0.000588	1.416	0.003667
0.073	0.001074	0.757	0.003966	1.440	0.010331
0.098	0.010840	0.781	0.029839	1.465	0.001469
0.122	0.004532	0.806	0.006731	1.489	0.030289
0.146	0.000695	0.830	0.000428	1.514	0.002610
0.171	0.001385	0.854	0.015252	1.538	0.029583
0.195	0.019307	0.879	0.003030	1.563	0.047909
0.220	0.013104	0.903	0.002829	1.587	0.110179
0.244	0.003750	0.928	0.002611	1.611	0.005328
0.269	0.009244	0.952	0.006579	1.636	0.059875
0.293	0.077625	0.977	0.000218	1.660	0.028473
0.317	0.026685	1.001	0.016722	1.685	0.003633
0.342	0.205105	1.025	0.003843	1.709	0.008026
0.366	0.029447	1.050	0.008583	1.733	0.000401
0.391	0.003183	1.074	0.004714	1.758	0.017935
0.415	0.035762	1.099	0.003117	1.782	0.019684
0.439	0.008750	1.123	0.000467	1.807	0.014191
0.464	0.028410	1.147	0.003806	1.831	0.019766
0.488	0.006579	1.172	0.001662	1.855	0.001357
0.513	0.002448	1.196	0.004494	1.880	0.008671
0.537	0.007701	1.221	0.002974	1.904	0.002295
0.562	0.003054	1.245	0.000921	1.929	0.005907
0.586	0.013829	1.270	0.020036	1.953	0.015363
0.610	0.000813	1.294	0.008118	1.978	0.002515
0.635	0.001989	1.318	0.002360	2.002	0.011991
0.659	0.000792	1.343	0.008729		

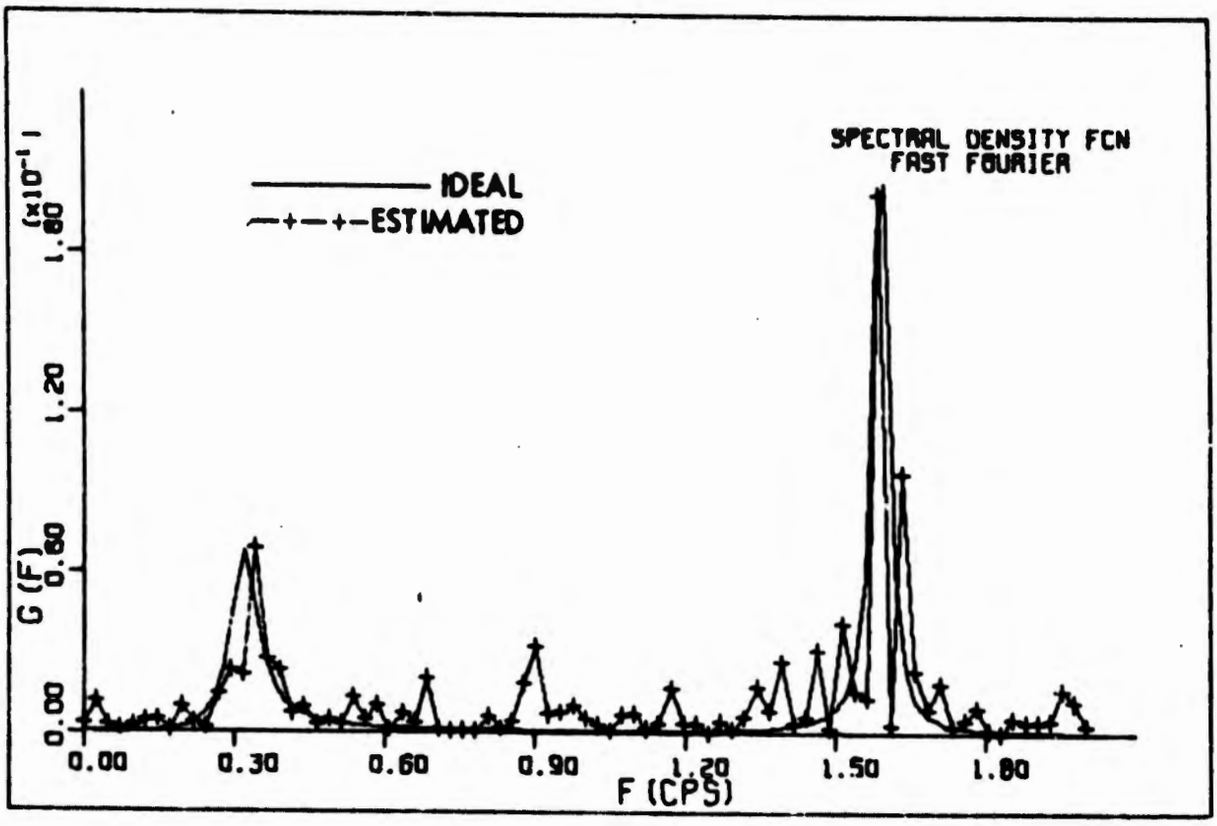


Fig. 42. Spectral density function estimates by Fast Fourier Transform for an open loop fourth-order process example.

Determination of Bending Mode

If a priori information is known that characterizes a bending mode to be in a restricted band, the spectral density function of the output process would be estimated at several frequencies as shown in Fig. 35 by either direct Fourier analysis, the spectral filter method, or the Fast Fourier Transform. The peak values of the spectral density function could then be determined by a suitable data processor that selected the peaks values of the generated spectral density function. For the given flight condition, the spectral density estimates were at 1.59 Hz and 1.58 Hz (open loop and closed loop) for the direct Fourier analysis; at 1.567 Hz and 1.563 Hz for the spectral filters method; and 1.587 Hz

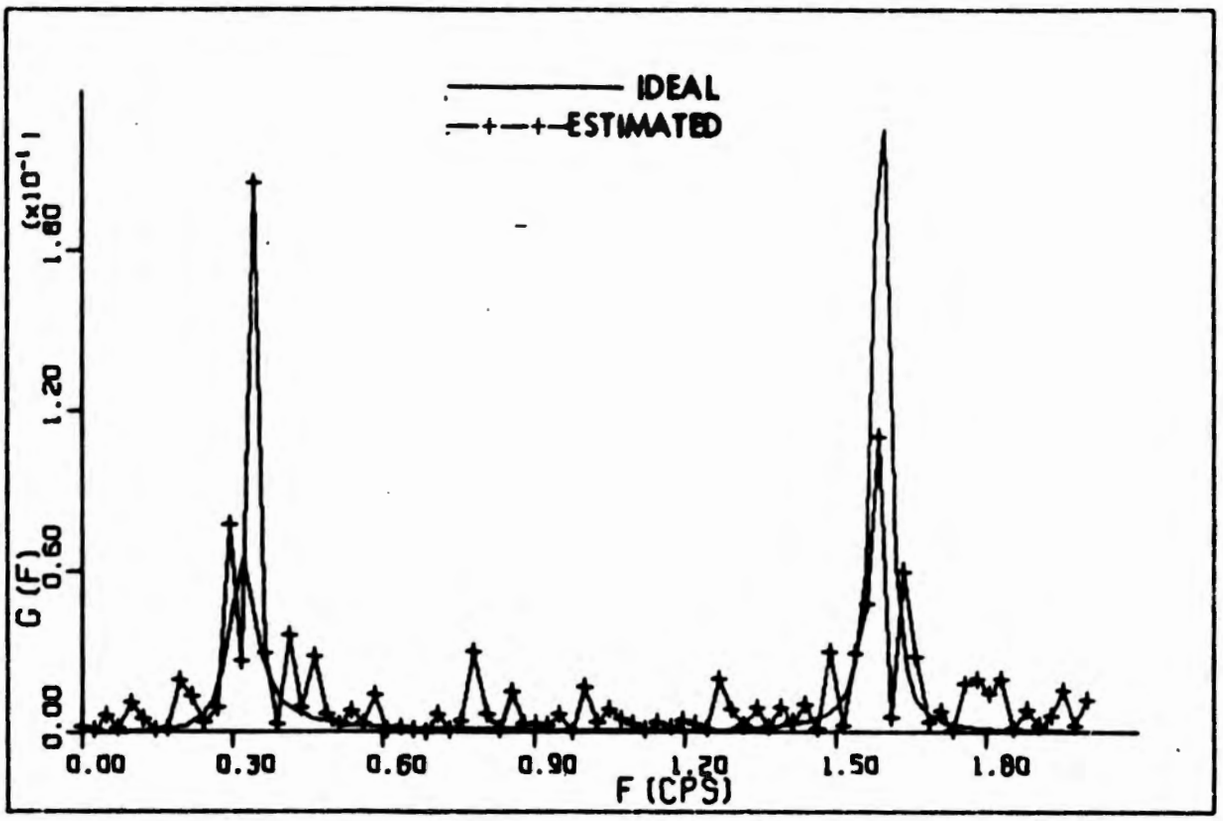


Fig. 43. Spectral density function estimates by Fast Fourier Transform for a closed loop fourth-order process example.

and 1.587 Hz for the Fast Fourier Transform method. In all three cases the bending mode estimate and in good agreement with the actual bending mode frequency at 1.59 Hz.

Conclusions

Both the adaptive digital frequency discriminator and spectral density identification methods provide suitable means of estimating the bending modes of the flexible airframe. The use of the adaptive digital frequency discriminator requires a minimum of special purpose digital computer hardware while the spectral density methods require a rather sophisticated digital computer. The possibility of implementing an

adaptive control system so that both the sampling rate for the adaptive compensator and the adaptive digital frequency discriminator are identical would provide an extremely neat package of flight hardware.

PART III. SIMULATION RESULTS OF THE ADAPTIVE CONTROL SCHEME

Introduction

Using an IBM 360 computer a digital simulation was made of the flexible airframe as well as the corresponding digital compensator; that is, a closed loop adaptive control system was simulated. Adaptation through variation of the sampling period T was made by identifying the first bending mode using the Fast Fourier Transform technique discussed in Part II, and using this information to set or "tune" the sampling period T to provide proper compensation. The block diagram of the control loop simulated is shown in Fig. 44.

Reiteration of the adaptive control scheme given in Part I is in order at this point. A control system is needed for a flexible airframe

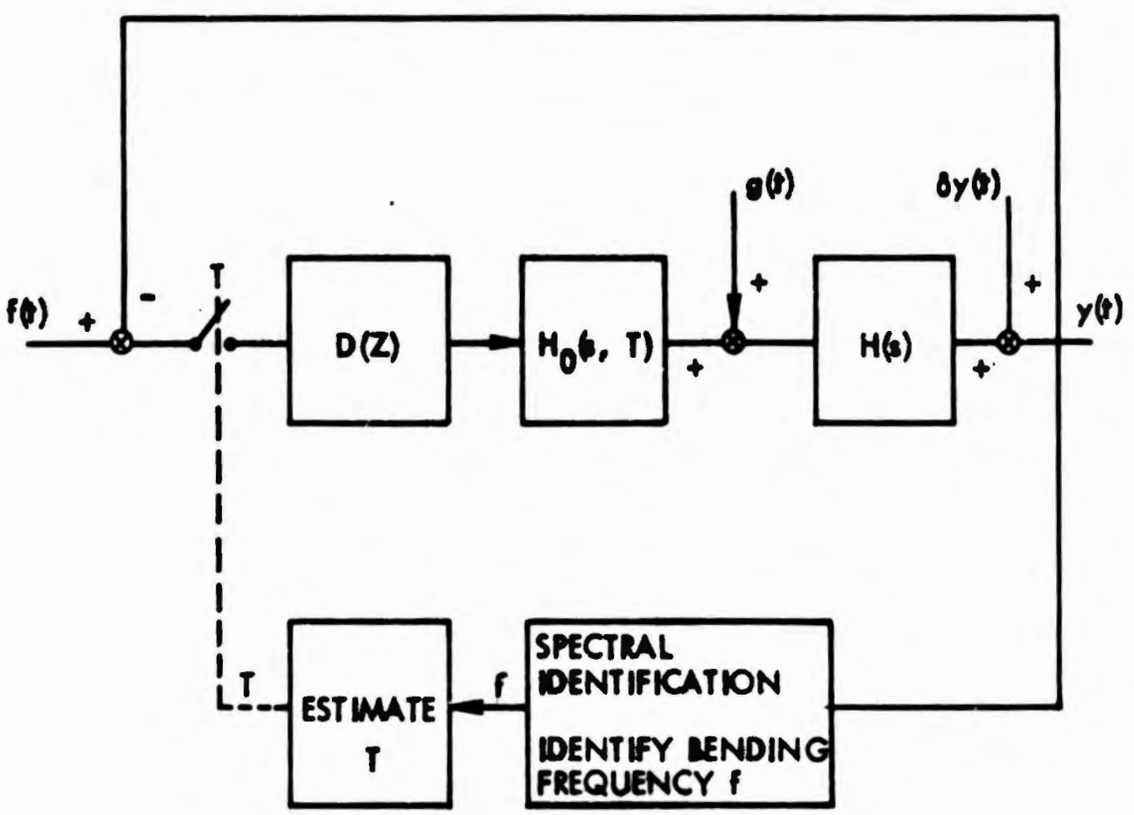


Fig. 44. Sampled data adaptive control system.

where first bending mode is time varying from 1.17 to 2.07 Hz. A digital compensator,

$$D(z) = \frac{(z - 0.9)(z - 0.8859 + j0.4536)}{(z - 1.0)(z + 0.4)(z + 0.6)} , \qquad (83)$$

was chosen to provide suitable system compensation for flight condition 2. By properly varying the sampling period as a function of the first bending mode frequency the closed loop system was shown to be stable over the flight profile the airframe experiences.

In Part III we will show by means of a digital computer simulation the spectral response of the compensated airframe process for the three flight conditions of Part I. It will be shown that by appropriately changing the sampling period T for each flight condition the resultant spectral density of the process remains essentially constant.

An Equivalent Continuous Compensator

Prior to discussion of the digital computer simulation an equivalent continuous compensator that represents the digital compensator given in Eq. (83) will be derived. This will allow the generation of a suitable airframe process that can be processed to determine the first bending mode's frequency. Figure 45 represents the digital compensator D(z) and the zero-order hold that follows it. Now the digital compensator can be represented as

$$D(z) = \frac{(1 + \alpha_1 z^{-1})(1 + \alpha_2 z^{-1})(1 + \alpha_3 z^{-1})}{(1 + \beta_1 z^{-1})(1 + \beta_2 z^{-1})(1 + \beta_3 z^{-1})} \qquad (84)$$

where

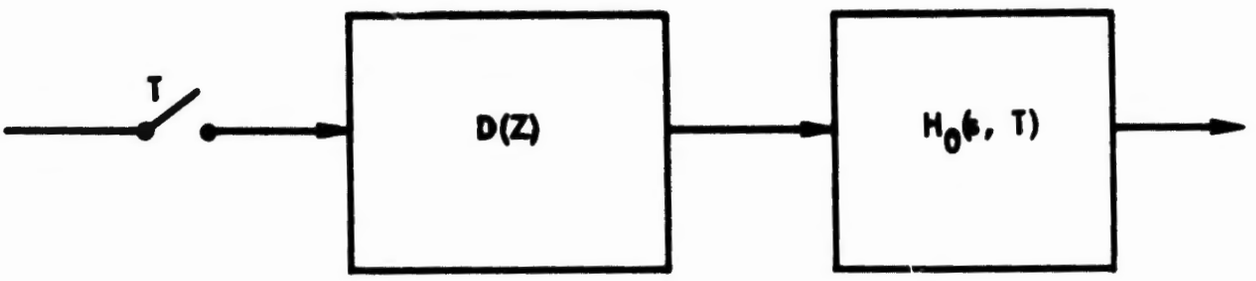


Fig. 45. Compensator and zero-order hold.

$\alpha_1 = (-0.9, 0.0)$ $\beta_1 = (-1.0, 0.0)$

$\alpha_2 = (-0.8859, 0.4536)$ $\beta_2 = (0.4, 0.0)$

$\alpha_3 = (-0.8859, -0.4536)$ $\beta_3 = (0.6, 0.0)$

Replacing z^{-1} by e^{-sT} in Eq. (84) we have

$$D(s, T) = \frac{(1 + \alpha_1 e^{-sT})(1 + \alpha_2 e^{-sT})(1 + \alpha_3 e^{-sT})}{(1 + \beta_1 e^{-sT})(1 + \beta_2 e^{-sT})(1 + \beta_3 e^{-sT})} .$$

The zero-order hold has the transfer function

$$H_0(s, T) = \frac{1 - e^{-sT}}{s} .$$

Therefore, the equivalent continuous transfer function representing the zero-order hold and the digital filter is given by

$$D_1(s, T) = \frac{(e^{sT} + \alpha_1)(e^{sT} + \alpha_2)(e^{sT} + \alpha_3)}{se^{sT}(e^{sT} + \beta_2)(e^{sT} + \beta_3)} .$$

Now $D_1(s, T)$ can be approximated by

$$D_A(s, T) = \frac{G_D(s + \alpha_4)(s + \alpha_5)(s + \alpha_6)}{(s + \beta_4)(s + \beta_5)(s + \beta_6)}$$

where

$$\alpha_4 = \left(-\frac{1}{T}\log a_1,\ 0.0\right) \qquad \beta_4 = (0.0,\ 0.0)$$

$$\alpha_5 = \left(-\frac{1}{T}\log\sqrt{a_4^2 + a_5^2},\ \frac{1}{T}\tan^{-1}\frac{a_5}{a_4}\right) \qquad \beta_5 = \left(-\frac{1}{T}\log a_3,\ \frac{\pi}{T}\right)$$

$$\alpha_6 = \left(-\frac{1}{T}\log\sqrt{a_4^2 + a_5^2},\ -\frac{1}{T}\tan^{-1}\frac{a_5}{a_4}\right) \qquad \beta_6 = \left(-\frac{1}{T}\log a_6,\ \frac{\pi}{T}\right)$$

and

$$a_1 = 0.9$$

$$a_2 = 0.6$$

$$a_3 = 0.4$$

$$a_4 = 0.45360$$

$$a_5 = 0.88590$$

The zeros of $D_1(s, T)$ are α_4, α_5, and α_6 while the poles of $D_1(s, T)$ are β_4, β_5, and β_6.

The unit white noise spectral responses of the compensator $D_1(s, T)$ and the approximated compensator $D_A(s, T)$ are compared in Figs. 46, 47, and 48 for the appropriate sampling periods chosen to provide proper compensation to the airframe for the three specified flight conditions.

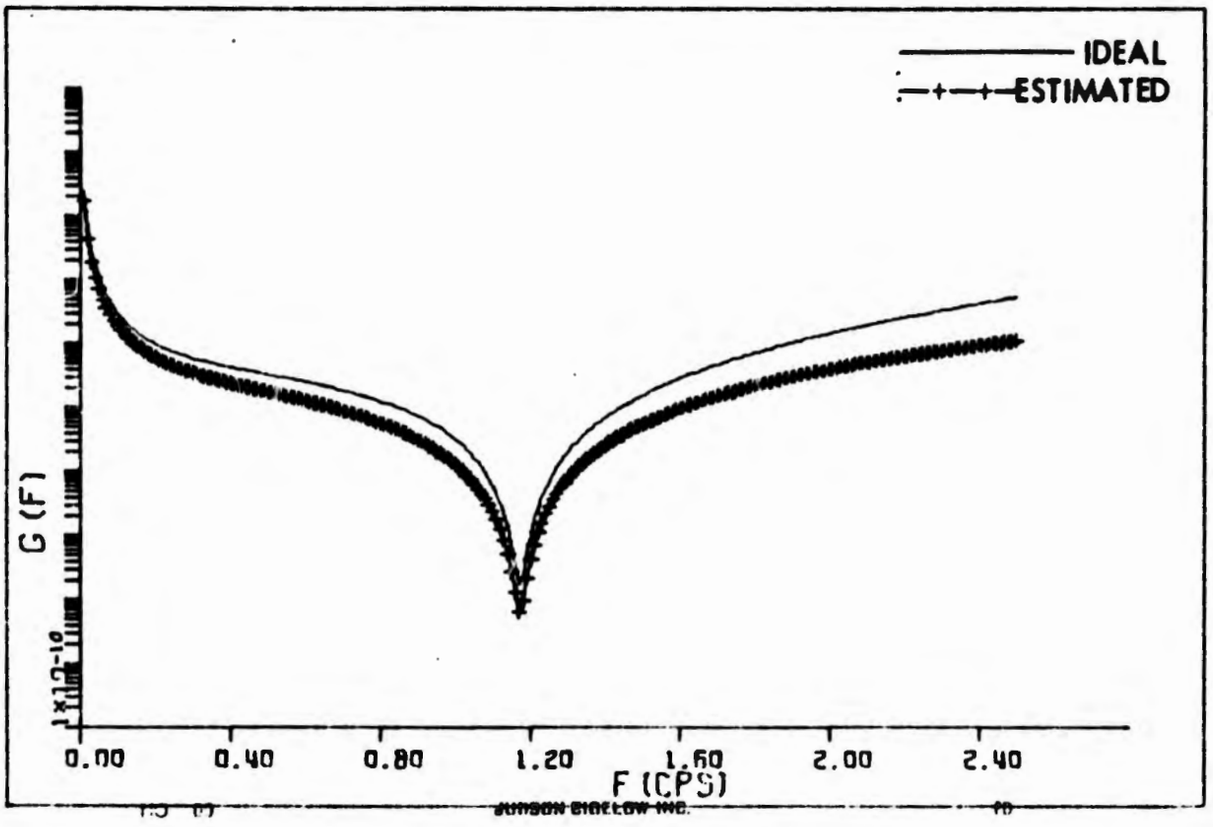


Fig. 46. Unit white noise spectral response approximation of the compensator when T = 0.064272 sec.

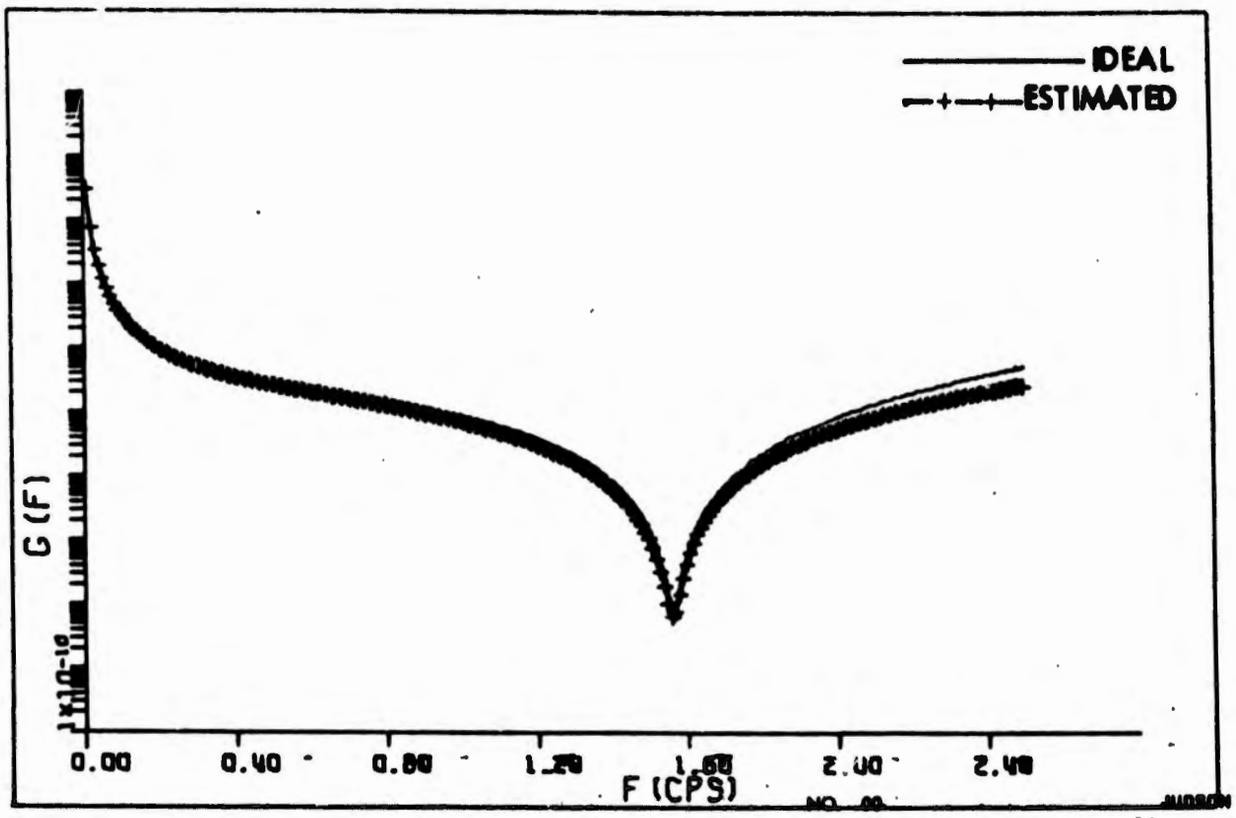


Fig. 47. Unit white noise spectral response approximation of the compensator when T = 0.048204 sec.

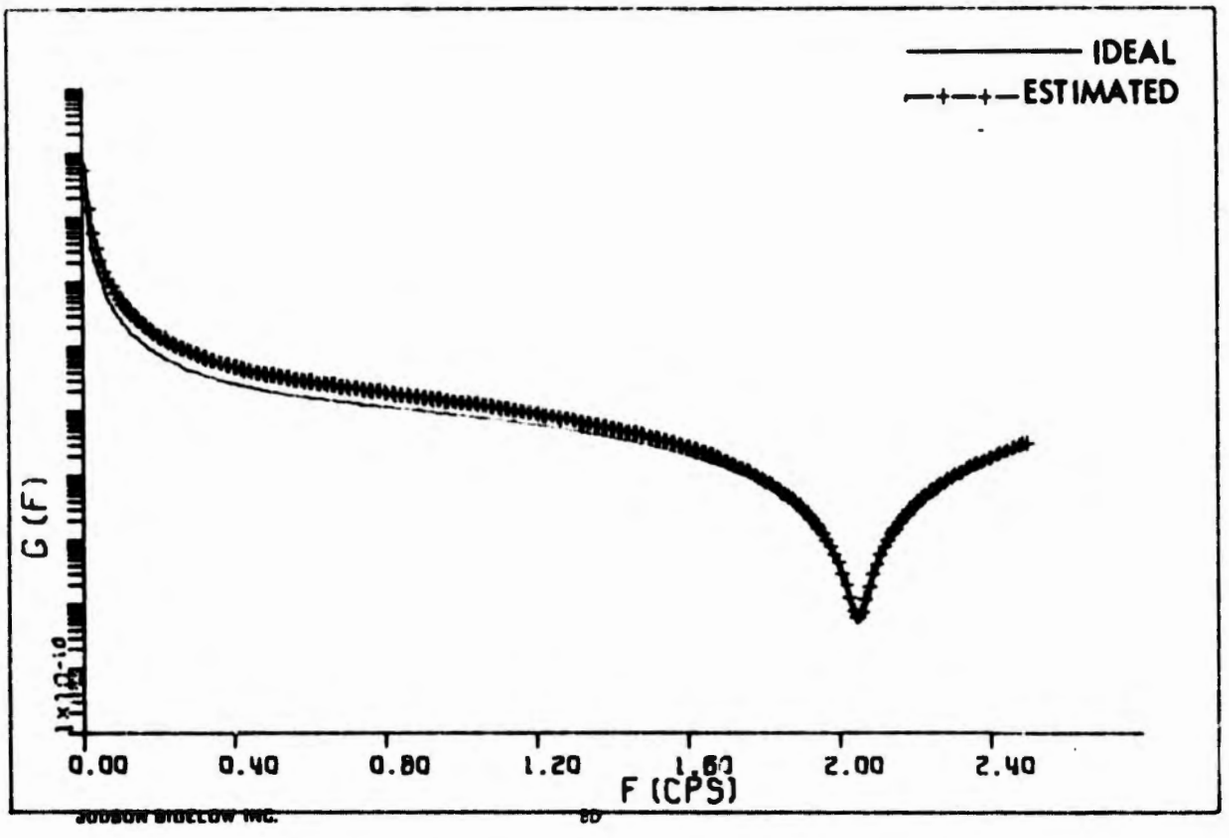


Fig. 48. Unit white noise spectral response approximation of the compensator when T = 0.036727 sec.

Digital Simulation of the Adaptive Control System

The overall features of the simulated spectral identification adaptive control system are summarized as follows:

1. The spectral identification system identifies the desired bending mode frequency, $f_{n\ell}$.
2. For a fixed digital compensator D(z) the sampling period T is estimated by the relationship

$$T = \frac{1}{2\pi f_{n\ell}} \left\{ \left[\log \left(a_4^2 + a_5^2 \right)^{1/2} \right]^2 + \left(\tan^{-1} \frac{a_5}{a_4} \right)^2 \right\}.$$

A block diagram of the simulated adaptive control system is shown in Fig. 49. In the digital simulation the digital compensation D(z)

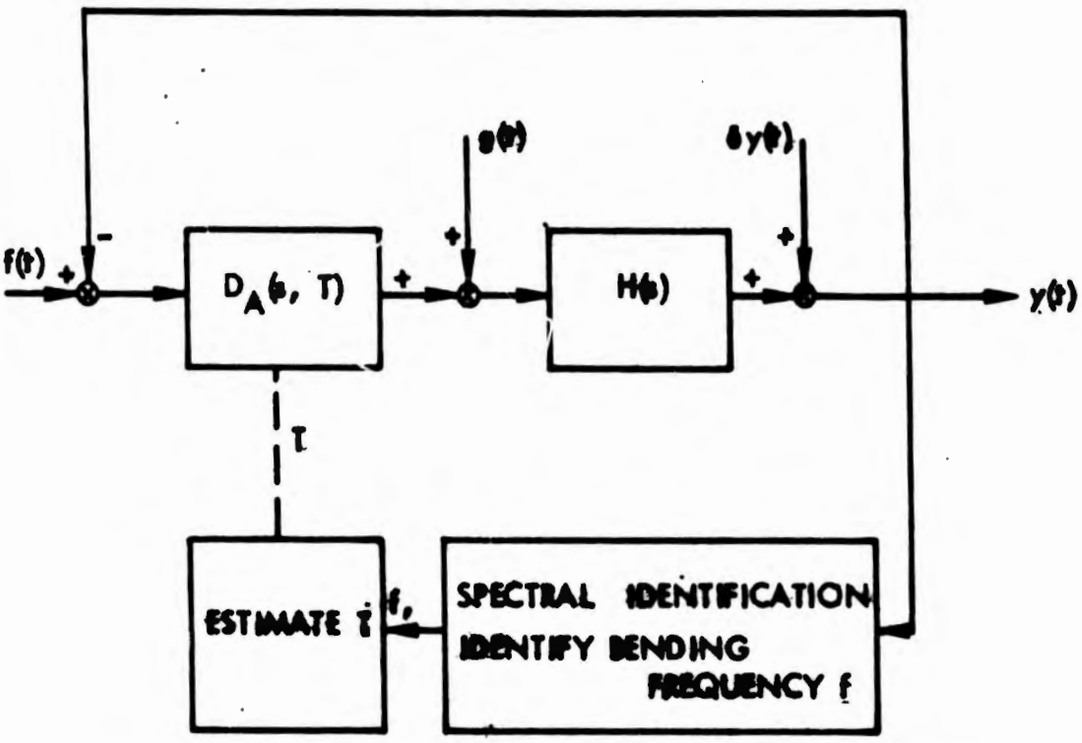


Fig. 49. Simulated spectral adaptive control system.

was approximated by the continuous compensation $D_A(s, T)$. The corresponding computer program is given in Appendix D. The input control f(t) is assumed to be zero and the disturbance and measurement noise are considered white.

For purposes of evaluation the spectral density function of the airframe process without a compensator (open loop) is compared to the process spectral density function in the closed loop mode which includes the digital compensator for the three flight conditions specified. Figures 50, 51 and 52 are the spectral density functions of the uncompensated airframe process, while Figs. 53, 54 and 55 are the spectral density functions of the compensated airframe process.

The identified bending mode frequencies in the digital simulation are

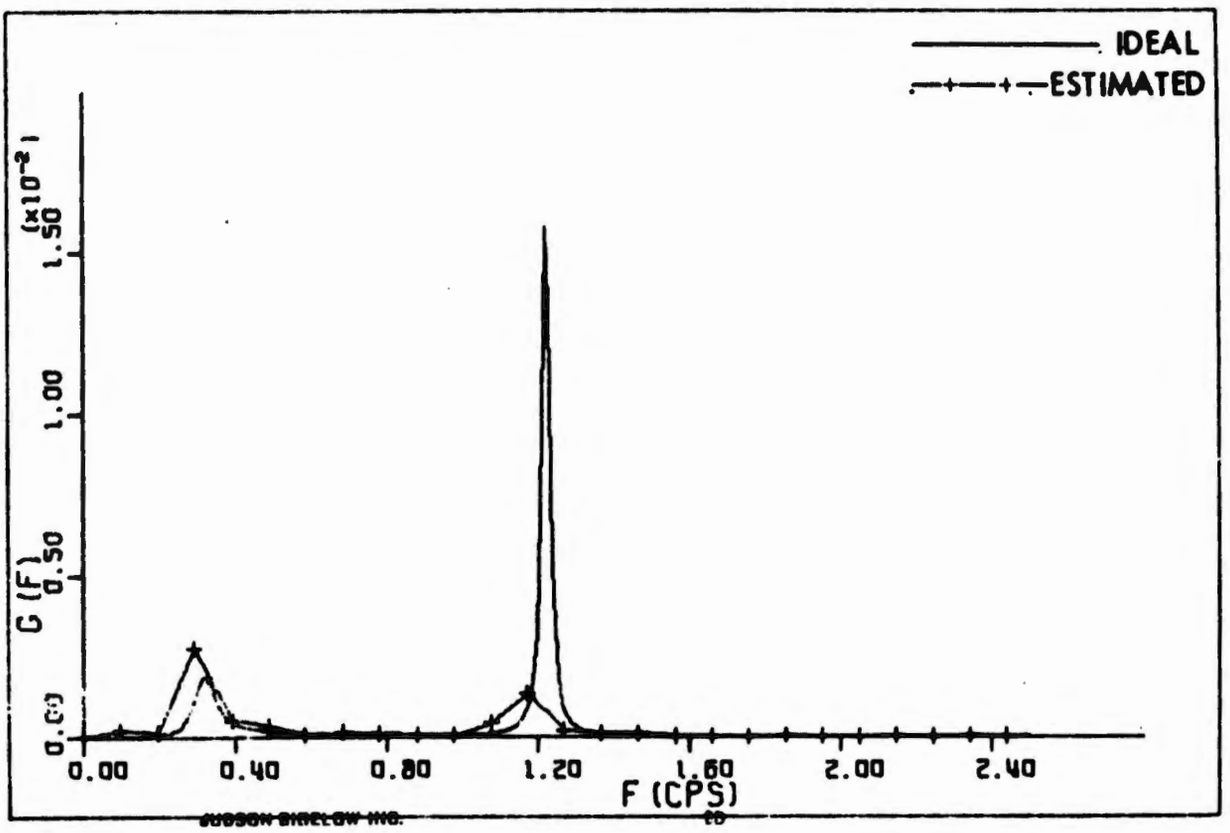


Fig. 50. Spectral identification of the flexible airframe at flight condition 1.

$f_{n\ell}$ = 1.171875 Hz at flight condition 1,

$f_{n\ell}$ = 1.562500 Hz at flight condition 2,

$f_{n\ell}$ = 2.050781 Hz at flight condition 3.

Although the identified bending mode frequencies are not exact, they are still effective in choosing the sampling period T. The corresponding sampling rates are

T_1 = 0.064272 sec at flight condition 1,

T_2 = 0.048204 sec at flight condition 2,

T_3 = 0.036727 sec at flight condition 3.

From Figs. 52, 53, and 54 we note that the first bending mode is completely compensated by merely varying the sampling period T.

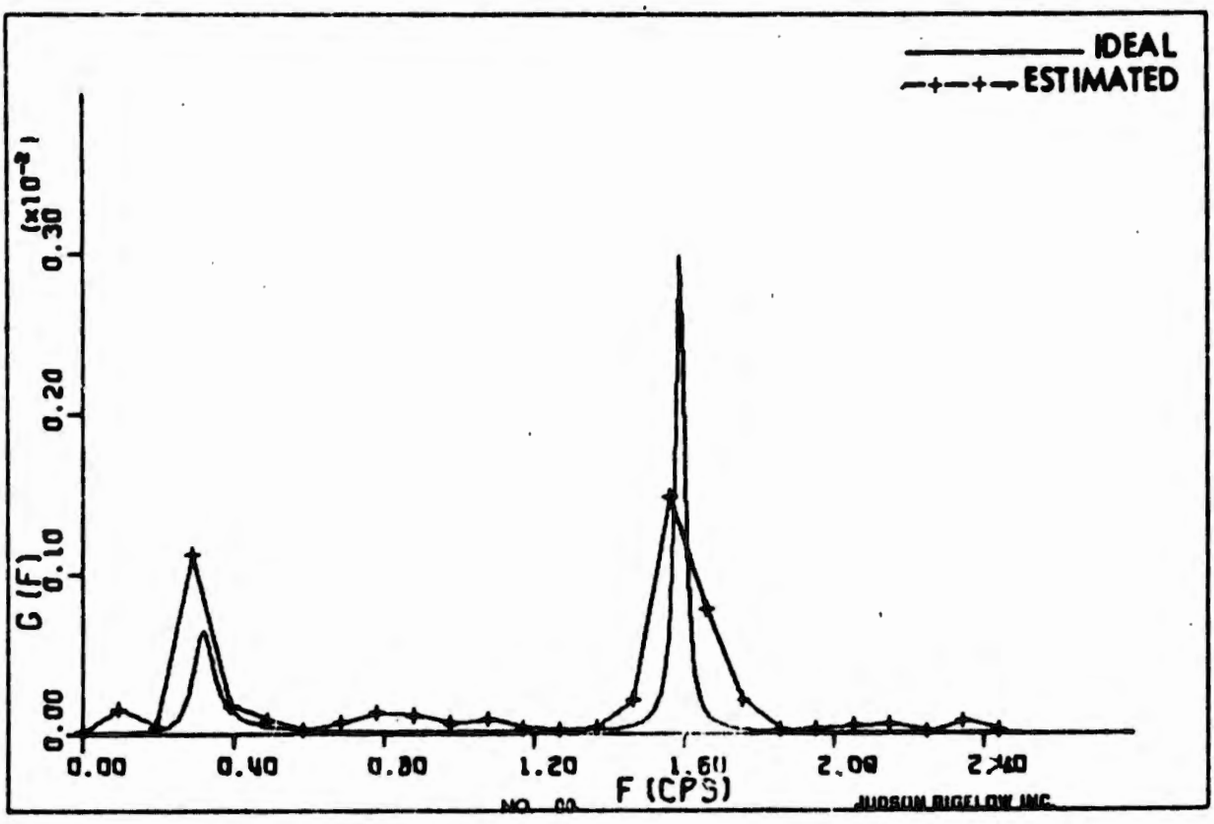


Fig. 51. Spectral identification of the flexible airframe at flight condition 2.

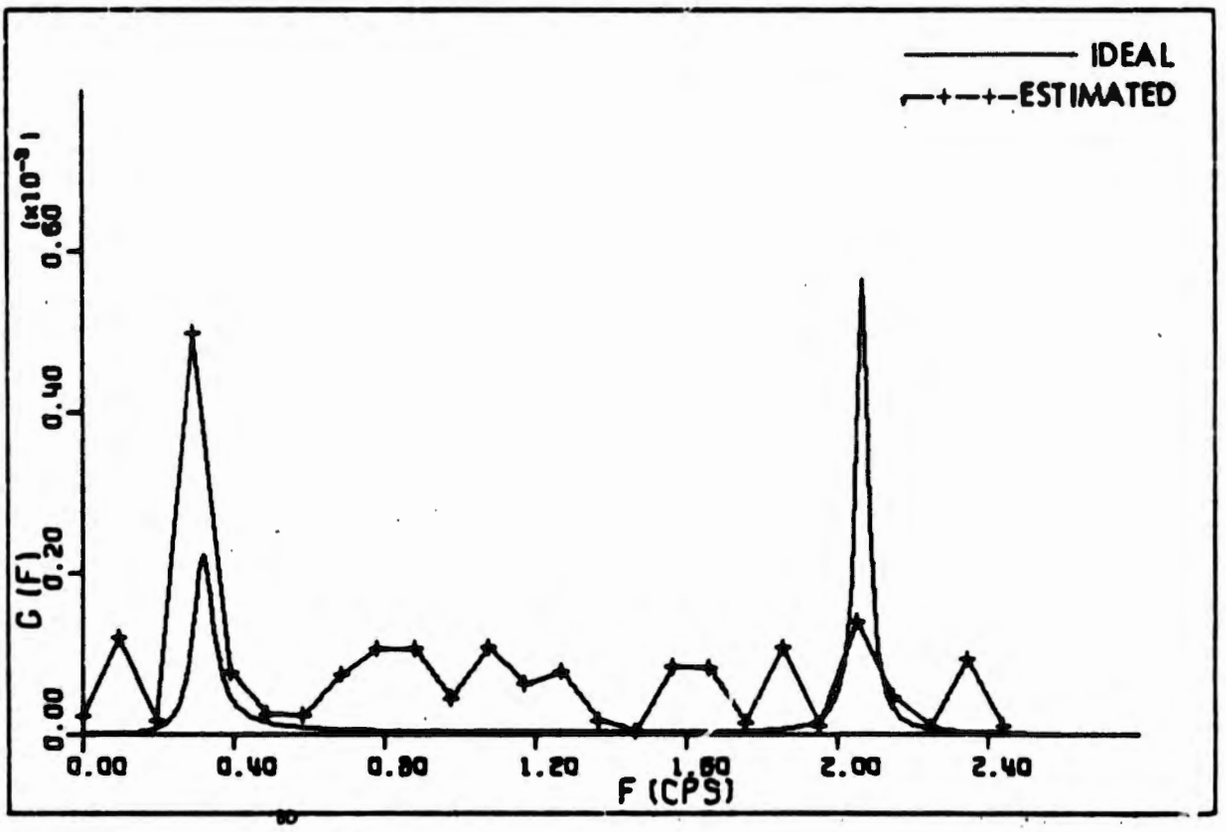


Fig. 52. Spectral identification of the flexible airframe at flight condition 3.

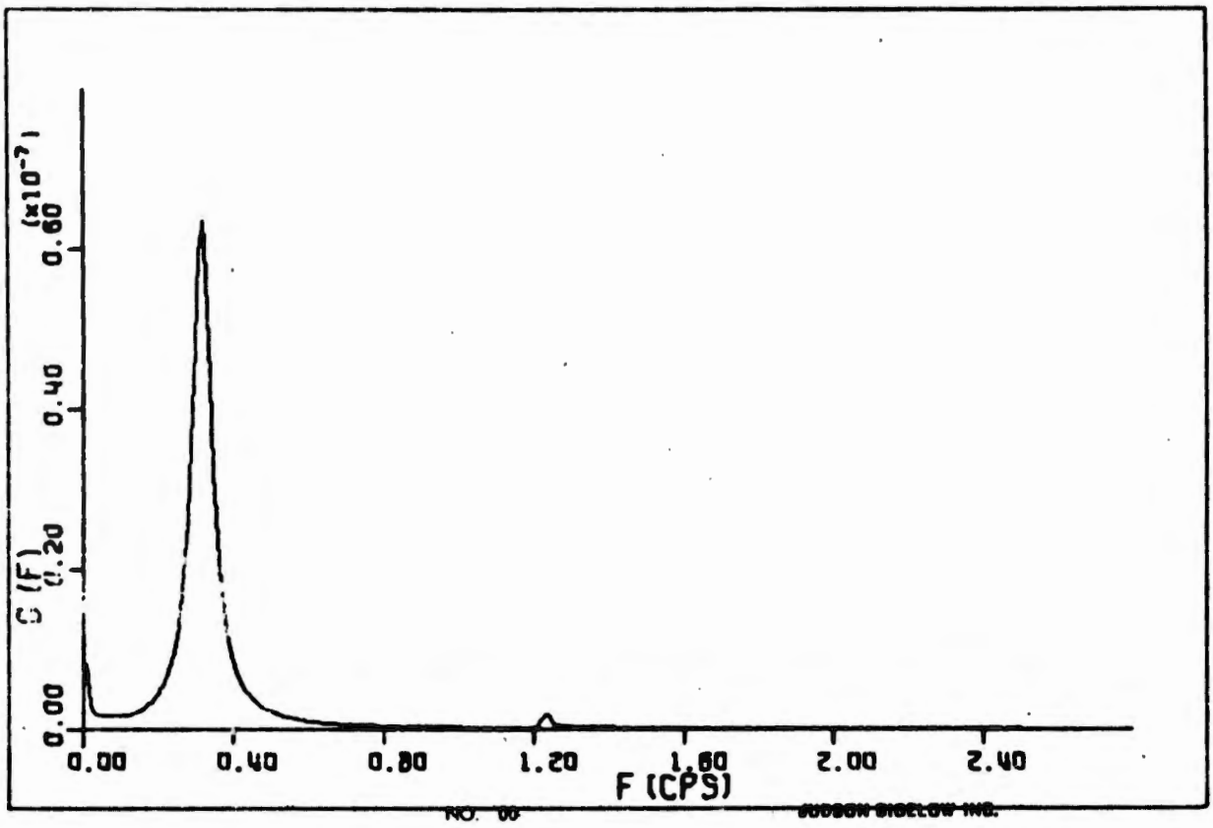


Fig. 53. Spectral identification with a compensator at flight condition 1.

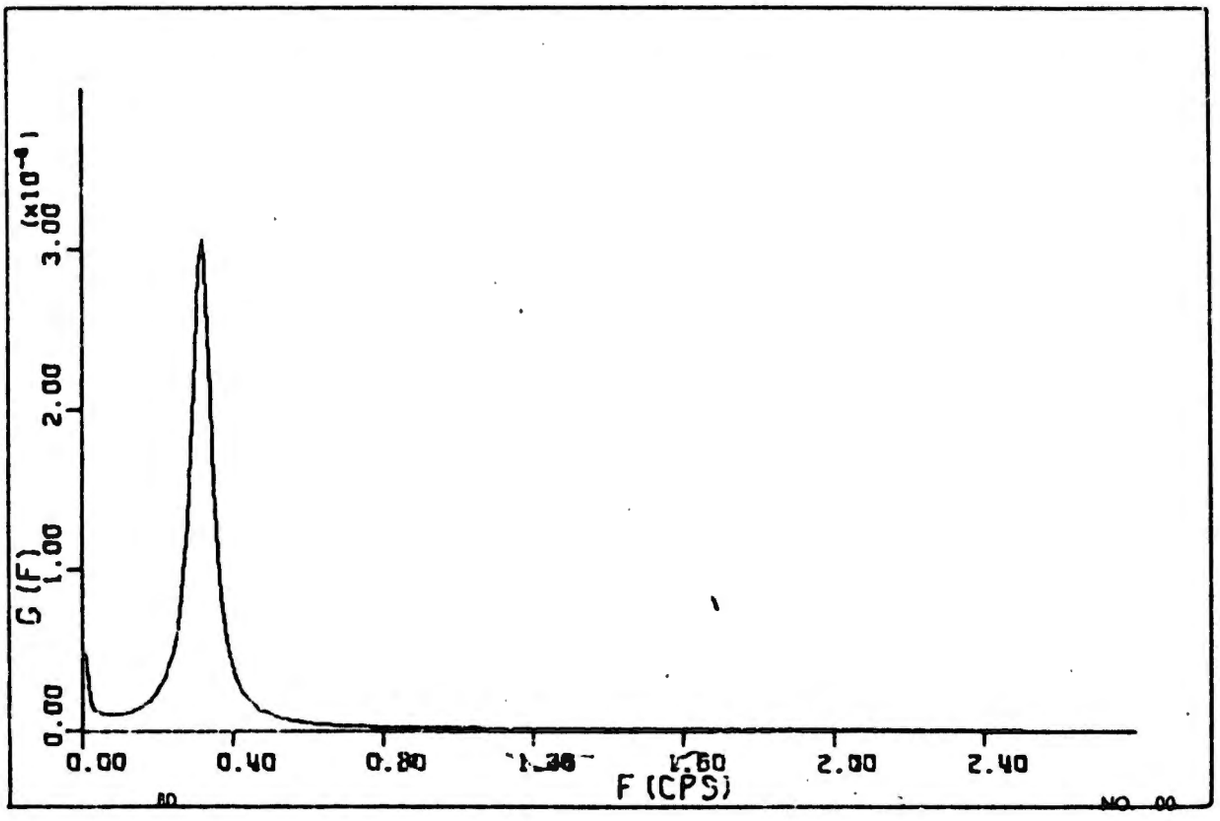


Fig. 54. Spectral identification with a compensator at flight condition 2.

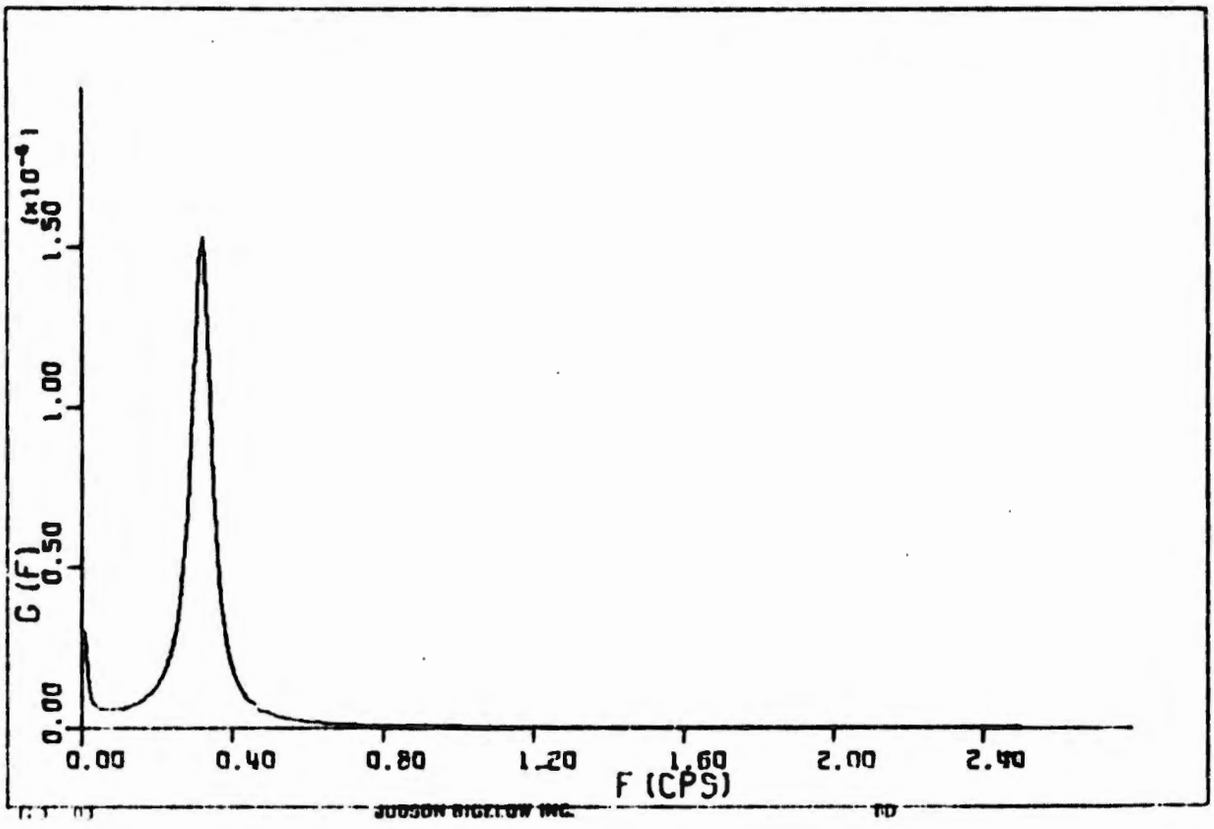


Fig. 55. Spectral identification with a compensator at flight condition 3.

Conclusions

The adaptive control of a flexible airframe system has been demonstrated utilizing a spectral identification for bending mode frequency identification. When a digital compensator is designed to compensate the bending modes, it is sufficient to vary the sampling rate in order to attain stable performance of the flexible airframe.

In the system simulation, the digital compensator is approximated by a continuous compensator. The unit white noise spectral response characteristic shows that the approximated transfer function is very similar to that of the original compensator. The total system simulation consists of a flexible airframe with bending mode, an approximated

compensator, and a closed feedback loop. The spectral identification system estimates the spectral density function of the output measurement by the Fast Fourier Transform method and identifies the first bending mode's frequency. The sampling rate is tuned according to this identified frequency to compensate the first bending mode.

In summary, the spectral identification adaptive control system offers a successful bending mode control.

ACKNOWLEDGMENTS

The authors would like to express their gratitude to the Office of Naval Research for its support of this work through the funds allocated to Project Themis under contract N00014-68-A-0162. A significant debt of gratitude is due to our colleagues who worked on this project. We wish to acknowledge the work of P. J. Hermann, Lorys Ascher, and R. P. Bhatia. In particular we would like to acknowledge the work of Mr. G. Y. Oak in the area of bending mode identification. Finally we would like to thank Dr. R. G. Brown, our program manager, for his guidance and suggestions in the preparation of this report.

REFERENCES

1. J. J. Bongiorno, Jr., "Adaptive Control Systems," in Eli Mishkin and Ludwig Braun, Jr. Adaptive Control Systems, Chapter 10, pp. 323-342, McGraw-Hill Book Co., New York (1961).

2. R. M. Willett, "Sampled-Data Control Utilizing Variable Sampling Rate," Unpublished PhD Thesis, Iowa State University (1966).

3. Lorys Ascher, "Analysis of Adaptive Sampled-Data Control Utilizing Variations of Sampling Rate by Means of Digital Simulation," Unpublished MS Thesis, Iowa State University (1969).

4. J. H. Wykes and J. H. Mori, "An Analysis of Flexible Aircraft Structural Mode Control — Part 1: Unclassified Data," Air Force Flight Dynamics Laboratory, Research and Technology Division, Air Force System Command, Wright-Patterson Air Force Base Report AFFDL-TR-65-190 Part 1 (1966).

5. P. M. DeRusso, R. J. Roy, and C. M. Close, State Variables for Engineers, John Wiley and Sons, Inc., New York (1967).

6. C. S. Weaver, J. von der Groeben, P. E. Mantey, J. G. Toole, C. A. Cole, J. W. Fitzgerald, and R. W. Lawrence, "Digital Filtering with Applications to Electrocardiogram Processing," IEEE Trans. on Audio and Electroacoustics, AU-16: 350-391 (1968).

7. W. B. Davenport and W. G. Root, "An Introduction to the Theory of Random Signals and Noise," McGraw-Hill, New York (1958).

8. G. Y. Oak, "A Simple Method of Estimating Spectral Density Functions from Experimental Data," MS Thesis, Iowa State University (1969).

9. E. O. Brighan and R. E. Morrow, "The Fast Fourier Transform," IEEE Spectrum, pp. 63-70 (1967).

APPENDIX A. METHOD OF SIMULATING DIGITAL FILTERS AND SAMPLED-DATA CONTROLLERS WITH ANALOG COMPONENTS

The basic device used is the track/store amplifier (also called a sample/hold unit) which is a standard component of the EAI 8812 analog computer. This track/store amplifier is driven by analog input signal and will be in the track mode (i.e., the output is the negative sum of the input signals) when a logic "1" is applied to the appropriate terminal on the logic patch panel. When a logic "0" appears at that terminal, the amplifier switches to the store mode and holds its output at whatever value it had when the switching occurred. Thus, a ramp function could be converted to a staircase function if very narrow logic pulses were used to drive the track/store amplifier. Figure A-1 shows the relationship between the input and output of a track/store amplifier and the logic pulses which would accomplish the switching. The logic pulse width is exaggerated to

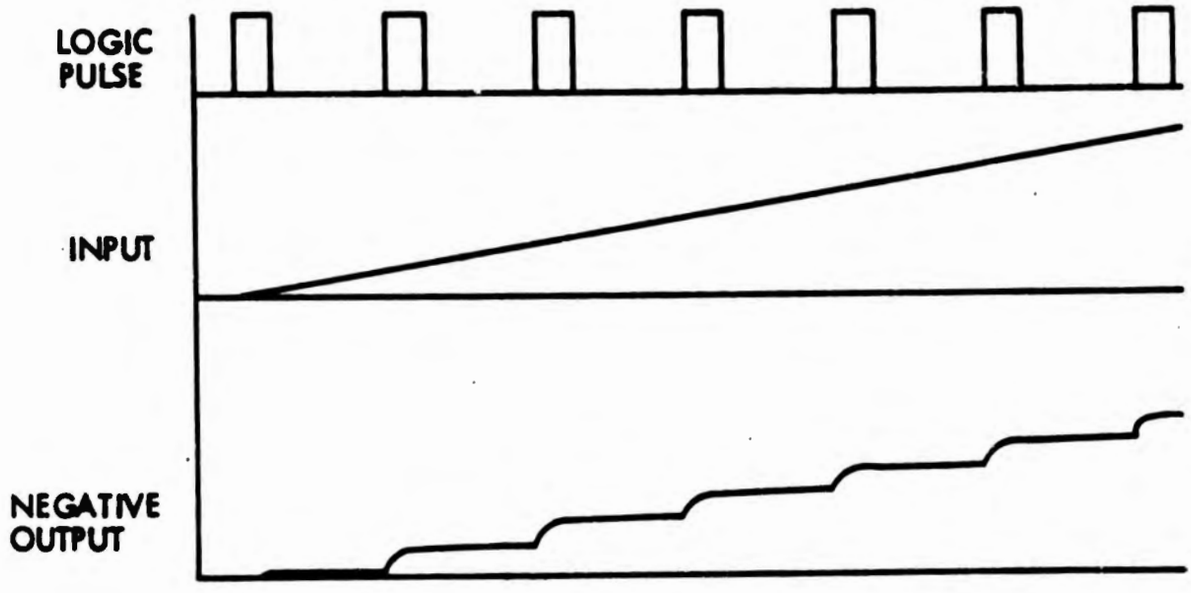


Fig. A-1. Relationship between the logic input, analog input, and analog output of a track/store amplifier.

show that there is a finite settling time for the device. If several track/store units are cascaded and driven by properly sequenced logic pulses, they can be used to simulate a delay function. To demonstrate this, refer to Fig. A-2 which shows the output of two cascaded track/store amplifiers, and their respective logic inputs, when they are driven by an analog ramp input. Note that when the logic pulse train b is such that its pulses just precede those of pulse train a, the output b is approximately the output a, delayed by one logic interval. In fact, at the instant of sampling for amplifier a, output b is exactly the value output a had one interval before. Therefore, for a sampled data system whose outputs are defined only at the sampling instant the track/store amplifier can be used to provide a delay function.

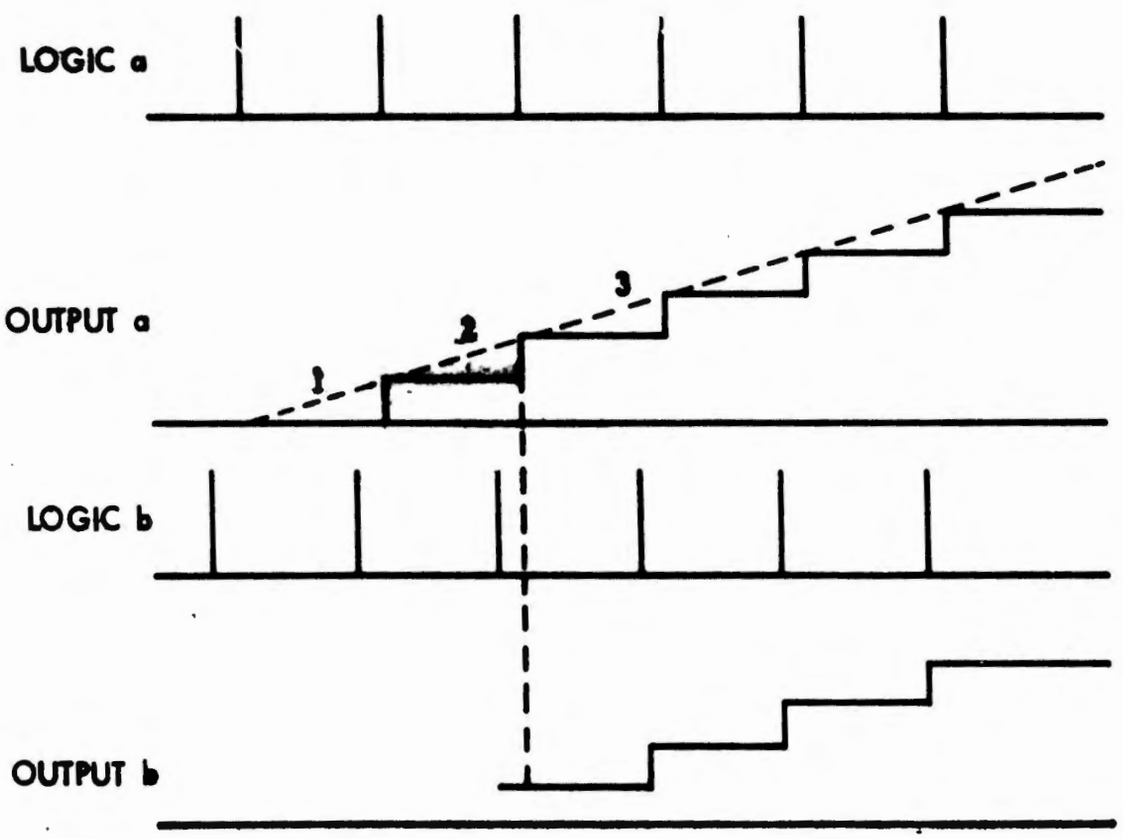


Fig. A-2. Output of two cascaded track/store amplifiers.

As an example of how these track/store amplifiers might be used, consider the case $D(z) = \frac{z - 1}{z} = \frac{1 - z^{-1}}{1} = \frac{X}{Y}$. This can be mechanized by summing the sampled input and the inverted delayed input. That is,

$$X^* = Y^* - Z^{-1}Y^* = Y^* - e^{-sT}Y^*.$$

If two track/store amplifiers are cascaded as described previously, then at the sampling instant, their sum will be the variable X. Note that the output must be sampled at the correct instants in time to give the correct results. This is the same instant that the first track/store amplifier samples the input signal. In nearly all physical systems the output of the sampled data controller must be held at some nonzero value between samples. For the case where this is accomplished by a zero order hold the summing amplifier and sampling switch on the output of Fig. A-3 can be replaced by another track/store amplifier driven by the same logic input as the first track/store amplifier. This also provides a signal which can be used to drive other track/store amplifiers so that the delayed output can be fed back when necessary.

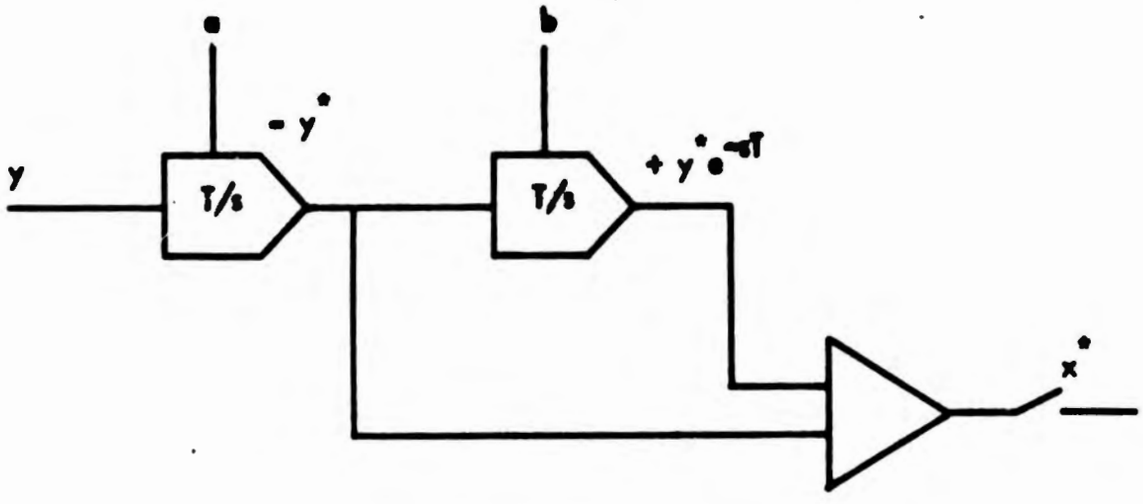


Fig. A-3. Mechanization of $\frac{X}{Y} = \frac{Z - 1}{Z}$.

For example, let

$$\frac{X}{Y} = \frac{Z + 0.1}{Z - 1} = \frac{1 + 0.1Z^{-1}}{1 - Z^{-1}}$$

or

$$X^* = Y^* + 0.1Y^*e^{-sT} + X^*e^{-sT} .$$

This can be mechanized as shown in Fig. A-4. The logic pulse L_1 must by definition occur at the sampling instant, and the pulse L_2 must occur an instant earlier.

This method can be used to generate reasonably complex functions so that fairly high-order sampled-data networks can be implemented. The basic limitation is that the track/store amplifiers have a tendency to drift so that the total storage time is somewhat limited. For the present application, this does not appear to be a problem of any great significance.

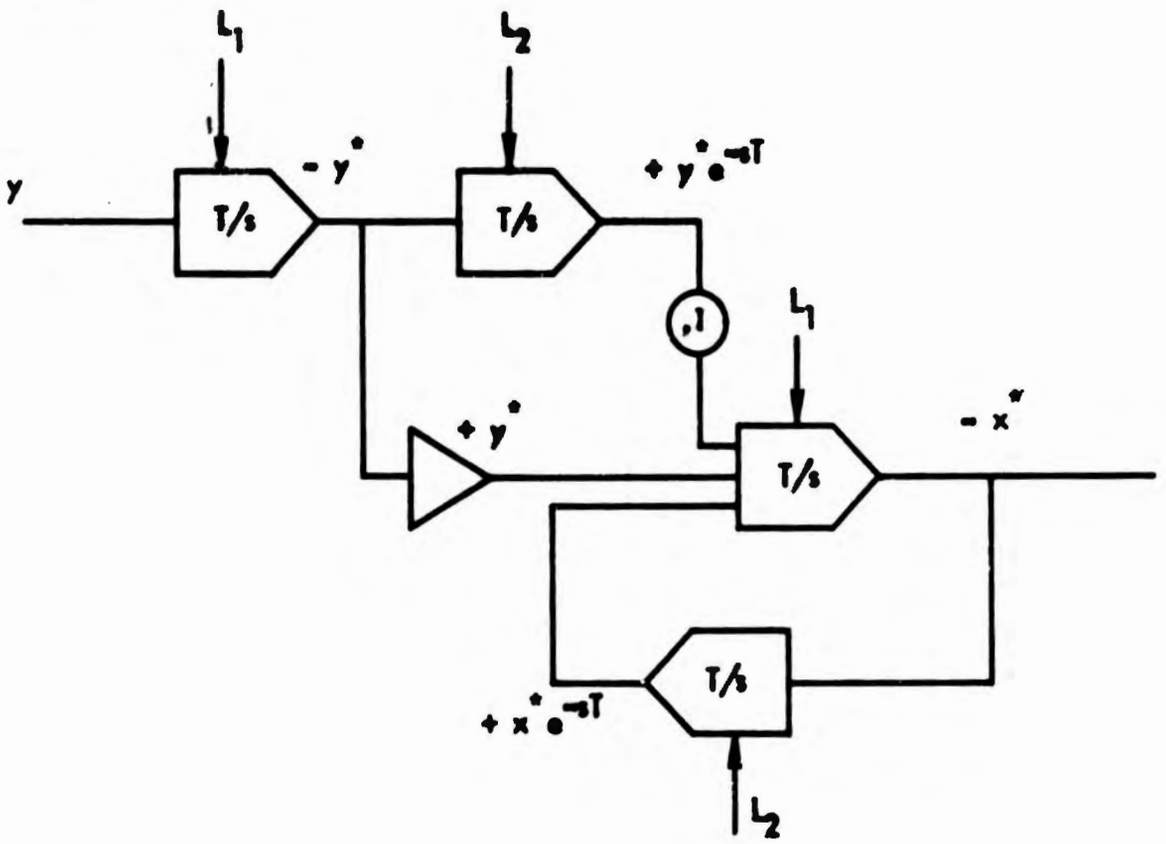


Fig. A-4. Mechanization of $\frac{X}{Y} = \frac{Z + 0.1}{Z - 1}$.

APPENDIX B

THIS PROGRAM DETERMINES THE SAMPLED RESPONSE OF A SAMPLED DATA SYSTEM TO AN ARBITRARY INPUT R(T)

THE PROGRAM REQUIRES THAT THE PULSE TRANSFER FUNCTION D(Z) BE EXPRESSED AS THE RATIO OF TWO POLYNOMIALS.

D(Z)=A(1)+A(2)Z*(-1)+...+A(M+1)Z*(-M)/1+B(2)Z*(-1)+...+B(N+1)Z*(-N)

LIMITATIONS...THE ORDER OF THE DENOMINATOR < +20

DATA INSTRUCTIONS

DATA CARD # 1 CONTAINS A '1' IN COLUMN 2 FOR EACH DATA SET

DATA CARD # 2 CONTAINS A NAME (1-80)

DATA CARD # 3 CONTAINS N,M,SFACTR (3X,I2,3X,I2,F5.2)
WHERE..N IS THE ORDER OF THE DENOMINATOR
M IS THE ORDER OF THE NUMERATOR
SFACTR IS THE SCALE FACTOR FOR THE PLOT

DATA CARD # 4 CONTAINS THE COEFFICIENTS OF THE NUMERATOR (4G15.7)
UP TO FIVE CARDS ALLOWED

THE NEXT DATA CARD CONTAINS THE COEFFICIENTS OF THE DENOMINATOR(4G15.7)
UP TO FIVE CARDS ALLOWED

THE NEXT DATA CARD CONTAINS T,TMAX,ICRM (3F12.6)
WHERE...T IS THE SAMPLING TIME

TMAX IS THE FINAL VALUE OF TIME
ICRM IS THE # OF PRINTOUT INCREMENTS...FOR A PRINTOUT EVERY SAMPLING INTERVAL ICRM = TMAX/T

THE NEXT DATA CARD CONTAINS NSIG (I2)
WHERE ...NSIG=1 RESPONSE TO A UNIT STEP
NSIG=2 RESPONSE TO A UNIT RAMP
NSIG=3 RESPONSE TO A UNIT ACCELERATION
NSIG=4 REQUIRES NEW SUBROUTINE

LAST DATA CARD CONTAINS A '0' IN COLUMN 2

```
      DIMENSION A(20),B(20),C(20),D(130),E(130),R(20),SNAM(80),F(1)
      DATA D/130*' '/,E/130*'.'/,F/1*'+'/
      REAL OMEGA,ICRM
1     READ(1,111) NTRIP
111   FORMAT(I2)
      IF(NTRIP.EQ.1) GO TO 112
      GO TO 1001
112   READ(1,113) (SNAM(II),II=1,20)
113   FORMAT(20A4)
      READ(1,100) N,M,SFACTR
100   FORMAT(3X,I2,3X,I2,F5.2)
      N=N+1
      M=M+1
      READ(1,101)(A(I),I=1,M)
101   FORMAT(4G15.7)
      READ(1,101)(B(J),J=1,N)
      READ(1,102) T,TMAX,ICRM
102   FORMAT(3F12.6)
      READ(1,111)NSIG
      WRITE(3,103)T,TMAX,ICRM
103   FORMAT('1',5X,'SAMPLING INTERVAL=',F12.6,5X,'FINAL TIME=',F12.6,5X,
     1'SCALE INCREMENTS=',F12.6)
      WRITE(3,104)(SNAM(II),II=1,20)
104   FORMAT(' ',20A4)
      WRITE(3,105)(E(I),I=1,100)
105   FORMAT('0',3X,'TIME',9X,'C(NT)',9X,100A1)
      TICRM=TMAX/ICRM
      NPTS=TMAX/T
      DUM=0.
      K=1
      DO1000 NN=1,NPTS
      TT=(NN-1)*T
      IF(NSIG.EQ.4) GO TO 124
      IF (NSIG.LT.4) GO TO 121
121   CALL SIGA(TT,R,C,K,T,NSIG)
      GO TO 125
124   CALL SIGB(TT,R,C,K,T,NSIG)
125   SUMC=0
      SUMR=0
      DO 2000 I=1,M
      L=K-I+1
      IF(L.EQ.0) GO TO 2001
      SUMR=SUMR+A(I)*R(L)
2000  CONTINUE
2001  DO 3000 I=2,N
      L=K-I+1
      IF(L.EQ.0) GO TO 3001
      SUMC=SUMC+B(I)*C(L)
3000  CONTINUE
3001  C(K)=SUMR-SUMC
      IF(DUM-TT) 106,106,114
106   WRITE(3,107)TT,C(K)
107   FORMAT( F10.3,5X,E12.5)
115   NSCALE=INT(C(K)*SFACTR)
      IF(NSCALE) 127,127,130
```

```
130   IF(NSCALE.GT.50) GO TO 116
      GO TO 118
116   WRITE(3,117)
117   FORMAT('+',30X,'SCALE FACTOR TOO LARGE')
      GO TO 108
118   WRITE(3,119) (D(I),I=1,NSCALE),F(1)
119   FORMAT('+',79X,60A1)
      GO TO 128
127   JJ=49+NSCALE
      IF(JJ.LT.1) GO TO 116
      WRITE(3,129) (D(I),I=1,JJ),F(1)
129   FORMAT('+',30X,50A1)
128   WRITE(3,120) E(1)
120   FORMAT('+',79X,1A1)
108   DUM=DUM+TICRM
114   IF(K-M) 109,109,110
109   K=K+1
      GO TO 1000
110   DO 4000 JJ=1,M
      C(JJ)=C(JJ+1)
      R(JJ)=R(JJ+1)
4000  CONTINUE
1000  CONTINUE
      GO TO 1
1001  CONTINUE
      STOP
      END
      SUBROUTINE SIGA(TT,R,C,K,T,NSIG)
      DIMENSION R(20),C(20)
      IF (NSIG-2) 100,101,102
100   R(K)=1.0
      GO TO 103
101   R(K)=1.0*TT
      GO TO 103
102   R(K)=TT*TT
103   CONTINUE
      RETURN
      END



      SUBROUTINE SIGB(TT,R,C,K,T,NSIG)
      DIMENSION R(20),C(20)
      RETURN
      END
```

APPENDIX C

```
A SPECTRAL IDENTIFICATION OF FLEXIBLE AIRFRAME BENDING MODES
AN EXAMPLE STUDIED HAS A FOURTH ORDER PROCESS
A SPECTRAL DENSITY FUNCTION IS ESTIMATED BY DIRECT FOURIER ANALYSIS,
SPECTRAL FILTERS AND FAST FOURIER TRANSFORM
     COMPLEX XI(4096),PI(4096),OMICRO,SIGMA,TAU
     REAL X1(5500),X2(5500),X3(5500),X4(5500)
    1,SF(100),LAMBDA(12)
    1,XCOF(5),COF(5),ROOTR(4),ROOTI(4),ALPA1,ALPA2,BETA1,BETA2
    1,XLAB(5),YLAB(5),GLAB(5),DATLAB(5),F(100)
     INTEGER IER
PROCESS CONSTANTS
     NDATA=5000
     NTRANS=500
     NRANDM=NDATA+NTRANS-1
     TIME=50.0
A FOURTH ORDER SYSTEM IS USED TO GENERATE THE PROCESS
     A=0.60
     B=104.130
     C=40.8120
     D=404.04040
     GAMMA=0.150
     GAINK=1.0
FOLLOWING TWO STATE A UNIT FEEDBACK FOR A CLOSED LOOP SYSTEM
     C=C+GAINK
     D=D+GAINK*GAMMA
CHARACTERISTIC VALUES OF THE SYSTEM
     XCOF(1)=D
     XCOF(2)=C
     XCOF(3)=B
     XCOF(4)=A
     XCOF(5)=1.0
     CALL POLRT (XCOF,COF,4,ROOTR,ROOTI,IER)
     ALPA1=-ROOTR(1)
     ALPA2=-ROOTR(3)
     BETA1=-ROOTI(1)
     BETA2=-ROOTI(3)
A TRANSFER FUNCTION METHOD IS USED FOR A STATE TRANSITION MATRIX
     OMEGA1=ALPA1*ALPA1+BETA1*BETA1
     OMEGA2=ALPA2*ALPA2+BETA2*BETA2
     DENOMI=(OMEGA1-OMEGA2)*(OMEGA1-OMEGA2)
    1+4.0*(ALPA1-ALPA2)*(ALPA1*OMEGA2-ALPA2*OMEGA1)
     A1=2.0*(ALPA1-ALPA2)/DENOMI
     B1=(4.0*ALPA1*(ALPA1-ALPA2)-(OMEGA1-OMEGA2))/DENOMI
     C1=-A1
     D1=(1.0-OMEGA2*B1)/OMEGA1
     A2=(OMEGA2-OMEGA1)/DENOMI
     B2=2.0*(ALPA2-ALPA1)*OMEGA1/DENOMI
     C2=-A2
     D2=-OMEGA2/OMEGA1*B2
     A3=2.0*(ALPA2*OMEGA1-ALPA1*OMEGA2)/DENOMI
     B3=OMEGA1*(OMEGA1-OMEGA2)/DENOMI
     C3=-A3
     D3=-OMEGA2/OMEGA1*B3
```

```
      A4=1.0+OMEGA2*(OMEGA1-OMEGA2)/DENOMI
      B4=2.0*OMEGA1*(ALPA1*OMEGA2-ALPA2*OMEGA1)/DENOMI
      C4=1.0-A4
      D4=-OMEGA2/OMEGA1*B4
THE STATE TRANSITION MATRIX OF THE SYSTEM
      DELTAT=1.0/100.0
      THETA1=1.0/EXP(ALPA1*DELTAT)*COS(BETA1*DELTAT)
      THETA2=1.0/EXP(ALPA1*DELTAT)*SIN(BETA1*DELTAT)
      THETA3=1.0/EXP(ALPA2*DELTAT)*COS(BETA2*DELTAT)
      THETA4=1.0/EXP(ALPA2*DELTAT)*SIN(BETA2*DELTAT)
      PHI14=A1*THETA1+(-ALPA1*A1+B1)/BETA1*THETA2
     1+C1*THETA3+(-ALPA2*C1+D1)/BETA2*THETA4
      PHI24=A2*THETA1+(-ALPA1*A2+B2)/BETA1*THETA2
     1+C2*THETA3+(-ALPA2*C2+D2)/BETA2*THETA4
      PHI34=A3*THETA1+(-ALPA1*A3+B3)/BETA1*THETA2
     1+C3*THETA3+(-ALPA2*C3+D3)/BETA2*THETA4
      PHI44=A4*THETA1+(-ALPA1*A4+B4)/BETA1*THETA2
     1+C4*THETA3+(-ALPA2*C4+D4)/BETA2*THETA4
      PHI11=PHI44+A*PHI34+B*PHI24+C*PHI14
      PHI12=PHI34+A*PHI24+B*PHI14
      PHI13=PHI24+A*PHI14
      PHI21=-D*PHI14
      PHI22=PHI44+A*PHI34+B*PHI24
      PHI23=PHI34+A*PHI24
      PHI31=-D*PHI24
      PHI32=-C*PHI24-D*PHI14
      PHI33=PHI44+A*PHI34
      PHI41=-D*PHI34
      PHI42=-C*PHI34-D*PHI24
      PHI43=-B*PHI34-C*PHI24-D*PHI14
COEFFICIENTS OF RECURSIVE EQUATIONS
      EPSIL1=1.0/EXP(0.50*ALPA1*DELTAT)*COS(0.50*BETA1*DELTAT)
      EPSIL2=1.0/EXP(0.50*ALPA1*DELTAT)*SIN(0.50*BETA1*DELTAT)
      EPSIL3=1.0/EXP(0.50*ALPA2*DELTAT)*COS(0.50*BETA2*DELTAT)
      EPSIL4=1.0/EXP(0.50*ALPA2*DELTAT)*SIN(0.50*BETA2*DELTAT)
      CHI14=A1*EPSIL1+(-ALPA1*A1+B1)/BETA1*EPSIL2
     1+C1*EPSIL3+(-ALPA2*C1+D1)/BETA2*EPSIL4
      CHI24=A2*EPSIL1+(-ALPA1*A2+B2)/BETA1*EPSIL2
     1+C2*EPSIL3+(-ALPA2*C2+D2)/BETA2*EPSIL4
      CHI34=A3*EPSIL1+(-ALPA1*A3+B3)/BETA1*EPSIL2
     1+C3*EPSIL3+(-ALPA2*C3+D3)/BETA2*EPSIL4
      CHI44=A4*EPSIL1+(-ALPA1*A4+B4)/BETA1*EPSIL2
     1+C4*EPSIL3+(-ALPA2*C4+D4)/BETA2*EPSIL4
      CHI13=CHI24+A*CHI14
      CHI23=CHI34+A*CHI24
      CHI33=CHI44+A*CHI34
      CHI43=-B*CHI34-C*CHI24-D*CHI14
      RHO1=GAINK*(CHI13+(GAMMA-A)*CHI14)
      RHO2=GAINK*(CHI23+(GAMMA-A)*CHI24)
      RHO3=GAINK*(CHI33+(GAMMA-A)*CHI34)
      RHO4=GAINK*(CHI43+(GAMMA-A)*CHI44)
RECURSIVE EQUATIONS GENERATE THE FOURTH ORDER PROCESS
      IX=1942827
      IY=946325
      X1(1)=0.0
      X2(1)=0.0
      X3(1)=0.0
      X4(1)=0.0
```

```
      DO 10 N=1,NRANDM
      CALL GAUSS (IX,1.0,0.0,GN)
FOLLOWING TWO STATE A UNIT FEEDBACK FOR A CLOSED LOOP PROCESS
      CALL GAUSS (IY,1.0,0.0,DELTYN)
      GN=GN-DELTYN
      X1(N+1)=PHI11*X1(N)+PHI12*X2(N)+PHI13*X3(N)+PHI14*X4(N)+RHO1*GN
      X2(N+1)=PHI21*X1(N)+PHI22*X2(N)+PHI23*X3(N)+PHI24*X4(N)+RHO2*GN
      X3(N+1)=PHI31*X1(N)+PHI32*X2(N)+PHI33*X3(N)+PHI34*X4(N)+RHO3*GN
   10 X4(N+1)=PHI41*X1(N)+PHI42*X2(N)+PHI43*X3(N)+PHI44*X4(N)+RHO4*GN
      DO 11 N=1,NDATA
      CALL GAUSS (IY,1.0,0.0,DELTYN)
   11 X2(N)=X1(N+NTRANS)+DELTYN
      WRITE (3,12) ALPA1,BETA1,ALPA2,BETA2
   12 FORMAT ('0','THE ALPAS AND BETAS ARE'/(2(5X,E12.5)))
      WRITE (3,13) IER
   13 FORMAT ('0','IER=',I2)
SPECTRAL DENSITY FUNCTION OF THE FOURTH ORDER PROCESS
      NPOINT=100
      TWOPI=2.0*3.14159
      DO 14 K=1,NPOINT
      CK=K
      F(K)=CK/50.0
   14 X1(K)=GAINK*GAINK*(TWOPI*F(K)*TWOPI*F(K)+GAMMA*GAMMA)/DELTAT
     1/((TWOPI*F(K)*TWOPI*F(K)+OMEGA1)*(TWOPI*F(K)*TWOPI*F(K)+OMEGA1)
     1-4.0*BETA1*BETA1*TWOPI*F(K)*TWOPI*F(K))
     1/((TWOPI*F(K)*TWOPI*F(K)+OMEGA2)*(TWOPI*F(K)*TWOPI*F(K)+OMEGA2)
     1-4.0*BETA2*BETA2*TWOPI*F(K)*TWOPI*F(K))
      WRITE (3,15) (F(K),X1(K),K=1,NPOINT)
   15 FORMAT ('1','SPECTRAL DENSITY FUNCTION'/(2X,F6.3,2X,F12.6))
DIRECT FOURIER ANALYSIS
      NSAMPL=201
      CDATA=NDATA
      NDATAM=NDATA-1
      CDATAM=NDATAM
      TWOPID=TWOPI*DELTAT
      WRITE (3,20)
   20 FORMAT ('1','DIRECT FOURIER ANALYSIS')
      DO 22 K=1,NSAMPL
      CK=K-1
      FREQDF=CK/100.0
      SS=0.50*SIN(TWOPID*FREQDF)*X2(1)
     1+0.50*SIN(CDATA*TWOPID*FREQDF)*X2(NDATA)
      SC=0.50*COS(TWOPID*FREQDF)*X2(1)
     1+0.50*COS(CDATA*TWOPID*FREQDF)*X2(NDATA)
      DO 21 L=2,NDATAM
      CL=L
      SS=SS+SIN(TWOPID*FREQDF*CL)*X2(L)
   21 SC=SC+COS(TWOPID*FREQDF*CL)*X2(L)
      SPECDF=(SS*SS+SC*SC)*DELTAT/2.0/CDATAM
      X3(K)=FREQDF
      X4(K)=SPECDF
   22 WRITE (3,23) FREQDF,SPECDF
   23 FORMAT (2X,F6.3,2X,F12.6)
      READ (1,24) XLAB,YLAB,GLAB,DATLAB
   24 FORMAT (20A4)
      CALL GRAPH (NPOINT,F,X1,K,2,7.0,4.0,0.3,0.0,0.06,0.0
     1,XLAB,YLAB,GLAB,DATLAB)
```

```
    CALL GRAPH (NSAMPL,X3,X4,3,3,0,0,0,0,0,0,0,0,0,0)
SPECTRAL FILTERS
    NSAMPL=90
    WRITE (3,50)
 50 FORMAT ('1','SPECTRAL FILTERS')
    DO 51 K=1,83
 51 SF(K)=2*K
    SF(84)=2*91
    SF(85)=2*101
    SF(86)=2*111
    SF(87)=2*125
    SF(88)=2*143
    SF(89)=2*167
    SF(90)=2*200
    DO 55 K=1,NSAMPL
    FREQSF=1.0/(2.0*DELTAT)/SF(K)
    SINIII=0.0
    COSIII=0.0
    SINII=0.0
    COSII=0.0
    UI=0.0
    VI=0.0
    SIGN=-1.0
    DO 54 L=1,9
    CL=L
    SIGN=-SIGN
    SINI=0.0
    COSI=0.0
    MUA=1.0+(CL-1.0)*SF(K)
    MUB=CL*SF(K)
    MUC=(CL-0.50)*SF(K)
    MUD=MUC+1
    DO 52 M=MUA,MUC
    SINI=SINI+X2(M)
    COSI=COSI+X2(M)
    DO 53 N=MUD,MUB
    SINI=SINI+X2(N)
    COSI=COSI-X2(N)
    UI=SIGN*SINI+SINII
    VI=SIGN*COSI+COSII
    SINII=SIGN*SINI
    COSII=SIGN*COSI
    SINIII=SINIII+UI
    COSIII=COSIII+VI
    B=SINIII*SINIII+COSIII*COSIII
    A=B/200.0/(SF(K)*SF(K))
    X3(NSAMPL-K+1)=FREQSF
    X4(NSAMPL-K+1)=A
    WRITE (3,56) FREQSF,A
    FORMAT (2X,F6.3,2X,F12.6)
    READ (1,57) XLAB,YLAB,GLAB,DATLAB
    FORMAT (20A4)
    CALL GRAPH (NPOINT,F,X1,K,2,7.0,4.0,0.3,0.0,0.06,0.0
   1,XLAB,YLAB,GLAB,DATLAB)
```

```
      CALL GRAPH (78,X3,X4,3,3,0,0,0,0,0,0,0,0,0,0)
FAST FOURIER TRANSFORM
      NSAMPL=83
      LAMBDA(1)=2048
      DO 79 K=1,11
   79 LAMBDA(K+1)=LAMBDA(K)/2
      WRITE (3,80)
   80 FORMAT ('1','FAST FOURIER TRANSFORM')
      DO 81 K=1,2048
      XI(K)=X2(K)+X2(K+2048)
   81 XI(K+2048)=X2(K)-X2(K+2048)
      DO 83 K=2,10,2
      IOTA=LAMBDA(K)
      KAPPA=2048/IOTA
      CKAPPA=KAPPA
      DO 82 L=1,IOTA
      DO 82 M=1,KAPPA
      MU=M-1
      CMU=MU
      CSIGMA=-CMU/CKAPPA*3.14159
      CTAU=-(CMU/CKAPPA+1.0)*3.14159
      SIGMA=CMPLX(0.0,CSIGMA)
      TAU=CMPLX(0.0,CTAU)
      NUA=L+MU*IOTA
      NUB=L+2*MU*IOTA
      NUC=NUB+IOTA
      NUD=NUA+2048
      PI(NUA)=XI(NUB)+XI(NUC)*CEXP(SIGMA)
   82 PI(NUD)=XI(NUB)+XI(NUC)*CEXP(TAU)
      IOTA=LAMBDA(K+1)
      KAPPA=2048/IOTA
      CKAPPA=KAPPA
      DO 83 L=1,IOTA
      DO 83 M=1,KAPPA
      MU=M-1
      CMU=MU
      CSIGMA=-CMU/CKAPPA*3.14159
      CTAU=-(CMU/CKAPPA+1.0)*3.14159
      SIGMA=CMPLX(0.0,CSIGMA)
      TAU=CMPLX(0.0,CTAU)
      NUA=L+MU*IOTA
      NUB=L+2*MU*IOTA
      NUC=NUB+IOTA
      NUD=NUA+2048
      XI(NUA)=PI(NUB)+PI(NUC)*CEXP(SIGMA)
      XI(NUD)=PI(NUB)+PI(NUC)*CEXP(TAU)
      DO 84 K=1,2048
      MU=K-1
      CMU=MU
      CSIGMA=-CMU/2048.0*3.14159
      CTAU=-(CMU/2048.0+1.0)*3.14159
      SIGMA=CMPLX(0.0,CSIGMA)
      TAU=CMPLX(0.0,CTAU)
      PI(MU+1)=XI(2*MU+1)+XI(2*MU+2)*CEXP(SIGMA)
      PI(MU+2049)=XI(2*MU+1)+XI(2*MU+2)*CEXP(TAU)
      DO 85 K=1,NSAMPL
      KFREQ=K-1
```

```
      AFREQ=KFREQ
      FREQFF=AFREQ/(4096.0*DELTAT)
      OMICRO=PI(K)
      OMICRA=CABS(OMICRO)
      OMICRS=OMICRA*OMICRA/(2.0*4096.0)*DELTAT
      X3(K)=FREQFF
      X4(K)=OMICRS
      WRITE (3,86) FREQFF,OMICRS
      FORMAT (2X,F6.3,2X,F12.6)
      READ (1,87) XLAB,YLAB,GLAB,DATLAB
      FORMAT (20A4)
      CALL GRAPH (NPOINT,F,X1,K,2,7.0,4.0,0.3,0.0,0.06,0.0
     1,XLAB,YLAB,GLAB,DATLAB)
      CALL GRAPH (NSAMPL,X3,X4,3,3,0,0,0,0,0,0,0,0,0,0)
      STOP
      END
```

APPENDIX D

```
AN ADAPTIVE CONTROL OF A FLEXIBLE AIRFRAME
UTILIZING A SPECTRAL IDENTIFICATION FOR BENDING MODES
AN EXAMPLE STUDIED HAS A SEVENTH ORDER PROCESS
AN AIRFRAME IS SIMULATED BY A FOURTH ORDER TRANSFER FUNCTION
A BENDING MODE COMPENSATOR HAS A THIRD ORDER TRANSFER FUNCTION
THE SIMULATED SYSTEM HAS A UNIT FEEDBACK LOOP
THE SPECTRAL DENSITY FUNCTION IS ESTIMATED BY A FAST FOURIER TRANSFORM
     COMPLEX A(7,7),B(7,2),C(7,2),D(7,7),E(7),F(7,7),G,GD,GH,H,U,V,W
    1,XI(1024),YI(1024)
     REAL P(7,7),Q(7,2),X(1524),Y(1024),ZA(7),ZB(7)
    1,XA(250),XB(250),XC(250),XD(250),XE(250)
    1,XL(5),YL(5),GL(5),DL(5)
     INTEGER I(10)
CHARACTERISTIC VALUES OF A FOURTH ORDER AIRFRAME SYSTEM
A(1,1) AND A(1,2), A(1,3) AND A(1,4) ARE BOTH COMPLEX CONJUGATES
A(1,1) AND A(1,2) REPRESENT THE AIRFRAME BENDING MODE
100  CONTINUE
     N=4
     PI=3.14159
     TP=2.0*PI
     T=0.010
     DO 101 K=1,7
     DO 101 L=1,7
     A(K,L)=(0.0,0.0)
101  CONTINUE
     READ (1,102) A(1,1)
102  FORMAT (F12.6,F12.6)
     A(1,2)=CONJG(A(1,1))
     A(1,3)=(0.20,2.0)
     A(1,4)=CONJG(A(1,3))
     GH=(1.0,0.0)
     A(2,1)=(0.15,0.0)
     A(3,1)=A(1,1)*A(1,2)*A(1,3)*A(1,4)
     A(3,2)=A(1,1)*A(1,2)*A(1,3)+A(1,1)*A(1,2)*A(1,4)
    1+A(1,1)*A(1,3)*A(1,4)+A(1,2)*A(1,3)*A(1,4)
     A(3,3)=A(1,1)*A(1,2)+A(1,1)*A(1,3)+A(1,1)*A(1,4)
    1+A(1,2)*A(1,3)+A(1,2)*A(1,4)+A(1,3)*A(1,4)
     A(3,4)=A(1,1)+A(1,2)+A(1,3)+A(1,4)
     A(4,1)=A(2,1)
     A(5,1)=A(3,1)+GH*A(4,1)
     A(5,2)=A(3,2)+GH
     A(5,3)=A(3,3)
     A(5,4)=A(3,4)
     A(6,1)=GH*A(4,1)
     A(6,2)=GH
     A(7,1)=GH*A(4,1)
     A(7,2)=GH
     B(1,1)=(0.0,0.0)
     B(2,1)=(0.0,0.0)
     B(3,1)=A(6,2)
     B(4,1)=A(6,1)-A(5,4)*B(3,1)
     B(1,2)=(0.0,0.0)
```

```
      B(2,2)=(0.0,0.0)
      B(3,2)=A(7,2)
      B(4,2)=A(7,1)-A(5,4)*B(3,2)
      DF=REAL(A(1,1))/SQRT(REAL(A(1,1)*CONJG(A(1,1))))
  103 CONTINUE
      NP=250
      GK=REAL(GH)
      DO 105 K=1,NP
      AK=K
      XA(K)=AK/100.0
      W=CMPLX(0.0,TP*XA(K))
      U=W+A(4,1)
      V=(1.0,0.0)
      DO 104 L=1,N
      V=V*W+A(5,N+1-L)
  104 CONTINUE
      U=U/V
      XB(K)=GK*GK*REAL(U*CONJG(U))
  105 CONTINUE
      WRITE (3,106) GH,DF
  106 FORMAT ('0','GH=',F12.6,2X,F12.6/' ','DF=',F12.6)
      WRITE (3,107) ((A(K,L),L=1,7),K=1,7)
  107 FORMAT ('0','A(K,L)'
     1/(' ',4(E12.5,2X,E12.5,6X)/' ',3(E12.5,2X,E12.5,6X)))
      WRITE (3,108) ((B(K,L),L=1,2),K=1,N)
  108 FORMAT ('0','B(K,L)'/(' ',2(E12.5,2X,E12.5,6X)))
      WRITE (3,109) (XA(K),XB(K),K=1,NP)
  109 FORMAT ('0','UNIT WHITE NOISE SPECTRAL RESPONSE'
     1/(' ',F12.3,2X,E12.5))
      READ (1,110) XL,YL,GL,DL
  110 FORMAT (20A4)
      CALL GRAPH (NP,XA,XB,K,2,7.0,4.0,0.4,0.0,5.0E-03,0.0,XL,YL,GL,DL)
ESTIMATE ROOTS OF A POLINOMIAL
  200 CONTINUE
      DO 201 K=1,N
      C(K,1)=-A(1,K)
      C(K,2)=A(5,K)
  201 CONTINUE
      NA=N-2
      DO 210 K=1,NA
      NB=N+1-K
      DO 204 L=1,20
      U=(1.0,0.0)
      V=(7.0,0.0)
      DO 202 M=2,NB
      AM=NB+1-M
      U=U*C(K,1)+C(N+2-M,2)
      V=V*C(K,1)+AM*C(N+2-M,2)
  202 CONTINUE
      U=U*C(K,1)+C(K,2)
      W=C(K,1)
      IF (REAL(V*CONJG(V)).LT.1.0E-35) V=100.0*V
      C(K,1)=C(K,1)-U/V
      G=W-C(K,1)
      AW=REAL(G*CONJG(G))-1.0E-06*REAL(C(K,1)*CONJG(C(K,1)))
      IF (AW.GT.0.0) GO TO 204
      AX=AIMAG(C(K,1))
```

```
      AY=REAL(C(K,1))
      AZ=AX*AX/(AY*AY)
      IF (AZ.GE.1.0E-09) GO TO 203
      C(K,1)=CMPLX(AY,0.0)
  203 CONTINUE
      IF (AW.LE.0.0) GO TO 205
  204 CONTINUE
  205 CONTINUE
      DO 207 L=1,30
      U=(1.0,0.0)
      V=(7.0,0.0)
      DO 206 M=2,NB
      AM=NB+1-M
      U=U*C(K,1)+C(N+2-M,2)
      V=V*C(K,1)+AM*C(N+2-M,2)
  206 CONTINUE
      U=U*C(K,1)+C(K,2)
      W=C(K,1)
      IF (REAL(V*CONJG(V)).LT.1.0E-35) V=30.0*V
      C(K,1)=C(K,1)-U/V
      G=W-C(K,1)
      AW=REAL(G*CONJG(G))-1.0E-06*REAL(C(K,1)*CONJG(C(K,1)))
      IF (AW.LT.0.0) GO TO 208
  207 CONTINUE
      GO TO 205
  208 CONTINUE
      H=(1.0,0.0)
      DO 209 L=1,NB
      H=H*C(K,1)+C(N+1-L,2)
      C(N+1-L,2)=H
  209 CONTINUE
  210 CONTINUE
      C(N-1,1)=(-C(N,2)-CSQRT(C(N,2)*C(N,2)-4.0*C(N-1,2)))/2.0
      C(N,1)=(-C(N,2)+CSQRT(C(N,2)*C(N,2)-4.0*C(N-1,2)))/2.0
      C(N-1,2)=(0.0,0.0)
      C(N,2)=(0.0,0.0)
      DO 211 K=1,N
      C(K,1)=-C(K,1)
  211 CONTINUE
      WRITE (3,212) ((C(K,L),L=1,2),K=1,N)
  212 FORMAT ('0','C(K,1)',26X,'C(K,2)'/(' ',2(E12.5,2X,E12.5,6X)))
ESTIMATE PARTIAL FRACTIONS
  220 CONTINUE
      DO 223 K=1,N
      U=(1.0,0.0)
      DO 222 L=1,N
      V=C(L,1)-C(K,1)
      IF (REAL(V*CONJG(V)).NE.0.0) GO TO 221
      V=(1.0,0.0)
  221 CONTINUE
      U=U*V
  222 CONTINUE
      D(1,K)=1.0/U
      DO 223 M=2,N
      D(M,K)=-C(K,1)*D(M-1,K)
  223 CONTINUE
      WRITE (3,224) ((D(K,L),L=1,N),K=1,N)
```

```
224 FORMAT ('0','D(K,L)'/(' ',E12.5,2X,E12.5))
ESTIMATE A STATE TRANSITION MATRIX
240 CONTINUE
    LOOP=1
    DO 241 K=1,N
    E(K)=-C(K,1)*T
241 CONTINUE
242 CONTINUE
    DO 244 K=1,N
     U=(0.0,0.0)
    DO 243 L=1,N
    U=U+D(K,L)*CEXP(E(L))
243 CONTINUE
    F(K,N)=U
244 CONTINUE
    NA=N-1
    DO 246 K=1,NA
    DO 246 L=K,NA
    LA=N+K-L
    LB=N-L
    U=F(LA,N)
    DO 245 M=1,LB
    U=U+A(5,N+1-M)*F(LA-M,N)
245 CONTINUE
    F(K,L)=U
246 CONTINUE
    DO 248 K=2,N
    KA=K-1
    DO 248 L=1,KA
    U=(0.0,0.0)
    DO 247 M=1,L
    U=U-A(5,L+1-M)*F(K-M,N)
247 CONTINUE
    F(K,L)=U
248 CONTINUE
    IF (LOOP-1) 262,249,262
249 CONTINUE
    DO 250 K=1,N
    DO 250 L=1,N
    P(K,L)=REAL(F(K,L))
250 CONTINUE
    WRITE (3,251) ((F(K,L),P(K,L),L=1,N),K=1,N)
251 FORMAT ('0','F(K,L)',26X,'P(K,L)'
   1/(' ',E12.5,2X,E12.5,6X,E12.5))
ESTIMATE A COEFFICIENT OF RECURSIVE EQUATIONS
260 CONTINUE
    LOOP=2
    DO 261 K=1,N
    E(K)=-C(K,1)*T/2.0
261 CONTINUE
    GO TO 242
262 CONTINUE
    DO 264 K=1,N
    DO 264 L=1,2
    U=(0.0,0.0)
    DO 263 M=1,N
    U=U+B(M,L)*F(K,M)
```

```
263 CONTINUE
    C(K,L)=U
264 CONTINUE
    DO 265 K=1,N
    DO 265 L=1,2
    Q(K,L)=REAL(C(K,L))
265 CONTINUE
    WRITE (3,266) ((C(K,L),Q(K,L),L=1,2),K=1,N)
266 FORMAT ('0','C(K,L)',26X,'Q(K,L)'
   1/(' ',E12.5,2X,E12.5,6X,E12.5))
RECURSIVE EQUATIONS GENERATE THE SYSTEM PROCESS
280 CONTINUE
    NDATA=1024
    NTRANS=500
    NRANDM=NDATA+NTRANS-1
    TIME=1024.0*T
    IX=1942827
    IY=946325
    X(1)=0.0
    DO 281 K=1,N
    ZA(K)=0.0
281 CONTINUE
    DO 285 K=1,NRANDM
    CALL GAUSS (IX,1.0,0.0,GN)
    CALL GAUSS (IY,1.0,0.0,DELTYN)
    DO 283 L=1,N
    U=Q(L,1)*GN-Q(L,2)*DELTYN
    DO 282 M=1,N
    U=U+P(L,M)*ZA(M)
282 CONTINUE
    ZB(L)=U
283 CONTINUE
    DO 284 L=1,N
    ZA(L)=ZB(L)
284 CONTINUE
    X(K+1)=ZB(1)
285 CONTINUE
    DO 286 K=1,NDATA
    CALL GAUSS (IY,1.0,0.0,DELTYN)
    Y(K)=X(K+NTRANS)+DELTYN
286 CONTINUE
ESTIMATE A SPECTRAL DENSITY FUNCTION BY A FAST FOURIER TRANSFORM
300 CONTINUE
    NS=2.50*TIME+1.0
    I(1)=512
    IA=I(1)
    DO 301 K=1,9
    I(K+1)=I(K)/2
301 CONTINUE
    DO 302 K=1,IA
    XI(K)=Y(K)+Y(K+IA)
    XI(K+IA)=Y(K)-Y(K+IA)
302 CONTINUE
    DO 304 K=2,8,2
    IB=I(K)
    IC=IA/IB
    AI=IC
```

```
      DO 303 L=1,IB
      DO 303 M=1,IC
      MA=M-1
      AM=MA
      AU=-(AM/AI)*PI
      AV=-(AM/AI+1.0)*PI
      U=CMPLX(0.0,AU)
      V=CMPLX(0.0,AV)
      NA=L+MA*IB
      NB=L+2*MA*IB
      NC=NB+IB
      ND=NA+IA
      YI(NA)=XI(NB)+XI(NC)*CEXP(U)
      YI(ND)=XI(NB)+XI(NC)*CEXP(V)
303   CONTINUE
      IB=I(K+1)
      IC=IA/IB
      AI=IC
      DO 304 L=1,IB
      DO 304 M=1,IC
      MA=M-1
      AM=MA
      AU=-(AM/AI)*PI
      AV=-(AM/AI+1.0)*PI
      U=CMPLX(0.0,AU)
      V=CMPLX(0.0,AV)
      NA=L+MA*IB
      NB=L+2*MA*IB
      NC=NB+IB
      ND=NA+IA
      XI(NA)=YI(NB)+YI(NC)*CEXP(U)
      XI(ND)=YI(NB)+YI(NC)*CEXP(V)
304   CONTINUE
      AI=IA
      DO 305 M=1,IA
      MA=M-1
      AM=MA
      AU=-(AM/AI)*PI
      AV=-(AM/AI+1.0)*PI
      U=CMPLX(0.0,AU)
      V=CMPLX(0.0,AV)
      YI(MA+1)=XI(2*MA+1)+XI(2*MA+2)*CEXP(U)
      YI(MA+1+IA)=XI(2*MA+1)+XI(2*MA+2)*CEXP(V)
305   CONTINUE
      DO 306 K=1,NS
      AK=K-1
      XC(K)=AK/TIME
      AX=CABS(YI(K))
      XD(K)=AX*AX*T*T/2048.0
306   CONTINUE
      WRITE (3,307) (XC(K),XD(K),K=1,NS)
307   FORMAT ('0','SYSTEM SPECTRAL RESPONSE'
     1/(' ',F12.3,2X,F12.5)
      CALL GRAPH (NS,XC,XD,3,3,0,0,0,0,0,0,0,0,0,0)
IDENTIFY THE BENDING FREQUENCY BF
320   CONTINUE
      NA=NS-1
```

```
      NB=1
      NC=0.50*TIME+1.0
      ND=TIME+1.0
      AX=XD(NC)
      AW=ND-NC+1
      DO 322 K=NC,NA
      AY=XD(K+1)
      IF (AX-AY) 321,322,322
321   CONTINUE
      AX=AY
      NB=K
322   CONTINUE
      AZ=0.0
      DO 323 K=NC,ND
      AZ=AZ+XD(K)
323   CONTINUE
      AZ=AZ/AW
      BF=NB
      BF=BF/TIME
      IF (AX.LT.2.00*AZ) BF=0.0
      WRITE (3,324) BF
324   FORMAT ('0','BF=',F12.6)
      IF (BF) 447,447,400
ESTIMATE A REAL PART OF A COMPENSATOR'S ZERO AR AND SAMPLING RATE T
REAL PART OF A COMPENSATOR'S ZERO AR IS ESTIMATED FROM
BENDING MODE DAMPING FACTOR DF AND IMAGINARY PART AI
400   CONTINUE
      AI=0.45360
      AR=0.88590
      AS=SQRT(1.0/DF/DF-1.0)
      DO 401 K=1,30
      AK=K
      AT=SQRT(AR*AR+AI*AI)
      AU=AS*ABS(ALOG(AT))
      AV=TAN(AU)
      AW=AI/AV
      AX=AR-AW
      IF (ABS(AX).LT.1.0E-10) GO TO 402
401   CONTINUE
      AR=AR-AX/100.0/AK
402   CONTINUE
      AY=ATAN(AI/AR)
      AZ=ABS(ALOG(SQRT(AR*AR+AI*AI)))
      T=(1.0/TP/BF)*SQRT(AY*AY+AZ*AZ)
      TI=1.0/T
      PT=PI*TI
      WRITE (3,403) T,AR,AI
403   FORMAT ('0','T =',F12.6/' ','AR=',F12.6/' ','AI=',F12.6)
CHARACTERISTIC VALUES OF A COMPENSATOR D(Z) IN Z-DOMAIN
A(5,5) HAS -1.0, A(5,6) AND A(5,7) HAVE ONLY REAL PARTS
A(6,6) AND A(6,7) ARE COMPLEX CONJUGATE
TRANSFORM THE COMPENSATOR IMPULSE RESPONSE FUNCTION D(Z)
INTO CONTINUOUS TRANSFER FUNCTION D1(S) INCLUDING ZERO ORDER HOLD
UNIT WHITE NOISE SPECTRAL RESPONSE OF THE COMPENSATOR D1(S)
GAIN OF THE COMPENSATOR GD IS APPROXIMATED UPTO T OF ORDER 3
420   CONTINUE
      A(5,5)=(-1.0,0.0)
```

```
      A(5,6)=(0.40,0.0)
      A(5,7)=(0.60,0.0)
      A(6,5)=(-0.90,0.0)
      A(6,6)=-CMPLX(AR,AI)
      A(6,7)=CONJG(A(6,6))
      AS=-TI*ALOG(REAL(A(5,6)))
      AT=-TI*ALOG(REAL(A(5,7)))
      AU=-TI*ALOG(ABS(REAL(A(6,5))))
      AV=-TI*ALOG(SQRT(AR*AR+AI*AI))
      AW=TI*ATAN(AIMAG(A(6,6))/REAL(A(6,6)))
      A(1,5)=(0.0,0.0)
      A(1,6)=CMPLX(AS,PT)
      A(1,7)=CMPLX(AT,PT)
      A(2,5)=CMPLX(AU,0.0)
      A(2,6)=CMPLX(AV,AW)
      A(2,7)=CONJG(A(2,6))
      A(7,1)=A(5,6)+A(5,7)
      A(7,2)=A(5,6)*A(5,7)
      A(7,3)=A(6,5)+A(6,6)+A(6,7)
      A(7,4)=A(6,5)*A(6,6)+A(6,5)*A(6,7)+A(6,6)*A(6,7)
      A(7,5)=(4.50,0.0)+(4.0/3.0)*A(7,1)+A(7,2)/6.0
      A(7,6)=(4.50,0.0)+(4.0/3.0)*A(7,3)+A(7,4)/6.0
      A(7,7)=A(7,6)/A(7,5)
      GD=A(7,7)
      A(3,5)=A(1,5)*A(1,6)*A(1,7)
      A(3,6)=A(1,5)*A(1,6)+A(1,5)*A(1,7)+A(1,6)*A(1,7)
      A(3,7)=A(1,5)+A(1,6)+A(1,7)
      A(4,5)=A(2,5)*A(2,6)*A(2,7)
      A(4,6)=A(2,5)*A(2,6)+A(2,5)*A(2,7)+A(2,6)*A(2,7)
      A(4,7)=A(2,5)+A(2,6)+A(2,7)
  421 CONTINUE
      GA=REAL(GD)
      DO 422 K=1,NP
      AK=K
      XA(K)=AK/100.0
      W=CMPLX(0.0,TP*XA(K))
      U=CEXP(T*W)
      G=((U+A(6,5))*(U+A(6,6))*(U+A(6,7)))/(W*U*(U+A(5,6))*(U+A(5,7)))
      H=((W+A(2,5))*(W+A(2,6))*(W+A(2,7)))/(W*(W+A(1,6))*(W+A(1,7)))
      XB(K)=REAL(G*CONJG(G))
      XC(K)=GA*GA*REAL(H*CONJG(H))
      XD(K)=ALOG10(XB(K))
      XE(K)=ALOG10(XC(K))
  422 CONTINUE
      WRITE (3,423) GH,GD,((A(K,L),L=1,7),K=1,7)
  423 FORMAT ('0','COEFFICIENTS'/' ',2(E12.5,2X,E12.5,6X)
     1/(' ',4(E12.5,2X,E12.5,6X)/' ',3(E12.5,2X,E12.5,6X)))
      WRITE (3,424) (XA(K),XB(K),XC(K),XD(K),XE(K),K=1,NP)
  424 FORMAT ('0','UNIT WHITE NOISE SPECTRAL RESPONSE'
     1/(F12.3,2X,4(E12.5,2X)))
      CALL GRAPH (NP,XA,XD,K,2,7.0,-4.0,0.4,0.0,10.0,-10.0,XL,YL,GL,DL)
      CALL GRAPH (NP,XA,XE,3,3,0,0,0,0,0,0,0,0,0,0)
CHARACTERISTIC VALUES OF A SEVENTH ORDER SYSTEM
  440 CONTINUE
      N=7
      A(5,1)=A(3,1)*A(3,5)+GH*GD*A(4,1)*A(4,5)
      A(5,2)=A(3,1)*A(3,6)+A(3,2)*A(3,5)+GH*GD*A(4,1)*A(4,6)
```

```
445 FORMAT ('0','B(K,L)'/(' ',2(E12.5,2X,E12.5,6X)))
    WRITE (3,446) (XA(K),XB(K),K=1,NP)
446 FORMAT ('0','UNIT WHITE NOISE SPECTRAL RESPONSE'
   1/(' ',F12.3,2X,E12.5))
    CALL GRAPH (NP,XA,XB,K,2,7.0,4.0,0.4,0.0,2.0E-08,0.0,XL,YL,GL,DL)
    GO TO 200
447 CONTINUE
    STOP
    END
```

```
     1+GH*GD*A(4,5)
      A(5,3)=A(3,1)*A(3,7)+A(3,2)*A(3,6)+A(3,3)*A(3,5)
     1+GH*GD*A(4,1)*A(4,7)+GH*GD*A(4,6)
      A(5,4)=A(3,1)+A(3,2)*A(3,7)+A(3,3)*A(3,6)+A(3,4)*A(3,5)
     1+GH*GD*A(4,1)+GH*GD*A(4,7)
      A(5,5)=A(3,2)+A(3,3)*A(3,7)+A(3,4)*A(3,6)+A(3,5)+GH*GD
      A(5,6)=A(3,3)+A(3,4)*A(3,7)+A(3,6)
      A(5,7)=A(3,4)+A(3,7)
      A(6,1)=GH*A(3,5)*A(4,1)
      A(6,2)=GH*(A(3,5)+A(3,6)*A(4,1))
      A(6,3)=GH*(A(3,6)+A(3,7)*A(4,1))
      A(6,4)=GH*(A(3,7)+A(4,1))
      A(6,5)=GH
      A(6,6)=(0.0,0.0)
      A(6,7)=(0.0,0.0)
      A(7,1)=GH*GD*A(4,1)*A(4,5)
      A(7,2)=GH*GD*(A(4,1)*A(4,6)+A(4,5))
      A(7,3)=GH*GD*(A(4,1)*A(4,7)+A(4,6))
      A(7,4)=GH*GD*(A(4,1)+A(4,7))
      A(7,5)=GH*GD
      A(7,6)=(0.0,0.0)
      A(7,7)=(0.0,0.0)
      B(1,1)=(0.0,0.0)
      B(2,1)=(0.0,0.0)
      B(3,1)=A(6,5)
      B(4,1)=A(6,4)-A(5,7)*B(3,1)
      B(5,1)=A(6,3)-A(5,7)*B(4,1)-A(5,6)*B(3,1)
      B(6,1)=A(6,2)-A(5,7)*B(5,1)-A(5,6)*B(4,1)-A(5,5)*B(3,1)
      B(7,1)=A(6,1)-A(5,7)*B(6,1)-A(5,6)*B(5,1)-A(5,5)*B(4,1)
     1-A(5,4)*B(3,1)
      B(1,2)=(0.0,0.0)
      B(2,2)=(0.0,0.0)
      B(3,2)=A(7,5)
      B(4,2)=A(7,4)-A(5,7)*B(3,2)
      B(5,2)=A(7,3)-A(5,7)*B(4,2)-A(5,6)*B(3,2)
      B(6,2)=A(7,2)-A(5,7)*B(5,2)-A(5,6)*B(4,2)-A(5,5)*B(3,2)
      B(7,2)=A(7,1)-A(5,7)*B(6,2)-A(5,6)*B(5,2)-A(5,5)*B(4,2)
     1-A(5,4)*B(3,2)
441  CONTINUE
      GK=GK*GA
      DO 443 K=1,NP
      AK=K
      XA(K)=AK/100.0
      W=CMPLX(0.0,TP*XA(K))
      U=(W+A(4,1))*(W*W*W+A(4,7)*W*W+A(4,6)*W+A(4,5))
      V=(1.0,0.0)
      DO 442 L=1,N
      V=V*W+A(5,N+1-L)
442  CONTINUE
      U=U/V
      XB(K)=GK*GK*REAL(U*CONJG(U))
443  CONTINUE
      WRITE (3,444) ((A(K,L),L=1,N),K=1,N)
444  FORMAT ('0','A(K,L)'
     1/(' ',4(E12.5,2X,E12.5,6X)/' ',3(E12.5,2X,E12.5,6X)))
      WRITE (3,445) ((B(K,L),L=1,2),K=1,7)
```