AD-A082 622 MISSOURI UNIV-COLUMBIA COLL OF ENGINEERING F/G 17/7

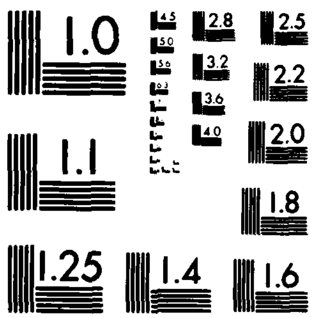
INHERENT ERROR IN ASYNCHRONOUS DIGITAL FLIGHT CONTROLS.(U)

FEB 80 C SLIVINSKY, S PATUMTAWAPIBAL AFOSR-76-2968

UNCLASSIFIED AFOSR-TR-80-[illegible]

1 of 2

AD A082622



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

80-0256

University of Missouri-Columbia
College of Engineering

AD A082622

DTIC
S ELECTE D
APR 2 1980
C

Inherent Error in Asynchronous
Digital Flight Controls
AFOSR-76-2968
Final Technical Report
1 Feb. 1976 - 31 July 1979

DTIC
ELECTE
APR 2 1980
S D
C

Inherent Error in Asynchronous
Digital Flight Controls

AFOSR-76-2968

Final Technical Report
1 Feb. 1976 - 31 July 1979

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

This research was sponsored by the Air Force Office of Scientific Research. The Program Manager was Major Charles L. Nefzger of the Directorate of Mathematical and Information Sciences, Air Force Office of Scientific Research (AFSC), Bolling Air Force Base, D.C. 20332. The Principal Investigator was Prof. Charles Slivinsky, Electrical Engineering Department, University of Missouri-Columbia, Columbia, Missouri 65211.

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFSC)
NOTICE OF TRANSMITTAL TO DDC
This technical report has been reviewed and is
approved for public release IAW AFR 190-12 (7b).
Distribution is unlimited.
A. D. BLOSE
Technical Information Officer

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: 18 19 AFOSR-TR-80-0256
2. GOVT ACCESSION NO.:
3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): 6 INHERENT ERROR IN ASYNCHRONOUS DIGITAL FLIGHT CONTROLS
5. TYPE OF REPORT & PERIOD COVERED: 9 Final Rept. 1 Feb 78-31 Jul 79
6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): 10 Charles Slivinsky, Sudhiporn Patumtawapibal
8. CONTRACT OR GRANT NUMBER(s): 15 AFOSR-76-2968
9. PERFORMING ORGANIZATION NAME AND ADDRESS: University of Missouri - Columbia, College of Engineering, Columbia, MO 65211
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 61102F 16 2304 17 A1
11. CONTROLLING OFFICE NAME AND ADDRESS: Air Force Office of Scientific Research/NM, Bolling AFB, Washington, DC 20332
12. REPORT DATE: 11 February 1980
13. NUMBER OF PAGES: 125
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): 12 127
15. SECURITY CLASS. (of this report): UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Redundancy Management
Digital Flight Control Systems
Digital Voter/Monitor
Comparison Monitor
Inherent Errors
Asynchronous

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Reference 1 provides the background information and results on the asynchronous operation and design of closed-loop digital flight control systems that have dual-redundant digital controllers. In the model use, the digital controllers have the same sample rate but there is a fixed time skew, or offset between their respective sample times. Also, this model requires that the same channel is selected as the output at all times. This latter assumption is roughly equivalent to a channel-voting scheme that selects the upper median (for a four-channel system) or the lower median (for a three or four-channel system) as the

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE 409 476 UNCLASSIFIED

18

20. Abstract cont.

output when the channel outputs are either monotonically increasing or decreasing in time.

Section II below describes the same model as in Reference 1 except that the input to the plant is the difference between the external input (pilot input) and the output of the first controller. An example, with a plot of the steady-state covariances of the errors due to the time skew between controllers, is shown at the end of the Section.

An extension to the model in Section II is developed in Section III. In this extended model, the first channel computes two outputs. The first output is the input to the plant and is exactly the same as the output of the first controller of the model in Section II. The second output, which is an estimate of the output of the second channel, is used to calculate the error due to the time skew between the two controllers. Like the model in SectionII, the second channel computes only one signal. In this model, the inherent errors depend on the difference of the second output of the first channel and the output of the second channel. An example, with a plot of the steady-state covariances of the second output of the first channel and the output of the second channel, is presented at the end of the Section.

An algorithm to estimate the time skew between two asynchronous systems is described in Section IV. The algorithm is based on the model in Section III. The comparison between the new configuration in SectionIII (with the algorithm to estimate the time skew in SectionIV) and the old configuration in Section II is shown in Section VI.

Section V describes the application of the new model in asynchrounous redundant digital flight control systems and Section VII contains the conclusions and summary. General descriptions, flowcharts, user instructions and listings for all the software in this report are shown in Appendices.

TABLE OF CONTENTS

Accession For
NTIS GRA&I
DDC TAB
Unannounced
Justification
By
Distribution/
Availability Codes
Dist. | Avail and/or special
A

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION I

INTRODUCTION

Reference 1 provides the background information and results on the asynchronous operation and design of closed-loop digital flight control systems that have dual-redundant digital controllers. In the model used, the digital controllers have the same sample rate but there is a fixed time skew, or offset between their respective sample times. Also, this model requires that the same channel is selected as the output at all times. This latter assumption is roughly equivalent to a channel-voting scheme that selects the upper median (for a four-channel system) or the lower median (for a three- or four-channel system) as the output when the channel outputs are either monotonically increasing or decreasing in time.

Section II below describes the same model as in Reference 1 except that the input to the plant is the difference between the external input (pilot input) and the output of the first controller. An example, with a plot of the steady-state covariances of the errors due to the time skew between controllers, is shown at the end of the Section.

An extension to the model in Section II is developed in Section III. In this extended model, the first channel computes two outputs. The first output is the input to the plant and is exactly the same as the output of the first controller of the model in Section II. The second output, which is an estimate of the output of the second channel, is used to calculate the error due to the time skew between the two controllers. Like the model in Section II, the second channel computes only one signal. In this model, the inherent errors depend on the difference of the second output of the first channel and the output of the second channel. An example, with a plot of the steady-state covariances of the second output of the first channel and the output of the second channel, is presented at the end of the Section.

An algorithm to estimate the time skew between two asynchronous systems is described in Section IV. The algorithm is based on the model in Section III. The comparison between the new configuration in Section III (with the algorithm to estimate the time skew in Section IV) and the old configuration in Section II is shown in Section VI.

Section V describes the application of the new model in asynchronous redundant digital flight control systems and Section VII contains the conclusions and summary. General descriptions, flow charts, user instructions and listings for all the software in this report are shown in Appendices.

SECTION II

STATE EQUATIONS, COVARIANCE, AND EXAMPLES FOR BASIC MODEL

The model illustrated in Figure 1 is labelled the basic model; the assumptions, techniques, and style of analysis are the foundation for the new model described in Section III of this report. The basic model is similar to the model in Reference 1, except that the input to the plant is the difference between the external input (pilot input) and the output of the first controller, while in the model of Reference 1, the input to the plant is the output of the first controller.

1. SYSTEM CONFIGURATION AND THE DYNAMIC EQUATION

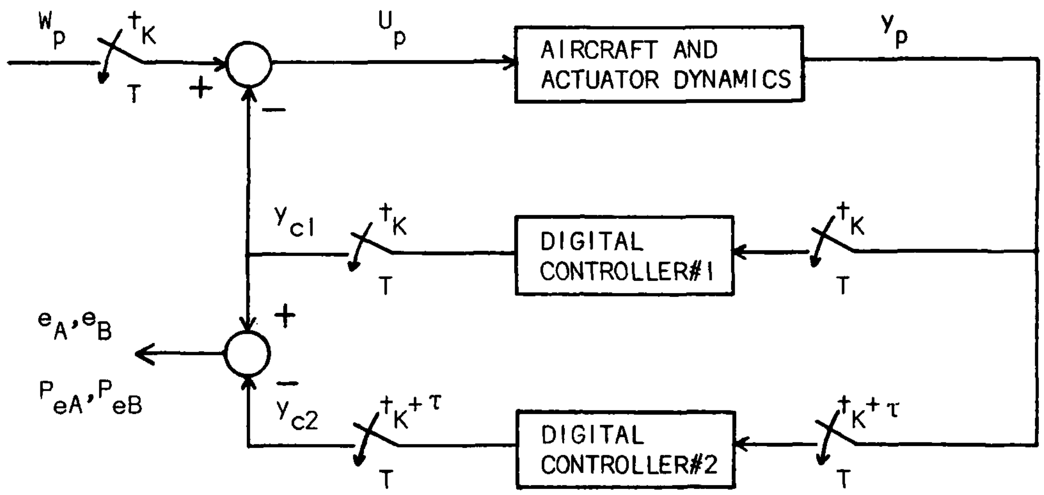
The system configuration for this closed-loop dynamic system consists of a continuous-time plant and dual-redundant, single-rate discrete-time controllers. The plant output is sampled by each of the controllers, using a common sample period but having a fixed time skew between them. The output of one of the controllers serves as the piecewise-constant input to the plant, along with an external input.

The plant equations include aircraft, sensor, and actuator dynamics, as well as any dynamics associated with the pilot input and wind-gust model input. The plant equations are assumed to be in the form

$$\dot{x}_p = A_p x_p + B_p u_p \tag{2-1}$$

$$y_p = C_p x_p \tag{2-2}$$

where



T : SAMPLE PERIOD

τ : SKEW TIME

FIGURE 1 : BLOCK DIAGRAM FOR THE BASIC MODEL

x_p = plant state vector (n_p x 1)

u_p = plant input vector (n_{up} x 1)

w_p = external input vector (n_{up} x 1)

y_{c1}= controller 1 output vector (n_{up} x 1)

y_p = plant output vector (n_{op} x 1)

A_p = plant state matrix (n_p x n_p)

B_p = plant input matrix (n_p x n_{up})

C_p = plant output matrix (n_{op} x n_p)

The solution to Equation 1 is

$$x_p(t) = \phi(t,t_0)x_p(t_0) + \int_{t_0}^{t} \phi(t,s)B_p u_p(s)ds \tag{2-3}$$

where $\phi(t,t_0)$ is the state transition matrix and for constant A_p is given by

$$\phi(t,t_0) = \exp\,[A_p(t-t_0)] \tag{2-4}$$

The plant input $u_p(t)$ is piecewise-constant over a given sampling interval; i.e.,

$$u_p(t) = u_p(t_k) \qquad t_k \leq t \leq t_{k+1}$$

and so for $t = t_k$, $t = t_{k+1}$, and $t_{k+1} - t_k = T$, the second term in Equation (2-3) can be written as

$$\int_{t_k}^{t_{k+1}} \phi(t_{k+1},s)B_p u_p(s)ds = \psi(t_{k+1},t_k)u_p(t_k) \tag{2-5}$$

where

$$\psi(t_{k+1},t_k) = \int_0^T \exp\,[A_p(t)]B_p dt \tag{2-6}$$

Substitution of (2-6) into (2-3) gives

$$x_p(t_{k+1}) = \phi(t_{k+1},t_k)x_p(t_k) + \psi(t_{k+1},t_k)u_p(t_k) \tag{2-7}$$

for k = 0, 1, . . .

The discrete-time equations for controller #1 are

$$x_{c1}(t_{k+1}) = F_c x_{c1}(t_k) + G_c u_{c1}(t_k) \tag{2-8}$$

$$y_{c1}(t_k) = H_c x_{c1}(t_k) + E_c u_{c1}(t_k) \tag{2-9}$$

for k = 0, 1, . . .

and for controller #2

$$x_{c2}(t_{k+1} + \tau) = F_c x_{c2}(t_k + \tau) + G_c u_{c2}(t_k + \tau) \tag{2-10}$$

$$y_{c2}(t_k + \tau) = H_c x_{c2}(t_k + \tau) + E_c u_{c2}(t_k + \tau) \tag{2-11}$$

for k = 0, 1, . . .

where

x_{c1} = controller 1 state vector (n_c x 1)

y_{c1} = controller 1 output vector (n_{up} x 1)

x_{c2} = controller 2 state vector (n_c x 1)

y_{c2} = controller 2 output vector (n_{up} x 1)

F_c = controller state matrix (n_c x n_c)

G_c = controller control input matrix ($n_c \times n_{op}$)

H_c = controller output matrix (states) ($n_{up} \times n_c$)

E_c = controller output matrix (inputs) ($n_{up} \times n_{op}$)

The plant (aircraft, actuator, and sensor dynamics) and the controllers are related by the equations

$$u_p(t_k) = w_p(t_k) - y_{c1}(t_k) \tag{2-12}$$

$$u_{c1}(t_k) = y_p(t_k) \tag{2-13}$$

$$u_{c2}(t_k + \tau) = y_p(t_k + \tau) \tag{2-14}$$

Substitution of Equation (2-12) into (2-7) gives

$$x_p(t_{k+1}) = \phi(t_{k+1},t_k)x_p(t_k) + \psi(t_{k+1},t_k)[w_p(t_k) - y_{c1}(t_k)] \tag{2-15}$$

and

$$y_p(t_{k+1}) = C_p x_p(t_{k+1}) \tag{2-16}$$

The quantity $x_p(t_k + \tau)$ can be written using the solution to Equation (2-7) as

$$x_p(t_k + \tau) = \phi(t_k + \tau,t_k)x_p(t_k) + \psi(t_k + \tau,t_k)[w_p(t_k) - y_{c1}(t_k)] \tag{2-17}$$

and

$$y_p(t_k + \tau) = C_p x_p(t_k + \tau) \tag{2-18}$$

The piecewise-constant inherent error $e(t)$ is written in two parts, $e_A(t)$ and $e_B(t)$ as

$$e_A(t) = y_{c1}(t_k) - y_{c2}(t_k + \tau) \qquad (2\text{-}19)$$

for $t_k + \tau \leq t < t_{k+1}$, $0 \leq \tau < T$, $k = 0, 1, \ldots$, and

$$e_B(t) = y_{c1}(t_{k+1}) - y_{c2}(t_k + \tau) \qquad (2\text{-}20)$$

for $t_{k+1} \leq t < t_{k+1} + \tau$, $0 < \tau \leq T$, $k = 0, 1, \ldots$

2. COVARIANCE ANALYSIS

Let the input $w_p(t_k)$ be a Gaussian white noise random process with zero mean, which is independent of $x(0)$ (Reference 3). Then, let P_{eA} and P_{eB} be the covariance of the errors e_A and e_B, respectively. Thus,

$$P_{eA}(t) = E[e_A(t)\, e_A^T(t)] \qquad (2\text{-}21)$$

for $t_k + \tau \leq t < t_{k+1}$, $0 \leq \tau < T$, $k = 0, 1, \ldots$, and

$$P_{eB}(t) = E[e_B(t)\, e_B^T(t)] \qquad (2\text{-}22)$$

for $t_{k+1} \leq t < t_{k+1} + \tau$, $0 < \tau \leq T$, $k = 0, 1, \ldots$

Since the input w_p is a Gaussian white noise and the controllers are discrete time controllers, the inherent errors e_A and e_B will be random variables $\{e_{A1}, e_{A2}, \ldots, e_{AN}\}$, and $\{e_{B1}, e_{B2}, \ldots, e_{BN}\}$, as

e_{A1} is the error in the interval $t_k + \tau$ and t_{k+1}

e_{A2} is the error in the interval $t_{k+1} + \tau$ and t_{k+2}

.

$\cdot$

$\cdot$

e_{AN} is the error in the interval $t_{k+N-1} + \tau$ and t_{k+N}

and

e_{B1} is the error in the interval t_{k+1} and $t_{k+1} + \tau$

e_{B2} is the error in the interval t_{k+2} and $t_{k+2} + \tau$

$\cdot$

$\cdot$

$\cdot$

e_{BN} is the error in the interval t_{k+N} and $t_{k+N} + \tau$

Let $\bar{e}_A$ and $\bar{e}_B$ represent the sample means based on N samples of $\{e_{A1}, e_{A2}, \ldots, e_{AN}\}$ and $\{e_{B1}, e_{B2}, \ldots, e_{BN}\}$. Thus,

$$\bar{e}_A = \frac{1}{N}[e_{A1} + e_{A2} + \ldots + e_{AN}] \tag{2-23}$$

and

$$\bar{e}_B = \frac{1}{N}[e_{B1} + e_{B2} + \ldots + e_{BN}] \tag{2-24}$$

From these two equations (2-23) and (2-24), the sample covariances of the errors based on the interval from t_k to t_{k+N} are

$$P_{eA} = \frac{1}{N}\sum_{i=1}^{N}[e_{Ai} - \bar{e}_A][e_{Ai} - \bar{e}_A]^T \tag{2-25}$$

and

$$P_{eB} = \frac{1}{N}\sum_{i=1}^{N}[e_{Ai} - \bar{e}_A][e_{Ai} - \bar{e}_A]^T \tag{2-26}$$

For the steady state sample covariance of errors, k is the value when the system is in steady-state.

3. EXAMPLE

As the example, consider the closed-loop system shown in Figure 2 with a second-order plant and a first-order controller. Assume w_p to be Gaussian white noise with zero mean and variance = 1 and let the sample period T equal 0.0125 seconds.

From the block diagram in Figure 2, the plant is described by

$$A_p = \begin{vmatrix} 0 & 1 \\ 0 & -10 \end{vmatrix}$$

$$B_p = \begin{vmatrix} 0 \\ 200 \end{vmatrix}$$

and

$$C_p = \begin{vmatrix} 1 & 0 \end{vmatrix}$$

The description of the digital controllers is obtained by starting with the Laplace transform transfer function of an analog controller and performing the Tustin transformation (also called the bilinear transformation) to obtain the z transformation and the discrete-time state equations.

The continuous controller transfer function is $\frac{1 + 0.03s}{1 + 0.02s}$.

The substitution $s = \frac{2}{T}\,\frac{z-1}{z+1}$ performs the Tustin transformation.

This yields

$$\frac{1 + 0.03s}{1 + 0.02s} \longrightarrow 1 + \frac{0.03 \times \frac{2}{T}\,\frac{z-1}{z+1}}{0.02 \times \frac{2}{T}\,\frac{z-1}{z+1}}$$

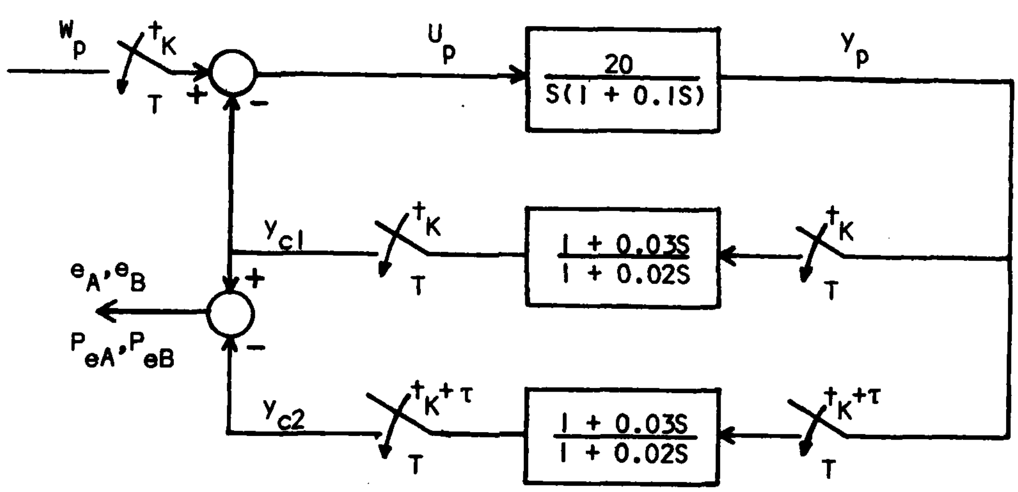


FIGURE 2 : EXAMPLE

Then, the transfer function can be written as

$$\frac{X_{c1}}{Y_p} = \frac{T + 0.06}{T + 0.04} - \frac{\dfrac{0.04T}{T^2 + 0.08T + 0.0016}}{z - \dfrac{0.04 - T}{0.04 + T}}$$

where $X_{c1}(z)$ is the controller output and $Y_p(z)$ is the controller input, which is also the plant output. A block diagram for digital controller 1 appears in Figure 3.

The state equations corresponding to Figure 3 are

$$x_{c1}(t_{k+1}) = F_c x_{c1}(t_k) + G_c y_p(t_k)$$

$$y_{c1}(t_k) = H_c x_{c1}(t_k) + E_c y_p(t_k)$$

where $t_{k+1} - t_k = T$ and

$$F_c = \frac{0.04 - T}{0.04 + T}$$

$$G_c = \frac{0.04T}{T^2 + 0.08T + 0.0016}$$

$$H_c = 1$$

$$E_c = \frac{0.06 + T}{0.04 + T}$$

The state transition matrix $\phi(t_{k+1},t_k)$ from equation (2-4) is

$$\phi(t_{k+1},t_k) = \begin{vmatrix} 1 & \frac{1}{10} - \frac{1}{10} e^{-10T} \\ 0 & e^{-10T} \end{vmatrix}$$

The steady state sample variance of errors P_{eAss} and P_{eBss} from Appendix B are plotted in Figure 4 as a funtion of τ. The diagrams to

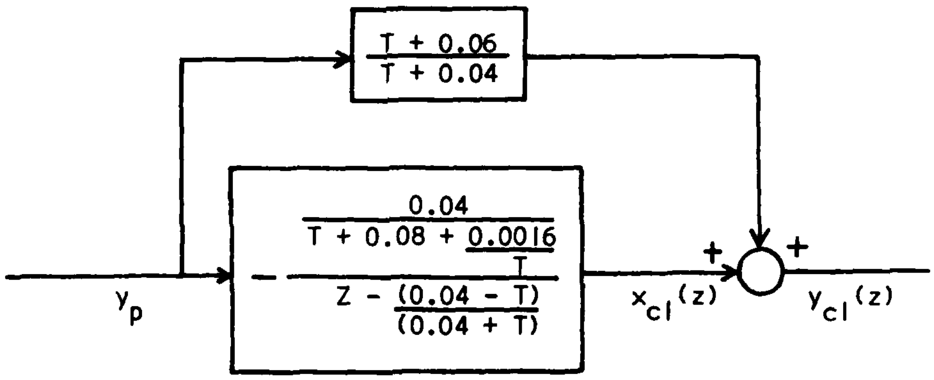


FIGURE 3 : BLOCK DIAGRAM OF DIGITAL CONTROLLER I

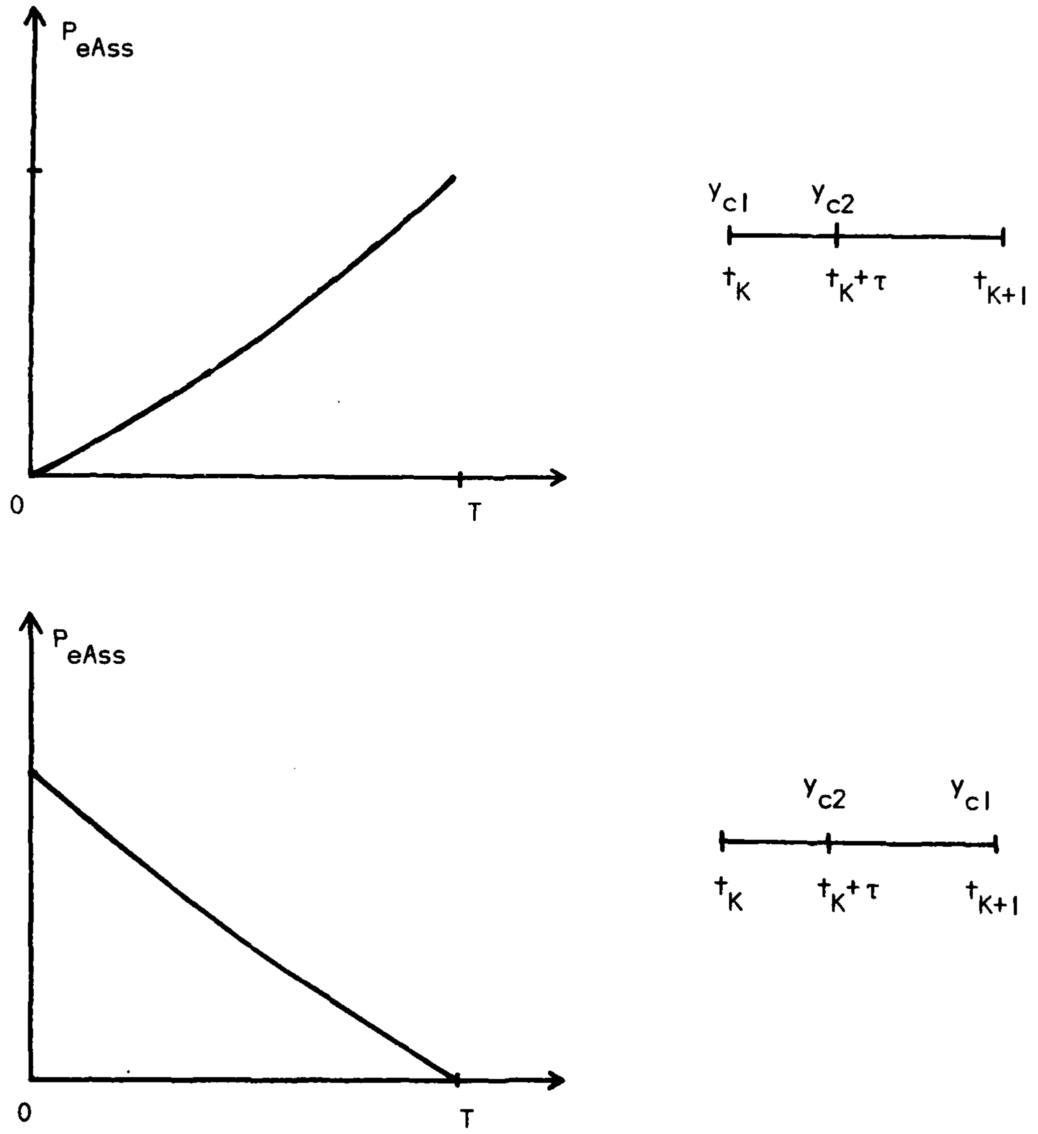


FIGURE 4 : P_{eAss} AND P_{eAss} OF THE BASIC MODEL FOR THE EXAMPLE IN FIGURE 2

the right of each plot show the times at which the controller outputs are sampled for the calculation of e_A and e_B. The sample variances are largest when the times at which y_{c1} and y_{c2} change are farthest apart, as expected. The results indicate that some combination of the two measurements of the channel inherent errors may be less affected by the amount of skews than either error taken alone.

SECTION III

STATE EQUATIONS, COVARIANCE, AND EXAMPLE FOR NEW MODEL

According to the example of the basic model in Section II, the tolerance value for two-channel operation is the maximum value of the steady-state sample covariance P_{eAss} and is reached at $\tau = T$. If the steady-state covariance is greater than the tolerance value, the error is due to a failed channel; if it is less than the tolerance, the error is the inherent error due to sampling skew. However, the above choice for the tolerance value may not be the best one to distinguish the inherent error from the error due to a failed channel.

For example, if the time skew of the second channel is small and the sample variance is greater than the average variance for that time skew, then it is likely that there is a failed channel. If that sample variance, which is greater than the average variance for that skew time, is less than the tolerance value (P_{eAss} at $\tau = T$), the basic model in Section II would indicate no failure.

To reduce the effect above, one possibility is for channel 1 to compute an approximation to the current output of channel 2. The difference between the true output of channel 2 and the estimated output of channel 2 will be close to zero, assuming that the estimate is a good one. The tolerance value for this approach can be a small value and it is equal to the maximum steady-state sample covariance of the difference between the estimated value and the actual value. The model described in this section estimates the output of channel 2 from the input of channel 1 and the details of this approach are in the following subsections.

1. SYSTEM CONFIGURATION AND DYNAMIC EQUATIONS

The closed-loop system configuration for this new model appears in Figure 5, and is almost the same as the basic model in Section II. The output y_p of the plant, which consists of the aircraft, the sensor, and the control actuator dynamics, is sampled by each of two digital controllers. They use the same fixed sample period T, but there is a constant skew τ between the starting points of the two samplers.

In Figure 5, the input of channel 1, $y_p(t_k)$, is used to compute $y_{c1}(t_k)$ and $y^*_{c2}(t_k + \tau^*)$, an estimate of $y_{c2}(t_k + \tau)$ (τ^* is an estimate of τ). The difference between the external input w_p and y_{c1} is the input to the plant. The block named OBSERVER (See Appendix D for details) computes $x^*_p(t_k)$, an estimate of $x_p(t_k)$ based on the two quantities $y_p(t_{k-1})$ and $u_p(t_{k-1})$, which are the previous inputs of channel 1 and the plant, respectively. $y^*_p(t_k + \tau^*)$ is an estimate of $y_p(t_k + \tau)$ calculated from τ^*, $x^*_p(t_k)$, and $y_{c1}(t_k)$; this estimate is an input to the block named 2nd DIGITAL CONTROLLER #1. Finally, $y^*_{c2}(t_k + \tau^*)$ is computed.

As in the basic model, the plant equations include the aircraft, sensor, and actuator dynamics, as well as any dynamics associated with the pilot input and the wind-gust model input. The plant equations are assumed to be in the form

$$\dot{x}_p = A_p x_p + B_p u_p \tag{3-1}$$

$$y_p = C_p x_p \tag{3-2}$$

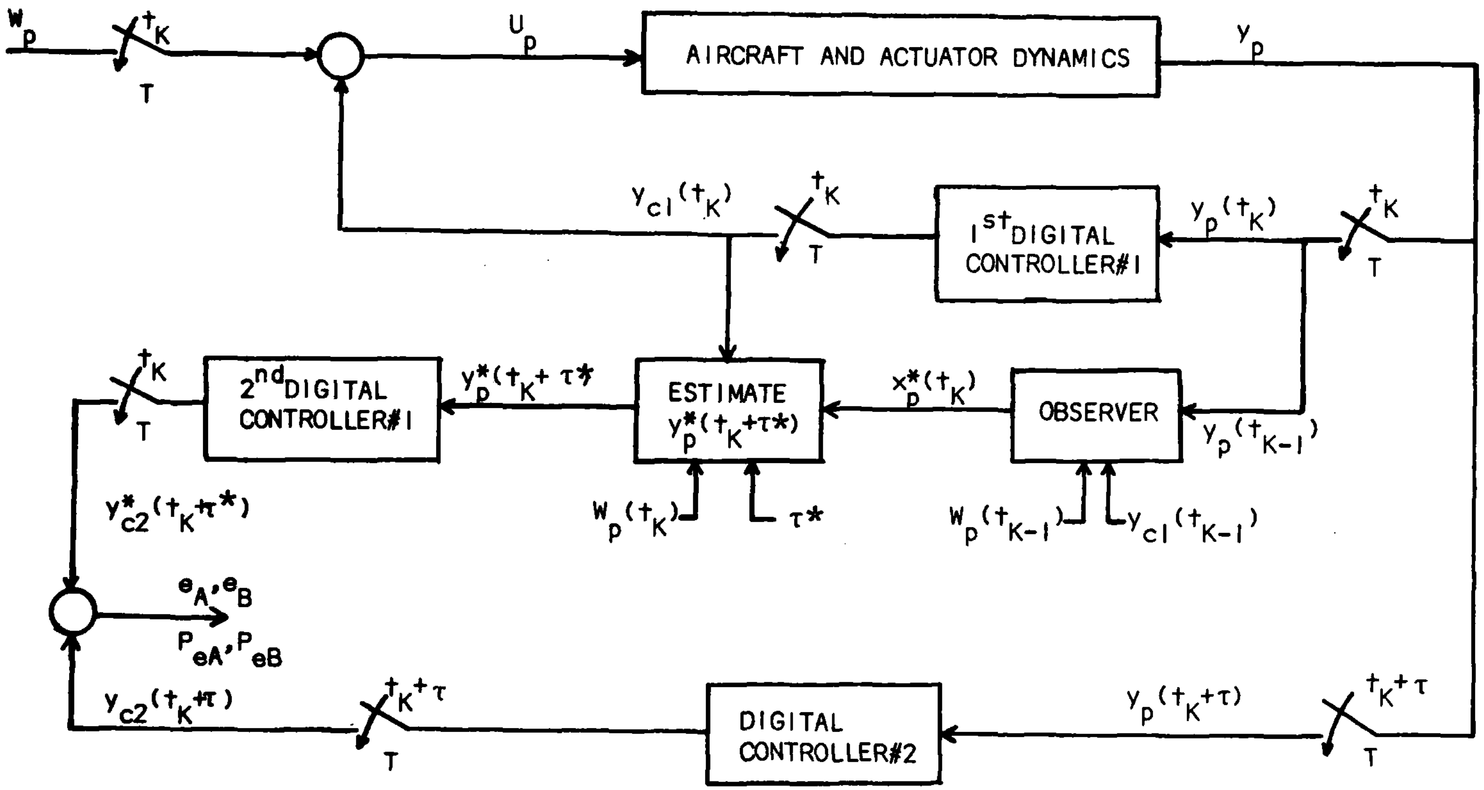


FIGURE 5 : BLOCK DIAGRAM OF THE NEW MODEL

Without showing the details of the derivation (the details are available in Section II), the solution of equation (3-1) is

$$x_p(t_{k+1}) = \phi(t_{k+1}, t_k)x_p(t_k) + \psi(t_{k+1}, t_k)u_p(t_k) \tag{3-3}$$

where

$$u_p(t_k) = w_p(t_k) - y_{c1}(t_k) \tag{3-4}$$

The first function of channel 1 is to compute the signal that is fed back to the plant according to the equation

$$y_{c1}(t_k) = H_c x_{c1}(t_k) + E_c y_p(t_k) \tag{3-5}$$

and

$$x_{c1}(t_{k+1}) = F_c x_{c1}(t_k) + G_c y_p(t_k) \tag{3-6}$$

for k = 0, 1, 2, . . .

The second function is to compute the signals that are used to calculate the inherent error according to

$$y^*_{c2}(t_k + \tau^*) = H_c x^*_{c2}(t_k + \tau^*) + E_c y^*_p(t_k + \tau^*) \tag{3-7}$$

and

$$x^*_{c2}(t_{k+1} + \tau^*) = F_c x^*_{c2}(t_k + \tau^*) + G_c y^*_p(t_k + \tau^*) \tag{3-8}$$

for k = 0, 1, 2, . . . Here,

$y^*_{c2}(t_k + \tau^*)$ is the estimate of $y_{c2}(t_k + \tau)$

$x^*_{c2}(t_k + \tau^*)$ is the estimate of $x_{c2}(t_k + \tau)$

and

$y^*_p(t_k + \tau^*)$ is the estimate of $y_p(t_k + \tau)$

Note: All * variables are in channel 1 and the computations are done at the times t_k, t_{k+1}, t_{k+2}, . . . , instead of $t_k + \tau$, $t_{k+1} + \tau$, $t_{k+2} + \tau$, . . .

Channel 2 computes the signal that is to calculate the inherent error according to

$$y_{c2}(t_k + \tau) = H_c x_{c2}(t_k + \tau) + E_c y_p(t_k + \tau) \tag{3-9}$$

and

$$x_{c2}(t_{k+1} + \tau) = F_c x_{c2}(t_k + \tau) + G_c y_p(t_k + \tau) \tag{3-10}$$

From equation (3-3), we can write the equation of $x_p(t_k + \tau)$ as

$$x_p(t_k + \tau) = \phi(t_k + \tau, t_k)x_p(t_k) + \psi(t_k + \tau, t_k)u_p(t_k) \tag{3-11}$$

and the input of controller #2 is

$$y_p(t_k + \tau) = C_p x_p(t_k + \tau) \tag{3-12}$$

From Appendix D, the equation of the observed state $x_p^*(t_{k+1})$ is

$$x_p^*(t_{k+1}) = \phi(t_{k+1}, t_k)x_p^*(t_k) + \psi(t_{k+1}, t_k)u_p(t_k)$$
$$+ \psi_1(t_{k+1}, t_k)y_p(t_k) \tag{3-13}$$

where

$x_p^*(t_k)$ is the estimate of $x_p(t_k)$

and

$$\psi_1(t_{k+1}, t_k) = \int_0^T \mathrm{EXP}(A_p(t))G_e dt$$

where G_e is the feedback matrix of the observer which y_p^* will approach y_p.

From equation (3-9), the equation of $x_p^*(t_k + \tau^*)$ can be written as

$$x_p^*(t_k + \tau^*) = \phi(t_k + \tau^*, t_k)x_p^*(t_k) + \psi(t_k + \tau^*, t_k)u_p(t_k) \qquad (3\text{-}15)$$

where τ^* is the estimate of τ.

In computing $y_{c2}^*(t_k + \tau^*)$, the variable $y_p^*(t_k + \tau^*)$ is calculated from the equation

$$y_p^*(t_k + \tau^*) = C_p x_p^*(t_k + \tau^*) \qquad (3\text{-}16)$$

The piecewise-constant inherent error e(t) is written in two parts, $e_A(t)$ and $e_B(t)$ as

$$e_A(t) = y_{c2}^*(t_k + \tau^*) - y_{c2}(t_k + \tau) \qquad (3\text{-}17)$$

for $t_k + \leq t < t_{k+1}$, $0 \leq \tau < T$, $k = 0, 1, \ldots$, and

$$e_B(t) = y_{c2}^*(t_{k+1} + \tau^*) - y_{c2}(t_k + \tau) \qquad (3\text{-}18)$$

for $t_{k+1} \leq t < t_{k+1} + \tau$, $0 < \tau \leq T$, $k = 0, 1, \ldots$

2. COVARIANCE ANALYSIS

As in the basic model, w_p is the Gaussian white noise, which has zero mean and variance = 1. Thus, the samples covariance of errors e_A and e_B are

$$P_{eA}(t) = \frac{1}{N} \sum_{i=1}^{N} (e_{Ai} - \bar{e}_A)(e_{Ai} - \bar{e}_A)^T \tag{3-19}$$

for $t_k + \tau \leq t < t_{k+1}$, $0 \leq \tau < T$, $k = 0, 1, \ldots$, and

$$P_{eB}(t) = \frac{1}{N} \sum_{i=1}^{N} (\bar{e}_{Ai} - \bar{e}_A)^T \tag{3-20}$$

for $t_{k+1} \leq t < t_{k+1} + \tau$, $0 < \tau \leq T$, $k = 0, 1, \ldots$

3. EXAMPLE

As the example, consider the system in Figure 2 of Section II. P_{eAss} and P_{eBss} of equations (3-19) and (3-20) are plotted as a function of τ for $\tau^* = 0$, T/5, 2T/5, 3T/5, 4T/5, and T respectively in Figure 6 (from Appendix E). The diagrams to the right of each plot show the times at which the controller output are used for the calculation of e_A (eq. 3-17) and e_B (eq. 3-18). The sample variance (P_{eAss}) are largest when τ^* is farthest apart from τ (or $y^*_{c2}(t_k + \tau^*)$ is farthest apart from $y_{c2}(t_k + \tau)$) as expected.

From the plot, if τ^* is equal to τ, the inherent error (e_A) of this model is zero and the main disadvantage of the asynchronous operation will be eliminated. If τ^* is close to τ, the inherent error (e_A) is a small value. Then the deficiency of the basic model, which is described at the beginning of this section, will be reduced. According to the plot, the sample variance of e_A (P_{eAss}) is directly proportional to the difference between τ and τ^*. Then τ can be estimated by comparing the sample variances of e_A for the values of τ^* in [0, T] and τ^* which corresponds to the smallest covariance will be the estimate of τ. The next section will describe the detail of the algorithm for estimating τ by using the result discussed above.

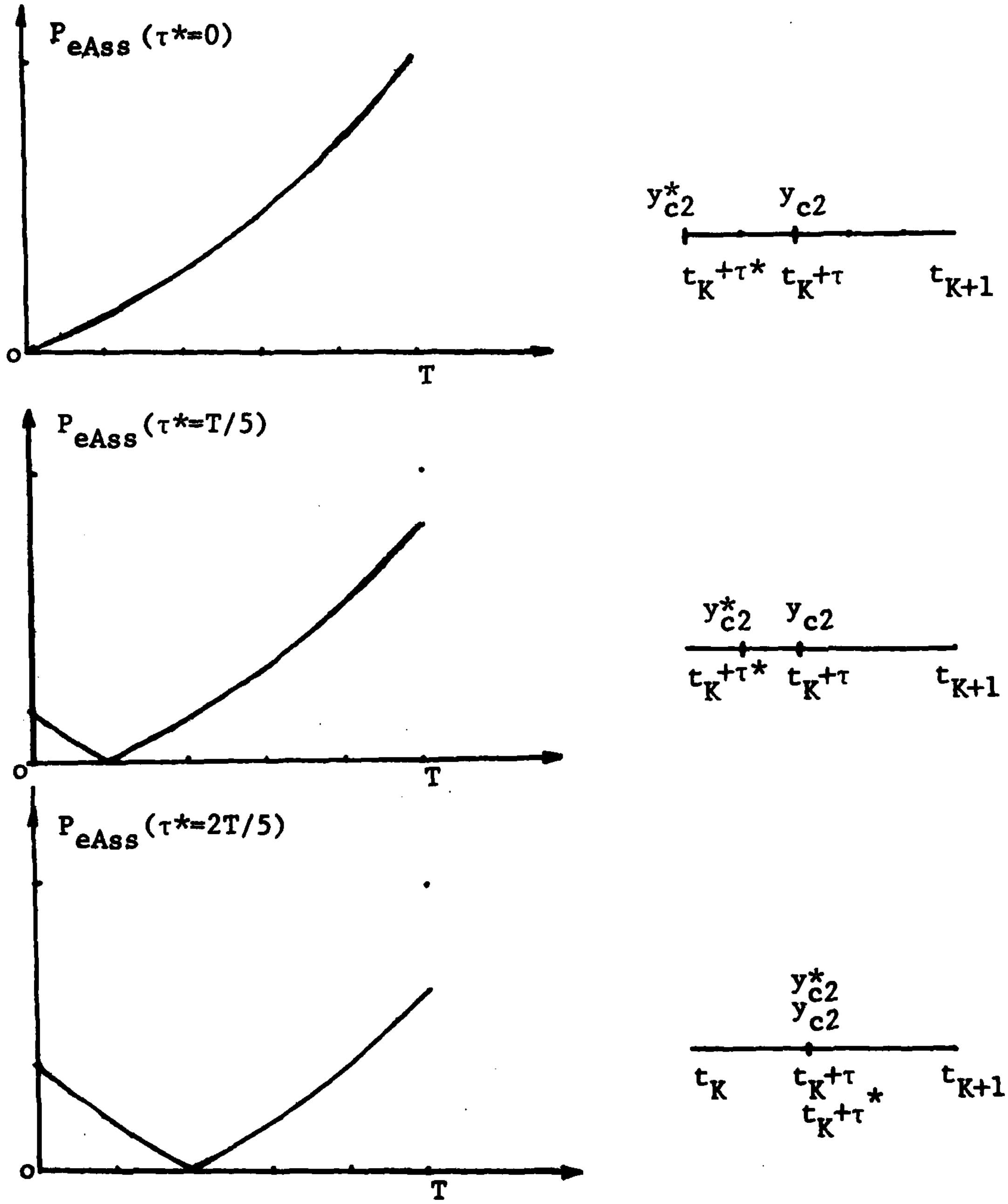


FIGURE 6 : P_{eAss} OF THE NEW MODEL FOR THE EXAMPLE IN FIGURE 2

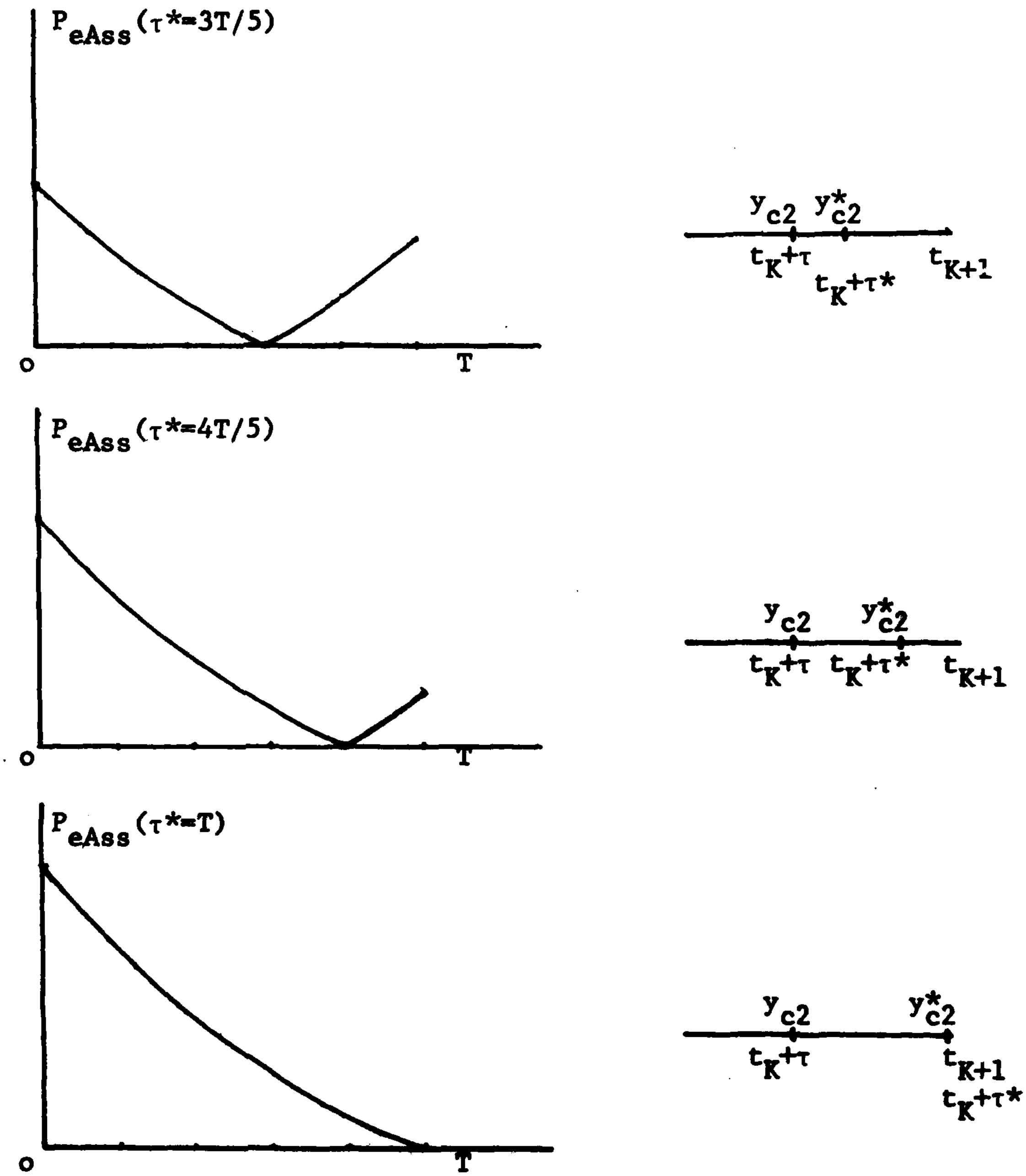


FIGURE 6 : (cont.)

SECTION IV

ALGORITHM FOR THE ESTIMATOR τ^*

According to the example in Section III, the steady-state sample covariance of e_A (eq. 3-19) appears to be directly proportional to the difference between τ and τ^*. That is, when the difference between τ and τ^* is large, the steady-state sample covariance of e_A is large; when the difference is small, the covariance is small; and when the difference is zero, the covariance is zero. This relationship is the basis for the technique, described in this section, to estimate τ^*. The technique uses the model of Figure 7 and the assumption that the steady-state sample covariance of e_A depends on the difference between τ and τ^*.

1. DESCRIPTION OF THE ALGORITHM

The basic procedure for estimating τ^* by using the model of Figure 7 is to change τ^* in an iterative manner until the smallest covariance of e_A is obtained. Let the single variable NT be the number of subintervals in the interval (0,T) so that the length of each subinterval in (0,T) is equal to $\frac{T}{NT}$ and (0,T) is divided into 0, T, . . . , $\frac{NT - 1}{NT} \times T$, T. Let 0 and T be the first lower and upper limits in which τ lies. Let τ^* be the estimate of τ. By comparing the steady-state sample covariance of e_A when τ^* is the midvalue between the lower and the upper limits and that of e_A when τ^* is the value which is greater than the midvalue by $\frac{T}{NT}$, one can determine whether to increase or to decrease τ^*, to reduce the steady-state sample covariance. If the previous steady-state sample covariance of e_A (eq. 3-19) when τ^* is

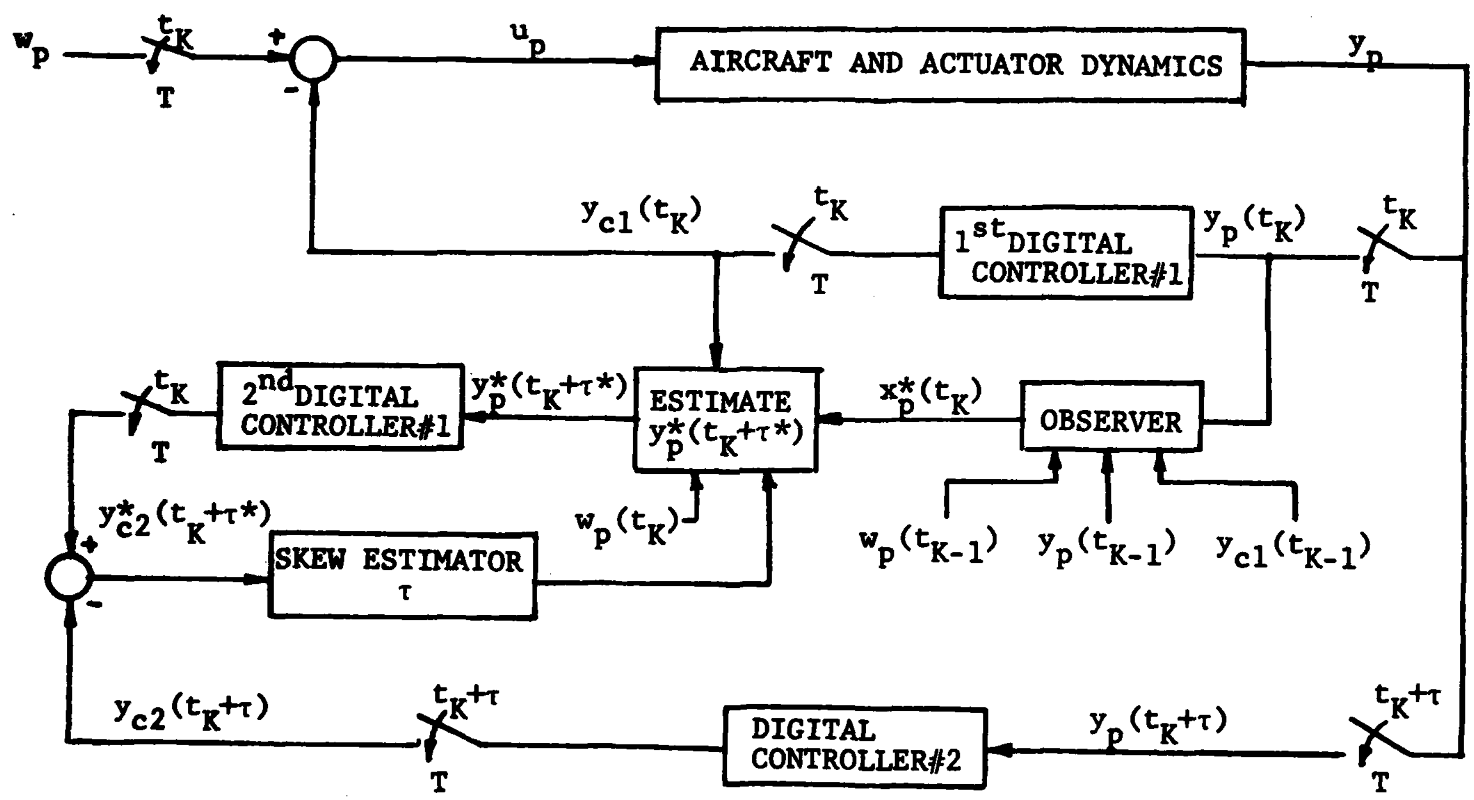


FIGURE 7 : THE MODEL FOR ESTIMATING τ

the midvalue is less than the current steady-state sample covariance of e_A, when τ^* is greater than the midvalue by $\frac{T}{NT}$, then τ must be between the lower limit and the midvalue. Therefore, this midvalue is selected as the new upper limit of the new interval of τ while the lower limit is unchanged. If the previous steady-state sample covariance of e_A is greater than the current steady-state sample covariance of e_A, then the lower limit is updated with the midvalue while the upper limit is maintained. (Note that the values of the lower limits, the upper limits, and the midvalues are restricted to the values 0, $\frac{T}{NT}$, $\frac{2T}{NT}$, . . . , T.) This procedure is repeated until the new midvalue differs from the previous midvalue by $\frac{T}{NT}$. Then the resulting interval is the smallest interval in which τ lies and any value of τ^* in this smallest interval can be used as the estimate of τ.

When the smallest interval in which τ lies is obtained, the lower limit or the upper limit will be the previous midvalue. Then the last current midvalue will differ from the lower or the upper limits by $\frac{T}{NT}$. However, the midvalue is not necessarily equal to the exact midpoint between the lower and the upper limits. Figure 8 shows the two possible locations of the midvalue. (Note: the midvalue in this report is the average value of the lower and upper limits in which the average value is a truncated-integer division.) Since the average value of the lower and upper limits is a truncated-integer division, then there are three smallest intervals of τ which is shown in Figure 9.

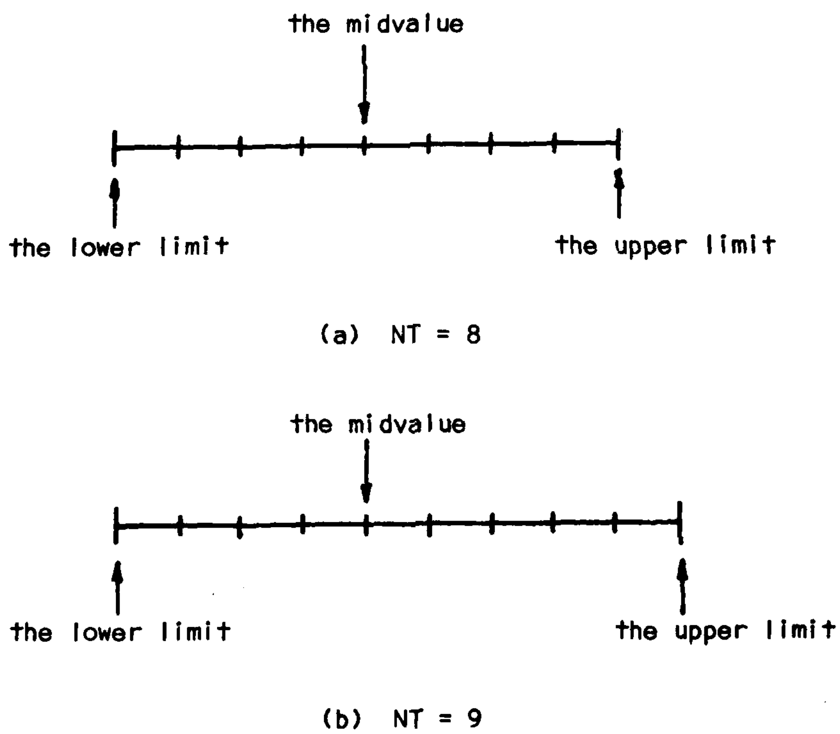


FIGURE 8 : TWO POSSIBLE LOCATIONS OF THE MIDVALUE

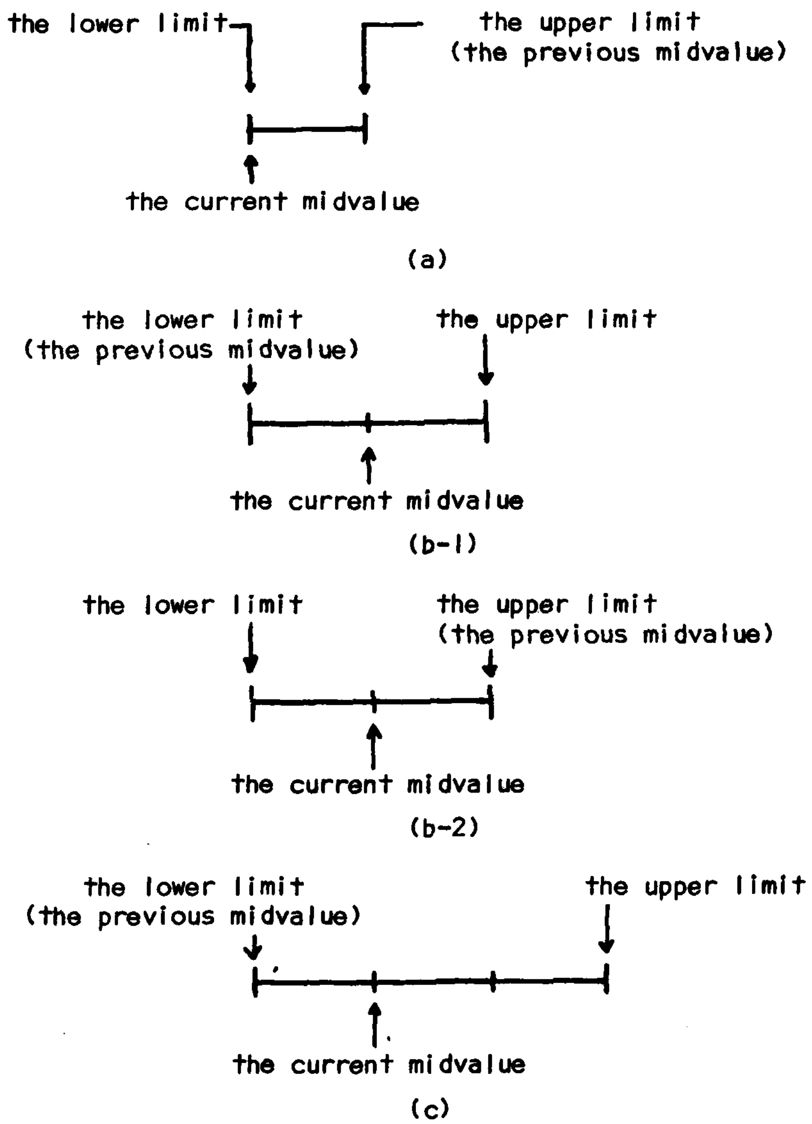


FIGURE 9 : THREE POSSIBLE SMALLEST INTERVALS OF τ

This technique is similar to the 'HALF-INTERVAL SEARCH' which is a method for obtaining an approximate solution to an equation $f(x) = 0$ and it is available in almost every numerical analysis book.

A flowchart of this procedure appears in Figure 10. Before discussing the flowchart, let us define the variables which are used in this flowchart.

As mentioned before, NT is the number of subintervals in (0,T) and instead of using the real values of the subintervals in (0,T); namely, $\frac{(1-1) \times T}{(NT1-1)}$, $\frac{(2-1) \times T}{(NT1-1)}$, . . . , $\frac{(NT1-1)}{(NT1-1)} \times T$, it is more convenient to refer to the numbers 1, 2, . . . , NT1. N4 and N5 are the integers which represent the lower and upper limits of τ respectively. The lower and the upper limits of the interval which τ lies. N3 is the truncated-integer midvalue of N4 and N5.

NTAU1 and NTAU2 are the integers which represent τ^* at the updated midvalue and the previous midvalue respectively. NTAU3 is also the integers which represents τ^* at the midvalue plus one. PEASS1 and PEASS2 are defined to be the steady-state sample covariances of e_A which correspond to NTAU1 and NTAU3 respectively.

The first step of the flowchart shows the initialization of the key variables.

The second step shows the computation of the covariance of e_A when τ^* is equal to the midvalue of the interval. NTAU1 represents the midvalue and PEASS1 is the corresponding covariance.

The third step provides the decision used in terminating the routine. This routine will terminate when the current midvalue (NTAU1)

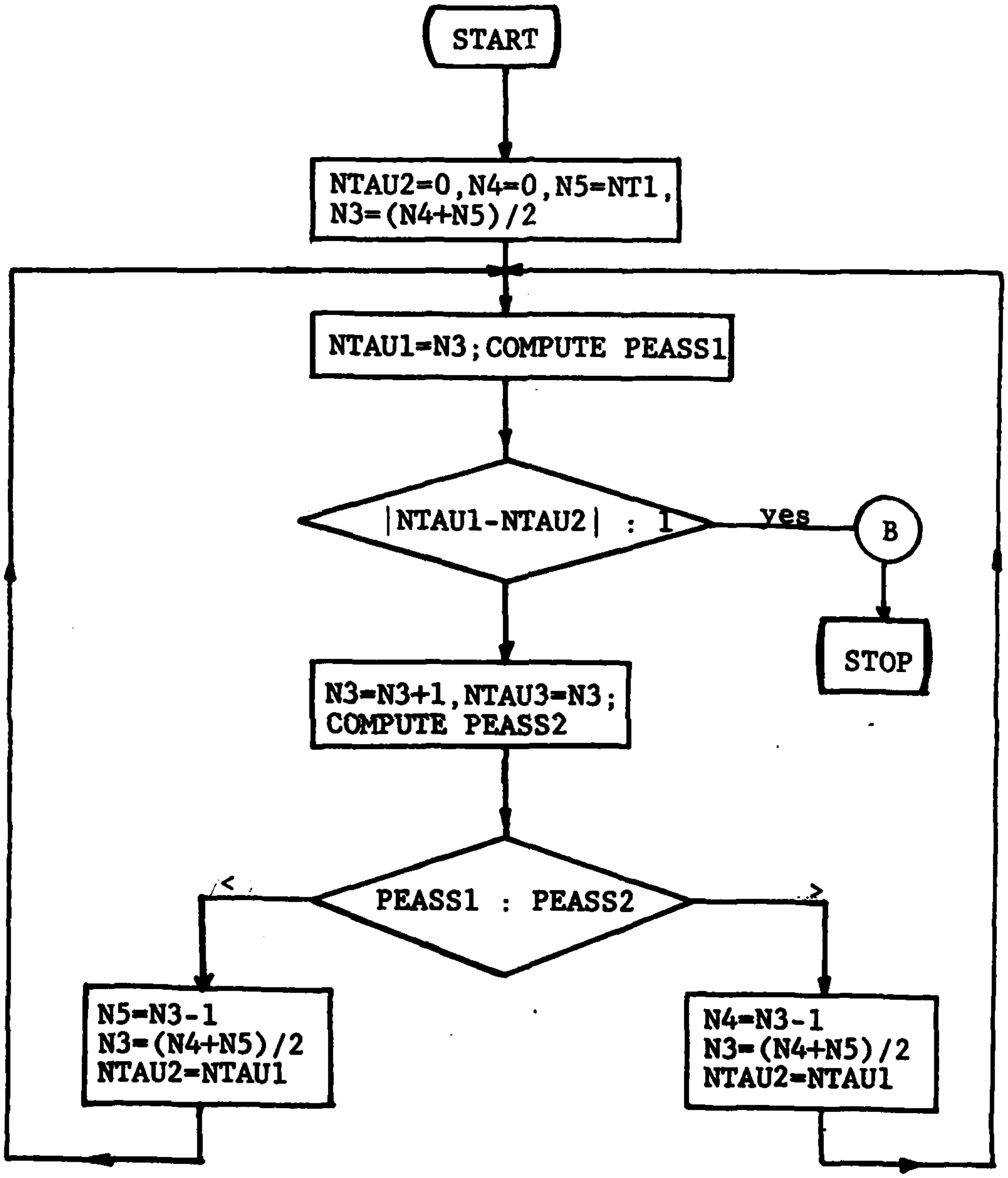


FIGURE 10 : FLOWCHART OF THE DETAILS OF THE PROCEDURE FOR ESTIMATING τ^*

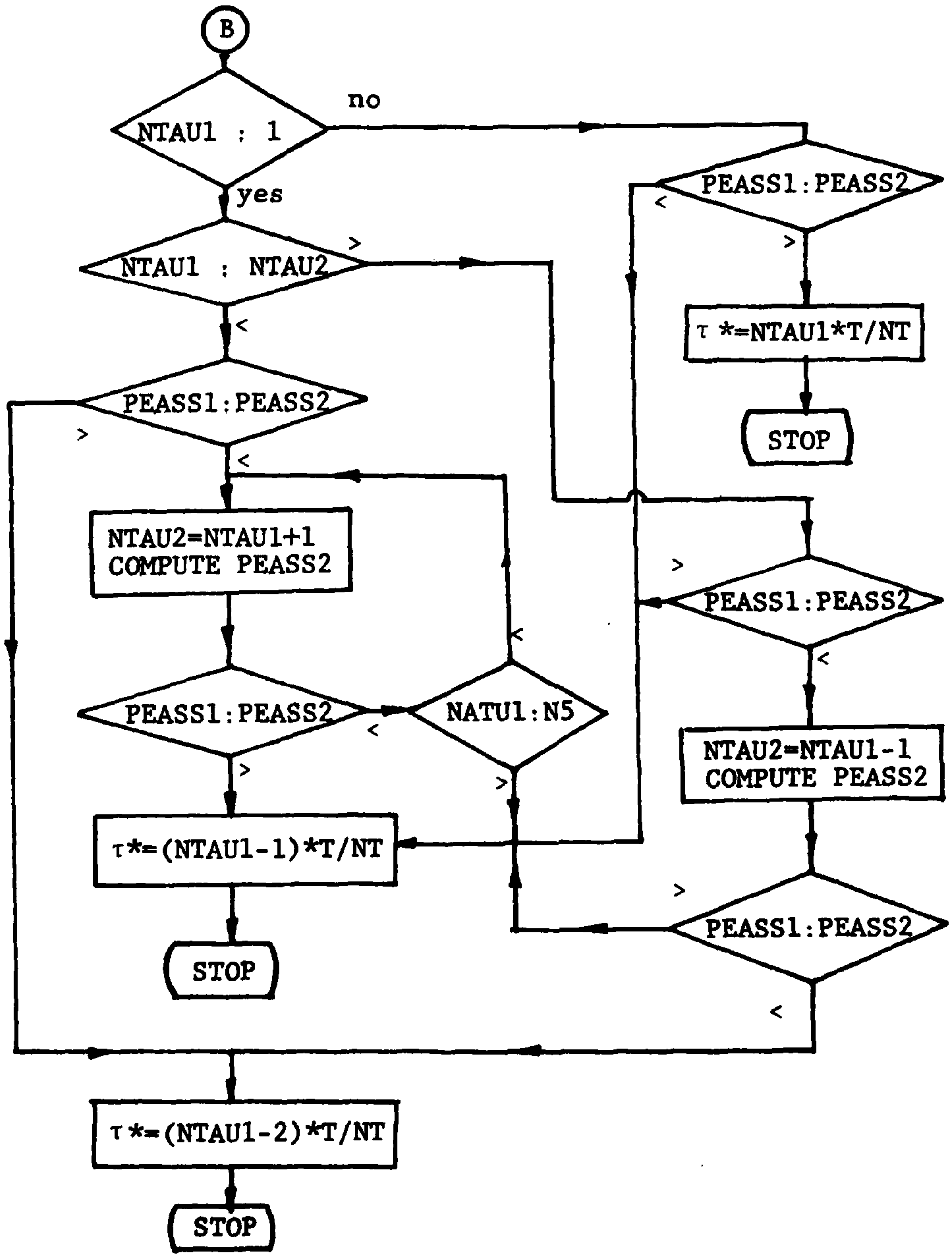


FIGURE 10 : (cont.)

is one subinterval apart from the previous midvalue (NTAU2).

The fourth step provides another value of the covariance when τ^* is equal to NTAU3.

The fifth step takes care of the comparison between the steady-state sample covariance obtained from step 2 and that obtained from step 4. The decision is made in this step in order to select a new interval of τ. The steady-state sample covariance of e_A is directly proportional to the difference between τ and τ^*; therefore, if PEASS1 is greater than PEASS2, then the lower limit N4 is updated with the midvalue while the upper limit, N5, is unchanged. If PEASS1 is less than PEASS2, then N5 will be updated with the midvalue while N4 is unchanged. The procedure goes back to step 2 and repeats until the condition in the third step is met.

The remainder of the flowchart (after the difference of the current midvalue and the previous midvalue is equal to one) shows the details of the technique for estimating τ. As discussed at the beginning of this section, the estimate of τ can be selected by comparing the steady-state sample covariance of every quantized number between the updated N4 and N5. The integer in this interval which corresponds to the smallest covariance denotes the estimate of τ.

If NT is an integer power of 2; i.e.,

$$NT = 2^V$$

then there is only one possible smallest interval in which τ lies and it is type-b smallest interval, which is shown in Figure 9-a. Furthermore, the maximum number of iterations to get the estimate of τ for any value of NT can be estimated by the maximum number of

* smallest interval type-b(FIGURE 9)

$\hat{*}$ smallest interval type-c(FIGURE 9)

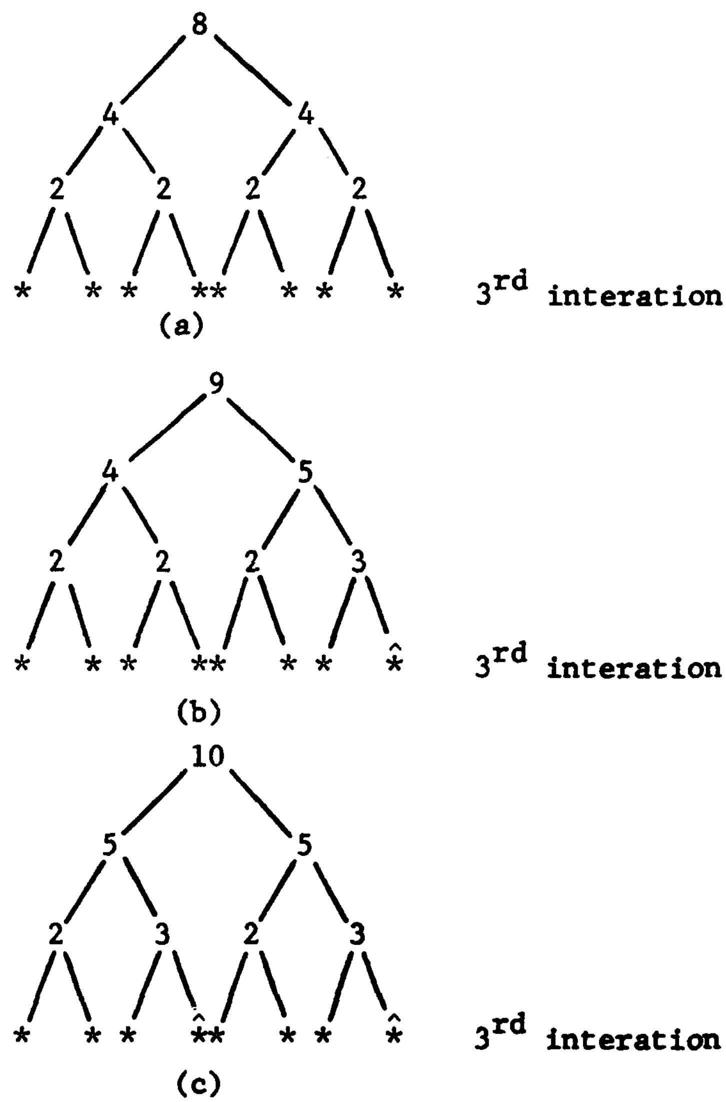


FIGURE 11 : DIAGRAM OF THE NUMBER OF INTERATIONS OF (a) NT=8, (b) NT=9, (c) NT=10

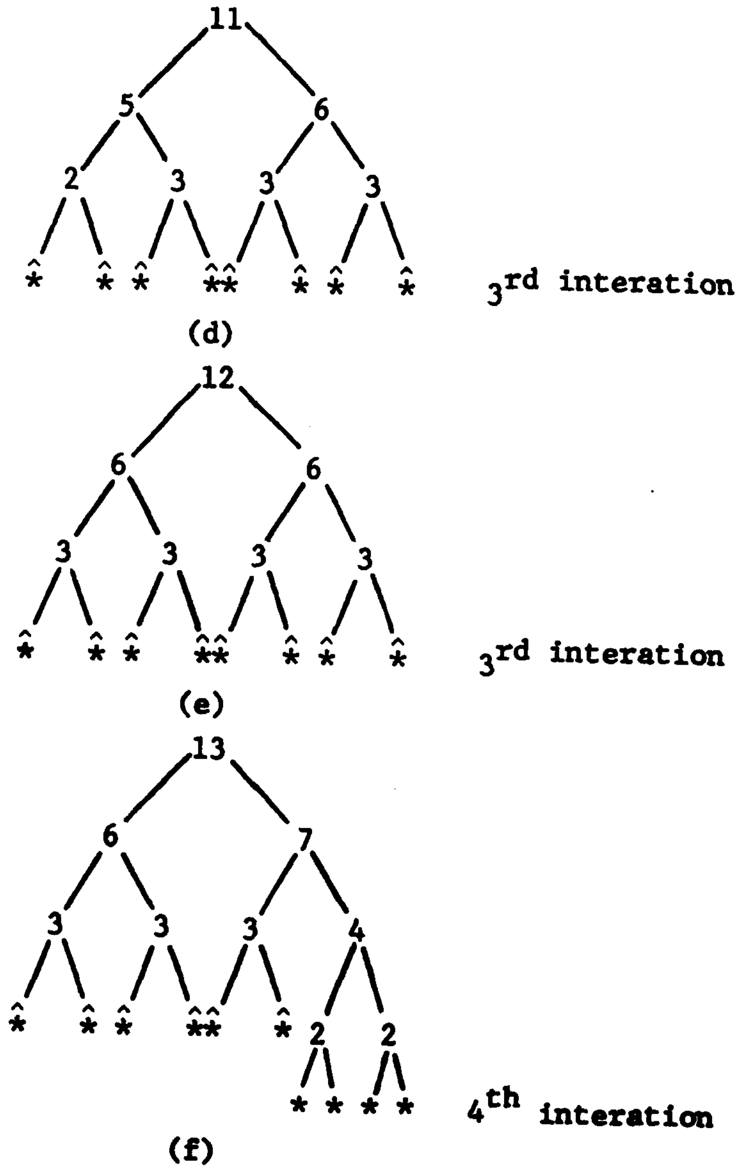


FIGURE 11 : (cont.) (d) NT=11, (e) NT=12
(f) NT=13

iterations when NT is equal to an integer power of 2. For example, if NT is 8, then the maximum iterations is 3 as shown in Figure 11-a. The last iteration (3rd iteration) is the smallest interval of τ when the difference between the previous midvalue and the current midvalue is one. All these intervals are type-b smallest intervals. If NT is increased by 1 (NT = 9), then the last iteration (3rd iteration) contains three type-b smallest interval and one type-c smallest interval and it is shown in Figure 11-b. The same as NT = 8, the maximum iterations of NT = 9 is equal to 3.

The situation when NT is 10, 11, and 12 are shown in Figure 11-c, d, and e, respectively, and the maximum iterations of these values of NT is still equal to 3. When NT is 13, the maximum iterations, which is shown in Figure 11-f, is 4. From these examples, the approximate maximum iterations of any values of NT between the midvalue of 2^{V-1} and 2^V and the midvalue of 2^V and 2^{V+1} is equal to V.

Another approach for calculating the estimate of τ is to determine the steady-state sample covariance of e_A for successive values of τ^* in (0,T) until that covariance begins to increase. The τ^* which corresponds to the smallest value of the steady-state sample covariance of e_A is the estimate of τ. Since τ^* for this approach starts from zero, the number of iterations to get the estimate of τ depends on the value of τ. If τ is close to zero or a small value, then τ^* can be estimated in a few iterations. The maximum number of iterations of this approach is equal to NT (when τ is equal to T). This maximum number of iterations is greater than the maximum number of iterations of the previous approach. For example, if NT is equal to 2^V, then the maximum number of iterations of this approach is equal to 2^V while that of the previous approach is only equal to V.

As an example, consider the system in Figure 2 of Section II with a second-order plant and a first-order controller. The external input w_p is Gaussian white noise, which has zero mean and variance = 1. Let's assume τ to be one of the values 0, $\frac{T}{NT}$, . . . , T and let NT be 50. A FORTRAN program to simulate the entire closed-loop system in Figure 7 and to implement the algorithm for estimating τ of the above example is in Appendix F. All arrays of this program are the same as those of the program of the new model.

The system in Figure 7 is simulated by the software in Appendix F. The software which implements the algorithm for estimating τ, will wait until the state observers (x_p^*) equal the state variables (x_p) and this system is in steady state. From Appendix A (this system is in steady state at the time 200T) and Appendix C (the state observers equal the state variables at time approximate equals 40T), the software will wait for 200T, then this software starts to implement the algorithm for estimate τ. As in the example in Section II and Section III, N is given the value 100 in the equation for calculating the sample covariance of the errors. With τ which is supposed to be equal to $\frac{3T}{NT}$, this software computes the estimate of this τ which is equal to $\frac{3T}{NT}$ and the number of iterations which is equal to 5.

In this example, τ is assumed to be one of the values 0, $\frac{T}{NT}$,

$\frac{2T}{NT}$, . . . , T, and τ^* is restricted to be any value among 0, $\frac{T}{NT}$, $\frac{2T}{NT}$, . . . , T. Therefore, τ can be estimated exactly. In general, τ will be between 0 and T, but it may not be one of the above values. Thus, the estimate of τ will generally not equal to τ and the maximum difference between τ and τ^* is $\frac{T}{NT}$. The numerical values of this difference can be reduced by increasing NT.

Since the time skew varies with time, it should be estimated at regular, short interval of times. Thus, the execution time of the algorithm in this subsection for estimating τ should be very small. From the above example, the total execution time for estimating τ is the sum of the time for computing P_{eAss} and the time for executing the algorithm for estimating τ. The time for executing the algorithm for estimating τ can not be reduced but the time for computing P_{eAss} can be reduced only by decreasing the number of values of e_A. (The difference between $y^*_{c2}(t_k + \tau^*)$ and $y_{c2}(t_k + \tau)$.)

Since the sample covariance of e_A in equation 3-19 is the estimated value of the actual covariance of e_A, the difference between the estimated value and the actual value depends on the number of values of e_A used. By the 'Law of Large Number' of Probability and Statistics, if N, which is the number of random variables of e_A used, is large, there is a high probability that $\bar{e}_A$ (the sample mean of e_A) will be closed to the actual mean of these random variables. Then, there is high probability that the sample covariance of e_A will be closed to the actual mean, too. Otherwise, if N is small, the estimated value may diverge from the actual value. According to the algorithm for estimating τ, the accuracy of this algorithm only depends on how smooth the curve P_{eAss} is. Then the covergence or divergence of the estimated value (the sample covariance of e_A) to the actual value (the covariance of e_A) does not concern to this algorithm. The value of N can be selected as small as the curve of P_{eAss} which is plotted as a function of τ and τ^* is still smooth. However, with N = 10, the curve of P_{eAss} is as smooth as the curve of P_{eAss} with N = 100.

2. CHARACTERISTICS OF THE TIME SKEW

By assumption, the time skew is constant over a short period of time. However, in reality the value of time skew will change slowly with time. As discussed in Reference 1, the time skew can be assumed to vary linearity with time and it is equal to T at the time which the two samplers return to synchronism. This characteristic variation of time skew is shown in Figure 12.

In the algorithm for estimating τ in last subsection, τ is assumed to be changing very slowly. However, this algorithm cannot estimate τ at that time at which τ changes from T to zero. Another means for estimating τ is needed for this time interval.

After the first estimate of τ is obtained, the characteristic curve in Figure 13 can be drawn. Let OA in Figure 13 represent the time that corresponds to the first τ^*, which is represented by AB. Since the time skew varies linearly with time, the estimate of the time skew also varies linearly with time. The characteristic of τ^* can be drawn by using the slope $\frac{AB}{OB}$ and the time which τ^* equals T is the approximate value which the two sample periods coincide. Thus,

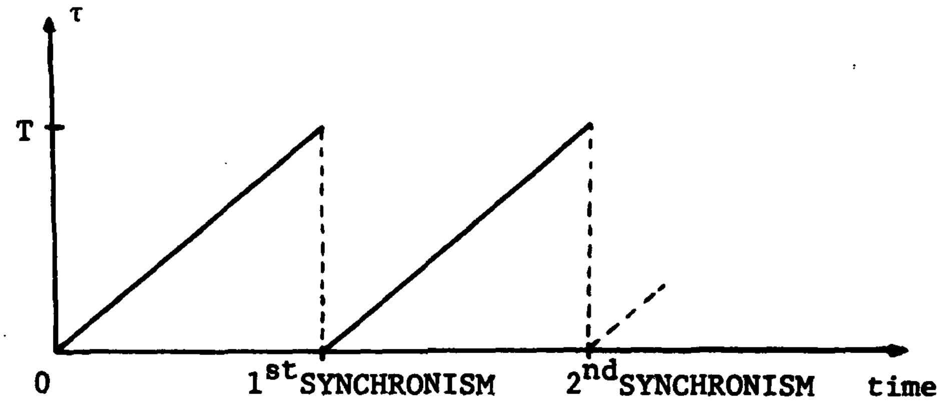


FIGURE 12 : CHARACTERISTIC OF TIME SKEW

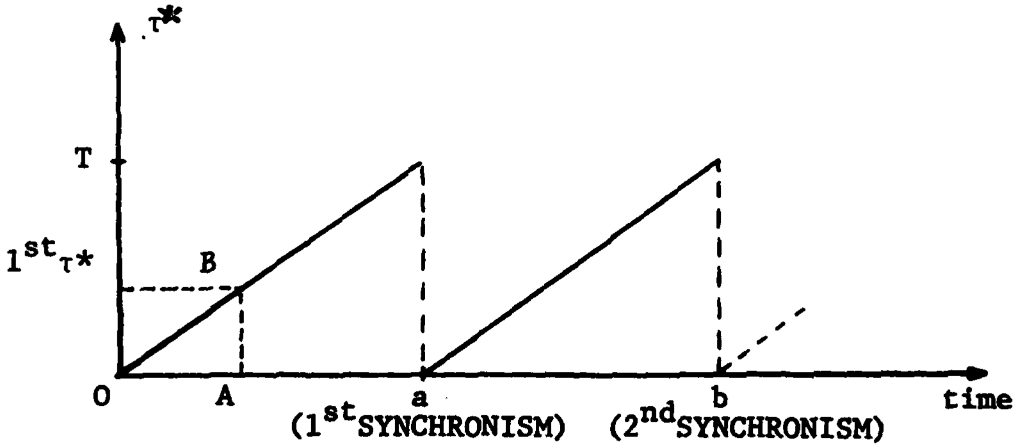


FIGURE 13 : CHARACTERISTIC OF τ*

an approximation to the time at which two sample periods coincide (point a in Figure 13) is given by $\frac{T \times AB}{OB}$. Thus, after the first τ^* has been estimated, the time at which two sample periods coincide is estimated from this latter equation.

SECTION V

OPERATION OF THE NEW MODEL

This section describes the asynchronous operation of a digital flight control system using the technique in Section III. The output of the first controller in the model used is always the input to the plant. The outputs of both controllers are sent to the monitor, which will compare these signals using the tolerance value from the technique in Section IV.

1. DUAL-REDUNDANT DIGITAL FLIGHT CONTROL SYSTEM.

A model for a dual-redundant digital flight control system is shown in Figure 14. The output of the plant is sampled by each of the controllers, using a common sample period but having a fixed time skew between them. The output of the first controller serves as the input to the plant. The outputs of the controllers go to the monitor and the monitor will first isolate the bad signal and then select or calculate the best signal from the remaining good signals.

Figure 15 shows the details of the monitor. All the previous values of output of the plant ($y_p(t_{k-1})$), the output of channel 1 ($y_{c1}(t_{k-1})$), and the external input to the plant ($w_p(t_{k-1})$) go to the block named OBSERVER, which is a subsystem designed to estimate the state variables of the plant. The observer produces $x_p^*(t_k)$, an estimate of $x_p(t_k)$. $x_p^*(t_k + \tau^*)$, an estimate of $x_p(t_k + \tau)$, is computed from $x_p^*(t_k)$, $y_{c1}(t_k)$, and τ^*. Then $y_{c2}^*(t_k + \tau^*)$, an estimate of $y_{c2}(t_k + \tau^*)$ that is used to calculate the inherent error, is computed from

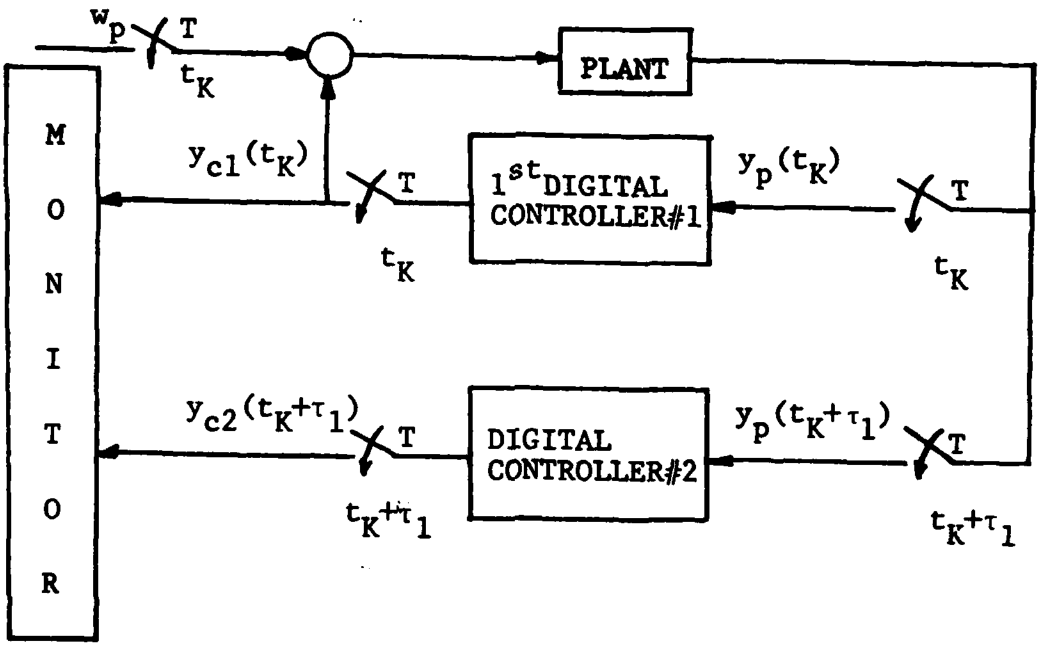


FIGURE 14 : VOTING OF DUAL-REDUNDANT CHANNEL

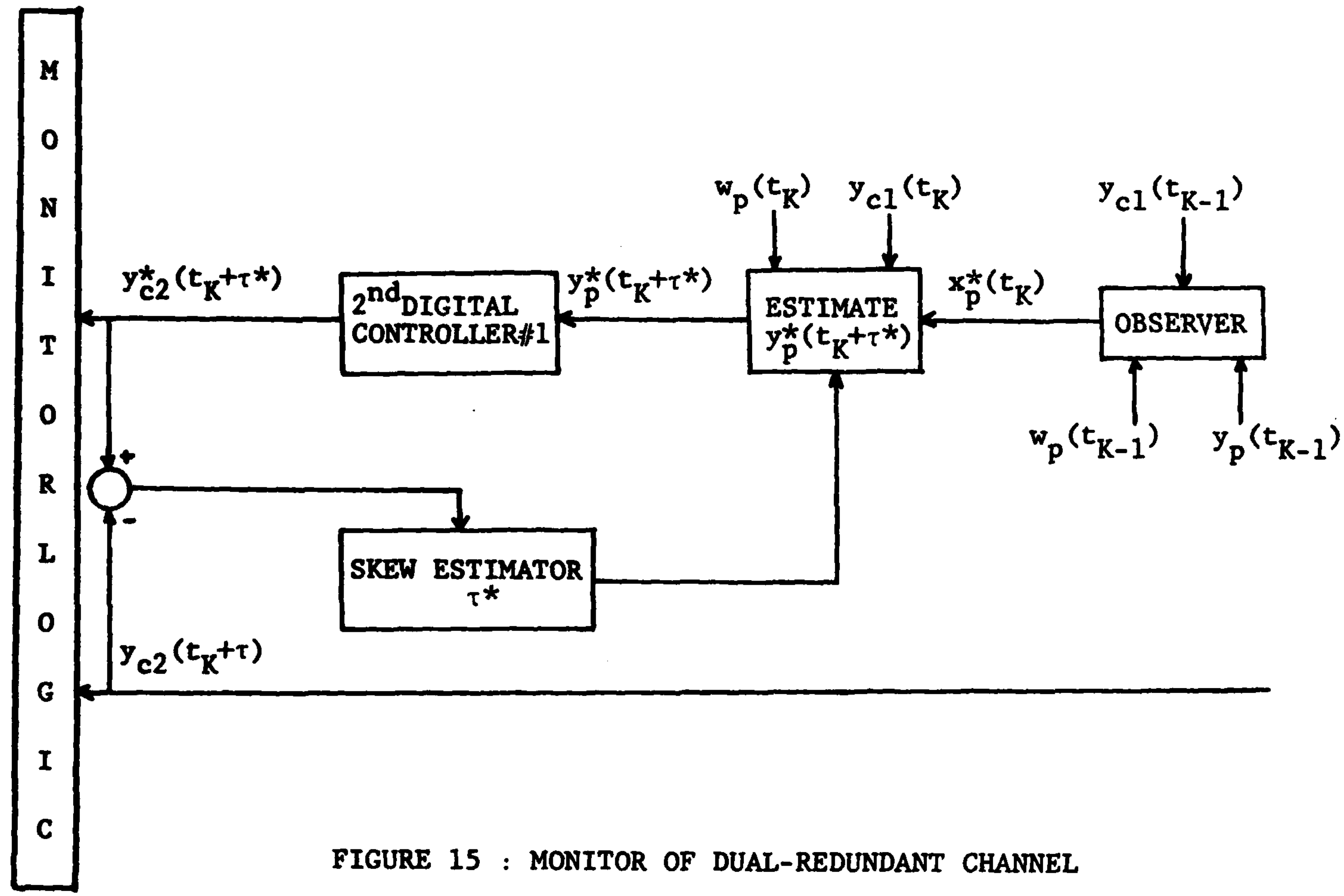


FIGURE 15 : MONITOR OF DUAL-REDUNDANT CHANNEL

$x_p^*(t_k + \tau^*)$ and the state variables of the 2nd DIGITAL CONTROLLER #1.

After the system in Figure 14 estimates τ, both $y_{c2}^*(t_k + \tau^*)$ and $y_{c2}(t_k + \tau)$ are sent to the monitor logic. If the sample covariance of the difference between $y_{c2}^*(t_k + \tau^*)$ and $y_{c2}(t_k + \tau)$ is greater than the tolerance value which is discussed in Section IV, then channel failures are probably present. The monitor logic will isolate the bad signal (y_{c1} or y_{c2}) from the channel failure.

As discussed at the end of Section IV, the execution time for estimating τ can be reduced by decreasing N in the equation of the sample covariance of e_A (eq. 3-19). Section IV also shows that there is no difference in the estimate of τ for N equal to 100 or 10. Thus, N in this section is selected to be equal to 10 for the purpose of reducing the execution time above.

To estimate τ, the 2nd DIGITAL CONTROLLER #1 executing the algorithm in Section IV for estimating τ will wait 50T seconds (T = 0.0125 second) until the state observer (x_p^*) is equal to the state variables (x_p) (The details are in Appendix C.). Then the 2nd DIGITAL CONTROLLER #1 will store the next ten values of e_A for estimating τ. After 50T seconds, τ can be estimated in the time required for obtaining 10 values of e_A plus the time for estimating τ. Since the ten values of e_A (from 51T second to 60T second) are used to estimate τ, then this τ^* is the estimated value of τ between 51T second and 60T second.

While the 2nd DIGITAL CONTROLLER #1 is in the process of estimating τ, y_{c1} and y_{c2} are always sent to the monitor. Then, for the time interval o to the time at which the first τ^* is estimated (50T seconds are required for the state observers to equal to the state variables.), the monitor logic should be disabled.

As in Reference 1, let P be equal to the period required for T_1 and T_2 to return to synchronism or

$$P = \frac{T_1 T_2}{T_1 - T_2}$$

Let e be the fractional error between the clock crystals controlling the separate processors; i.e.,

$$e = \frac{T_1 - T_2}{T_1}$$

then $$P = \frac{T_2}{e} = \frac{T_1}{e} - T_1 \qquad \frac{T_1}{e}$$

With T_1 = 0.0125 second and e = 0.017, it requires 10,000 samples of T_1 and T_2 to return to synchronism. Thus, τ changes very slowly with time, then the τ^* in the previous time interval can reasonably be assumed to be the estimated time skew over the current time interval.

As discussed in Section IV, the time at which the two sample periods return to synchronism is important because τ at this time changes abruptly from T to 0. For the time at which τ is equal to T or 0, it can be estimated by the technique described in the last subsection of Section IV.

After τ^* is estimated, the monitor will compare $y^*_{c2}(t_k + \tau^*)$ and $y_{c2}(t_k + \tau)$. If the difference of these two signals is greater than the given tolerance value discussed in Section IV, then a channel failure has occurred.

2. TRI-REDUNDANT DIGITAL FLIGHT CONTROL SYSTEM

A model for a tri-redundant digital flight control system is shown in Figure 16. The output of the plant is sampled by each of the controllers, using a common sample period but having two fixed time skews τ_1 and τ_2 between channel 1 and 2, and channel 1 and 3, respectively. The output of the first controller serves as the input to the plant and the outputs of these three controllers go to the monitor, as in the model of the previous subsection.

Figure 17 shows the details of the monitor. The output of the first controller is used to calculate $y^*_{c2}(t_k + \tau^*_1)$, an estimate of $y_{c2}(t_k + \tau_1)$, $y^*_{c2}(t_k + \tau^*_2)$, an estimate of $y_{c3}(t_k + \tau_2)$, where τ^*_1 and τ^*_2 are the estimated values of τ_1 and τ_2, respectively. Using the same technique as discussed in the last subsection, τ_1 can be estimated from $y^*_{c3}(t_k + \tau^*_2)$ and $y_{c3}(t_k + \tau_2)$. Since the maximum differences between τ^*_1 and τ_1, and τ^*_2 and τ_2 are equal, then the tolerance values between channel 1 and 2, and channel 1 and 3 are equal. In comparing channel 2 and 3, the tolerance value is approximately equal to the maximum covariance of the difference between $y^*_{c2}(t_k + \tau^*_1)$ and $y^*_{c3}(t_k + \tau^*_2)$. Thus, if the covariance of the difference between $y_{c2}(t_k + \tau_1)$ and $y_{c3}(t_k + \tau_2)$ is greater than the tolerance value between channel 2 and 3 described above, the malfunction must have occurred in channel 2 or channel 3.

The model for more than three channels can be described in a manner similar to the model for the three-channel system. For example, consider a four-channel system. This system has three time skews: the time skew between channels 1 and 2 (τ_1), the time skew between

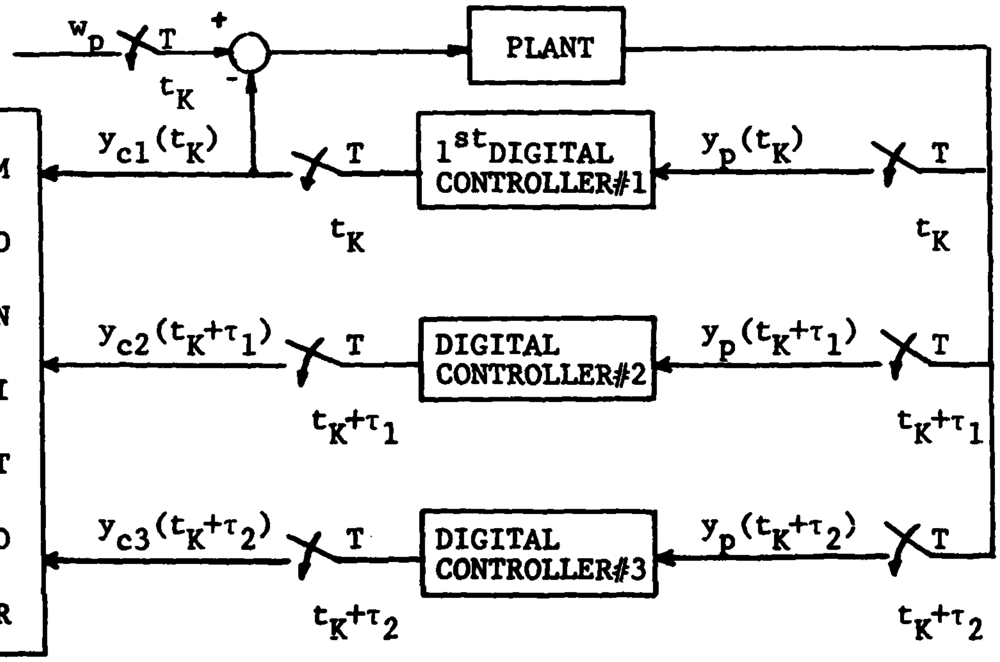


FIGURE 16 : VOTING OF TRI-REDUNDANT CHANNEL

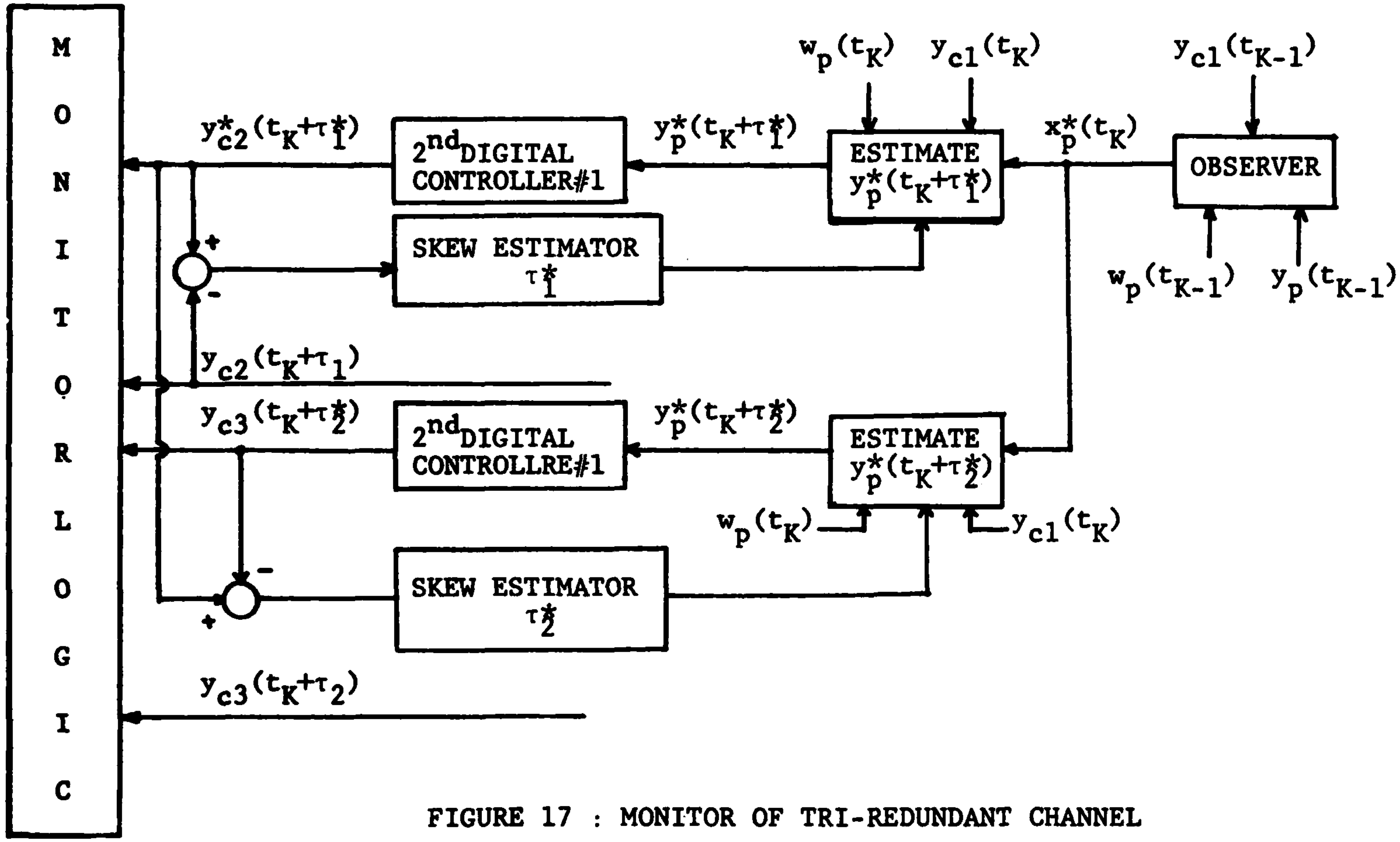


FIGURE 17 : MONITOR OF TRI-REDUNDANT CHANNEL

channels 1 and 3 (τ_2), and the time skew between channels 1 and 4 (τ_3). The 2nd DIGITAL CONTROLLER will estimate these three time skews for the estimated value of the output of channel 2, the estimated value of the output of channel 3, and the estimated value of the output of channel 4. As in the three-channel system, the maximum differences between τ_1 and τ_1^*, the estimate of τ_1, τ_2 and τ_2^*, the estimate of τ_2, τ_3 and τ_3^*, the estimate of τ_3 are equal to T/NT. Then the tolerance values of channels 1 and 2, channel 1 and 3, and channel 1 and 4 are equal to the maximum sample covariance between the estimated value and the actual value of any one of the outputs of the controller (the maximum sample covariance of the differences between the estimated and the actual values of the output of the second channel, the estimated and the actual values of the output of the third channel and the estimated and the actual values of the fourth channel are equal). For channels 2 and 3, the tolerance value is equal to the maximum covariance of the difference between the estimated values of channel 2 and 3. Similarly, the tolerance value of channels 2 and 4, and the tolerance value of channel 2 and 4 are equal to the maximum covariance of the differences between the estimated values of channel 2 and 4 and channel 3 and 4, respectively.

SECTION VI

COMPARISON BETWEEN BASIC MODEL AND NEW MODEL

The basic model can distinguish inherent errors from the errors induced by channel failures by using the maximum steady-state sample covariance of e_A as the tolerance value. If the measured covariance of e_A is greater than this tolerance value, then the monitor indicates a channel failure. Otherwise, if the measured covariance of e_A is less than this tolerance value, the monitor indicates that only inherent errors are present. However, this tolerance value is not the best value to use to distinguish inherent errors from errors induced by a channel failure. If the measured covariance of e_A is greater than the covariance of e_A of the present τ but is less than the maximum steady-state sample covariance of e_A ($\tau = T$), then the basic model would not indicate the channel failure. To reduce this deficiency, the new model computes a tolerance value equal to the maximum steady-state sample covariance of the difference between $y^{*}_{c2}(t_k + \tau^{*})$ and $y_{c2}(t_k + \tau)$. Since this tolerance value is very small when it is compared with the tolerance value of the basic model, the deficiency of the basic model discussed above can be reduced.

The number of tolerance values of the new model depends on the number of channels. There is one tolerance value for two channels, two tolerance values for three channels, four tolerance values for four channels, and so on. For the basic model, there is only one tolerance value for any number of channels.

The next comparison is the hardware structure. Both models (the basic model and the new model) require a computer to calculate the covariance of e_A. But the software of the new model is more complicated than that of the basic model. The software of the new model is used to compute x_p^*, an estimate of x_p, $x_p^*(t_k + \tau^*)$, an estimate of $x_p(t_k + \tau)$, $y_{c2}^*(t_k + \tau^*)$, an estimate of $y_{c2}(t_k + \tau)$, τ^*, an estimate of τ, and the sample covariance of e_A (The difference between $y_{c1}(t_k)$ and $y_{c2}(t_k + \tau)$ is e_A of the basic model. The difference between $y_{c2}^*(t_k + \tau^*)$ and $y_{c2}(t_k + \tau)$ is e_A of the new model.).

SECTION VII

SUMMARY AND RECOMMENDATIONS

Two models for the asynchronous digital flight control system are described in this report. The basic model is almost the same as the basic model in Reference 1 and 2 except that the input to the plant is the difference between the external input and the output of the controller. However, the input to the plant is always the output of the first controller.

As discussed in Reference 1, this basic model is roughly equivalent to the voter named 'median select' (the upper median for a four-channel system or the lower median for a three- or four-channel system is used as the voter output) when the channel outputs are either monotonically increasing or decreasing in time. Figure 18 illustrates the example of an asynchronous, dual-redundant digital flight control system which produces a monotonically increasing output. Channel 1 produces the sampled outputs at times t_k, t_{k+1}, . . . , for k = 0, 1, . . . and channel 2 produces the sampled outputs at times t_k, $t_k + \tau$, $t_{k+1} + \tau$, . . . , for k = 0, 1, ... A comparison monitor which compares the magnitudes of the outputs of the two channels will observe differences illustrated as e_A and e_B in Figure 18.

Since the pilot's command or wind - gust which changes all the time is the external input, then the random signal (Gaussian white noise) is chosen to be the external input of the models. A comparison monitor in this case (Gaussian white noise is the external input.) will compare the signal by using the covariance of e_A instead of e_A.

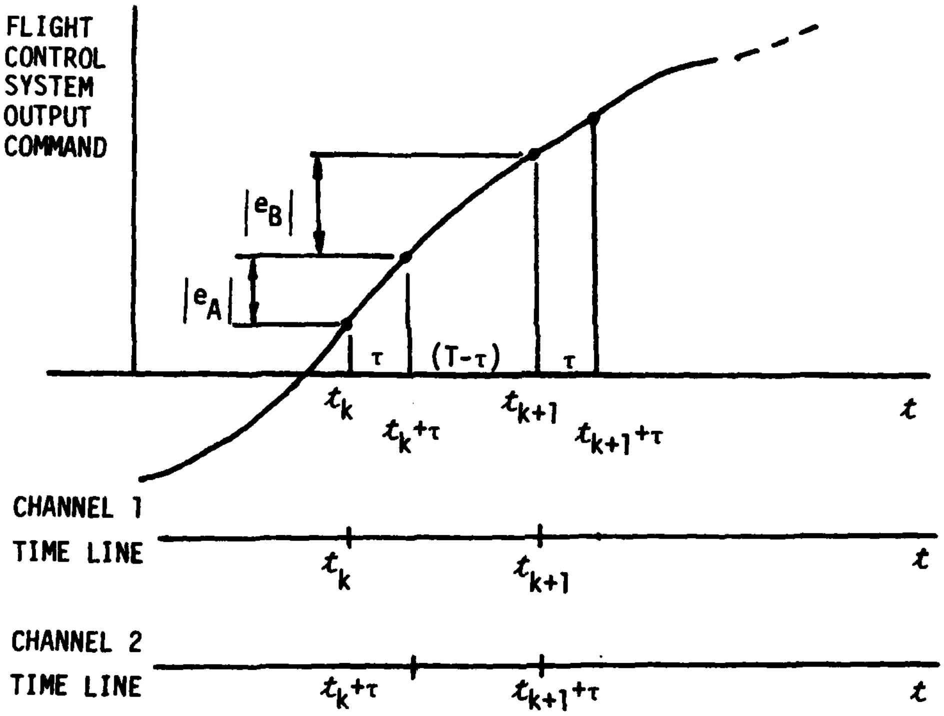


FIGURE 18 INHERENT ERRORS IN ASYNCHRONOUS OPERATION

According to the results of the basic model in Section II, the steady-state sample covariance of e_A (P_{eAss}) is largest when the times at which y_{c1} (output of the first controller) and y_{c2} (output of the second controller) change are farthest apart. Then the tolerance value of this model is equal to the maximum steady-state sample covariance of the difference between $y_{c1}(t_k)$ and $y_{c2}(t_k + \tau)$ (when $\tau = T$).

The new model described in Section III is an extension of the basic model in Section II. As in the basic model, the external input to this model is a Gaussian white noise. This model tries to reduce the deficiency of the basic model described at the beginning of Section III by decreasing the tolerance value. There are two functions performed by the first channel of this model. The first function is to compute the control output to the plant. The second function is to compute a signal used to calculate the inherent error; this signal is an estimated value of the output of the second channel.

According to the results of the new model in Section III, the steady-state sample covariance of the difference between $y^*_{c2}(t_k + \tau^*)$ and $y_{c2}(t_k + \tau)$ is directly proportional to the difference between τ and τ^*. If τ^* equals τ, the difference between $y^*_{c2}(t_k + \tau^*)$ and $y_{c2}(t_k + \tau)$ is equal to zero. Then the tolerance value of the new model depends on the difference between τ and τ^*.

The algorithm in Section IV for estimating τ is based on the results above. Let τ and τ^* be one of the values $0, \frac{T}{NT}, \ldots, T$. Then this algorithm computes the steady-state sample covariance of the difference between $y^*_{c2}(t_k + \tau^*)$ and $y_{c2}(t_k + \tau)$ when τ^* is the midvalue between the first lower limit (0) and the first upper limit (T). If the covariance of the difference between $y^*_{c2}(t_k + \tau^*)$ and $y_{c2}(t_k + \tau)$ of this first τ^* is less than that of the next value of this first τ^*, then must be between 0 and the new upper limit (the midvalue). Otherwise, if that of this first τ^* is greater than that of the next value of this first τ^*, then τ must be between the new lower limit (the midvalue) and T. By using this scheme, the interval containing τ can be reduced by half for each iteration. The algorithm will repeat this technique until the current midvalue differs from the previous midvalue by $\frac{T}{NT}$. In the last interval, any values of τ^* which corresponds to the smallest steady-state sample covariance of the difference between $y^*_{c2}(t_k + \tau^*)$ and $y_{c2}(t_k + \tau)$ is the estimate of τ. In general, τ may not be one of these values 0, $\frac{T}{NT}$, . . . , T, then the maximum difference between τ and τ^* is $\frac{T}{NT}$.

The results from Section III and Section IV can be applied to the asynchronous operation of digital flight control system. For a two-channel system, the DIGITAL CONTROLLER #1 in the monitor will wait until the state observers (x^*_p) equal the state variables (x_p). Then the DIGITAL CONTROLLER #1 will execute the algorithm for estimating τ by using the next 10 values of the difference between $y^*_{c2}(t_k + \tau^*)$ and $y_{c2}(t_k + \tau)$. During the time DIGITAL CONTROLLER #1 is estimating the first τ^*, the monitor will not compare the signals ($y^*_{c2}(t_k + \tau^*)$ and $y_{c2}(t_k + \tau)$). After the first τ^* is estimated, the monitor will compare the signal by using the first τ^* until the new τ^* is estimated. Then the monitor will compare the signals by using the new τ^* and so on. However, during the time in which the first τ^* is estimating (after the state observers (x_p^*) equal the state variables (x_p)), the monitor can compare the signals by using the tolerance value which is equal to the sample covariance when $\tau^* = 0$ and $\tau = T$. τ^* can be estimated for every time period except the time at which τ changes from its maximum value to its minimum value (The details are in Section IV.).

For a system with three or more channels, the number of time skews depends on the number of channels. For example, there are two time skews for the three-channel system: τ_1, the time skew between channels 1 and 2; and τ_2, the time skew between channels 1 and 3. For the four-channel system, there is one more time skew: τ_3, the time skew between channels 1 and 4. All these time skews can be estimated by using the same technique as in the two-channel system. After these time skews are estimated, the monitor will compare channels 1 and 2, channels 1 and 3, and so on by using the estimated output of channel 2 and the actual output of channel 2 for channels 1 and 2 and so on. If the sample covariance of the difference between the estimated and the actual values of channels 1 and 2 or channels 1 and 3 and so on is greater than the tolerance value of the new model, then the channel failure is occurred. For a pair of the channels 2, 3, . . . , the monitor will compare a pair of the channels by using the estimated outputs of that pair of the channels. For example, the monitor will compare channels 2 and 3 by using the tolerance value which is the maximum covariance of the difference between the estimated values of channels 2 and 3. If the sample covariance of the actual outputs of channels 2 and 3 is greater than the above tolerance value, then a channel failure has occurred.

The last section describes the comparison between the basic model and the new model. The disadvantage of the basic model is that the basic model would not indicate the channel failure although the sample covariance is greater than the sample covariance of the present τ. The new model can reduce this deficiency by reducing the tolerance value. However, the new model requires more hardware and is more complicated to simulate.

It is recommended that the work be continued to accomplish the following:

1. The software in this report can only implement the example in Figure 2. Then the software for simulating the models should be developed to be the general software. Thus, software for simulating a class of systems is needed.

2. Increase the complexity and generality of the models and covariance analysis to include such features as multirate sampling, computational delays, processor word-length effects, sensor noise and additional voter algorithms.

3. To reduce the complexity of the asynchronous operation of a digital flight control system using the technique in Section IV, a model of a new algorithm for a dual-redundant system is shown in Figure 19. In this model, τ^* is constant and equal to T/2. After the state observers are equal to the state variables, the signals $y^*_{c2}(t_k + T/2)$, and $y_{c2}(t_k + \tau)$ are sent to the monitor logic. The tolerance value to

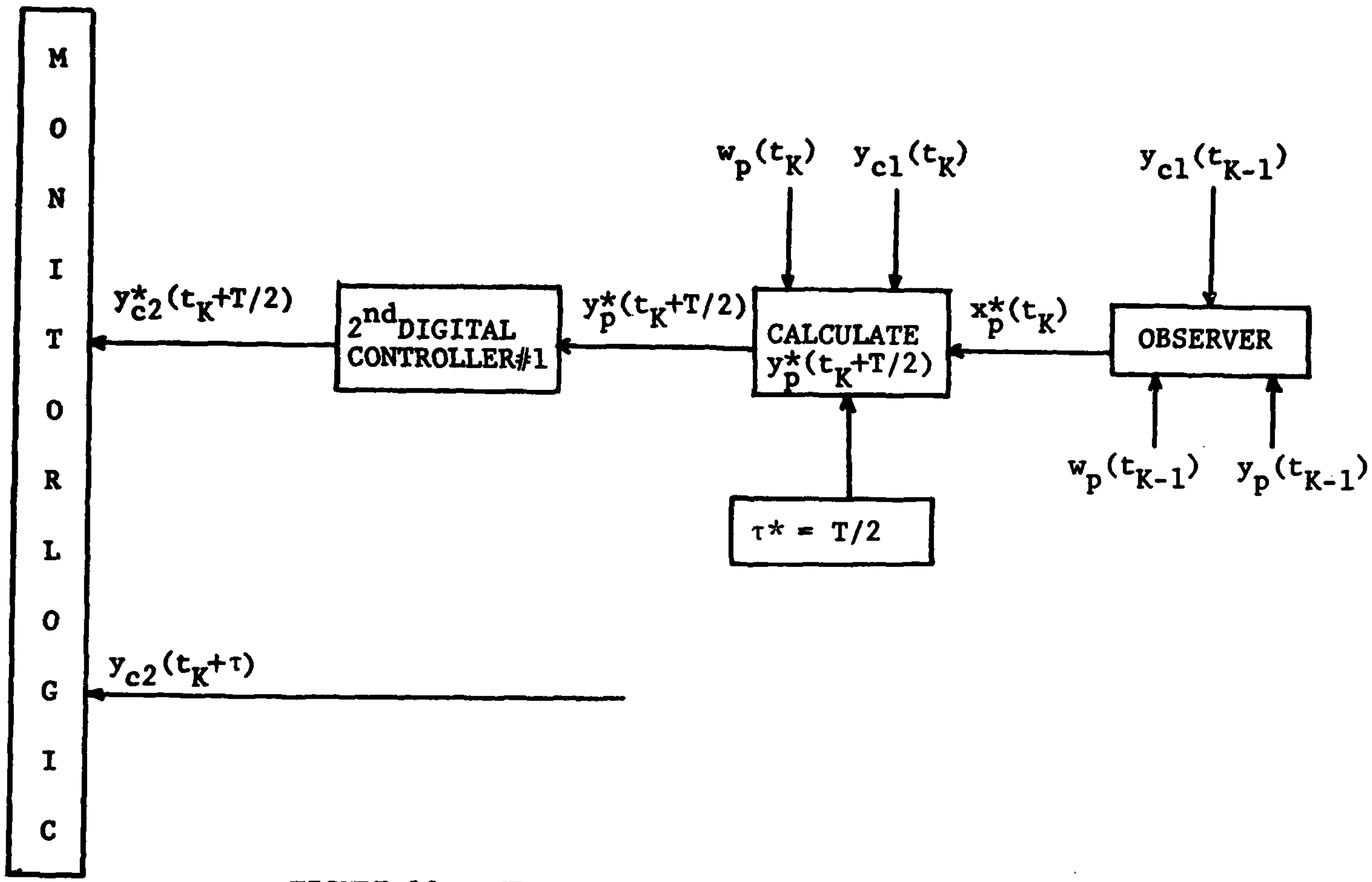


FIGURE 19 : NEW ALGORITHM FOR MONITOR OF DUAL-REDUNDANT CHANNEL

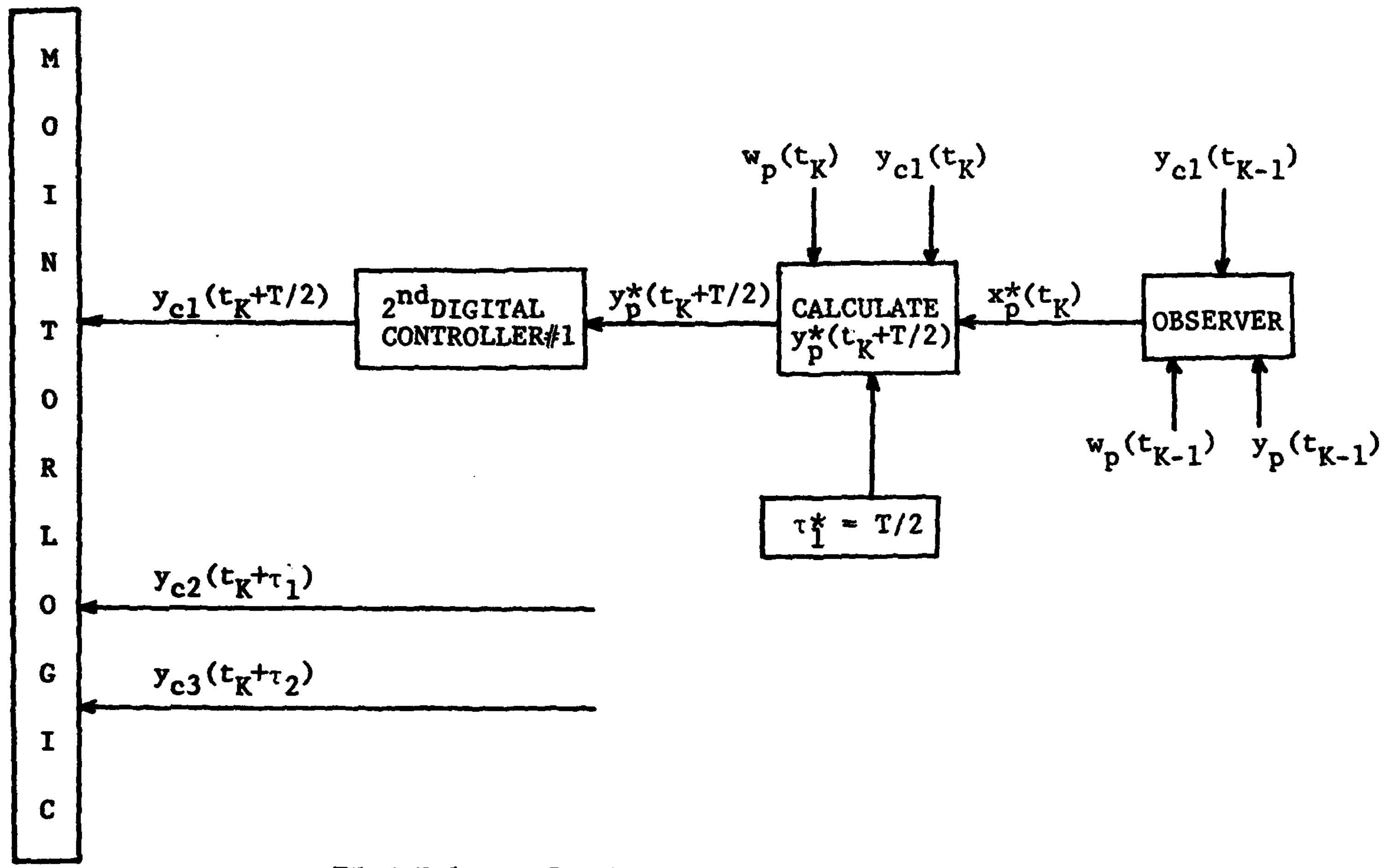


FIGURE 20 : NEW ALGORITHM FOR MONITOR OF TRI-REDUNDANT CHANNEL

use in this case is equal to the sample covariance of the difference between $y_{c2}(t_k + T/2)$ and $y_{c2}(t_k + 0)$. This algorithm is simpler than the algorithm described in Section V because τ^* is constant. But the tolerance value of this new algorithm is larger than the tolerance value of the algorithm in Section V. However, the tolerance value of this algorithm is less than the tolerance value of the basic model by half.

A model of a new algorithm for a tri-redundant system is shown in Figure 20. In this system, the tolerance value between channels 1 and 2, and channels 1 and 3 are equal to the sample covariance of the difference between $y_{c2}(t_k + T/2)$ and $y_{c2}(t_k + \tau)$. It is not necessary to estimate $y_{c3}(t_k + T/2)$ because $y_{c3}(t_k + T/2)$ is equal to $y_{c2}(t_k + T/2)$. However, the tolerance value of channels 2 and 3 is the same as in the basic model.

4. The number of values of e_A for computing the sample covariance should be studied with the objective of using as few values as needed for a reasonable reduction in the variability of the estimate.

APPENDIX A

SOFTWARE FOR THE BASIC MODEL

1. FLOWCHART AND DESCRIPTION OF MAJOR COMPONENTS AND SUBROUTINE

As in Reference 1, the main program of the software for the basic model is called PROGRAM SKEW. Its major computational tasks are to develop the state variable model of the complete closed-loop system and to compute the controller outputs, the errors e_A and e_B, and the steady-state covariances of the states.

A flowchart for the program appears in Figure 21. The blocks in this figure correspond to the clearly identified components of the main program.

The first block shows the data input. The variables are self-explanatory except for the quantities NT, NTAU, and NT2. Since a numerical integration is required to compute $\psi(\tau)$, and $\psi(T)$, it is necessary to quantize the time interval [0,T]. The user specifies the degree of quantization by specifying NT, the number of subintervals in [0,T] which are to be used in the computation. For convenience, the subintervals of [0,T] are designated 1, 2, . . . to NT1; where NT1 is equal to NT + 1. NTAU represents the value of which is computed within the program as

$$\tau_i = \frac{(NTAU_i - 1)}{(NT1 - 1)} * T \qquad \text{B-1}$$

and

$$NTAU_i = ITAU * \frac{NT}{NT2} + 1 \qquad \text{B-2}$$

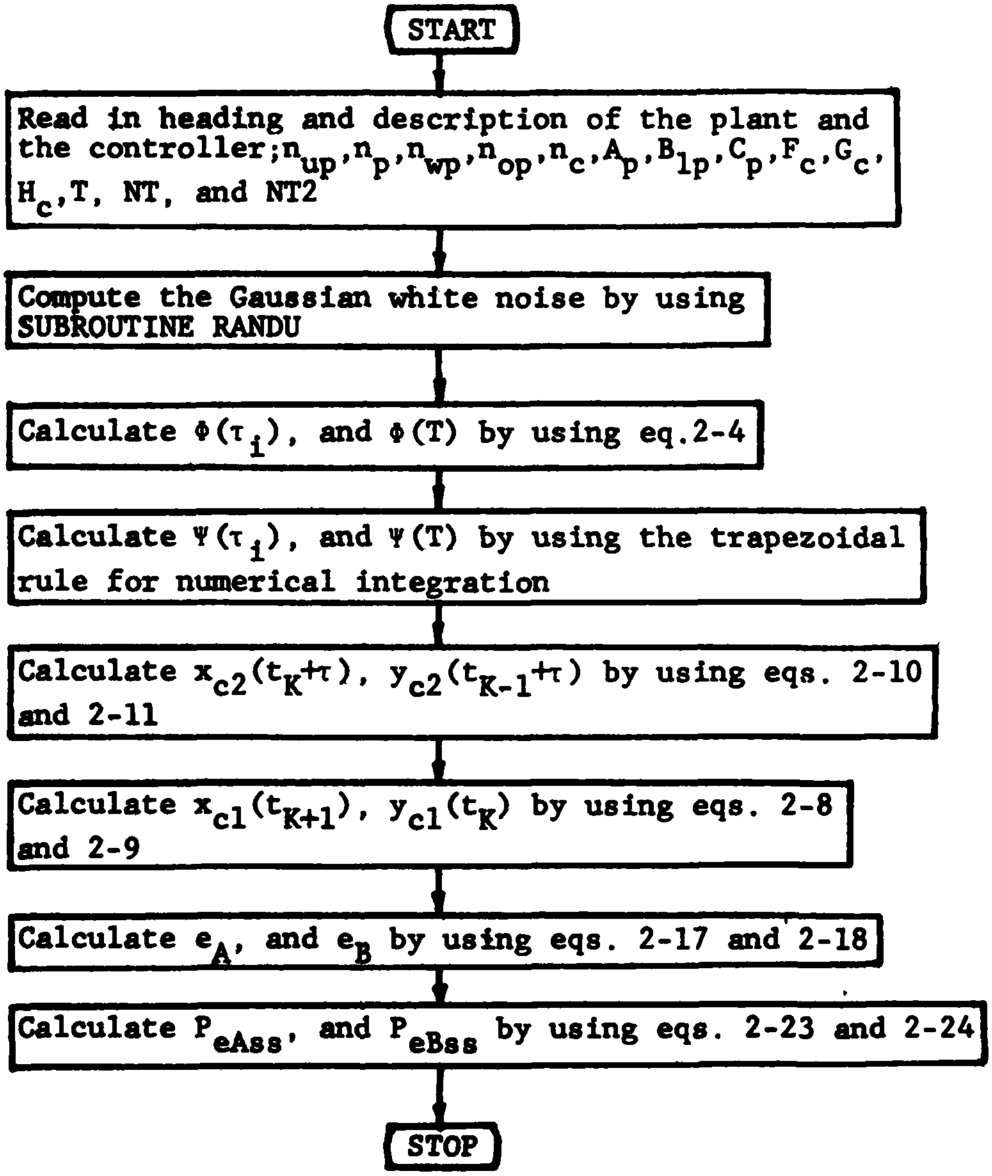


FIGURE 21 : FLOWCHART DESCRIBING THE MAJOR COMPUTATIONS OF PROGRAM SKEW

for ITAU = 0, 1, 2, . . . , NT2

where $\frac{NT}{NT2}$ must be an integer.

In block two, the gaussian noise is computed by using the subroutine named RANDU. This subroutine is available in most IBM-based computer systems.

The third block specifies the calculation of $\phi(\tau_i)$ and $\phi(T)$. The computations require ϕ from block 2, so that the required numerical integrations can be performed. The numerical integrations use the trapezoidal approximation.

In block five of Figure 21, the second controller state variable x_{c2} and the second controller y_{c2} are calculated by using equations

$$x_{c2}(t_k + \tau) = F_c x_{c2}(t_{k-1} + \tau) + G_c U_{c2}(t_{k-1} + \tau)$$

$$y_{c2}(t_{k-1} + \tau) = H_c x_{c2}(t_{k-1} + \tau) + E_c U_{c2}(t_{k-1} + \tau)$$

In block six, the first controller state variable x_{c1} and the first controller output y_{c1} are calculated by using equations

$$x_{c1}(t_{k+1}) = F_c x_{c1}(t_k) + G_c U_{c1}(t_k)$$

$$y_{c1}(t_k) = H_c x_{c2}(t_k) + E_c U_{c2}(t_k)$$

In block seven, the inherent errors e_A and e_B are calculated by using equations

$$e_A(t) = y_{c1}(t_k) - y_{c2}(t_k + \tau)$$

for $t_k + \tau \leq t < t_{k+1}$, $0 \leq \tau < T$, $k = 0, 1, \ldots$

$$e_B(t) = y_{c1}(t_{k+1}) - y_{c2}(t_k + \tau)$$

for $t_{k+1} \leq t < t_{k+1} + \tau$, $0 < \tau \leq T$, $k = 0, 1, \ldots$

The final set of computations is given in block eight. The steady state covariance PEASS and PEBSS are calculated by using the equations

$$P_{eAss} = \frac{1}{N} \sum_{i=1}^{N} [e_{Ai} - \bar{e}_A][e_{Ai} - \bar{e}_A]^T$$

$$P_{eBss} = \frac{1}{N} \sum_{i=1}^{N} [e_{Bi} - \bar{e}_B][e_{Bi} - \bar{e}_B]^T$$

where

e_{Ai} and e_{Bi}: are the errors when the system is in steady state

and

$\bar{e}_A$ and $\bar{e}_B$: are the sample means of N samples of e_{Ai} and e_{Bi}

2. INSTRUCTIONS FOR USING THE PROGRAM

The first data card is used to provide a message which will be printed at the top of a new page of output. The next card specifies NP, NUP, NWP, NOP, and NC using the format (5I3). These quantities are the actual dimensions of the plant and controller. Next, the matrices A_p, B_p, C_p, E_c, F_c, G_c, and H_c are specified in succession, one row and one card at a time, using the FORMAT (F10.4, 2I3). The next card specifies T, NT, and NT2 using FORMAT (F10.4, 3I5). Table 1 shows what values to assign to the given arrays.

The computer program listing for PROGRAM SKEW of the basic model written in FORTRAN appears in appendix B. PEASS and PEBSS for each value of τ are shown at the end of the listing.

TABLE 1

REQUIRED DIMENSIONS OF ALL ARRAYS

AP (NP, NP)	YP (NOP)
BP (NP, NWP)	XPTAU (NP)
CP (NOP, NP)	YPTAU (NOP)
FC (NC, NC)	UP (NUP)
GC (NC, NOP)	W (1000)
HC (NUP, NC)	YC1 (NUP)
EC (NUP, NOP)	XC1 (NC)
ECCP (NUP, NP)	YC2 (NC)
PHST1 (NP, NP)	XC2 (NC)
PHTAU (NP, NP)	E1 (NUP)
PHTAU1 (NP, NP)	E2 (NUP)
PSST (NP, NUP)	EA (1000)
PSTAU (NP, NUP)	EB (1000)
PSTAU1 (NP, NUP)	PEASS (30)
XP (NP)	PEBSS (30)

Since the example is in the steady-state at time approximately equal 200T (T = 0.0125 second), then the first value of i in equation (2-23), (2-24), (2-25), and (2-26) is 201 and let's assume the value of N in these equations equals 100.

APPENDIX B

COMPUTER PROGRAM LISTING
FOR
PROGRAM SKEW AND EXAMPLE
OF OUTPUT WRITTEN IN FORTRAN

```
      COMMON ELEMX,MAXI,MAXJ
      DIMENSION AP(2,2),B1P(2,2),CP(1,2),FC(2,2),GC(2,1),PHIT(4,4),
     1 HC(2,2),EC(2,1),PHIT1(4,4,101),PSIT1(4,4),PHTAU(4,4),PSTAU(4,4),
     4 INDEX(4),W(4000),PS(4,4),YC1(2),PSIT(4,4),XW3(2),
     5 YC2(2),E1(2),E2(2),XP(2),XPTAU(2),XP1(2),YP(2),YPTAU(2),
     6 XC1(2),XC2(2),AM(4,4),PT(4,4),P1(4,4),D1(4,4),
     7 D2(4,4),D3(4),XW1(2),XW2(2),ECCP(4,4),D(4,4),
     8 FEASS(50),FEBSS(50),EA(1100),EB(1100)
CCCCC PROVIDE MAXIMA FOR CALLED ARRAYS
      NPM = 2
      NUPM = 2
      NWPM = 2
      NOPM = 1
      NCM = 2
      NHM = NPM + NUPM
      NFM = NPM + 2*NCM
      NRRM = 2*NPM +4
C
C     ******************************************************************
C     READ INPUT DATA
C     ******************************************************************
C
      WRITE(6,899)
899   FORMAT('1')
  100 READ(5,900) ID
  900 FORMAT(20A4)
      WRITE(6,902) ID
  902 FORMAT('1',20A4)
      READ(5,906)NP,NUP,NWP,NOP,NC
  906 FORMAT(5I3)
      WRITE(6,908) NP,NUP,NWP,NOP,NC
  908 FORMAT('ONO. OF PLANT STATES = ',I3/
     1 ' NO. OF PLANT INPUTS = ',I3/
     2 ' NO. OF DISTURBANCE INPUTS = ', I3/
     4 ' NO. OF PLANT OUTPUTS = ', I3/
     5 ' NO. OF CONTROLLER STATES (EACH CONTROLLER) = ',I3)
       WRITE(6,910)
  910 FORMAT('OPLANT STATE MATRIX -- AP')
  110 DO 112 I = 1,NP
      READ(5,914) (AP(I,J),J=1,NP)
112   WRITE(6,913) (AP(I,J),J=1,NP)
913   FORMAT(' ',8G13.6)
914   FORMAT(6F12.7)
915   FORMAT(8G13.6)
      WRITE(6,916)
  916 FORMAT('OPLANT CONTROL INPUT MATRIX -- B1P')
  120 DO 122 I = 1,NP
      READ(5,914)(B1P(I,J),J=1,NUP)
122   WRITE(6,913)(B1P(I,J),J=1,NUP)
      WRITE(6,918)
918   FORMAT('OOBSERVER MATRIX-- GE')
  130 DO 132 I=1,NP
      READ(5,914)(GE(I,J),J=1,NWP)
132   WRITE(6,913)(GE(I,J),J=1,NWP)
```

```
      WRITE(6,920)
  920 FORMAT('OPLANT OUTPUT MATRIX -- CP')
  140 DO 142 I=1,NOP
      READ(5,914)(CP(I,J),J=1,NP)
142   WRITE(6,913)(CP(I,J),J=1,NP)
      WRITE(6,922)
  922 FORMAT('OCONTROLLER STATE MATRIX -- FC')
  150 DO 152 I =1,NC
      READ(5,914)(FC(I,J),J=1,NC)
152   WRITE(6,913)(FC(I,J),J=1,NC)
      WRITE(6,924)
  924 FORMAT('OCONTROLLER CONTROL INPUT MATRIX -- GC ')
  160 DO 162 I=1,NC
      READ(5,914)(GC(I,J),J=1,NOP)
  162 WRITE(6,913)(GC(I,J),J=1,NOP)
      WRITE(6,925)
  925 FORMAT('OCONTROLLER OUTPUT MATRIX (STATES) -- HC')
  170 DO 172 I=1,NUP
      READ(5,914)(HC(I,J),J=1,NC)
  172 WRITE(6,913)(HC(I,J),J=1,NC)
      WRITE(6,926)
  926 FORMAT('OCONTROLLER OUTPUT MATRIX (INPUTS) -- EC')
  180 DO 182 I=1,NUP
      READ(5,914)(EC(I,J),J=1,NOP)
  182 WRITE(6,913)(EC(I,J),J=1,NOP)
      READ(5,928) T,NT
928   FORMAT(F10.4,I5)
      XNT = NT
      DELTA = T/(XNT-1)
      WRITE(6,930) T,NT
930   FORMAT('1T = ',F10.4/
     1 ' NT = ',I5/
     3 ' T = SAMPLE RATE.'/
     6 ' DELTA = T/(NT-1) = INCREMENT USED IN THE NUMERICAL'/
     7 '                    INTEGRATIONS TO COMPUTE PSITAU,PSIT,PSIT2'/
     9 '                    PSITAU1 USING TRAPEZOIDAL RULE.'//)
      WRITE(6,931)
931   FORMAT(2X,' W IS THE EXTERNAL INPUT (WHITE GAUSSIAN NOISE WITH
     1 '                MEAN = 0.0, AND VARIANCE = 1.0')
C
C     ****************************************************************
C     GENERATE WHITE GAUSSIAN NOISE WITH MEAN = 0 AND VARIANCE = 1
C     ****************************************************************
C
      IX = 11111
      DO 192 I = 1,1200
      A = 0.0
      DO 193 J = 1,12
      CALL RANDU(IX,IY,Y)
      IX = IY
193   A = A+Y
192   W(I) = A-6
      DO 1199 I = 1,NUP
      DO 1199 J = 1,NP
```

```
      ECCP(I,J) = 0.0
      DO 1199 K = 1,NOP
1199  ECCP(I,J) = ECCP(I,J) + EC(I,K)*CP(K,J)
802   FORMAT(5X,5G13.6)
835   FORMAT(5X,5G13.6)
C
C     ****************************************************************
C     CALCULATE PHIT(0),PHIT(DELTA),PHIT(2*DELTA),...,PHIT(T)
C     ****************************************************************
C
      DELHLF = DELTA/2.0
      TI = 0.0
      DO 402 II = 1,NP
      DO 402 JJ = 1,NP
      IF(II.EQ.JJ) GO TO 403
      PHIT1(II,JJ,1) = 0.0
      GO TO 402
403   PHIT1(II,JJ,1) = 1.0
402   CONTINUE
      DO 4 I1 = 2,NT
      TI = TI+DELTA
      PHIT1(1,1,I1) = 1.
      PHIT1(1,2,I1) = (1./10.)*(1.-EXP(-10.*TI))
      PHIT1(2,1,I1) = 0.0
4     PHIT1(2,2,I1) = EXP(-10.*TI)
      DO 400 II = 1,NP
      DO 400 JJ = 1,NP
400   PHIT(II,JJ) = PHIT1(II,JJ,NT)
      WRITE(6,860)
860   FORMAT(5X,'PHIT')
      DO 861 I = 1,NP
861   WRITE(6,802) (PHIT(I,J),J=1,NP)
      N1 = 0
      DO 1800 KK2 = 1,6
      TAU = N1*0.0125/5.0
      N1 = N1 + 1
      NTAU = (KK2-1)*10
      NTAU = NTAU + 1
C     INITIAL VALUE YC2(-TIME+TAU), YC3(-TIME), YC1(-TIME)
      DO 31 I = 1,NUP
      YC2(I) = 0.0
31    YC1(I) = 0.0
      DO 32 I = 1,NUP
32    E1(I) = YC1(I) - YC2(I)
C     FROM INITIAL VALUE XP(TIME), AND YP(TIME) ARE EQUAL TO ZERO
      DO 35 I = 1,NP
35    XP(I) = 0.0
      DO 36 I = 1,NOP
36    YP(I) = 0.0
C     FROM INITIAL VALUE XC1(TIME+T), YC1(TIME) ARE EQUAL TO ZERO
      DO 37 I = 1,NC
      XC1(I) = 0.0
37    XC2(I) = 0.0
      DO 52 I = 1,NUP
```

```
52    YC1(I) = 0.0
C     ********************************************************************
C
C     CALCULATE PHIT(TAU)
C
C     ********************************************************************
      IDEL = 0
      DO 5000 I = 1,NT
      IDEL = IDEL + 1
      IF(IDEL.EQ.NTAU) GO TO 16
      GO TO 5000
16    DO 17 II = 1,NP
      DO 17 JJ = 1,NP
17    PHTAU(II,JJ) = PHIT1(II,JJ,IDEL)
5000  CONTINUE
C
C     ********************************************************************
C     CALCULATE PSIT(TAU),PSIT(T)
C     *********************************************************************
C
      DO 550 I = 1,NP
      DO 550 J = 1,NWP
550   PS(I,J) = 0.0
      DO 551 I = 1,NP
      DO 551 J = 1,NWP
      PSTAU(I,J) = 0.0
      PSIT(I,J) = 0.0
      DO 551 K = 1,NP
      PSTAU(I,J) = PSTAU(I,J) + PHIT1(I,K,NTAU)*B1P(K,J)
551   PSIT(I,J) = PSIT(I,J) + PHIT1(I,K,NT)*B1P(K,J)
      DO 552 I = 1,NP
      DO 552 J = 1,NWP
      PSTAU(I,J) = DELHLF*PSTAU(I,J)
552   PSIT(I,J) = DELHLF*PSIT(I,J)
60    DO 61 I1 = 2,NT
      I2 = NT-I1+1
      DO 62 I = 1,NP
      DO 62 J = 1,NWP
62    PSIT(I,J) = PSIT(I,J) + PS(I,J)
      DO 63 I = 1,NP
      DO 63 J = 1,NWP
      PS(I,J) = 0.0
      DO 63 K = 1,NP
63    PS(I,J) = PS(I,J) + PHIT1(I,K,I2)*B1P(K,J)
      DO 64 I = 1,NP
      DO 64 J = 1,NWP
64    PS(I,J) = DELHLF*PS(I,J)
      DO 66 I = 1,NP
      DO 66 J = 1,NWP
66    PSIT(I,J) = PS(I,J) + PSIT(I,J)
61    CONTINUE
67    DO 68 I = 1,NP
      DO 68 J = 1,NWP
68    PS(I,J) = 0.0
```

```
      IF(NTAU.EQ.1) GO TO 55
      GO TO 69
55    DO 58 II = 1,NP
      DO 58 JJ = 1,NWP
58    PSTAU(II,JJ) = 0.0
      GO TO 77
69    DO 70 I1 = 2,NTAU
      I2 = NTAU-I1+1
      DO 71 I = 1,NP
      DO 71 J = 1,NWP
71    PSTAU(I,J) = PSTAU(I,J) + PS(I,J)
      DO 72 I = 1,NP
      DO 72 J = 1,NWP
      PS(I,J) = 0.0
      DO 72 K = 1,NP
72    PS(I,J) = PS(I,J) + PHIT1(I,K,I2)*B1P(K,J)
      DO 73 I = 1,NP
      DO 73 J = 1,NWP
73    PS(I,J) = DELHLF*PS(I,J)
      DO 76 I = 1,NP
      DO 76 J = 1,NWP
76    PSTAU(I,J) = PS(I,J) + PSTAU(I,J)
70    CONTINUE
77    CONTINUE
C
C     ****************************************************************
C     START TIME LOOP
      TIME = 0
C     ****************************************************************
C
      NN1 = 1
      DO 412 I = 1,NOP
412   UP(I) = W(NN1)-YC1(I)
C     ****************************************************************
C     CALCULATE XP(TIME+TAU), AND YP(TIME+TAU)
C     ****************************************************************
      DO 430 I = 1,NUP
430   E2(I) = YC1(I) - YC2(I)
      DO 220 II1 = 1,400
      DO 253 I = 1,NP
      Q = 0.0
      DO 254 J = 1,NP
254   Q = Q + PHTAU(I,J)*XP(J)
253   XW1(I) = Q
      DO 255 I = 1,NP
      Q = 0.0
      DO 256 J = 1,NUP
256   Q = Q + PSTAU(I,J)*UP(J)
255   XW2(I) = Q
      DO 257 I = 1,NP
257   XPTAU(I) = XW1(I) + XW2(I)
      DO 260 I = 1,NOP
      Q = 0.0
      DO 261 J = 1,NP
```

```
261   Q = Q + CP(I,J)*XPTAU(J)
260   YPTAU(I) = Q
C     ******************************************************************
C     TIME = TIME + T
C     CALCULATE XC2(TIME+T+TAU),AND YC2(TIME+TAU)
C     ******************************************************************
      DO 300 I = 1,NUP
      Q = 0.0
      DO 301 J = 1,NC
301   Q = Q + HC(I,J)*XC2(J)
300   XW1(I) = Q
      DO 302 I = 1,NUP
      Q = 0.0
      DO 303 J = 1,NOP
303   Q = Q + EC(I,J)*YPTAU(J)
302   XW2(I) = Q
      DO 304 I = 1,NUP
304   YC2(I) = XW1(I) + XW2(I)
      DO 305 I = 1,NC
      Q = 0.0
      DO 306 J = 1,NC
306   Q = Q + FC(I,J)*XC2(J)
305   XW1(I) = Q
      DO 307 I = 1,NC
      Q = 0.0
      DO 308 J = 1,NOP
308   Q = Q + GC(I,J)*YPTAU(J)
307   XW2(I) = Q
      DO 309 I = 1,NC
309   XC2(I) = XW1(I) + XW2(I)
C     ******************************************************************
C     CALCULATE E1
C     ******************************************************************
      DO 290 I = 1,NUP
290   E1(I) = YC1(I) - YC2(I)
C     ******************************************************************
C     CALCULATE XP(TIME), AND YP(TIME)
C     ******************************************************************
373   DO 500 I = 1,NP
      Q = 0.0
      DO 501 J = 1,NP
501   Q = Q + PHIT(I,J)*XP(J)
500   XW1(I) = Q
      DO 502 I = 1,NP
      Q = 0.0
      DO 503 J = 1,NUP
503   Q = Q + PSIT(I,J)*UP(J)
502   XW2(I) = Q
      DO 504 I = 1,NP
504   XP(I) = XW1(I) + XW2(I)
      DO 507 I = 1,NOP
      Q = 0.0
      DO 508 J = 1,NP
508   Q = Q + CP(I,J)*XP(J)
```

```
507   YP(I) = Q
C     ******************************************************************
C     CALCULATE XC1(2*TIME), AND YC1(TIME)
C     ******************************************************************
      DO 700 I = 1,NUP
      Q = 0.0
      DO 701 J = 1,NC
701   Q = Q + HC(I,J)*XC1(J)
700   XW1(I) = Q
      DO 702 I = 1,NUP
      Q = 0.0
      DO 703 J = 1,NOP
703   Q = Q + EC(I,J)*YP(J)
702   XW2(I) = Q
      DO 704 I = 1,NUP
704   YC1(I) = XW1(I) + XW2(I)
      DO 705 I = 1,NC
      Q = 0.0
      DO 706 J = 1,NC
706   Q = Q + FC(I,J)*XC1(J)
705   XW1(I) = Q
      DO 707 I = 1,NC
      Q = 0.0
      DO 708 J = 1,NOP
708   Q = Q + GC(I,J)*YP(J)
707   XW2(I) = Q
      DO 709 I = 1,NC
709   XC1(I) = XW1(I) + XW2(I)
      NN1 = NN1 + 1
      DO 221 I = 1,NOP
221   UP(I) = W(NN1)-YC1(I)
C     ******************************************************************
C     CALCULATE E2
C     ******************************************************************
      DO 540 I = 1,NUP
540   E2(I) = YC1(I) - YC2(I)
      DO 560 I = 1,NUP
      EA(II1) = E1(I)
560   EB(II1) = E2(I)
220   CONTINUE
C
C     ******************************************************************
C     CALCULATE THE STEADY STATE SAMPLE COVARIANCE OF ERRORS
C     ******************************************************************
C
      SMEANA = 0.0
      SMEANB = 0.0
      DO 561 I = 501,800
      SMEANA = SMEANA + EA(I)
      SMEANB = SMEANB + EB(I)
      SMEANA = SMEANA/300
      SMEANB = SMEANB/300
      [illegible]A = 0.0
      [illegible]B = 0.0
```

```
      DO 563 I = 501,800
      VEA = VEA + (EA(I)-SMEANA)**2
563   VEB = VEB + (EB(I)-SMEANB)**2
      PEASS(KK2) = VEA/300
      PEBSS(KK2) = VEB/300
1800  CONTINUE
      WRITE(6,570) TAU
570   FORMAT(5X,'PEASS( TAU = ',F12.8,')')
      WRITE(6,571)(PEASS(I),I=1,6)
571   FORMAT(5X,F18.10)
      WRITE(6,572) TAU
572   FORMAT(5X,'PEBSS( TAU = ',F12.8,')')
      WRITE(6,571)(PEBSS(I),I=1,6)
1801  CONTINUE
      STOP
      END
      SUBROUTINE RANDU(IX,IY,YFL)
      IY = IX*65539
      IF(IY)5,6,6
5     IY = IY + 2147483647+1
6     YFL = IY
      YFL = YFL*0.4656613E-9
      RETURN
      END
```

```
EXAMPLE------2TH ORDER PLANT, 1ST ORDER CONTROLLER
NO. OF PLANT STATES = 2
NO. OF PLANT INPUTS = 1
NO. OF EXTERNAL INPUT = 1
NO. OF PLANT OUTPUTS = 1
NO. OF CONTROLLER STATES ( EACH CONTROLLER) = 1

PLANT STATE MATRIX -- AP
   .0            1.0
   .0          -10.0

PLANT CONTROL INPUT MATRIX -- BP
   .0
  200.0


PLANT OUTPUT MATRIX -- CP
  1.0            .0


CONTROLLER STATE MATRIX -- FC
  .523810

CONTROLLER CONTROL INPUT MATRIX -- GC
 -.18162

CONTROLLER OUTPUT MATRIX (STATES) -- HC
  1.0

CONTROLLER OUTPUT MATRIX (INPUTS) -- EC
  1.381
NT = 51
T = SAMPLE PERIOD = 0.0125 SEC
DELTA = T/(NT-1) = INCREMENT USED IN THE NUMERICAL
                   INTEGRATIONS TO COMPUTE PSITAU,PSIT,PSIT2,
                   PSITAU1 USING TRAPEZOIDAL RULE.
W IS THE EXTERNAL INPUT (WHITE GAUSSIAN NOISE WITH
  MEAN = 0.0, AND VARIANCE = 1.0
THE STEADY STATE SAMPLE VARIANCE OF ERRORS
    PEASS
           0.0
           0.0002195683
           0.0008547062
           0.0018880414
           0.0033245913
           0.0051899776
    PEBSS
           0.0051899776
           0.0033477221
           0.0019137899
           0.0008709673
           0.0002244494
           0.0
```

APPENDIX C

DESIGN OF THE STATE OBSERVER

In section III, $y_p(t_k)$ the input to channel 1, go through the blocks named 1st DIGITAL CONTROLLER AND OBSERVER. The output of the first block serves as the input to the plant and the output of the second is the estimate of the state variables of the plant and is used to estimate $y^*_{c2}(t_k + \tau^*)$. This appendix describes how to estimate the state variables $x_p(t_k)$ from $y_p(t_k)$.

1. DESIGN OF STATE OBSERVER

Reference 3 defines an observer as the subsystem that estimates the state variables of a dynamical system, based on measurements of the input $u_p(t)$ and the output $y_p(t)$. Figure 22 shows the block diagram of an observer which is formulated as a feedback control with G_c as the feedback matrix. The design objective is to select the feedback matrix G_e such that $y^*_p(t)$, the estimate of $y_p(t)$, will approach $y_p(t)$ as fast as possible. When $y_p(t)$ equals $y^*_p(t)$, the dynamics of the state observer are described by

$$\dot{x}^*_p(t) = A_p x^*_p(t) + B_p u_p(t) \qquad \text{C-1}$$

which is identical to the state equation of the system (plant) to be observed. In general, with $u_p(t)$ and $y_p(t)$ as inputs to the observer, the dynamics of the observer are represented by

$$\dot{x}^*_p(t) = [A_p - G_e C_p] x^*_p(t) + B_p u_p(t) + G_e y_p(t) \qquad \text{C-2}$$

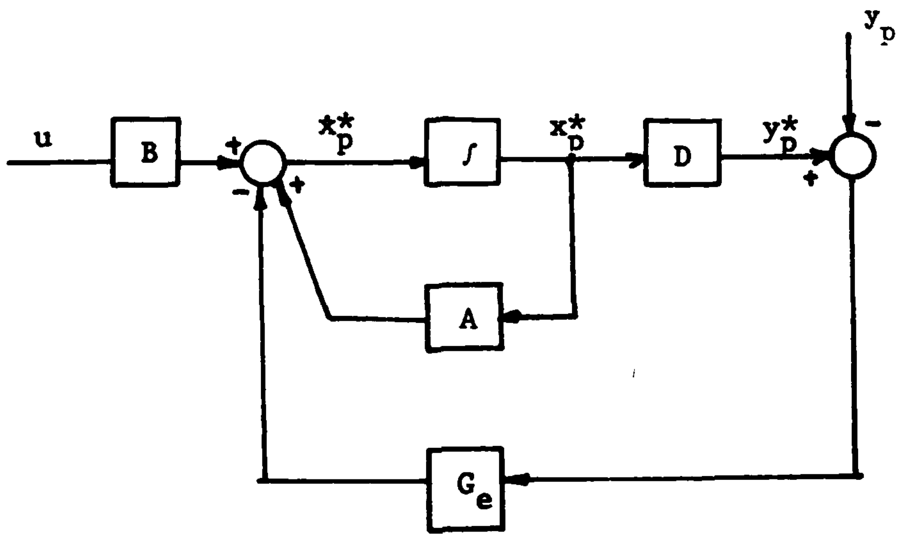


FIGURE 22 : BLOCK DIAGRAM OF AN OBSERVER

Since $y_p(t)$ equals $C_p x_p(t)$, the equation C-2 is written as

$$\dot{x}_p^*(t) = A_p x_p^*(t) + B_p u_p(t) + G_e C_p[x_p(t) - x_p^*(t)] \qquad \text{C-3}$$

The significance of this expression is that if the initial values of $x_p(t)$ and $x_p^*(t)$ are identical, the equation reverts to that of equation C-1, and the response of the observer will be identical to that of the original system. [In the model in Section III, the initial value of $x_p^*(t_k)$ is unknown. Therefore, the design of the feedback matrix G_e for the observer is significant only if the initial conditions of $x_p(t)$ and $x_p^*(t)$ are different.

If we subtract equations (C-3) from (C-1), we have

$$[\dot{x}_p(t) - x_p^*(t)] = [A_p - G_e C_p][x_p(t) - x_p^*(t)] \qquad \text{C-4}$$

which may be regarded as the homogeneous state equation of a linear system with the coefficient matrix $[A - G_e C_p]$. The characteristic equation of $[A - G_e C_p]$ and of the state observer is then

$$|\lambda - (A - G_e C_p)| = 0 \qquad \text{C-5}$$

Since we are interested in driving $x_p^*(t)$ as close to $x_p(t)$ as possible, the objective of the observer design may be stated as to select the elements of G_e so that the natural response of equation C-4 decays to zero as quickly as possible. In other words, the eigenvalues of $[A - G_e C_p]$ should be selected so that $x_p^*(t)$ approaches $x_p(t)$ rapidly. However, it must be kept in mind that the approach of assigning the eigenvalues of $[A - G_e C_p]$ may not always be satisfactory for the purpose of matching all the observed states to the real state, since the eigenvalues control only the denominator or polynomial of the transfer relation, while the numerator polynomial is not controlled. More details and an example of the statement above are available in Reference 4.

2. EXAMPLE

The following example, which is the example used in Section II and III, is used to illustrate the technique described above.

From the example in Section II, we have

$$A_p = \begin{vmatrix} 0 & 1 \\ 0 & -10 \end{vmatrix} \tag{C-6}$$

$$B_p = \begin{vmatrix} 0 \\ 200 \end{vmatrix} \tag{C-7}$$

and

$$C_p = \begin{vmatrix} 1 & 0 \end{vmatrix} \tag{C-8}$$

Let the feedback matrix be designated as

$$G_e = \begin{vmatrix} g_{e1} \\ g_{e2} \end{vmatrix} \tag{C-9}$$

Substitution of equations (C-6), (C-8), and (C-9) into (C-5), gives

$$[\lambda I - (A_p - G_e C_p)] = \begin{vmatrix} \lambda & 0 \\ 0 & \lambda \end{vmatrix} - \left\{ \begin{vmatrix} 0 & 1 \\ 0 & -10 \end{vmatrix} - \begin{vmatrix} g_{c1} & 0 \\ g_{c2} & 0 \end{vmatrix} \right\}$$

$$= \begin{vmatrix} \lambda & 0 \\ 0 & \lambda \end{vmatrix} - \begin{vmatrix} -g_{c1} & 1 \\ -g_{c2} & -10 \end{vmatrix}$$

$$= \begin{vmatrix} \lambda + g_{c1} & -1 \\ g_{c2} & \lambda + 10 \end{vmatrix} \tag{C-10}$$

Then the characteristic equation of the state observer is

$$\lambda x^2 + (10 + g_{e1})\lambda + (10g_{e1} + g_{e2}) = 0 \qquad \text{C-11}$$

Let the eigenvalues of $\lambda I - (A_p - G_eC_p)$ be $\lambda = -15, -15$ then the characteristic equation should be

$$\lambda^2 + 30\lambda + 225 = 0 \qquad \text{C-12}$$

Equating like terms of equations (C-11) and (C-12) gives

$$g_{e1} = 20$$

$$g_{e2} = 25$$

Figure 23 illustrates the responses $x_{p1}(t)$ and $x^*_{p1}(t)$ for the following initial states

$$x_p(0) = \begin{vmatrix} 0 \\ 0 \end{vmatrix}, \qquad x^*_p(0) = \begin{vmatrix} 2 \\ 0 \end{vmatrix}$$

Shown in the same figure is the response of $x^*_{p1}(t)$ when the state observer is designed for eigenvalues at $\lambda = -20, -20$; in this case $g_{e1} = 30$ and $g_{e2} = 100$. However, it is seen from the figure that x^*_{p1} for both $g_{e1} = 20$, $g_{e2} = 25$ and $g_{e1} = 30$, $g_{e2} = 100$ are approximately the same deviation from x_p.

Figure 24 illustrates the response $x_{p2}(t)$ and $x^*_{p2}(t)$ for the two cases of observer design. The characteristics of x^*_{p2} for both $g_{e1} = 20$, $g_{e2} = 25$ and $g_{e1} = 30$, $g_{e2} = 100$ are the same as that of both x^*_{p1} which are approximately the same deviation from x_{p2}.

As mentioned earlier, the eigenvalues may not always be satis-

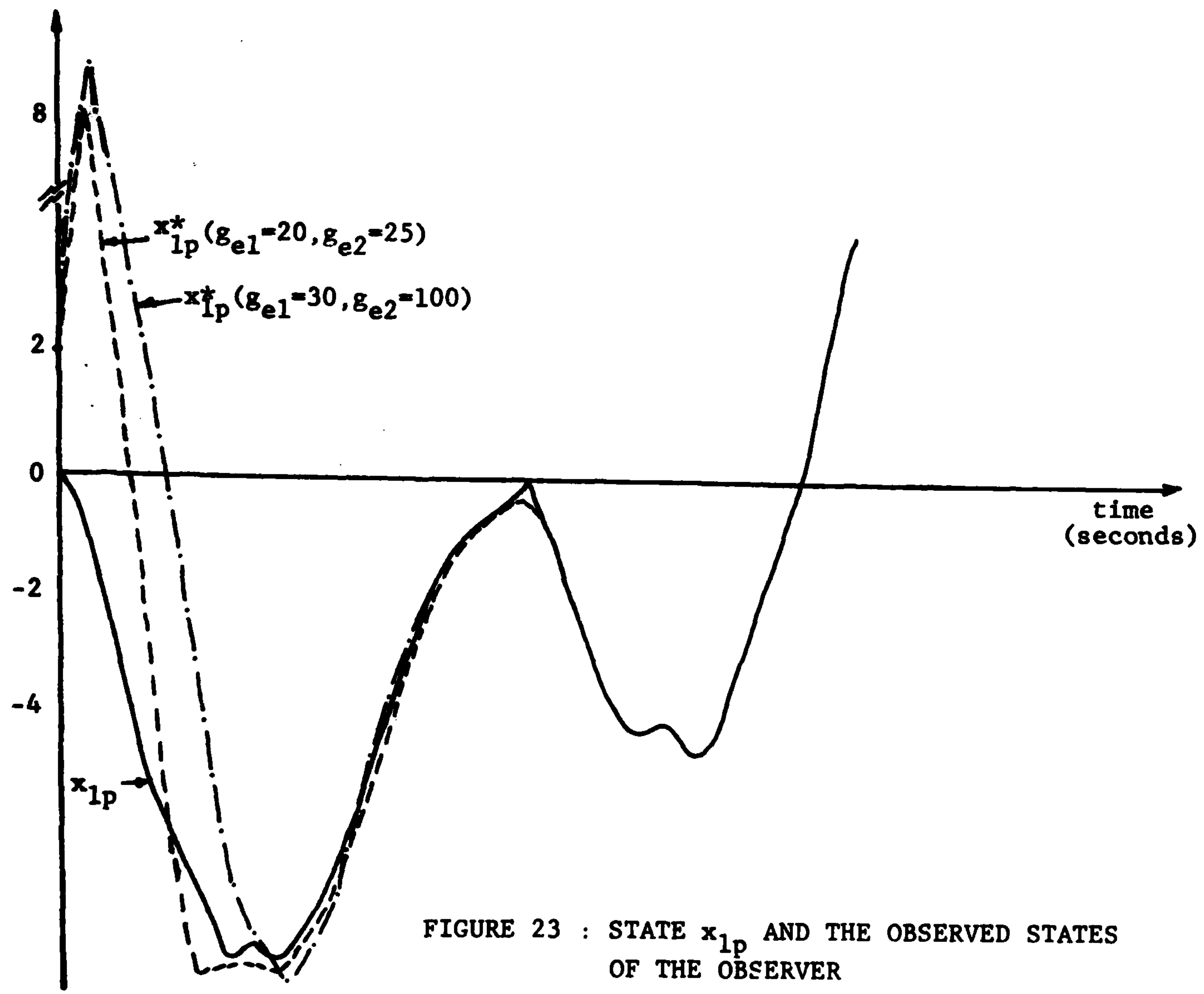


FIGURE 23 : STATE x_{1p} AND THE OBSERVED STATES OF THE OBSERVER

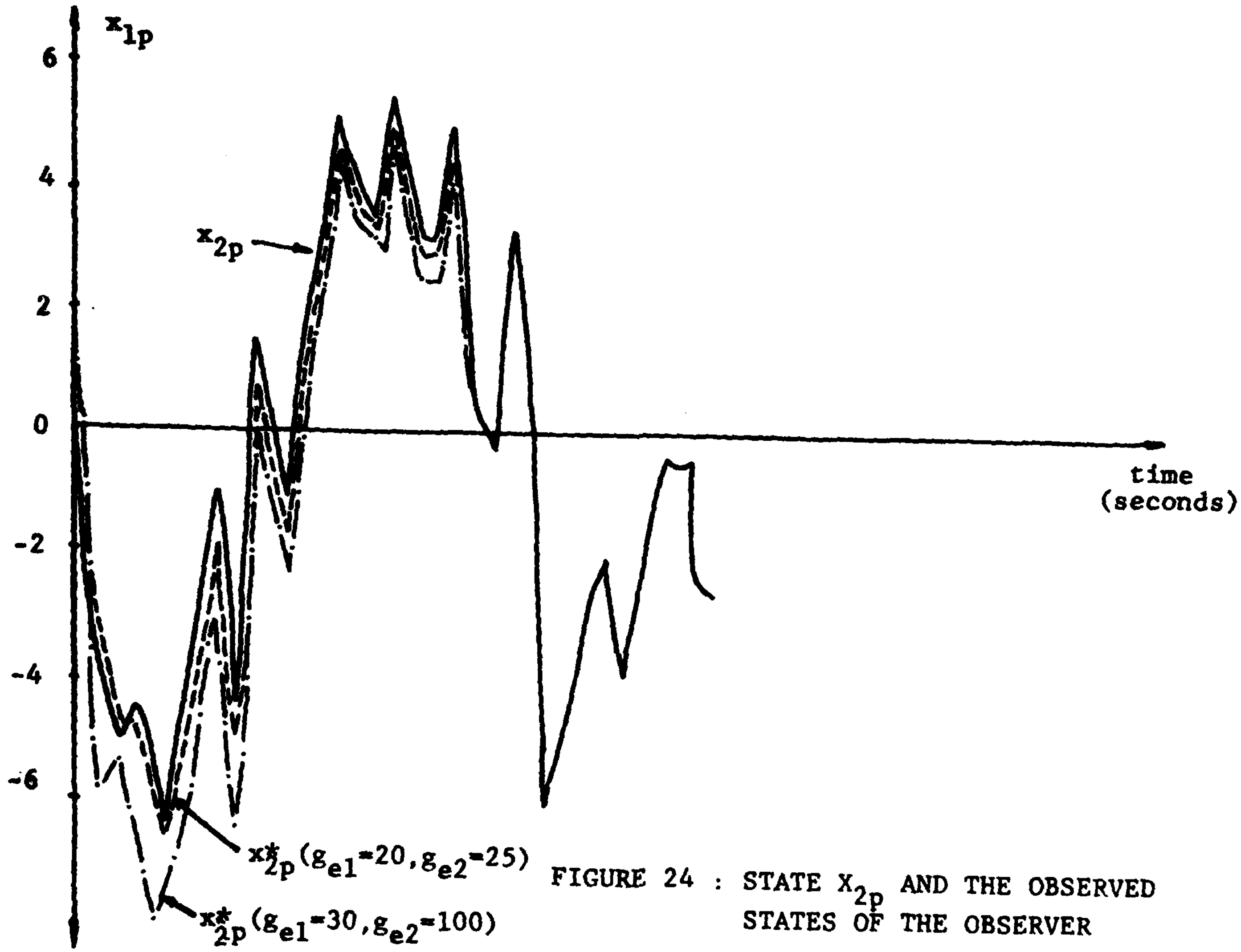


FIGURE 24 : STATE X_{2p} AND THE OBSERVED STATES OF THE OBSERVER

factory for the purpose of matching all the observed states to the real states. Reference 4 shows that the selecting larger values for g_{e1} and g_{e2} to give faster transient response for the observer is not always best. Sometimes, the large values of g_{e1} and g_{e2} will only give faster transient response for one of the observed states but not the other and the details are in Reference 4.

Since both x^*_{p1} and x^*_{p2} converge to x_{p1} and x_{p2} respectively at the same rate for λ = -20, -20 and λ = -15, -15, then we can select either set of eigenvalues for this example.

APPENDIX D

SOFTWARE FOR THE NEW MODEL

1. FLOWCHART AND DESCRIPTION OF MAJOR COMPONENTS AND SUBROUTINE

The main program in the software for the new model is called PROGRAM SKEW1. Its major computational tasks are to develop the state variable model of the complete closed-loop system and to compute the steady-state covariance of the states, the controller outputs and the errors e_A and e_B. This program is almost the same as PROGRAM SKEW, described in Appendix A, except that the inherent errors are computed from the output of channel 2, $y_{c2}(t_k + \tau)$, and its estimated value from the input of channel 1, $y^*_{c2}(t_k + \tau^*)$. All the variables from Appendix A plus the variables for calculating the estimated value of the output of the output of channel 2 are used in this program.

A flowchart for this program appears in Figure 25. The first block shows the data input. The variables are self-explanatory and discussed in Appendix A except for a new variable NTAU1, which represents the value of τ^* as

$$\tau^* = \frac{(\text{NTAU1i} - 1) *}{(\text{NT1} - 1)} T \qquad \text{D-1}$$

and

$$\text{NTAU1i} = \text{ITAU1} * \frac{\text{NT}}{\text{NT2}} = 1 \qquad \text{D-2}$$

for ITAU1 = 0, 1, 2, . . . , NT2

where $\frac{\text{NT}}{\text{NT2}}$ must be an integer.

In block two, gaussian noise is computed by using subroutine RANDU.

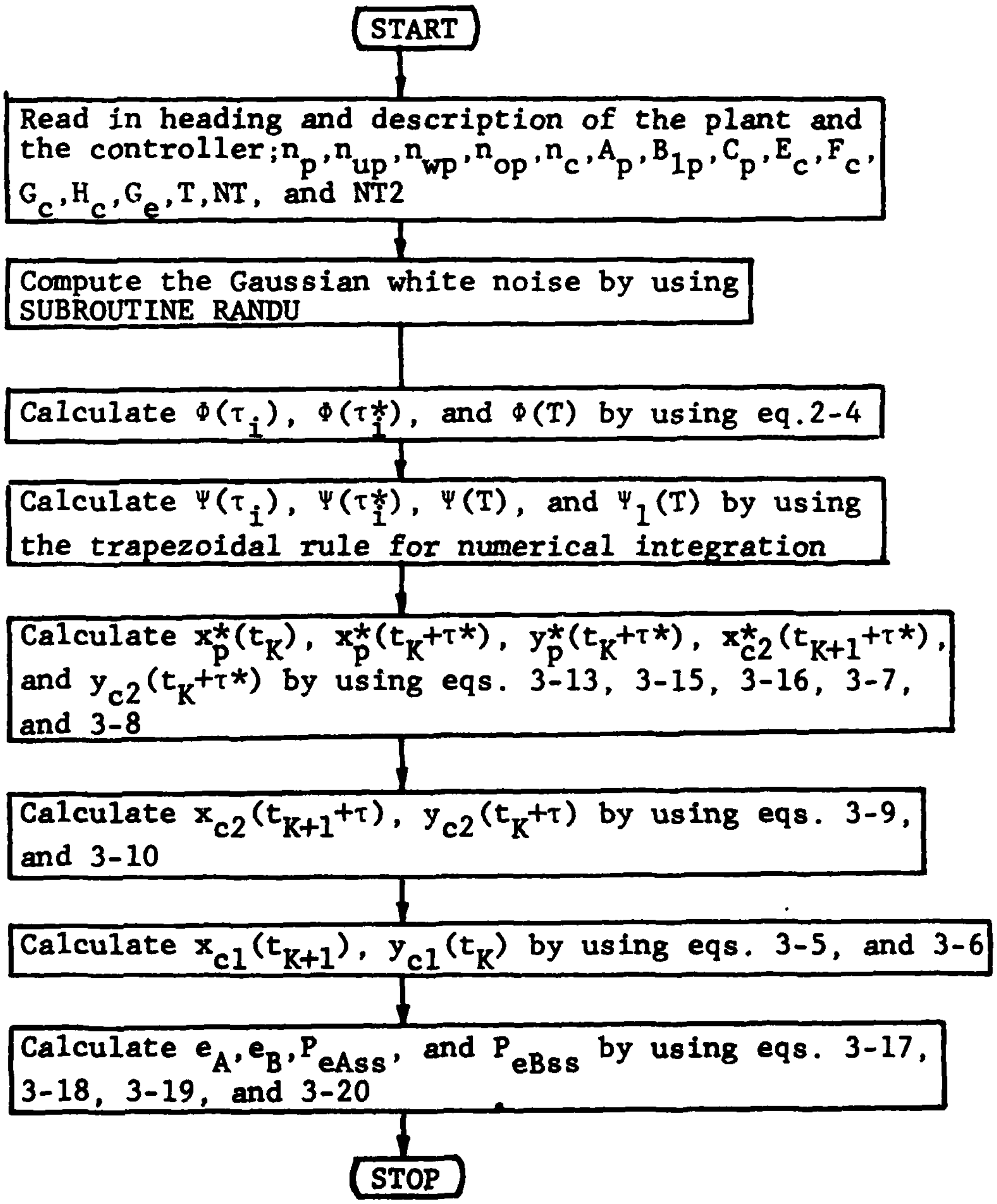


FIGURE 25 : FLOWCHART DESCRIBING THE MAJOR COMPUTATIONS OF PROGRAM SKEW1

The third block specifies the calculation of $\phi(\tau_i)$, $\phi(\tau_i^*)$, and $\phi(T)$.

The fourth block specifies the calculation of $\psi(\tau_i)$, $\psi(\tau_i^*)$, and $\psi(T)$ by using the same technique in Appendix A.

In block five, $x_p^*(t_k)$, $x_p^*(t_k + \tau^*)$, $x_{c2}^*(t_{k+1} + \tau^*)$, and $y_{c2}^*(t_k + \tau^*)$ are calculated from equations

$$x_p^*(t_k) = \phi(t_k, t_{k-1})x_p^*(t_{k-1}) + \psi(t_k, t_{k-1})U_p(t_{k-1}) + \psi_1(t_k, t_{k-1})y_p(t_{k-1})$$

$$x_p^*(t_k + \tau^*) = \phi(t_k + \tau^*, t_{k-1})x_p^*(t_{k-1}) + \psi(t_k + \tau^*, t_{k-1})U_p(t_{k-1}) + \psi_1(t_k + \tau^*, t_{k-1})y_p(t_{k-1})$$

$$x_{c2}^*(t_{k+1} + \tau^*) = F_c x_{c2}^*(t_k + \tau^*) + G_c y_p^*(t_k + \tau^*)$$

$$y_{c2}^*(t_k + \tau^*) = H_c x_{c2}^*(t_k + \tau^*) + E_c y_p^*(t_k + \tau^*)$$

In block six, $x_{c2}(t_{k+1} + \tau)$ and $y_{c2}(t_k + \tau)$ are calculated from equations

$$x_{c2}(t_{k+1} + \tau) = F_c x_{c2}(t_k + \tau) + G_c y_p(t_k + \tau)$$

$$y_{c2}(t_k + \tau) = H_c x_{c2}(t_k + \tau) + E_c y_p(t_k + \tau)$$

In block seven, $x_{c1}(\dot{t}_{k+1})$ and $y_{c1}(t_k)$ are calculated from equations

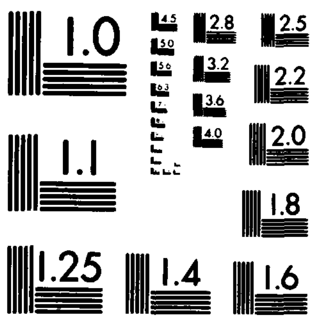
$$x_{c1}(t_{k+1}) = F_c x_{c1}(t_k) + G_c y_p(t_k)$$

$$y_{c1}(t_k) = H_c x_{c1}(t_k) + E_c y_p(t_k)$$

AD-A082 622 MISSOURI UNIV-COLUMBIA COLL OF ENGINEERING F/G 17/7
INHERENT ERROR IN ASYNCHRONOUS DIGITAL FLIGHT CONTROLS.(U)
FEB 80 C SLIVINSKY, S PATUNTAWAPIBAL AFOSR-76-2966
UNCLASSIFIED AFOSR-TR-80-[illegible] NL

2 of 2
AD A082622

END
DATE FILMED
5-80
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

In block eight, the inherent errors, e_A, e_B, and the steady-state covariance of errors: PEASS, PEBSS, are computed from equations

$$e_A(t) = y_{c1}(t_k) - y_{c2}(t_k + \tau)$$

$$t_k + \tau \leq t < t_{k+1},\ 0 \leq \tau < T,\ k = 0, 1, \ldots$$

$$e_B(t) = y_{c1}(t_{k+1}) - y_{c2}(t_k + \tau)$$

$$t_{k+1} \leq t < t_{k+1} + \tau,\ 0 < \tau < T,\ k = 0, 1, \ldots$$

$$\text{PEASS} = \frac{1}{N} \sum_{i=201}^{N} [e_{Ai} - \bar{e}_A]\,[e_{Ai} - \bar{e}_A]^T$$

$$\text{PEBSS} = \frac{1}{N} \sum_{i=201}^{N} [e_{Bi} - \bar{e}_B]\,[e_{Bi} - \bar{e}_B]^T$$

Since the majority of this program is the same as the program in Appendix A, then only new arrays will be shown in the following

XP1 (NP)	XC3 (NC)
YP1 (NOP)	XPTAU1 (UP)
YC3 (NUP)	PSIT2 (NP, NUP)

2. EXAMPLE

The computer listing for PROGRAM SKEW1, of the new model, written in FORTRAN, appears in Appendix E. For each value of τ_i, PEASS and PEBSS for each value of τ_i^* are shown at the end of the listing. To compute PEASS and PEBSS of the new model, the system has been waiting until the state observer x_p^* is equal or close to x_p and the system is in steady-state transition.

From Appendix C, the state observer x_p^* equals x_p at approximately

30T (g_{e1} = 20, g_{e2} = 25). Thus, PEASS and PEBSS can be computed at the same time (between 200T and 300T) as PEASS and PEBSS of the basic model in Appendix A are computed.

APPENDIX E

COMPUTER PROGRAM LISTING
FOR
PROGRAM SKEW1 AND EXAMPLE
OF OUTPUT WRITTEN IN FORTRAN

```
CCCCC PROVIDE MAXIMA FOR CALLED ARRAYS
      NPM = 2
      NUPM = 2
      NWPM = 2
      NOPM = 1
      NCM = 2
      NHM = NPM + NUPM
      NFM = NPM + 2*NCM
      NRRM = 2*NPM +4
C
C     ***********************************************************************
C     READ INPUT DATA
C     ***********************************************************************
C
      WRITE(6,899)
899   FORMAT('1')
  100 READ(5,900) ID
  900 FORMAT(20A4)
      WRITE(6,902) ID
  902 FORMAT('1',20A4)
      READ(5,906)NP,NUP,NWP,NOP,NC
  906 FORMAT(5I3)
      WRITE(6,908) NP,NUP,NWP,NOP,NC
  908 FORMAT('ONO. OF PLANT STATES = ',I3/
     1 ' NO. OF PLANT INPUTS = ',I3/
     2 ' NO. OF DISTURBANCE INPUTS = ', I3/
     4 ' NO. OF PLANT OUTPUTS = ', I3/
     5 ' NO. OF CONTROLLER STATES (EACH CONTROLLER) = ',I3)
       WRITE(6,910)
  910 FORMAT('OPLANT STATE MATRIX -- AP')
  110 DO 112 I = 1,NP
      READ(5,914) (AP(I,J),J=1,NP)
112   WRITE(6,913) (AP(I,J),J=1,NP)
913   FORMAT(' ',8G13.6)
914   FORMAT(6F12.7)
915   FORMAT(8G13.6)
      WRITE(6,916)
  916 FORMAT('OPLANT CONTROL INPUT MATRIX -- B1P')
  120 DO 122 I = 1,NP
      READ(5,914)(B1P(I,J),J=1,NUP)
122   WRITE(6,913)(B1P(I,J),J=1,NUP)
      WRITE(6,918)
918   FORMAT('OOBSERVER MATRIX-- GE')
  130 DO 132 I=1,NP
      READ(5,914)(GE(I,J),J=1,NWP)
132   WRITE(6,913)(GE(I,J),J=1,NWP)
      WRITE(6,920)
  920 FORMAT('OPLANT OUTPUT MATRIX -- CP')
  140 DO 142 I=1,NOP
      READ(5,914)(CP(I,J),J=1,NP)
142   WRITE(6,913)(CP(I,J),J=1,NP)
      WRITE(6,922)
  922 FORMAT('OCONTROLLER STATE MATRIX -- FC')
  150 DO 152 I =1,NC
```

```
      READ(5,914)(FC(I,J),J=1,NC)
152   WRITE(6,913)(FC(I,J),J=1,NC)
      WRITE(6,924)
  924 FORMAT('OCONTROLLER CONTROL INPUT MATRIX -- GC ')
  160 DO 162 I=1,NC
      READ(5,914)(GC(I,J),J=1,NOP)
  162 WRITE(6,913)(GC(I,J),J=1,NOP)
      WRITE(6,925)
  925 FORMAT('OCONTROLLER OUTPUT MATRIX (STATES) -- HC')
  170 DO 172 I=1,NUP
      READ(5,914)(HC(I,J),J=1,NC)
  172 WRITE(6,913)(HC(I,J),J=1,NC)
      WRITE(6,926)
  926 FORMAT('OCONTROLLER OUTPUT MATRIX (INPUTS) -- EC')
  180 DO 182 I=1,NUP
      READ(5,914)(EC(I,J),J=1,NOP)
  182 WRITE(6,913)(EC(I,J),J=1,NOP)
      READ(5,928) T,NT
928   FORMAT(F10.4,I5)
      XNT = NT
      DELTA = T/(XNT-1)
      WRITE(6,930) T,NT
930   FORMAT('1T = ',F10.4/
     1 ' NT = ',I5/
     3 ' T = SAMPLE RATE.'/
     6 ' DELTA = T/(NT-1) = INCREMENT USED IN THE NUMERICAL'/
     7 '                    INTEGRATIONS TO COMPUTE PSITAU,PSIT,PSIT2'/
     9 '                    PSITAU1 USING TRAPEZOIDAL RULE.'//)
      WRITE(6,931)
931   FORMAT(2X,' W IS THE EXTERNAL INPUT (WHITE GAUSSIAN NOISE WITH
     1 '            MEAN = 0.0, AND VARIANCE = 1.0')
C
C     ****************************************************************
C     GENERATE WHITE GAUSSIAN NOISE WITH MEAN = 0 AND VARIANCE = 1
C     ****************************************************************
C
      IX = 11111
      DO 192 I = 1,1200
      A = 0.0
      DO 193 J = 1,12
      CALL RANDU(IX,IY,Y)
      IX = IY
193   A = A+Y
192   W(I) = A-6
      DO 1199 I = 1,NUP
      DO 1199 J = 1,NP
      ECCP(I,J) = 0.0
      DO 1199 K = 1,NOP
1199  ECCP(I,J) = ECCP(I,J) + EC(I,K)*CP(K,J)
802   FORMAT(5X,5G13.6)
835   FORMAT(5X,5G13.6)
C
C     ****************************************************************
C     CALCULATE PHIT(0),PHIT(DELTA),PHIT(2*DELTA),...,PHIT(T)
```

```
C      ***************************************************************
C
       DELHLF = DELTA/2.0
       TI = 0.0
       DO 402 II = 1,NP
       DO 402 JJ = 1,NP
       IF(II.EQ.JJ) GO TO 403
       PHIT1(II,JJ,1) = 0.0
       GO TO 402
403    PHIT1(II,JJ,1) = 1.0
402    CONTINUE
       DO 4 I1 = 2,NT
       TI = TI+DELTA
       PHIT1(1,1,I1) = 1.
       PHIT1(1,2,I1) = (1./10.)*(1.-EXP(-10.*TI))
       PHIT1(2,1,I1) = 0.0
4      PHIT1(2,2,I1) = EXP(-10.*TI)
       DO 400 II = 1,NP
       DO 400 JJ = 1,NP
400    PHIT(II,JJ) = PHIT1(II,JJ,NT)
       N1 = 0
       DO 1801 KK1 = 1,6
       NTAU1 = (KK1-1)*10
       NTAU1 = NTAU1 + 1
        TAUU = N1*0.0125/5.
       N1 = N1 + 1
       DO 1800 KK2 = 1,6
       NTAU = (KK2-1)*10
       NTAU = NTAU + 1
C      INITIAL VALUE YC2(-TIME+TAU), YC3(-TIME), YC1(-TIME)
       DO 31 I = 1,NUP
       YC2(I) = 0.0
       YC3(I) = 0.0
31     YC1(I) = 0.0
       DO 32 I = 1,NUP
32     E1(I) = YC3(I) - YC2(I)
C      FROM INITIAL VALUE XP(TIME), AND YP(TIME) ARE EQUAL TO ZERO
       DO 35 I = 1,NP
       XP1(I) = 2.0
35     XP(I) = 0.0
       DO 435 I = 1,NOP
       Q = 0.0
       DO 436 J = 1,NP
436    Q = Q + CP(I,J)*XP1(J)
435    YP1(I) = Q
       DO 36 I = 1,NOP
36     YP(I) = 0.0
C      FROM INITIAL VALUE XC1(TIME+T), YC1(TIME) ARE EQUAL TO ZERO
       DO 37 I = 1,NC
       XC1(I) = 0.0
       XC3(I) = 0.0
37     XC2(I) = 0.0
       DO 52 I = 1,NUP
52     YC1(I) = 0.0
```

```
C       ****************************************************************
C
C       CALCULATE PHIT(TAU)
C
C       ****************************************************************
        IDEL = 0
        DO 5000 I = 1,NT
        IDEL = IDEL + 1
        IF(NTAU.EQ.NTAU1.AND.IDEL.EQ.NTAU) GO TO 7
        IF(IDEL.EQ.NTAU) GO TO 16
        IF(IDEL.EQ.NTAU1) GO TO 22
        GO TO 5000
7       DO 8 II = 1,NP
        DO 8 JJ = 1,NP
        PHTAU(II,JJ) = PHIT1(II,JJ,IDEL)
8       PHTAU1(II,JJ) = PHIT1(II,JJ,IDEL)
        GO TO 5000
16      DO 17 II = 1,NP
        DO 17 JJ = 1,NP
17      PHTAU(II,JJ) = PHIT1(II,JJ,IDEL)
        GO TO 5000
22      DO 23 II = 1,NP
        DO 23 JJ = 1,NP
23      PHTAU1(II,JJ) = PHIT1(II,JJ,IDEL)
5000    CONTINUE
C       ****************************************************************
C
C       CALCULATE PSIT(TAU),PSIT(T)
C       ****************************************************************
C
        DO 550 I = 1,NP
        DO 550 J = 1,NWP
        P1(I,J) = 0.0
550     PS(I,J) = 0.0
        DO 551 I = 1,NP
        DO 551 J = 1,NWP
        PSIT2(I,J) = 0.0
        PSTAU(I,J) = 0.0
        PSTAU1(I,J) = 0.0
        PSIT(I,J) = 0.0
        DO 551 K = 1,NP
        PSIT2(I,J) = PSIT2(I,J) + PHIT1(I,K,NT)*GE(K,J)
        PSTAU(I,J) = PSTAU(I,J) + PHIT1(I,K,NTAU)*B1P(K,J)
        PSTAU1(I,J) = PSTAU1(I,J) + PHIT1(I,K,NTAU1)*B1P(K,J)
551     PSIT(I,J) = PSIT(I,J) + PHIT1(I,K,NT)*B1P(K,J)
        DO 552 I = 1,NP
        DO 552 J = 1,NWP
        PSIT2(I,J) = DELHLF*PSIT2(I,J)
        PSTAU(I,J) = DELHLF*PSTAU(I,J)
        PSTAU1(I,J) = DELHLF*PSTAU1(I,J)
552     PSIT(I,J) = DELHLF*PSIT(I,J)
60      DO 61 I1 = 2,NT
        I2 = NT-I1+1
        DO 62 I = 1,NP
```

```
      DO 62 J = 1,NWP
      PSIT2(I,J) = PSIT2(I,J) + P1(I,J)
62    PSIT(I,J) = PSIT(I,J) + PS(I,J)
      DO 63 I = 1,NP
      DO 63 J = 1,NWP
      P1(I,J) = 0.0
      PS(I,J) = 0.0
      DO 63 K = 1,NP
      P1(I,J) = P1(I,J) + PHIT1(I,K,I2)*GE(K,J)
63    PS(I,J) = PS(I,J) + PHIT1(I,K,I2)*B1P(K,J)
      DO 64 I = 1,NP
      DO 64 J = 1,NWP
      P1(I,J) = DELHLF*P1(I,J)
64    PS(I,J) = DELHLF*PS(I,J)
      DO 66 I = 1,NP
      DO 66 J = 1,NWP
      PSIT2(I,J) = P1(I,J) + PSIT2(I,J)
66    PSIT(I,J) = PS(I,J) + PSIT(I,J)
61    CONTINUE
      DO 80 I = 1,NP
      DO 80 J = 1,NWP
80    PS(I,J) = 0.0
      IF(NTAU1.EQ.1) GO TO 85
      GO TO 90
85    DO 86 II = 1,NP
      DO 86 JJ = 1,NWP
86    PSTAU1(II,JJ) = 0.0
      GO TO 67
90    DO 91 I1 = 2,NTAU1
      I2 = NTAU1-I1+1
      DO 92 I = 1,NP
      DO 92 J = 1,NWP
92    PSTAU1(I,J) = PSTAU1(I,J) + PS(I,J)
      DO 93 I = 1,NP
      DO 93 J = 1,NWP
      PS(I,J) = 0.0
      DO 93 K = 1,NP
93    PS(I,J) = PS(I,J) + PHIT1(I,K,I2)*B1P(K,J)
      DO 94 I = 1,NP
      DO 94 J = 1,NWP
94    PS(I,J)=DELHLF*PS(I,J)
      DO 96 I = 1,NP
      DO 96 J = 1,NWP
96    PSTAU1(I,J) = PS(I,J) + PSTAU1(I,J)
91    CONTINUE
67    DO 68 I = 1,NP
      DO 68 J = 1,NWP
68    PS(I,J) = 0.0
      IF(NTAU.EQ.1) GO TO 55
      GO TO 69
55    DO 58 II = 1,NP
      DO 58 JJ = 1,NWP
58    PSTAU(II,JJ) = 0.0
      GO TO 77
```

```
69     DO 70 I1 = 2,NTAU
       I2 = NTAU-I1+1
       DO 71 I = 1,NP
       DO 71 J = 1,NWP
71     PSTAU(I,J) = PSTAU(I,J) + PS(I,J)
       DO 72 I = 1,NP
       DO 72 J = 1,NWP
       PS(I,J) = 0.0
       DO 72 K = 1,NP
72     PS(I,J) = PS(I,J) + PHIT1(I,K,I2)*B1P(K,J)
       DO 73 I = 1,NP
       DO 73 J = 1,NWP
73     PS(I,J) = DELHLF*PS(I,J)
       DO 76 I = 1,NP
       DO 76 J = 1,NWP
76     PSTAU(I,J) = PS(I,J) + PSTAU(I,J)
70     CONTINUE
77     CONTINUE
C
C      *****************************************************************
C      START TIME LOOP
C      TIME = 0
C      *****************************************************************
C
       NN1 = 1
       DO 412 I = 1,NOP
412    UP(I) = W(NN1)-YC1(I)
       DO 413 I = 1,NP
       Q = 0.0
       DO 414 J = 1,NP
414    Q = Q + PHTAU1(I,J)*XP1(J)
413    XW1(I) = Q
       DO 415 I = 1,NP
       Q = 0.0
       DO 416 J = 1,NUP
416    Q = Q + PSTAU1(I,J)*UP(J)
415    XW2(I) = Q
       DO 417 I = 1,NP
417    XPTAU1(I) = XW1(I) + XW2(I)
       DO 418 I = 1,NOP
       Q = 0.0
       DO 419 J = 1,NP
419    Q = Q + CP(I,J)*XPTAU1(J)
418    YPTAU1(I) = Q
       DO 420 I = 1,NUP
       Q = 0.0
       DO 421 J = 1,NC
421    Q = Q + HC(I,J)*XC3(J)
420    XW1(I) = Q
       DO 422 I = 1,NUP
       Q = 0.0
       DO 423 J = 1,NOP
423    Q = Q + EC(I,J)*YPTAU1(J)
422    XW2(I) = Q
```

```
      DO 424 I = 1,NUP
424   YC3(I) = XW1(I) + XW2(I)
      DO 425 I = 1,NC
      Q = 0.0
      DO 426 J = 1,NC
426   Q = Q + FC(I,J)*XC3(J)
425   XW1(I) = Q
      DO 427 I = 1,NC
      Q = 0.0
      DO 428 J = 1,NOP
428   Q = Q + GC(I,J)*YPTAU1(J)
427   XW2(I) = Q
      DO 429 I = 1,NC
429   XC3(I) = XW1(I) + XW2(I)
      DO 430 I = 1,NUP
430   E2(I) = YC3(I) - YC2(I)
      DO 220 II1 = 1,300
C     ******************************************************************
C     CALCULATE XP(TIME+TAU), AND YP(TIME+TAU)
C     ******************************************************************
      DO 253 I = 1,NP
      Q = 0.0
      DO 254 J = 1,NP
254   Q = Q + PHTAU(I,J)*XP(J)
253   XW1(I) = Q
      DO 255 I = 1,NP
      Q = 0.0
      DO 256 J = 1,NUP
256   Q = Q + PSTAU(I,J)*UP(J)
255   XW2(I) = Q
      DO 257 I = 1,NP
257   XPTAU(I) = XW1(I) + XW2(I)
      DO 260 I = 1,NOP
      Q = 0.0
      DO 261 J = 1,NP
261   Q = Q + CP(I,J)*XPTAU(J)
260   YPTAU(I) = Q
C     ******************************************************************
C     TIME = TIME + T
C     CALCULATE XC2(TIME+T+TAU),AND YC2(TIME+TAU)
C     ******************************************************************
      DO 300 I = 1,NUP
      Q = 0.0
      DO 301 J = 1,NC
301   Q = Q + HC(I,J)*XC2(J)
300   XW1(I) = Q
      DO 302 I = 1,NUP
      Q = 0.0
      DO 303 J = 1,NOP
303   Q = Q + EC(I,J)*YPTAU(J)
302   XW2(I) = Q
      DO 304 I = 1,NUP
304   YC2(I) = XW1(I) + XW2(I)
      DO 305 I = 1,NC
```

```
      Q = 0.0
      DO 306 J = 1,NC
306   Q = Q + FC(I,J)*XC2(J)
305   XW1(I) = Q
      DO 307 I = 1,NC
      Q = 0.0
      DO 308 J = 1,NOP
308   Q = Q + GC(I,J)*YPTAU(J)
307   XW2(I) = Q
      DO 309 I = 1,NC
309   XC2(I) = XW1(I) + XW2(I)
C     ****************************************************************
C     CALCULATE E1
C     ****************************************************************
      DO 290 I = 1,NUP
290   E1(I) = YC3(I) - YC2(I)
C     *****************************************************************
C     ESTIMATE XP*(TIME), AND YP*(TIME)
C     *****************************************************************
      DO 432 I = 1,NUP
432   YP2(I) = YP(I) - YP1(I)
      DO 405 I = 1,NP
      Q = 0.0
      DO 406 J = 1,NP
406   Q = Q + PHIT(I,J)*XP1(J)
405   XW1(I) = Q
      DO 407 I = 1,NP
      Q = 0.0
      DO 408 J = 1,NUP
408   Q = Q + PSIT(I,J)*UP(J)
407   XW2(I) = Q
      DO 409 I = 1,NP
      Q = 0.0
      DO 410 J = 1,NUP
410   Q = Q + PSIT2(I,J)*YP2(J)
409   XW3(I) = Q
      DO 411 I = 1,NP
411   XP1(I) = XW1(I) + XW2(I) + XW3(I)
      DO 433 I = 1,NOP
      Q = 0.0
      DO 434 J = 1,NP
434   Q = Q + CP(I,J)*XP1(J)
433   YP1(I) = Q
C     ****************************************************************
C     CALCULATE XP(TIME), AND YP(TIME)
C     ****************************************************************
373   DO 500 I = 1,NP
      Q = 0.0
      DO 501 J = 1,NP
501   Q = Q + PHIT(I,J)*XP(J)
500   XW1(I) = Q
      DO 502 I = 1,NP
      Q = 0.0
      DO 503 J = 1,NUP
```

```
503    Q = Q + PSIT(I,J)*UP(J)
502    XW2(I) = Q
       DO 504 I = 1,NP
504    XP(I) = XW1(I) + XW2(I)
       DO 507 I = 1,NOP
       Q = 0.0
       DO 508 J = 1,NP
508    Q = Q + CP(I,J)*XP(J)
507    YP(I) = Q
C      ****************************************************************
C      CALCULATE XC1(2*TIME), AND YC1(TIME)
C      ****************************************************************
       DO 700 I = 1,NUP
       Q = 0.0
       DO 701 J = 1,NC
701    Q = Q + HC(I,J)*XC1(J)
700    XW1(I) = Q
       DO 702 I = 1,NUP
       Q = 0.0
       DO 703 J = 1,NOP
703    Q = Q + EC(I,J)*YP(J)
702    XW2(I) = Q
       DO 704 I = 1,NUP
704    YC1(I) = XW1(I) + XW2(I)
       DO 705 I = 1,NC
       Q = 0.0
       DO 706 J = 1,NC
706    Q = Q + FC(I,J)*XC1(J)
705    XW1(I) = Q
       DO 707 I = 1,NC
       Q = 0.0
       DO 708 J = 1,NOP
708    Q = Q + GC(I,J)*YP(J)
707    XW2(I) = Q
       DO 709 I = 1,NC
709    XC1(I) = XW1(I) + XW2(I)
C      *****************************************************************
C      ESTIMATE XP*(TIME+TAU), AND YP*(TIME+TAU)
C      *****************************************************************
       NN1 = NN1 + 1
       DO 221 I = 1,NOP
221    UP(I) = W(NN1)-YC1(I)
       DO 520 I = 1,NP
       Q = 0.0
       DO 521 J = 1,NP
521    Q = Q + PHTAU1(I,J)*XP1(J)
520    XW1(I) = Q
       DO 522 I = 1,NP
       Q = 0.0
       DO 523 J = 1,NUP
523    Q = Q + PSTAU1(I,J)*UP(J)
522    XW2(I) = Q
       DO 524 I = 1,NP
524    XPTAU1(I) = XW1(I) + XW2(I)
```

```
      DO 525 I = 1,NOP
      Q = 0.0
      DO 526 J = 1,NP
526   Q = Q + CP(I,J)*XPTAU1(J)
525   YPTAU1(I) = Q
C     ******************************************************************
C     CALCULATE XC*(2*TIME+TAU*), AND YC*(TIME+TAU*)
C     ******************************************************************
      DO 529 I = 1,NUP
      Q = 0.0
      DO 530 J = 1,NC
530   Q = Q + HC(I,J)*XC3(J)
529   XW1(I) = Q
      DO 531 I = 1,NUP
      Q = 0.0
      DO 532 J = 1,NOP
532   Q = Q + EC(I,J)*YPTAU1(J)
531   XW2(I) = Q
      DO 533 I = 1,NUP
533   YC3(I) = XW1(I) + XW2(I)
      DO 535 I = 1,NC
      Q = 0.0
      DO 536 J = 1,NC
536   Q = Q + FC(I,J)*XC3(J)
535   XW1(I) = Q
      DO 537 I = 1,NC
      Q = 0.0
      DO 538 J = 1,NOP
538   Q = Q + GC(I,J)*YPTAU1(J)
      DO 539 I = 1,NC
539   XC3(I) = XW1(I) + XW2(I)
C     *****************************************************************
C     CALCULATE E2
C     *****************************************************************
      DO 540 I = 1,NUP
540   E2(I) = YC3(I) - YC2(I)
      DO 560 I = 1,NUP
      EA(II1) = E1(I)
560   EB(II1) = E2(I)
220   CONTINUE
      SMEANA = 0.0
      SMEANB = 0.0
      DO 561 I = 201,300
      SMEANA = SMEANA + EA(I)
561   SMEANB = SMEANB + EB(I)
      SMEANA = SMEANA/100
      SMEANB = SMEANB/100
      VEA = 0.0
      VEB = 0.0
      DO 563 I = 201,300
      VEA = VEA + (EA(I)-SMEANA)**2
563   VEB = VEB + (EB(I)-SMEANB)**2
      PEASS(KK2) = VEA/100
      PEBSS(KK2) = VEB/100
```

```
1800  CONTINUE
570   FORMAT(5X,'PEASS( TAU* = ',F12.8,')')
      WRITE(6,571)(PEASS(I),I=1,6)
571   FORMAT(5X,F18.10)
      WRITE(6,572) TAU1
572   FORMAT(5X,'PEBSS( TAU* = ',F12.8,')')
      WRITE(6,571)(PEBSS(I),I=1,6)
1801  CONTINUE
      STOP
      END
      SUBROUTINE RANDU(IX,IY,YFL)
      IY = IX*65539
      IF(IY)5,6,6
5     IY = IY + 2147483647+1
6     YFL = IY
      YFL = YFL*0.4656613E-9
      RETURN
      END
```

```
EXAMPLE------2TH ORDER PLANT, 1ST ORDER CONTROLLER
NO. OF PLANT STATES = 2
NO. OF PLANT INPUTS = 1
NO. OF EXTERNAL INPUT = 1
NO. OF PLANT OUTPUTS = 1
NO. OF CONTROLLER STATES ( EACH CONTROLLER) = 1

PLANT STATE MATRIX -- AP
  .0            1.0
  .0          -10.0

PLANT CONTROL INPUT MATRIX -- BP
  .0
  200.0


PLANT OUTPUT MATRIX -- CP
  1.0           .0


CONTROLLER STATE MATRIX -- FC
  .523810

OBSERVER MATRIX -- GE
  20.0
  25.0
CONTROLLER CONTROL INPUT MATRIX -- GC
 -.18162

CONTROLLER OUTPUT MATRIX (STATES) -- HC
  1.0

CONTROLLER OUTPUT MATRIX (INPUTS) -- EC
  1.381
NT = 51
T = SAMPLE PERIOD = 0.0125 SEC
DELTA = T/(NT-1) = INCREMENT USED IN THE NUMERICAL
                   INTEGRATIONS TO COMPUTE PSITAU,PSIT,PSIT2,
                   PSITAU1 USING TRAPEZOIDAL RULE.
W IS THE EXTERNAL INPUT (WHITE GAUSSIAN NOISE WITH
  MEAN = 0.0, AND VARIANCE = 1.0
THE STEADY STATE SAMPLE VARIANCE OF ERRORS
   PEASS( TAU* = 0.0 )
          0.0
          0.0002195683
          0.0008547062
          0.0018880414
          0.0033245913
          0.0051899776
   PEBSS( TAU* = 0.0 )
          0.0051899776
          0.0033477221
          0.0019137899
          0.0008709673
```

0.0002244494
0.0

PEASS(TAU* = 0.0025)
0.0002195683
0.0
0.0002097336
0.0008310268
0.0018685847
0.0033477221
PEBSS(TAU* = 0.0025)
0.0074686036
0.0052524656
0.0034256792
0.0019714807
0.0008955190
0.0002240368

PEASS(TAU* = 0.005)
0.0008547062
0.0002097336
0.0
0.0002076342
0.0008371675
0.0019137899
PEBSS(TAU* = 0.005)
0.0100931898
0.0075152330
0.0053092465
0.0034589050
0.0019702637
0.0008699684

PEASS(TAU* = 0.0075)
0.0018880414
0.0008310268
0.0002076342
0.0
0.0002127137
0.0008709673
PEBSS(TAU* = 0.0075)
0.0130441934
0.0101163760
0.0075445175
0.0053128712
0.0034278994
0.0019165913

PEASS(TAU* = 0.01)
0.0033245913
0.0018685847
0.0008371675
0.0002127137
0.0

```
        0.0002244494
PEBSS(  TAU* = 0.01 )
        0.0163251571
        0.0130585469
        0.0101340003
        0.0075355470
        0.0052701645
        0.0033652126

PEASS(  TAU* = 0.0125 )
        0.0051899776
        0.0033477221
        0.0019137899
        0.0008709673
        0.0002244494
        0.0
PEBSS(  TAU* = 0.0125 )
        0.0199597441
        0.0163656212
        0.0131007358
        0.0101497732
        0.0075195357
        0.0052378476
```

APPENDIX F

COMPUTER PROGRAM LISTING
FOR
ALGORITHM FOR ESTIMATING τ
AND EXAMPLE OF OUTPUT WRITTEN
IN FORTRAN

```
      DIMENSION AP(2,2),B1P(2,2),GE(2,2),CP(1,2),FC(2,2),GC(2,1),
     1 HC(2,2),EC(2,1),PHIT1(4,4,101),PSIT1(4,4),PHTAU(4,4),YP2(2),
     2 PHIT(4,4),PSTAU(4,4),PSIT(4,4),
     4 INDEX(4),W(4000),PS(4,4),PHTAU1(4,4),PSTAU1(4,4),YC1(2),
     5 YC2(2),E1(2),E2(2),XP(2),XPTAU(2),XP1(2),YP(2),YPTAU(2),
     6 YP1(2),XC1(2),XC2(2),AM(4,4),PT(4,4),P1(4,4),D1(4,4),
     7 D2(4,4),D3(4),XW1(2),XW2(2),ECCP(2,2),D(4,4),YPTAU1(2),
     8 PEASS(50),PEBSS(50),EA(300),EB(300),YC3(2),XC3(2),UP(2),
     9 XW3(2),XPTAU1(2)
CCCCC PROVIDE MAXIMA FOR CALLED ARRAYS
      NPM = 2
      NUPM = 2
      NWPM = 2
      NOPM = 1
      NCM = 2
      NHM = NPM + NUPM
      NFM = NPM + 2*NCM
      NRRM = 2*NPM +4
C
C     ******************************************************************
C     READ INPUT DATA
C     ******************************************************************
C
      WRITE(6,899)
899   FORMAT('1')
  100 READ(5,900) ID
  900 FORMAT(20A4)
      WRITE(6,902) ID
  902 FORMAT('1',20A4)
      READ(5,906)NP,NUP,NWP,NOP,NC
  906 FORMAT(5I3)
      WRITE(6,908) NP,NUP,NWP,NOP,NC
  908 FORMAT('ONO. OF PLANT STATES = ',I3/
     1 ' NO. OF PLANT INPUTS = ',I3/
     2 ' NO. OF DISTURBANCE INPUTS = ', I3/
     4 ' NO. OF PLANT OUTPUTS = ', I3/
     5 ' NO. OF CONTROLLER STATES (EACH CONTROLLER) = ',I3)
       WRITE(6,910)
  910 FORMAT('OPLANT STATE MATRIX -- AP')
  110 DO 112 I = 1,NP
      READ(5,914) (AP(I,J),J=1,NP)
112   WRITE(6,913) (AP(I,J),J=1,NP)
913   FORMAT(' ',8G13.6)
914   FORMAT(6F12.7)
915   FORMAT(8G13.6)
      WRITE(6,916)
  916 FORMAT('OPLANT CONTROL INPUT MATRIX -- B1P')
  120 DO 122 I = 1,NP
      READ(5,914)(B1P(I,J),J=1,NUP)
122   WRITE(6,913)(B1P(I,J),J=1,NUP)
      WRITE(6,918)
918   FORMAT('OOBSERVER MATRIX-- GE')
  130 DO 132 I=1,NP
      READ(5,914)(GE(I,J),J=1,NWP)
```

```
132   WRITE(6,913)(GE(I,J),J=1,NWP)
      WRITE(6,920)
  920 FORMAT('OPLANT OUTPUT MATRIX -- CP')
  140 DO 142 I=1,NOP
      READ(5,914)(CP(I,J),J=1,NP)
142   WRITE(6,913)(CP(I,J),J=1,NP)
      WRITE(6,922)
  922 FORMAT('OCONTROLLER STATE MATRIX -- FC')
  150 DO 152 I =1,NC
      READ(5,914)(FC(I,J),J=1,NC)
152   WRITE(6,913)(FC(I,J),J=1,NC)
      WRITE(6,924)
  924 FORMAT('OCONTROLLER CONTROL INPUT MATRIX -- GC ')
  160 DO 162 I=1,NC
      READ(5,914)(GC(I,J),J=1,NOP)
  162 WRITE(6,913)(GC(I,J),J=1,NOP)
      WRITE(6,925)
  925 FORMAT('OCONTROLLER OUTPUT MATRIX (STATES) -- HC')
  170 DO 172 I=1,NUP
      READ(5,914)(HC(I,J),J=1,NC)
  172 WRITE(6,913)(HC(I,J),J=1,NC)
      WRITE(6,926)
  926 FORMAT('OCONTROLLER OUTPUT MATRIX (INPUTS) -- EC')
  180 DO 182 I=1,NUP
      READ(5,914)(EC(I,J),J=1,NOP)
  182 WRITE(6,913)(EC(I,J),J=1,NOP)
      READ(5,928) T,NT
928   FORMAT(F10.4,I5)
      XNT = NT
      DELTA = T/(XNT-1)
      WRITE(6,930) T,NT
930   FORMAT('1T = ',F10.4/
     1 ' NT = ',I5/
     3 ' T = SAMPLE RATE.'/
     4 ' NT-1 = NB. OF EVENLY-SPACED SUBINTERVALS INTO WHICH T IS'/
     5 '         DIVIDED.'/
     6 ' DELTA = T/(NT-1) = INCREMENT USED IN THE NUMERICAL'/
     7 '                    INTEGRATIONS TO COMPUTE VZT, VZTAU, AND'/
     9 '                    VZTAU1 USING THE TRAPEZOIDAL RULE.'//)
      WRITE(6,931)
931   FORMAT(2X,' W IS THE DISTURBANCE VECTOR (WHITE GAUSSIAN NOISE'/
     1 '              WITH MEAN = 0 AND VARIANCE = 1)')
C
C     ******************************************************************
C     GENERATE WHITE GAUSSIAN NOISE WITH MEAN = 0 AND VARIANCE = 1
C     ******************************************************************
C
      IX = 11111
      DO 192 I = 1,1200
      A = 0.0
      DO 193 J = 1,12
      CALL RANDU(IX,IY,Y)
      IX = IY
193   A = A+Y
```

```
192     W(I) = A-6
        DO 1199 I = 1,NUP
        DO 1199 J = 1,NP
        ECCP(I,J) = 0.0
        DO 1199 K = 1,NOP
1199    ECCP(I,J) = ECCP(I,J) + EC(I,K)*CP(K,J)
802     FORMAT(5X,5G13.6)
835     FORMAT(5X,5G13.6)
C
C       ***************************************************************
C       CALCULATE PHIT(T), PSIT(TAU),
C                 PHIT(0), PHIT(DELTA), PHIT(2DELTA),...,PHIT(T-DELTA),
C                 PSIT(0), PSIT(DELTA), PSIT(2DELTA),...,PSIT(T-DELTA)
C       ***************************************************************
C
        TI = 0.0
        DELHLF = DELTA/2.0
        DO 402 II = 1,NP
        DO 402 JJ = 1,NP
        IF(II.EQ.JJ) GO TO 403
        PHIT1(II,JJ,1) = 0.0
        GO TO 402
403     PHIT1(II,JJ,1) = 1.0
402     CONTINUE
        DO 4 I1 = 2,NT
        TI = TI+DELTA
        PHIT1(1,1,I1) = 1.
        PHIT1(1,2,I1) = (1./10.)*(1.-EXP(-10.*TI))
        PHIT1(2,1,I1) = 0.0
4       PHIT1(2,2,I1) = EXP(-10.*TI)
        DO 400 II = 1,NP
        DO 400 JJ = 1,NP
400     PHIT(II,JJ) = PHIT1(II,JJ,NT)
        NTAU2 = 0
        N7 = 2000
        N8 = 300
        N9 = 50
        ICOUNT = 0
        ICUNT1 = 1
        N4 = 0
        N5 = 51
        NTAU = 4
        N3 = (N4+N5)/2
600     NTAU1 = N3
        ICOUNT = ICOUNT + 1
         CALL COVAR(AP,B1P,CP,FC,GC,HC,EC,GE,ECCP,W,PHIT,PHIT1,NTAU,NTAU1
       1 EA,PEASS,NPM,NUPM,NOPM,NCM,NHM,NWPM,N7,N8,ICOUNT,N9,DELHLF)
        WRITE(6,655) NTAU1,NTAU2
655     FORMAT(5X,I5,5X,I5)
        NERROR = IABS(NTAU1-NTAU2)
        IF(NERROR.EQ.1) GO TO 605
        IF(ICUNT1.NE.1) GO TO 601
        N3 = N3 + 1
        NTAU3 = N3
```

```
      NTAU1 = NTAU3
      ICUNT2 = ICOUNT
      ICUNT1 = 0
      GO TO 600
601   IF(PEASS(ICUNT2).LT.PEASS(ICOUNT)) GO TO 602
      N4 = N3 - 1
      N3 = ( N4 + N5 )/2
      NTAU2 = NTAU1
      ICUNT1 = 1
      GO TO 600
602   N5 = N3 - 1
      N3 = (N4+N5)/2
      NTAU2 = NTAU1
      ICUNT1 = 1
      GO TO 600
605   IF(NTAU1.EQ.1) GO TO 606
      IF(NTAU1.LT.NTAU2) GO TO 608
      ICUNT2 = ICOUNT - 1
      IF(PEASS(ICUNT2).LT.PEASS(ICOUNT)) GO TO 613
616     NTAU1 = NTAU1 + 1
      ICOUNT = ICOUNT + 1
        CALL COVAR(AP,B1P,CP,FC,GC,HC,EC,GE,ECCP,W,PHIT,PHIT1,NTAU,NTAU1
     1 EA,PEASS,NPM,NUPM,NOPM,NCM,NHM,NWPM,N7,N8,ICOUNT,N9,DELHLF)
      ICUNT2 = ICOUNT - 1
      IF(PEASS(ICUNT2).LT.PEASS(ICOUNT)) GO TO 615
      TAU1 = (NTAU1-2)*0.0125/50
      GO TO 612
 615    IF(NTAU1.LT.N5) GO TO 616
      TAU1 = (NTAU1-1)*0.0125/50
      GO TO 612
613   TAU1 = (NTAU1-2)*0.0125/50
      GO TO 612
606   ICUNT2 = ICOUNT - 1
      IF(PEASS(ICUNT2).LT.PEASS(ICOUNT)) GO TO 607
      TAU1 = (NTAU1-1)*0.0125/50
      GO TO 612
607   TAU1 = NTAU1*0.0125/50
      GO TO 612
608   ICUNT2 = ICOUNT - 1
      IF(PEASS(ICUNT2).LT.PEASS(ICOUNT)) GO TO 609
      NTAU1 = NTAU1 - 1
      ICOUNT = ICOUNT + 1
        CALL COVAR(AP,B1P,CP,FC,GC,HC,EC,GE,ECCP,W,PHIT,PHIT1,NTAU,NTAU1
     1 EA,PEASS,NPM,NUPM,NOPM,NCM,NHM,NWPM,N7,N8,ICOUNT,N9,DELHLF)
      ICUNT2 = ICOUNT - 1
      IF(PEASS(ICUNT2).LT.PEASS(ICOUNT)) GO TO 610
      TAU1 = NTAU1*0.0125/50
      GO TO 612
610   TAU1 = (NTAU1-1)*0.0125/50
      GO TO 612
609   TAU1 = (NTAU1-1)*0.0125/50
612   WRITE(6,650) TAU1
650   FORMAT(5X,' THE SKEW OF  THE SECOND CONTROLLER IS = ',F13.8)
657   FORMAT(5X,'THE NUMBER OF ITERATIONS IS ',I5)
```

```
      WRITE(6,657) ICOUNT
      STOP
      END
      SUBROUTINE COVAR(AP,B1P,CP,FC,GC,HC,EC,GE,ECCP,W,PHIT,PHIT1,
     1 NTAU,NTAU1,EA,PEASS,NPM,NUPM,NOPM,NCM,NHM,NWPM,N7,N8,ICOUNT,N9,
     2 DELHLF)
      DIMENSION ECCP(NPM,NUPM)
      DIMENSION AP(NPM,NPM),B1P(NPM,NUPM),CP(NOPM,NPM),FC(NCM,NCM),
     1 GC(NCM,NOPM),HC(NUPM,NCM),EC(NUPM,NOPM),GE(NPM,NWPM),W(N7),
     2 PHIT(NHM,NHM),PHIT1(NHM,NHM,101),EA(N8),PEASS(N9),
     3 PHTAU(4,4),PHTAU1(4,4),PSIT1(484),PSTAU(4,4),YC1(2),YP2(2),
     5 YC2(2),E1(2),E2(2),XP(2),XPTAU(2),XP1(2),YP(2),YPTAU(2),
     6 YP1(2),XC1(2),XC2(2),AM(4,4),PS(4,4),P1(4,4),D1(4,4),
     7 D2(4,4),D3(4),XW1(2),XW2(2),D(4,4),YPTAU1(2),YC3(2),XC3(2),
     9 XW3(2),XPTAU1(2),PSTAU1(4,4),PSIT2(4,4),PSIT(4,4),UP(2)
      NP = 2
      NOP = 1
      NC = 1
      NWP = 1
      NUP = 1
      NT = 50
      DO 31 I = 1,NUP
      YC2(I) = 0.0
      YC3(I) = 0.0
31    YC1(I) = 0.0
      DO 32 I = 1,NUP
32    E1(I) = YC3(I) - YC2(I)
C     FROM INITIAL VALUE XP(TIME), AND YP(TIME) ARE EQUAL TO ZERO
      DO 35 I = 1,NP
      XP1(I) = 0.5
35    XP(I) = 0.0
      DO 435 I = 1,NOP
      Q = 0.0
      DO 436 J = 1,NP
436   Q = Q + CP(I,J)*XP1(J)
435   YP1(I) = Q
      DO 36 I = 1,NOP
36    YP(I) = 0.0
C     FROM INITIAL VALUE XC1(TIME+T), YC1(TIME) ARE EQUAL TO ZERO
      DO 37 I = 1,NC
      XC1(I) = 0.0
      XC3(I) = 0.0
37    XC2(I) = 0.0
      DO 52 I = 1,NUP
52    YC1(I) = 0.0
      IDEL = 0
      DO 5000 I = 1,NT
      IDEL = IDEL + 1
      IF(NTAU.EQ.NTAU1.AND.IDEL.EQ.NTAU) GO TO 7
      IF(IDEL.EQ.NTAU) GO TO 16
      IF(IDEL.EQ.NTAU1) GO TO 22
      GO TO 5000
7     DO 8 II = 1,NP
      DO 8 JJ = 1,NP
```

```
      PHTAU(II,JJ) = PHIT1(II,JJ,IDEL)
8     PHTAU1(II,JJ) = PHIT1(II,JJ,IDEL)
      GO TO 5000
16    DO 17 II = 1,NP
      DO 17 JJ = 1,NP
17    PHTAU(II,JJ) = PHIT1(II,JJ,IDEL)
      GO TO 5000
22    DO 23 II = 1,NP
      DO 23 JJ = 1,NP
23    PHTAU1(II,JJ) = PHIT1(II,JJ,IDEL)
5000  CONTINUE
      DO 550 I = 1,NP
      DO 550 J = 1,NWP
      P1(I,J) = 0.0
550   PS(I,J) = 0.0
      DO 551 I = 1,NP
      DO 551 J = 1,NWP
      PSIT2(I,J) = 0.0
      PSTAU(I,J) = 0.0
      PSTAU1(I,J) = 0.0
      PSIT(I,J) = 0.0
      DO 551 K = 1,NP
      PSIT2(I,J) = PSIT2(I,J) + PHIT1(I,K,NT)*GE(K,J)
      PSTAU(I,J) = PSTAU(I,J) + PHIT1(I,K,NTAU)*B1P(K,J)
      PSTAU1(I,J) = PSTAU1(I,J) + PHIT1(I,K,NTAU1)*B1P(K,J)
551   PSIT(I,J) = PSIT(I,J) + PHIT1(I,K,NT)*B1P(K,J)
      DO 552 I = 1,NP
      DO 552 J = 1,NWP
      PSIT2(I,J) = DELHLF*PSIT2(I,J)
      PSTAU(I,J) = DELHLF*PSTAU(I,J)
      PSTAU1(I,J) = DELHLF*PSTAU1(I,J)
552   PSIT(I,J) = DELHLF*PSIT(I,J)
60    DO 61 I1 = 2,NT
      I2 = NT-I1+1
      DO 62 I = 1,NP
      DO 62 J = 1,NWP
      PSIT2(I,J) = PSIT2(I,J) + P1(I,J)
62    PSIT(I,J) = PSIT(I,J) + PS(I,J)
      DO 63 I = 1,NP
      DO 63 J = 1,NWP
      P1(I,J) = 0.0
      PS(I,J) = 0.0
      DO 63 K = 1,NP
      P1(I,J) = P1(I,J) + PHIT1(I,K,I2)*GE(K,J)
63    PS(I,J) = PS(I,J) + PHIT1(I,K,I2)*B1P(K,J)
      DO 64 I = 1,NP
      DO 64 J = 1,NWP
      P1(I,J) = DELHLF*P1(I,J)
64    PS(I,J) = DELHLF*PS(I,J)
      DO 66 I = 1,NP
      DO 66 J = 1,NWP
      PSIT2(I,J) = P1(I,J) + PSIT2(I,J)
66    PSIT(I,J) = PS(I,J) + PSIT(I,J)
61    CONTINUE
```

```
      DO 80 I = 1,NP
      DO 80 J = 1,NWP
80    PS(I,J) = 0.0
      IF(NTAU1.EQ.1) GO TO 85
      GO TO 90
85    DO 86 II = 1,NP
      DO 86 JJ = 1,NWP
86    PSTAU1(II,JJ) = 0.0
      GO TO 67
90    DO 91 I1 = 2,NTAU1
      I2 = NTAU1-I1+1
      DO 92 I = 1,NP
      DO 92 J = 1,NWP
92    PSTAU1(I,J) = PSTAU1(I,J) + PS(I,J)
      DO 93 I = 1,NP
      DO 93 J = 1,NWP
      PS(I,J) = 0.0
      DO 93 K = 1,NP
93    PS(I,J) = PS(I,J) + PHIT1(I,K,I2)*B1P(K,J)
      DO 94 I = 1,NP
      DO 94 J = 1,NWP
94    PS(I,J)=DELHLF*PS(I,J)
      DO 96 I = 1,NP
      DO 96 J = 1,NWP
96    PSTAU1(I,J) = PS(I,J) + PSTAU1(I,J)
91    CONTINUE
67    DO 68 I = 1,NP
      DO 68 J = 1,NWP
68    PS(I,J) = 0.0
      IF(NTAU.EQ.1) GO TO 55
      GO TO 69
55    DO 58 II = 1,NP
      DO 58 JJ = 1,NWP
58    PSTAU(II,JJ) = 0.0
      GO TO 77
69    DO 70 I1 = 2,NTAU
      I2 = NTAU-I1+1
      DO 71 I = 1,NP
      DO 71 J = 1,NWP
71    PSTAU(I,J) = PSTAU(I,J) + PS(I,J)
      DO 72 I = 1,NP
      DO 72 J = 1,NWP
      PS(I,J) = 0.0
      DO 72 K = 1,NP
72    PS(I,J) = PS(I,J) + PHIT1(I,K,I2)*B1P(K,J)
      DO 73 I = 1,NP
      DO 73 J = 1,NWP
73    PS(I,J) = DELHLF*PS(I,J)
      DO 76 I = 1,NP
      DO 76 J = 1,NWP
76    PSTAU(I,J) = PS(I,J) + PSTAU(I,J)
70    CONTINUE
C
C     ****************************************************************
```

```
C       START TIME LOOP
        TIME = 0
C       ****************************************************************
C
77      NN1 = 1
        DO 412 I = 1,NOP
412     UP(I) = W(NN1)-YC1(I)
        DO 413 I = 1,NP
        Q = 0.0
        DO 414 J = 1,NP
414     Q = Q + PHTAU1(I,J)*XP1(J)
413     XW1(I) = Q
        DO 415 I = 1,NP
        Q = 0.0
        DO 416 J = 1,NUP
416     Q = Q + PSTAU1(I,J)*UP(J)
415     XW2(I) = Q
        DO 417 I = 1,NP
417     XPTAU1(I) = XW1(I) + XW2(I)
        DO 418 I = 1,NOP
        Q = 0.0
        DO 419 J = 1,NP
419     Q = Q + CP(I,J)*XPTAU1(J)
418     YPTAU1(I) = Q
        DO 420 I = 1,NUP
        Q = 0.0
        DO 421 J = 1,NC
421     Q = Q + HC(I,J)*XC3(J)
420     XW1(I) = Q
        DO 422 I = 1,NUP
        Q = 0.0
        DO 423 J = 1,NOP
423     Q = Q + EC(I,J)*YPTAU1(J)
422     XW2(I) = Q
        DO 424 I = 1,NUP
424     YC3(I) = XW1(I) + XW2(I)
        DO 425 I = 1,NC
        Q = 0.0
        DO 426 J = 1,NC
426     Q = Q + FC(I,J)*XC3(J)
425     XW1(I) = Q
        DO 427 I = 1,NC
        Q = 0.0
        DO 428 J = 1,NOP
428     Q = Q + GC(I,J)*YPTAU1(J)
427     XW2(I) = Q
        DO 429 I = 1,NC
429     XC3(I) = XW1(I) + XW2(I)
        DO 430 I = 1,NUP
430     E2(I) = YC3(I) - YC2(I)
        DO 220 II1 = 1,60
        DO 253 I = 1,NP
        Q = 0.0
        DO 254 J = 1,NP
```

```
254   Q = Q + PHTAU(I,J)*XP(J)
253   XW1(I) = Q
      DO 255 I = 1,NP
      Q = 0.0
      DO 256 J = 1,NUP
256   Q = Q + PSTAU(I,J)*UP(J)
255   XW2(I) = Q
      DO 257 I = 1,NP
257   XPTAU(I) = XW1(I) + XW2(I)
      DO 260 I = 1,NOP
      Q = 0.0
      DO 261 J = 1,NP
261   Q = Q + CP(I,J)*XPTAU(J)
260   YPTAU(I) = Q
      DO 300 I = 1,NUP
      Q = 0.0
      DO 301 J = 1,NC
301   Q = Q + HC(I,J)*XC2(J)
300   XW1(I) = Q
      DO 302 I = 1,NUP
      Q = 0.0
      DO 303 J = 1,NOP
303   Q = Q + EC(I,J)*YPTAU(J)
302   XW2(I) = Q
      DO 304 I = 1,NUP
304   YC2(I) = XW1(I) + XW2(I)
      DO 305 I = 1,NC
      Q = 0.0
      DO 306 J = 1,NC
306   Q = Q + FC(I,J)*XC2(J)
305   XW1(I) = Q
      DO 307 I = 1,NC
      Q = 0.0
      DO 308 J = 1,NOP
308   Q = Q + GC(I,J)*YPTAU(J)
307   XW2(I) = Q
      DO 309 I = 1,NC
309   XC2(I) = XW1(I) + XW2(I)
      DO 290 I = 1,NUP
290   E1(I) = YC3(I) - YC2(I)
      DO 432 I = 1,NUP
432   YP2(1) = YP(I) - YP1(I)
      DO 405 I = 1,NP
      Q = 0.0
      DO 406 J = 1,NP
406   Q = Q + PHIT(I,J)*XP1(J)
405   XW1(I) = Q
      DO 407 I = 1,NP
      Q = 0.0
      DO 408 J = 1,NUP
408   Q = Q + PSIT(I,J)*UP(J)
407   XW2(I) = Q
      DO 409 I = 1,NP
      Q = 0.0
```

```
      DO 410 J = 1,NUP
410   Q = Q + PSIT2(I,J)*YP2(J)
409   XW3(I) = Q
      DO 411 I = 1,NP
411   XP1(I) = XW1(I) + XW2(I) + XW3(I)
      DO 433 I = 1,NOP
      Q = 0.0
      DO 434 J = 1,NP
434   Q = Q + CP(I,J)*XP1(J)
433   YP1(I) = Q
373   DO 500 I = 1,NP
      Q = 0.0
      DO 501 J = 1,NP
501   Q = Q + PHIT(I,J)*XP(J)
500   XW1(I) = Q
      DO 502 I = 1,NP
      Q = 0.0
      DO 503 J = 1,NUP
503   Q = Q + PSIT(I,J)*UP(J)
502   XW2(I) = Q
      DO 504 I = 1,NP
504   XP(I) = XW1(I) + XW2(I)
      DO 507 I = 1,NOP
      Q = 0.0
      DO 508 J = 1,NP
508   Q = Q + CP(I,J)*XP(J)
507   YP(I) = Q
      DO 700 I = 1,NUP
      Q = 0.0
      DO 701 J = 1,NC
701   Q = Q + HC(I,J)*XC1(J)
700   XW1(I) = Q
      DO 702 I = 1,NUP
      Q = 0.0
      DO 703 J = 1,NOP
703   Q = Q + EC(I,J)*YP(J)
702   XW2(I) = Q
      DO 704 I = 1,NUP
704   YC1(I) = XW1(I) + XW2(I)
      DO 705 I = 1,NC
      Q = 0.0
      DO 706 J = 1,NC
706   Q = Q + FC(I,J)*XC1(J)
705   XW1(I) = Q
      DO 707 I = 1,NC
      Q = 0.0
      DO 708 J = 1,NOP
708   Q = Q + GC(I,J)*YP(J)
707   XW2(I) = Q
      DO 709 I = 1,NC
709   XC1(I) = XW1(I) + XW2(I)
      NN1 = NN1 + 1
      DO 221 I = 1,NOP
221   UP(I) = W(NN1)-YC1(I)
```

```
      Q = 0.0
      DO 521 J = 1,NP
521   Q = Q + PHTAU1(I,J)*XP1(J)
520   XW1(I) = Q
      DO 522 I = 1,NP
      Q = 0.0
      DO 523 J = 1,NUP
523   Q = Q + PSTAU1(I,J)*UP(J)
522   XW2(I) = Q
      DO 524 I = 1,NP
524   XPTAU1(I) = XW1(I) + XW2(I)
      DO 525 I = 1,NOP
      Q = 0.0
      DO 526 J = 1,NP
526   Q = Q + CP(I,J)*XPTAU1(J)
525   YPTAU1(I) = Q
      DO 529 I = 1,NUP
      Q = 0.0
      DO 530 J = 1,NC
530   Q = Q + HC(I,J)*XC3(J)
529   XW1(I) = Q
      DO 531 I = 1,NUP
      Q = 0.0
      DO 532 J = 1,NOP
532   Q = Q + EC(I,J)*YPTAU1(J)
531   XW2(I) = Q
      DO 533 I = 1,NUP
533   YC3(I) = XW1(I) + XW2(I)
      DO 535 I = 1,NC
      Q = 0.0
      DO 536 J = 1,NC
536   Q = Q + FC(I,J)*XC3(J)
535   XW1(I) = Q
      DO 537 I = 1,NC
      Q = 0.0
      DO 538 J = 1,NOP
538   Q = Q + GC(I,J)*YPTAU1(J)
537   XW2(I) = Q
      DO 539 I = 1,NC
539   XC3(I) = XW1(I) + XW2(I)
      DO 540 I = 1,NUP
540   E2(I) = YC3(I) - YC2(I)
      DO 560 I = 1,NUP
560   EA(II1) = E1(I)
220   CONTINUE
      SMEANA = 0.0
      DO 561 I = 51,60
561   SMEANA = SMEANA + EA(I)
      SMEANA = SMEANA/10
      VEA = 0.0
      DO 563 I = 51,60
563   VEA = VEA + (EA(I)-SMEANA)**2
      PEASS(ICOUNT) = VEA/10
      RETURN
```

```
      END
      SUBROUTINE RANDU(IX,IY,YFL)
      IY = IX*65539
      IF(IY)5,6,6
5     IY = IY + 2147483647+1
6     YFL = IY
      YFL = YFL*0.4656613E-9
      RETURN
      END
```

```
EXAMPLE-------2TH ORDER PLANT, 1ST ORDER CONTROLLER
NO. OF PLANT STATES = 2
NO. OF PLANT INPUTS = 1
NO. OF EXTERNAL INPUT = 1
NO. OF PLANT OUTPUTS = 1
NO. OF CONTROLLER STATES ( EACH CONTROLLER) = 1

PLANT STATE MATRIX -- AP
  .0            1.0
  .0          -10.0

PLANT CONTROL INPUT MATRIX -- BP
  .0
  200.0

PLANT OUTPUT MATRIX -- CP
  1.0           .0

CONTROLLER STATE MATRIX -- FC
  .523810

OBSERVER MATRIX -- GE
  20.0
  25.0
CONTROLLER CONTROL INPUT MATRIX -- GC
 -.18162

CONTROLLER OUTPUT MATRIX (STATES) -- HC
  1.0

CONTROLLER OUTPUT MATRIX (INPUTS) -- EC
  1.381
NT = 51
T = SAMPLE PERIOD = 0.0125 SEC
DELTA = T/(NT-1) = INCREMENT USED IN THE NUMERICAL
                   INTEGRATIONS TO COMPUTE PSITAU,PSIT,PSIT2,
                   PSITAU1 USING TRAPEZOIDAL RULE.
W IS THE EXTERNAL INPUT (WHITE GAUSSIAN NOISE WITH
  MEAN = 0.0, AND VARIANCE = 1.0
THE SKEW OF THE SECOND CONTROLLER IS 0.00725
THE NUMBER OF ITERATIONS IS 5
```

REFERENCES

1. Vincent J. Darcy and Charles R. Slivinsky, "Analysis of Inherent Errors in Asynchronous Redundant Digital Flight Control Systems." Technical Report AFFDL-TR-76-16, Air Force Dynamics Laboratory, Wright-Patterson Air Force Base, April 1976.

2. Charles R. Slivinsky, "Inherent Errors in Asynchronous Digital Flight Controls." Technical Report AFOSR-76-2968A, Directorate of Mathematical and Information Sciences Air Force Office of Scientific Research (AFSC), Bolling Air Force Base, March 1978.

3. Meditch, J. S., Stochastic Optimal Linear Estimation and Control, McGraw-Hill Book Company, 1969.

4. Benjamin, C. Kuo, Automatic Control System, Prentice-Hall, Inc. 1975.