UTILIZATION OF A KALMAN OBSERVER WITH LARGE SPACE STRUCTURES

by

Bruce M. Jackson

December 1988

Thesis Advisor

Jeff B. Burl

Approved for public release; distribution is unlimited.
Control of the motions and vibrations of large space structures require the knowledge of state values that may not be available due either to inability to measure the states or, the high cost of the sensors to measure the required states. One solution is the use of an observer to estimate the states from limited sensor input.

The physical characteristics of large space structures and the environment they operate in will cause large amounts of noise in the measurements. The obvious observer for such an environment is the Kalman Filter which is specifically designed to produce optimal estimates in a noisy environment.

A straightforward application of the Kalman Filter will be examined utilizing a steady state Kalman gain matrix. The observer performance will be examined in both matched filter-plant and reduced order filter configurations.
Utilization of a Kalman Observer with Large Space Structures

by

Bruce M. Jackson
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 1976

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ENGINEERING SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
December 1988

Author: Bruce M. Jackson

Approved by: Jeff B. Burl, Thesis Advisor
               H. A. Titus, Second Reader
               John P. Powers, Chairman,
               Department of Electrical and Computer Engineering

Gordon E. Schacher,
Dean of Science and Engineering
ABSTRACT

Control of the motions and vibrations of large space structures require the knowledge of state values that may not be available due either to inability to measure the states or, the high cost of the sensors to measure the required states. One solution is the use of an observer to estimate the states from limited sensor input.

The physical characteristics of large space structures and the environment they operate in will cause large amounts of noise in the measurements. The obvious observer for such an environment is the Kalman Filter which is specifically designed to produce optimal estimates in a noisy environment.

A straightforward application of the Kalman Filter will be examined utilizing a steady state Kalman gain matrix. The observer performance will be examined in both matched filter/plant and reduced order filter configurations.
THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.
TABLE OF CONTENTS

I. INTRODUCTION .................................................. 1
   A. BACKGROUND ............................................. 1
   B. PROBLEM STATEMENT ..................................... 1
   C. ORGANIZATION ........................................... 2

II. MATHEMATICAL MODEL ............................................. 3
    A. INTRODUCTION .......................................... 3
    B. MODAL MODEL .......................................... 3
    C. DISCRETE-TIME MODEL ................................... 6

III. THE OBSERVER .................................................. 10
    A. INTRODUCTION .......................................... 10
    B. KALMAN FILTER EQUATIONS ............................... 10
    C. STEADY-STATE SOLUTION ................................ 12
    D. OBSERVER PERFORMANCE ................................ 12

IV. SIMULATION ...................................................... 14
    A. INTRODUCTION .......................................... 14
    B. PLANT AND OBSERVER DATA ............................... 14
    C. SIMULATION PROGRAMS .................................. 14
    D. EFFECT OF PLANT TO MEASUREMENT NOISE RATIO ON OB-
       SERVER PERFORMANCE ................................... 15
    E. EFFECTS OF INCREASED MODES ON OBSERVER PERFORMANCE 16
    F. REDUCED ORDER KALMAN OBSERVER ......................... 16

V. CONCLUSIONS AND RECOMMENDATIONS ............................. 35
    A. CONCLUSIONS ........................................... 35
    B. RECOMMENDATIONS ...................................... 35

APPENDIX A. KALMAN GAIN MATRIX GENERATION PROGRAM ........ 36
APPENDIX B. KALMAN OBSERVER AND PLANT SIMULATION ........ 49

APPENDIX C. PROGRAM TO ESTIMATE NOISE IN KALMAN FILTER FROM UNOBSERVED MODES ........................................ 62

LIST OF REFERENCES ........................................... 73

INITIAL DISTRIBUTION LIST .................................... 74
LIST OF TABLES

Table 1. CUMULATIVE UNANTICIPATED NOISE FROM UNOBSERVED MODES ............................................. 18
Table 2. UNANTICIPATED NOISE FROM UNOBSERVED MODES BY MODE ............................................. 18
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Observer Performance (J) $PN/MN = 1.0d12$</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>Observer Performance (J) $PN/MN = 1.0d11$</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>Observer Performance (J) $PN/MN = 1.0d10$</td>
<td>21</td>
</tr>
<tr>
<td>4</td>
<td>Observer Performance (J) $PN/MN = 5.0d09$</td>
<td>22</td>
</tr>
<tr>
<td>5</td>
<td>Observer Performance (J) $PN/MN = 2.5d09$</td>
<td>23</td>
</tr>
<tr>
<td>6</td>
<td>Observer Performance (J) $PN/MN = 1.0d09$</td>
<td>24</td>
</tr>
<tr>
<td>7</td>
<td>Mode 7 (Position) Observer Performance versus $PN/MN$</td>
<td>25</td>
</tr>
<tr>
<td>8</td>
<td>Settling Time versus $PN/MN$</td>
<td>26</td>
</tr>
<tr>
<td>9</td>
<td>Observer Performance (J) 4 Modes (7 - 10)</td>
<td>27</td>
</tr>
<tr>
<td>10</td>
<td>Observer Performance (J) 5 Modes (7 - 11)</td>
<td>28</td>
</tr>
<tr>
<td>11</td>
<td>Observer Performance (J) 6 Modes (7 - 12)</td>
<td>29</td>
</tr>
<tr>
<td>12</td>
<td>Observer Performance (J) 7 Modes (7 - 13)</td>
<td>30</td>
</tr>
<tr>
<td>13</td>
<td>Observer Performance (J) 8 Modes (7 - 14)</td>
<td>31</td>
</tr>
<tr>
<td>14</td>
<td>Observer Performance (J) 9 Modes (7 - 15)</td>
<td>32</td>
</tr>
<tr>
<td>15</td>
<td>Observer Performance (J) 10 Modes (7 - 16)</td>
<td>33</td>
</tr>
<tr>
<td>16</td>
<td>Settling Time versus number of Modes Observed</td>
<td>34</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENTS

I would like to express my appreciation to the McDonnel Douglas Astronautics Company of Huntington Beach, California, for allowing the use of their dynamic model of a preliminary space station configuration. Also, I would like to thank Major William J. Preston, USMC for allowing the use of the mathematical model for the station dynamics which makes up the majority of Chapter II of this paper as well as portions of his simulation program. Finally, I would like to thank Professor J. B. Burl for his assistance and most of all his patience in the completion of this work.

This paper is dedicated to Christina Marie Jackson (USNA '10).
I. INTRODUCTION

A. BACKGROUND

The advent of large space structures poses a number of problems for the control engineer. Previously, the objects put into space could be treated as rigid bodies so that a single three axis sensor package could be used to tell the motion of all components. The large space structures will not be rigid, instead they will have considerable flexibility and multiple modes of vibration [Ref. 1: p. 51]

Control of the structure's attitude and vibrations requires knowing the motions of the components. One approach would be to heavily instrument the space structure, but weight and cost make this approach impractical. An alternative is to use a limited number of sensors to measure only certain states and to deduce the other required states by use of an observer algorithm.

This thesis will address the production of estimates of the states needed for control of the structure. The model used will be a early design study by McDonnell Douglas Astronautics for a dual keel space station. The techniques and problems of observation for this model are generic to all large space structures.

B. PROBLEM STATEMENT

Design of an observer for estimating the states of a large space structure breaks down into several steps. First, a mathematical model is developed for the system behavior over time. Modal analysis is used to form a system composed of decoupled second order differential equations. The use of decoupled equations allows a reduced order model to be generated by truncating the number of modal equations. A reduced order model will have all of the same mathematical qualities (and problems) but reduces the amount of time and computer resources required to do simulation.

Second, the observer is designed. The observer is designed to obtain a minimum variance estimate of the desired state values from the measurements.

Third, the observer is simulated to verify performance. Simulation runs of both a matched observer/plant system and a reduced order observer are employed. That is, the system is run where the observer is used to estimate all of the plant states and run where there are more plant states than the observer estimates.
Fourth, results are analysed and conclusions drawn based on these results. Recommendations for further areas of research are suggested based on the results and conclusions.

C. ORGANIZATION

The model of the space station is developed in Chapter II. The modal model was developed using modal analysis and discretized to form the discrete-time state equations. The data for this model was from an early design study by McDonnell Douglas Astronautics Company for a dual-keel space station. The observer and its equations are developed in Chapter III. Chapter IV is the simulation runs of the observer versus the plant. Chapter V presents conclusions and recommendations for further research.
II. MATHEMATICAL MODEL

A. INTRODUCTION

Prior to the proposed space station almost all of the objects put into space could be treated as simple rigid bodies for the purpose of mathematical modelling of their motions. The design constraints imposed by the high cost of lifting mass to orbit dictates a light, open structure with considerable flexing. Large space structures such as the space station, therefore, cannot be treated as rigid bodies. The structure is in fact lightly damped with multiple natural frequencies. The result is a structure that will vibrate for considerable periods of time whenever external forces are applied.

The space station structure can be modeled as an n-DOF (degree of freedom) system consisting of n masses, springs, and dashpots [Ref. 2: p. 173-176]. This straightforward modelling of the coupled masses produces a system of unworkable complexity. As a result, the system will be modelled in terms of the structure's natural modes of vibration. The resulting system, while still complex, is at least workable.

The model will be developed in two steps. The first will be to generate the continuous-time model of the natural modes. The second will yield the discrete-time model, developed from the first model, for use in the simulation.

B. MODAL MODEL

The space station structure can be modeled as a system of discrete masses coupled by springs and dashpots. The major mechanism of damping in the structure is structural damping, the internal dissipation of energy within the members, as the structure vibrates. Structural damping can be shown to be equivalent to viscous damping and this equivalency is used in the model [Ref. 2: p. 72-73].

The energy dissipated by structural damping is:

\[ W_d = \alpha X^2 \]  

\( W_d \) = energy dissipated by structural damping  
\( \alpha \) = constant (force/displacement)  
\( X \) = displacement

The energy dissipated by viscous damping is:

\[ W_v = nc\omega X^2 \]  

\( W_v \) = energy dissipated by viscous damping  
\( n \) =  
\( \omega \) = natural frequency  
\( X \) = displacement
We can equate the two
\[ \pi C_{eq} \omega X^2 = aX^2 \] (3)
yielding an equivalent viscous damping coefficient:
\[ C_{eq} = \frac{a}{\pi \omega} \] (4)
The second order differential equation for a single viscously damped mass is:
\[ m\ddot{x} + c\dot{x} + kx = F(t) \] (5)
Substituting \( C_{eq} \) for \( c \)
\[ m\ddot{x} + \frac{a}{\pi \omega} \dot{x} + kx = F(t) \] (6)
For multiple mass systems \( C_{eq} \) becomes \( \frac{d}{\omega_f} K \) where \( \omega_f \) is the natural frequency of vibration.

The displacement of masses can be represented by the second order matrix differential equation [Ref. 3: p. 3-9],
\[ M\ddot{q}(t) + \frac{d}{\omega_f} K\dot{q}(t) + Kq(t) = F(t) \] (7)
\[ q \] = coordinate vector
\[ M \] = system mass matrix (diagonal)
\[ \frac{d}{\omega_f} K \] = equivalent damping
\[ d \] = damping coefficient
\[ \omega_f \] = frequency of oscillation of the system
\[ K \] = symmetric system stiffness matrix
\[ F(t) \] = system forcing function

The above equation represents a system of second order differential equations coupled through the stiffness matrix. Decoupling can be done by expressing \( q \) in terms of natural modes of vibration. The process is called modal analysis. The independent differential equations can then be treated individually. The modal equations are derived below.

First, the undamped, homogeneous form of Eq. (7)
\[ M\ddot{q}(t) + Kq(t) = 0 \]  
(8)

is solved. Let

\[ q(t) = Ax \sin(\omega t + \Theta) \]  
(9)

\[ \dot{q}(t) = A\omega \cos(\omega t + \Theta) \]  
(10)

\[ \ddot{q}(t) = -A\omega^2 \sin(\omega t + \Theta) \]  
(11)

substituting Eq. (9) and Eq. (10) into Eq. (11)

\[ [-\omega^2 M + K]Ax \sin(\omega t + \Theta) = 0 \]  
(12)

This equation has a non-trivial solution for all time if and only if:

\[ [K - \omega^2 M]x = 0 \]  
(13)

Equation (12) has \( n \) combinations of \( x \) (natural mode shapes) and \( \omega \) (natural frequencies) as solutions. These can be grouped into matrices:

\[ X = \begin{bmatrix} x_1 & x_2 & \ldots & x_n \end{bmatrix}^T \]  
(14)

\[ \Omega^2 = \text{diag}[\omega_{o1}^2, \omega_{o2}^2, \ldots, \omega_{on}^2] \]  
(15)

which satisfy the equation:

\[ KX = \Omega^2 MX \]  
(16)

Several useful relations can be derived from Eq. (16). Premultiplying Eq. (16) by \( X^T \),

\[ X^TKX = \Omega^2X^TMX \]  
(17)

The eigenvectors can be normalized

\[ X^TMX = I \]  
(18)

which yields

\[ X^TKX = \Omega^2 \]  
(19)
The equations of motion can be uncoupled through the linear transformation of the coordinate system

\[ q(t) = \sum_{i=1}^{n} x_i \eta_i(t) = X \eta(t) \]  

\[ X = \text{modal matrix} \]
\[ n = \text{maximum number of degrees of freedom} \]
\[ \eta(t) = \text{transformed coordinate vector} \]

Application of the transformation to the system Eq. (7) yields

\[ X^T MX \ddot{\eta}(t) + \frac{d}{d\omega} X^T K X \dot{\eta}(t) + X^T K X \eta(t) = X^T F(t) \]  

Using Eq.(18) and Eq. (19)

\[ X^T MX \dot{\eta}(t) = I \dot{\eta}(t) = \dot{\eta} \]
\[ \frac{d}{d\omega} X^T K X \dot{\eta}(t) = \frac{d}{d\omega} \Omega^2 \dot{\eta}(t) = d\Omega \dot{\eta} \]
\[ X^T K X \eta(t) = \Omega^2 \eta \]

therefore

\[ \ddot{\eta} + d\Omega \dot{\eta} + \Omega^2 \eta = X^T F \]  

Equation (25) is the modal model of uncoupled second order differential equations. The motion of the structure can be found from the modal amplitudes, \( \eta(t) \), using Eq. (20).

C. DISCRETE-TIME MODEL

The discrete-time state space model is found by solving the continuous-time equations. The \( i \)th equation of motion is

\[ \ddot{\eta}_i(t) + \omega_i^2 \dot{\eta}_i(t) + \omega_i^2 \eta_i(t) = X_i^T F(t) \]

\[ X_i^T = \text{transpose of the ith mode shape vector} \]
\[ F(t) = \text{torquing force applied at a point} \]

The homogeneous solution (\( F(t) = 0 \)) for Eq. (26) is [Ref. 4: p. 475-476]
\[ \eta(t) = C_1 e^{-\gamma t} \cos(\omega_d t) + C_2 e^{-\gamma t} \sin(\omega_d t) \]  

(27)

where

\[ \gamma = \frac{d\omega_{ol}}{2} \]  

(28)

\[ \omega_d = \sqrt{\omega_{ol}^2 - \gamma^2} \]  

(29)

The constants in Eq. (27) can be found by taking the derivative

\[ \dot{\eta}(t) = (C_2 \omega_d - C_1 \gamma) e^{-\gamma t} \cos(\omega_d t) - (C_1 \omega_d - C_2 \gamma) e^{-\gamma t} \sin(\omega_d t) \]  

(30)

and evaluating at \( t = 0 \)

\[ \eta(0) = C_1 \]  

(31)

\[ \dot{\eta}(0) = C_2 \omega_d - C_1 \gamma \]  

(32)

Solving for \( C_1 \) and \( C_2 \)

\[ C_1 = \eta(0) \]  

(33)

\[ C_2 = \frac{\dot{\eta}(0)}{\omega_d} + \frac{\eta(0) \gamma}{\omega_d} \]  

(34)

In matrix form

\[
\begin{bmatrix}
C_1 \\
C_2
\end{bmatrix} = \begin{bmatrix}
1 & 0 \\
\frac{\gamma}{\omega_d} & \frac{1}{\omega_d}
\end{bmatrix} \begin{bmatrix}
\eta(0) \\
\dot{\eta}(0)
\end{bmatrix}
\]  

(35)

Rewriting Eq. (27) and Eq. (30) in matrix form

\[
\begin{bmatrix}
\eta(t) \\
\dot{\eta}(t)
\end{bmatrix} = \begin{bmatrix}
e^{-\gamma t} \cos(\omega_d t) & e^{-\gamma t} \sin(\omega_d t) \\
e^{-\gamma t}[\gamma \cos(\omega_d t) + \omega_d \sin(\omega_d t)] & e^{-\gamma t}[\omega_d \cos(\omega_d t) - \gamma \sin(\omega_d t)]
\end{bmatrix} \begin{bmatrix}
C_1 \\
C_2
\end{bmatrix}
\]  

(36)

Substituting Eq. (35) into Eq. (36), the solution can be written in terms of the initial conditions

7
\[
\begin{bmatrix}
\eta(t) \\
\dot{\eta}(t)
\end{bmatrix} =
\begin{bmatrix}
 e^{-\gamma t}\left[ \cos(\omega_d t) + \frac{\gamma}{\omega_d} \sin(\omega_d t) \right] & \frac{1}{\omega_d} e^{-\gamma t} \sin(\omega_d t) \\
-\frac{\omega_o}{\omega_d} e^{-\gamma t} \sin(\omega_d t) & e^{-\gamma t}\left[ \cos(\omega_d t) - \frac{\gamma}{\omega_d} \sin(\omega_d t) \right]
\end{bmatrix}
\begin{bmatrix}
\eta(0) \\
\dot{\eta}(0)
\end{bmatrix}
\] (37)

Letting

\[
X_i(t) = \begin{bmatrix}
\eta_i(t) \\
\dot{\eta}_i(t)
\end{bmatrix}
\] (38)

and

\[
A_i = \begin{bmatrix}
 e^{-\gamma t}\left[ \cos(\omega_d t) + \frac{\gamma}{\omega_d} \sin(\omega_d t) \right] & \frac{1}{\omega_d} e^{-\gamma t} \sin(\omega_d t) \\
-\frac{\omega_o}{\omega_d} e^{-\gamma t} \sin(\omega_d t) & e^{-\gamma t}\left[ \cos(\omega_d t) - \frac{\gamma}{\omega_d} \sin(\omega_d t) \right]
\end{bmatrix}
\] (39)

the solution can be written as

\[
X_i(t) = A_i(t)X_i(0)
\] (40)

where \(A_i\) is the state transition matrix of the \(i\)th mode. The non-homogeneous solution is

\[
X_i(t) = A_i(t)X_i(0) + B_iX_i^T F(0)
\] (41)

where the discrete-time input matrix, for constant \(F\), is given by

\[
B_i = \int_0^T B_i(t) \Gamma \, dt
\] (42)

and \(\Gamma = [0 \ 1]^T\) is the input matrix for the continuous-time system, and \(T\) is the sampling time. Solving Eq. (42) yields

\[
B_i = \begin{bmatrix}
\frac{-1}{\omega_o} \left[ 1 - e^{-\gamma T} \cos(\omega_d t) - \frac{\gamma}{\omega_d} e^{-\gamma T} \sin(\omega_d t) \right] \\
\frac{1}{\omega_d} e^{-\gamma T} \sin(\omega_d t)
\end{bmatrix}
\] (43)

The discrete-time state equation for the \(i\)th equation of motion can be written as

\[
X_i(kT + 1) = A_i(T)X_i(kT) + B_i(T)x_i^T F(kT)
\] (44)

where \(A_i\) and \(B_i\) are evaluated at \(t = T\). Here,

\[
X_i = \text{vector of the } i\text{th modal amplitude and the } i\text{th modal velocity}
\]
Equation (44) can be expanded to include the disturbance input, $w(kT)$:

$$X_i(kT + 1) = A_i(T)X_i(kT) + B_i(T)x_i^T[F(kT) + w(kT)]$$

Equation (45) is the discrete-time mathematical model describing the motion of the structure in terms of its natural modes of vibration.
III. THE OBSERVER

A. INTRODUCTION

The observer design will be required to estimate the modal states in a noisy environment. Kalman filtering is the most widely used technique for accomplishing the production of state estimates in a noisy environment [Ref. 5: p.159]. The steady state Kalman filter was selected to minimize the computations during the actual plant observer operation. Use of a steady-state gain matrix for the observer allows the matrix to be computed separately from the operational observer, reducing the computer power required for the observer and allowing the algorithm to operate more rapidly.

B. KALMAN FILTER EQUATIONS

The discrete Kalman filter provides state estimates for the following dynamic system [Ref. 5: p. 159-162],

\[ X(k+1) = AX(k) + BU(k) + BnW(k) \]

\[ Y(k+1) = CX(k+1) + V(k+1) \]

\[ X = n \times 1 \text{ state vector} \]
\[ U = P \times 1 \text{ control vector} \]
\[ W = r \times 1 \text{ plant noise vector} \]
\[ Y = m \times 1 \text{ measurement vector} \]
\[ V = m \times 1 \text{ measurement noise vector} \]
\[ A = n \times n \text{ state transition matrix} \]
\[ B = n \times p \text{ control input matrix} \]
\[ Bn = n \times r \text{ plant noise input matrix} \]
\[ C = m \times n \text{ measurement matrix} \]

The plant noise vector \( W(k) \) is gaussian white noise with

\[ E(W(k)) = 0 \]

\[ E(W(k)W^T(k)) = Q \]
for all $k = 0,1,2,...$, and $Q$ is a positive semi-definite $r \times r$ matrix. $V(k)$ is gaussian white noise with

$$E\{V(k)\} = 0$$  \hfill (50)$$

$$E\{V(k)V^T(k)\} = R$$  \hfill (51)$$

for all $k = 0,1,2,...$, and $R$ is a positive definite $m \times m$ matrix. The two random processes $W(k)$ and $V(k)$ are assumed to be independent, so that

$$E\{V(j)W(k)\} = 0$$  \hfill (52)$$

for all $j = 1,2,...$, and $k = 0,1,2,...$. The initial state $X(0)$ is assumed to be a gaussian random vector with

$$E\{X(0)\} = 0$$  \hfill (53)$$

It is assumed that $X(0)$ is independent of $W(k)$ and $V(k)$.

The optimal estimate of $X(k+1)$ is denoted $\hat{X}(k+1 | k+1)$. The Kalman filter is designed to minimize

$$J = E\{(X(k+1) - \hat{X}(k+1 | k+1))\}^T[X(k+1) - \hat{X}(k+1 | k+1)]$$  \hfill (54)$$

The recursive realtions for generating $\hat{X}(k+1 | k+1)$ are

$$\hat{X}(k+1 | k) = A\hat{X}(k | k) + BU(k)$$  \hfill (55)$$

$$\hat{X}(k+1 | k+1) = \hat{X}(k+1 | k) + G(k+1)[Y(k+1) - C\hat{X}(k+1 | k)]$$  \hfill (56)$$

for $k = 0,1,2,...$, where $\hat{X}(0 | 0) = 0$. $\hat{X}(0 | 0)$ is set equal to zero since the expectation of $X(0)$ is zero.

$G(k+1)$ is an $n \times m$ matrix, called the Kalman Gain Matrix which is specified by the realtions:

$$P(k+1 | k) = AP(k | k)A^T + BnQ(k)Bn^T$$  \hfill (57)$$

$$G(k+1) = P(k+1 | k)C^T[CPC(k+1 | k)C^T + R(k+1)]^{-1}$$  \hfill (58)$$

$$P(k+1 | k+1) = [I - G(k+1)C]P(k+1 | k)$$  \hfill (59)$$
\( P(k | k) \) is the covariance matrix of the error between the states and their estimates

\[
P(k | k) = E\{[X(k) - \hat{X}(k | k)][X(k) - \hat{X}(k | k)]^T\} \quad (60)
\]

Since we are using the steady state gains the choice of \( P(0 | 0) \) is irrelevant. \( P(0 | 0) \) is initialized to zero in the gain derivation program for simplicity [Ref. 6: p. 139-140].

C. STEADY-STATE SOLUTION

If Equations (57), (58), and (59) are repeatedly iterated, \( G(k + 1) \) will converge to a steady state value [Ref. 7: p. 263].

\[
G_{ss} = \lim_{k \to \infty} G(k + 1) \quad (61)
\]

The values of \( G_{ss} \) (or \( G \)) can be substituted into Eq. (56) making the steady state Kalman filter

\[
\hat{X}(k + 1 | k) = A\hat{X}(k | k) + BU(k) \quad (62)
\]

\[
\hat{X}(k + 1 | k + 1) = \hat{X}(k + 1 | k) + G[Y(k + 1) - C\hat{X}(k + 1 | k)] \quad (63)
\]

D. OBSERVER PERFORMANCE

The performance of an observer is judged by how accurately and rapidly it estimates the desired states. The performance measure of the observer as a whole is shown in equation (54). The normalized performance of the observer for individual states is

\[
J_i = E[(x_i - \hat{x}_i)^2]/E[x_i^2] \quad (64)
\]

which can be found using Eq. (65)

\[
J_i = \sum_{k=0}^{\infty} (x_i(k) - \hat{x}_i(k))^2 T_s + \sum_{k=0}^{\infty} x_i^2(k) T_s \quad (65)
\]

\( J_i \) = performance measure for the \( i \)th state

\( x_i(k) \) = value of the \( i \)th state at \( k \)

\( \hat{x}_i(k) \) = observer estimate of \( i \)th state at \( k \)
\[ T_s = \text{sample interval} \]

A normalized performance measure is used to aid comparison of the performance of the observer in estimating various states. From Eq. (65) it can be shown that if \( \hat{x}_i(k) = 0 \) for all \( k = 0,1,2,... \) that \( J_i \) would be unity. Therefore, the better the performance of the observer, the smaller the fraction of one \( J_i \) will be.
IV. SIMULATION

A. INTRODUCTION

The objectives of the simulation were to

- determine the sensitivity of the observer performance and settling time to changes in the ratio of plant noise to measurement noise,
- determine the effect on observer performance and settling time of increasing the number of modes observed in the matched plant/observer, and
- determine the performance for the reduced order observer.

B. PLANT AND OBSERVER DATA

The dynamic model is a truncated form of a preliminary space station configuration; the phase II dual keel structure.\(^1\) The full model consists of an infinite number of natural modes but this was restricted to the first ten active modes for this study due to limitations on computer resources. As will be shown reasonable data can be obtained with this simplification in examining the observer performance.

C. SIMULATION PROGRAMS

The simulation was broken down into two segments due to the large memory and computational time requirements. The first program computed the steady state observer gain matrix (\(G\)). The second program ran the observer and the plant when the plant was subjected to an impulse excitation.

The steady state observer gain matrix (\(G\)) was obtained by repeated iteration of equations (57), (58), and (59). The equations were run until the values of the matrix changed by less than a set fraction. The following formula was used to check the changes in the gain matrix elements

\[
\Delta g_{ij} = \frac{|g_{ij}(k+1) - g_{ij}(k)|}{g_{ij}(k+1)}
\]

The program was terminated when \(\Delta g_{ij}\) was less than \(10^{-10}\).

The settling time for the estimates of the states to be within 2% of the actual states was determined by finding the eigenvalues of \(A - G*C\) then computing as follows [Ref. 6: p. 139-143]

\(^1\) The model for preliminary station configuration was provided courtesy of McDonnel Douglas Astronautics Company, 5301 Bolsa Avenue, Huntington Beach, CA 92647.
The expected error in the sensor, i.e., the standard deviation of the noise in the measurement, was chosen as \(10^{-3}\) feet per sec. per sec. based on the natural frequencies in the structure and reasonable sensor sensitivity [Ref. 2: p. 79-80]. The expected plant noise was varied to find the range of ratios between plant and measurement noise that the filter would be effective. This approach was taken since the plant noise contributors are not currently well defined.

The second program subjected the plant as modeled in Eq. (45) to an impulse excitation and then had the observer estimate the selected states using observer equations (62) and (63). Observer performance was computed using Eq. (65).

A third program was used to find the contribution of unobserved modes to the noise in the Kalman observer. The program ran the plant subject to an impulse excitation and computed the product of the measurement matrix \(C\) times the unobserved modes of the state vector \(X(k)\) for a measure of the noise contributed by the unobserved modes.

The three programs are listed in the appendices.

D. EFFECT OF PLANT TO MEASUREMENT NOISE RATIO ON OBSERVER PERFORMANCE

The ratio of the variance of the plant noise (PN) to the variance of the measurement noise (MN) was found to have a strong effect on the Kalman Observer performance \((J)\) and settling time \((T_s)\). Figures 1 through 6 show the observer performance for a 3 mode matched plant and observer system for progressively smaller PN/MN ratios. Figure 7 shows the performance for the seventh mode (position) versus several values of PN/MN. Figure 8 is the settling time versus the same PN/MN ratios.

The figures show that, for all of the plotted performance values, the observer performance is at least marginally acceptable regardless of the PN/MN ratio. Decreasing the PN/MN ratio leads to an even more rapid degradation in observer performance. The settling times also rapidly increase as the PN/MN ratio decreases.

\[ T_s = \frac{\log(0.02)}{\log(\lambda_{AGC_{min}})} \]
E. EFFECTS OF INCREASED MODES ON OBSERVER PERFORMANCE

The matched plant/observer was run with increasing numbers of modes to see if there was an effect on observer performance \( J \) or settling time \( T \). Figures 7 through 17 are of observer performance for systems with increasing numbers of modes in the system being observed. Figure 18 is of settling time versus the number of modes in the system. The ratio of \( PN/MN \) was kept constant at \( PN/MN = 2.5 \times 10^9 \).

The increasing of the number of modes for the matched plant/observer had negligible effect on the performance for the individual modes. The performance value for the modes was effectively constant. Settling times for the observers increased as the number of modes was increased.

F. REDUCED ORDER KALMAN OBSERVER

The Kalman Observer has been shown to be effective where the number of modes observed matches the number of modes in the plant. The Kalman Observer was then run with the one less mode observed than the number of modes in the plant. The gain matrix \( G \) from the matched system was used. The observer failed with the state estimates produced by the observer becoming excessively large and having settling times of hours vice minutes. Since the purpose of the observer was to provide estimates for use in controlling the plant the time delay makes the estimates unusable.

The cause of the observer failure is apparent when you look at the last portion of Eq. (56) of the Kalman Observer

\[
G[Y(k + 1) - CX(k + 1 | k)]
\]  \hspace{1cm} (68)

This portion of the observer equation is the correction of \( \dot{X}(k + 1 | k) \) to produce \( \dot{X}(k + 1 | k + 1) \). The design of the Kalman observer is to produce an estimate despite the measurement noise but, with the reduced order filter there is additional unanticipated noise which causes over correction of the values of \( \dot{X} \) leading to the state estimates being excessively large and settling times being too long. This can be shown by examining what composes \( Y(k + 1) - CX(k + 1 | k) \)
\[
\begin{bmatrix}
    x_1(k) \\
    x_2(k) \\
    \vdots \\
    x_{m-1}(k) \\
    x_m(k) \\
    x_{m+1}(k) \\
    x_{m+2}(k) \\
    \vdots \\
    x_{n-1}(k) \\
    x_n(k)
\end{bmatrix}
- C
\begin{bmatrix}
    \hat{x}_1(k) \\
    \hat{x}_2(k) \\
    \vdots \\
    \hat{x}_{m-1}(k) \\
    \hat{x}_m(k) \\
    \vdots \\
    \hat{x}_{n-1}(k) \\
    \hat{x}_n(k)
\end{bmatrix}
\]

(69)

\[
\begin{bmatrix}
    x_1(k) \\
    x_2(k) \\
    \vdots \\
    x_{m-1}(k) \\
    x_m(k) \\
    \vdots \\
    x_{m+1}(k) \\
    x_{m+2}(k) \\
    \vdots \\
    x_{n-1}(k) \\
    x_n(k)
\end{bmatrix}
- C
\begin{bmatrix}
    \hat{x}_1(k) \\
    \hat{x}_2(k) \\
    \vdots \\
    \hat{x}_{m-1}(k) \\
    \hat{x}_m(k) \\
    \vdots \\
    \hat{x}_{n-1}(k) \\
    \hat{x}_n(k)
\end{bmatrix}
= 0
\]

(70)

C times the state \(x_{m-1}(k)\) through \(x_n(k)\) is unanticipated noise so if

the remaining portion of the C matrix times the modal states is an equivalent noise.

Table (1) shows the growth of the unanticipated noise in the filter as the number of unobserved modes in the plant grows. Table (2) shows the individual contributions of the individual modes when left unobserved. Table (1) shows that the unanticipated noise is much larger than that expected by the filter \(10^{-2}\). Table (2) shows that there are modes that do not markedly contribute to the noise and that they might successfully be left unobserved if the measurement noise estimate was already much larger than these noise sources.
Table 1. CUMULATIVE UNANTICIPATED NOISE FROM UNOBSERVED MODES

<table>
<thead>
<tr>
<th>Number of Unobserved Modes</th>
<th>Unobserved Modes</th>
<th>E1</th>
<th>E2</th>
<th>E2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>0.647</td>
<td>97.440</td>
<td>3.277</td>
</tr>
<tr>
<td>2</td>
<td>10-11</td>
<td>366.354</td>
<td>95.764</td>
<td>3.142</td>
</tr>
<tr>
<td>3</td>
<td>10-12</td>
<td>366.355</td>
<td>95.764</td>
<td>3.142</td>
</tr>
<tr>
<td>4</td>
<td>10-13</td>
<td>365.565</td>
<td>95.855</td>
<td>3.143</td>
</tr>
<tr>
<td>5</td>
<td>10-14</td>
<td>365.426</td>
<td>96.032</td>
<td>5.704</td>
</tr>
<tr>
<td>6</td>
<td>10-15</td>
<td>144195.7</td>
<td>116.205</td>
<td>2201.94</td>
</tr>
<tr>
<td>7</td>
<td>10-16</td>
<td>148006.8</td>
<td>170.142</td>
<td>5475.21</td>
</tr>
<tr>
<td>8</td>
<td>10-17</td>
<td>473974.3</td>
<td>39424.1</td>
<td>5692.50</td>
</tr>
<tr>
<td>9</td>
<td>10-18</td>
<td>474344.2</td>
<td>60078.8</td>
<td>9419.77</td>
</tr>
<tr>
<td>10</td>
<td>10-19</td>
<td>474358.5</td>
<td>68987.7</td>
<td>9865.27</td>
</tr>
</tbody>
</table>

Table 2. UNANTICIPATED NOISE FROM UNOBSERVED MODES BY MODE

<table>
<thead>
<tr>
<th>Unobserved Mode</th>
<th>E1</th>
<th>E2</th>
<th>E3</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.64716</td>
<td>97.440</td>
<td>3.27761</td>
</tr>
<tr>
<td>11</td>
<td>359.342</td>
<td>0.20929</td>
<td>0.21429</td>
</tr>
<tr>
<td>12</td>
<td>0.9353E-08</td>
<td>0.1364E-07</td>
<td>0.2196E-07</td>
</tr>
<tr>
<td>13</td>
<td>0.3852E-02</td>
<td>0.4409E-02</td>
<td>0.9017E-03</td>
</tr>
<tr>
<td>14</td>
<td>0.5615E-03</td>
<td>0.2592E-01</td>
<td>2.57554</td>
</tr>
<tr>
<td>15</td>
<td>143090.9</td>
<td>17.9682</td>
<td>2167.02</td>
</tr>
<tr>
<td>16</td>
<td>195.736</td>
<td>83.3458</td>
<td>5675.91</td>
</tr>
<tr>
<td>17</td>
<td>324471.2</td>
<td>39252.26</td>
<td>221.170</td>
</tr>
<tr>
<td>18</td>
<td>216.240</td>
<td>21133.6</td>
<td>3736.93</td>
</tr>
<tr>
<td>19</td>
<td>7.69298</td>
<td>8829.95</td>
<td>458.504</td>
</tr>
<tr>
<td>20</td>
<td>2.3194</td>
<td>3949.16</td>
<td>108.981</td>
</tr>
</tbody>
</table>

18
PERFORMANCE MEASURE (J) PN/MN=1.0D12

Figure 1. Observer Performance (J) PN/MN = 1.0d12
Figure 2. Observer Performance (J) $PN/MN = 1.0d11$
Figure 3. Observer Performance (J) PN/MN = 1.0d10
Figure 4. Observer Performance (J) PN/MN = 5.0d09
Figure 5. Observer Performance (J) $PN/MN = 2.5 \times 10^9$
Figure 6. Observer Performance (J) PN/MN = 1.0D09
Figure 7. Mode 7 (Position) Observer Performance versus PN/MN
Figure 8. Settling Time versus PN/MN
Figure 9. Observer Performance (J) 4 Modes (7 - 10)
Figure 10. Observer Performance (J) 5 Modes (7 - 11)
Figure 11. Observer Performance (J) 6 Modes (7 - 12)
Figure 12. Observer Performance (J) 7 Modes (7 - 13)
Figure 13. Observer Performance (J) 8 Modes (7 - 14)
Figure 14. Observer Performance (J) 9 Modes (7 - 15)
Figure 15. Observer Performance (J) 10 Modes (7 - 16)
Figure 16. Settling Time versus number of Modes Observed
V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

Simulations runs showed that a matched plant/observer can work if the following criterions are meet:

- The ratio of plant noise to measurement noise is sufficiently high to produce a usable settling time.
- Sufficient computational power is available to run the matched observer. The amount of memory and number of computation goes up as the number of modes observed increases.

Utilizing a reduce order observer for an arbitrarily selected set of modes is not feasible. The non-observed modes add so much noise to the system that settling times and observer performance are so poor as to render the observer useless for obtaining state values for plant control.

B. RECOMMENDATIONS

The work on the Kalman Observer for Large Space Structures lead to the following recommendations for further research:

- Identify those modes that contribute the largest noise to the Kalman observer and set the observer to estimate these states in addition to those required for plant control. A possible method for identifying the modes that contribute the largest noise to the observer would be the Karhunen-Loeve expansion.
- Modifying the plant:observer to model the use of sensors at additional positions to see if the increase in the data rate will help decrease settling time.
- Modify the model to incorporate noise injection into more than one location. The current model has noise injected at only one position, a useful simplification for initial analysis but not realistic.
APPENDIX A. KALMAN GAIN MATRIX GENERATION PROGRAM

C ***************************************************************
C ***** GGAIN **
C ***** ADAPTED TO RUN KALMAN FILTER AND COMPUTE THE G MATRIX BY ITERATION STOPPING WHEN THE THE MATRIX GOES TO STEADY STATE
C ***************************************************************
C
C ***************************************************************
C ***** VARIABLE DECLARATIONS **
C ***************************************************************
C
EXTERNAL STMTRX,DLINRG,EXCMS, DEVCRG
CHARACTER*6 NAM
CHARACTER*1 AGAIN,CORECT,RAGAIN
INTEGER ROWN1,ROWN2,ROWN3,COUNT,NODE,MODE,QK,EMODE,SMODE,R2M,C2M
INTEGER CT,CF,KADJ,CFADJ,LOOP,PRNT,JJ,JK,N1,JR,KR,MR,ISEED,M2
INTEGER JL,J1,JM
REAL LAMA(100), UGVEX(684,100),RNODE,RMODE,MIN
REAL*8 PHI(2,2,00),GAMMA(2,100),EGT,GMA,W1,X1T,X2T,TIME
REAL*8 A(200,200),B(200,3),F(3, 50),IMPLSE,ENERGY
REAL*8 C(6,200), IDENT( 50, 50), RMN(6,6), QPN(3,3)
REAL*8 PK( 50, 50), Y(6), BN(200,3)
REAL*8 PNVARX, PNVARY, PNVARZ
REAL*8 MNVX1, MNVY1, MNVZ1, SUM, BQBT(50,50)
REAL*8 TMP1( 50,3), TMP2(3,3), TMP3( 50, 50)
REAL*8 PK1( 50, 50),G( 50,3)
REAL*8 DY(3), ES,ED,ESUM,CGN,PRT
REAL*8 SF, N9, TCHK, ACKH, H1, H2, H3, H4, H5, H6
REAL*8 AGC(100,100)
REAL*8 EVAL(100), EVEC (100,100)

C COMPLEX*8 EVAL(100), EVEC (100,100)

C ***************************************************************
C ***** VARIABLE DEFINITIONS **
C ***************************************************************
C
STMTRX = SUBROUTINE EXTABLISHES STATE TRANSITION MATRICIES
LAMA = VECTOR OF THE SQUARE OF THE NATURAL FREQUENCIES
UGVEX = MODE POSITONS AND SLOPES OF THE NODAL POINTS
PHI = STATE TRANSITION MATRICIES FOR EACH MODE
GAMMA = INPUT TRANSITION MATRIX
A = DIAGONAL MATRIX CONSISTING OF PHI
B = INPUT MATRIX OF GAMMA AND CONTROL SLOPES
DAMP = DAMPING FACTOR
SAMPT = SAMPLING TIME
C TCX, TCY, TCZ = CONTROL TORQUE VALUES
C ENERGY = TOTAL SYSTEM ENERGY
C IMPLSE = IMPULSE INPUT FUNCTION
C MIN = NUMBER OF MINUTES SYSTEM WILL BE OBSERVED
C SMODE = NUMBER OF STARTING MODE (INT)
C MODE = NUMBER OF MODES (INT)
C EMODE = NUMBER OF THE LAST MODE (INT)
C NODE = NUMBER OF THE NOISE INPUT MODE (INT)
C *** NOISE SLOPE LOCATIONS IN DATA MATRIX ***
C ROWN1 = X-SLOPE LOCATION
C ROWN2 = Y-SLOPE LOCATION
C ROWN3 = Z-SLOPE LOCATION
C C = OUTPUT MATRIX FOR Y
C IDENT = IDENTITY MATRIX
C RNM = MEASUREMENT NOISE COVARIANCE MATRIX
C QPN = PLANT NOISE COVARIANCE MATRIX
C PNVARX = PLANT NOISE X-SLOPE VARIANCE
C PNVARY = PLANT NOISE Y-SLOPE VARIANCE
C PNVARZ = PLANT NOISE Z-SLOPE VARIANCE
C MNVARX = MEASUREMENT NOISE X-SLOPE VARIANCE
C MNVARY = MEASUREMENT NOISE Y-SLOPE VARIANCE
C MNVARZ = MEASUREMENT NOISE Z-SLOPE VARIANCE
C ISEED = INITIALIZATION FOR RANDOM NUMBER GENERATOR
C XKAL = X MATRIX
C Y = OUTPUT MATRIX
C RNDM = RANDOM NUMBERS USED FOR WHITE NOISE IN MEASUREMENTS AND
C IN PLANT FORCES
C BN = B MATRIX TO MULTIPLY NOISE DISTURBANCES
C TNX,TNY,TNZ= NOISE TORQUES X,Y,Z SLOPES
C M2=2*MODE
C **************** SAMPLE OF SPACE EXEC FILE **********************
C THIS FILE MUST BEGIN IN COLUMN 1 AND RUN WITH THE FOLLOWING
C SEQUENCE FOR THE INITIAL RUN OF THE PROGRAM:
C FORTVS SPACE (COMPILES PROGRAM)
C SPACE (EXECUTES EXEC FILE)
C LOAD SPACE (START (LOADS AND EXECUTES PROGRAM)
C SUBSEQUENT PROGRAM RUNS CAN ELIMINATE "FORTVS SPACE" IF NO
C CHANGES HAVE BEEN MADE TO THE PROGRAM, AND CAN ELIMINATE
C RUNNING THE EXEC FILE.
C FI 4 DISK THESIS INPUT B (PERM)
C FI 8 DISK UTILITY DATA (RECFM VS BLOCK 133 PERM)
C FI 11 DISK CNTRL OUTPUT (RECFM F BLOCK 80 LRECL 80 PERM)
C FI 13 DISK GAMMA OUTPUT (RECFM VS BLOCK 133 PERM)
C FI 14 DISK MODE OUTPUT (RECFM F BLOCK 80 LRECL 80 PERM)
C FI 16 DISK COST OUTPUT (RECFM F BLOCK 80 LRECL 80 PERM)
C FI 17 DISK PRT OUTPUT (RECFM F BLOCK 80 LRECL 80 PERM)
C FI 18 DISK ERROR DATA (RECFM F BLOCK 80 LRECL 80 PERM)
C FI 19 DISK END FILE (RECFM F BLOCK 80 LRECL 80 PERM)
C FI 20 DISK GMAT FILE (RECFM F BLOCK 80 LRECL 80 PERM)
PARAMETER (JR=5243, KR=5397, MR=262139)

MIN =1200.0
WT=1.0D00
PI = 4.0D00 * ATAN(1.0D0)

READ LAMA AND UGVEX MATRICIES

CALL EXCMS ('CLRSCRN')
WRITE(6,1008) READING LAMA AND UGVEX MATRICIES'
WRITE(6,*)

THIS SECTION READS THE LAMA VECTOR AND THE UGVEX
MATRIX AND STORES THEM IN MEMORY FOR FURTHER RECALL OF
DESIRED LOCATION DATA.

READ(4,1001) NAM
READ(4,1002)(LAMA(I),I=1,100)
READ(4,1001) NAM
DO 5 J = 1,100
    READ(4,1002)(UGVEX(I,J),I=1,684)
5 CONTINUE

** NUMBER OF MODES SCANNED: ',I2)

** MODE 1 TO 93 (INTEGER)

MODE= 3
EMODE = SMODE + MODE - 1
WRITE (16,701) MODE

** NUMBER OF MODES SCANNED: ',I2)

** NODE 1 TO 114 (INTEGER) (IF 0 THEN NO NOISE INPUT)
NODE= 8
WRITE (16,702) NODE
702  FORMAT (',', 'NOISE NODE LOCATION: ',I5)
C
C  ***************  SAMPLING TIME  ***************
C  ** SAMPT MUST BE LESS THAN OR EQUAL TO SAMPTM **
C  SAMPT = .05
C  SAMPTM = ((2.0D0*PI)/SORT(LAMA(EMODE)))/2.0D0
C  IF (SAMPT.GE.SAMPTM) THEN
C  SAMPT=SAMPTM
C  ENDIF
C
C  WRITE (16,900) MIN
900  FORMAT (',',2X,'MIN: ',F8.3)
C
C  WRITE (16,703) SAMPT
703  FORMAT (',', 'SAMPLING TIME: ',D12.4)
C
C  ***************  DAMPING FACTOR  ***************
C  ** DAMP 0.0 TO 1.0 (REAL*8) **
C  DAMP=. 01
C
C  WRITE (16,704) DAMP
704  FORMAT (',', 'DAMPING FACTOR: ',D12.4)
C
C  *** PLANT NOISE VARIANCE ***
C  ** PNVARX, PNVARY, PNVARZ GT 0.0 **
C
C  SF1=2.5D06
C
C  PNVARX=1.0D00*SF1
C  PNVARY=1.0D00*SF1
C  PNVARZ=1.0D00*SF1
C
C  *** MEASUREMENT NOISE VARIANCE ***
C  ** MNVX1, MNVY1, MNVZ1 GT 0.0 **
C  SF=1. 0
C  MNVX1=5.5D-03*SF
C  MNVY1=5.5D-03*SF
C  MNVZ1=5.5D-03*SF
C
C  CALL EXCMS ('CLRSCRN')
510  WRITE (6,1008)
1008  WRITE (6,*) ' PROGRAM RUNNING'
C
C  ***************  NOISE INPUT LOCATION  ***************
C
C  ROWN3 = NODE*6
C  ROWN2 = (NODE*6) - 1
C  ROWN1 = (NODE*6) - 2
C  COUNT = 0
C
C  ************  INITIALIZE MATRICES  ************  
C
DO 40 I = 1,3  
DO 45 J = 1,3  
RMN(I,J)=0.0  
45  CONTINUE  
40  CONTINUE  
C
DO 47 I=1,50  
DO 46 J=1,50  
IDENT(I,J)=0.0  
PK(I,J)=0.0  
46  CONTINUE  
47  CONTINUE  
C
DO 60 I=1,3  
DO 58 J=1,3  
QPN(I,J)=0.0  
58  CONTINUE  
60  CONTINUE  
C
RMN(1,1)=MNVX1**2  
RMN(2,2)=MNVY1**2  
RMN(3,3)=MNVZ1**2  
QPN(1,1)=PNVARX**2.0  
QPN(2,2)=PNVARY**2.0  
QPN(3,3)=PNVARZ**2.0  
C
9999 FORMAT (')  
C
C  9999 FORMAT (')  
C
C
C
C  ******************  BEGIN MAIN PROGRAM  ******************  
C
CALL STMTRX(EMODE,SMODE,SAMPT,DAMP,PHI,GAMMA,A,B,LAMA,UGVEX,C,  
+ ROWN1,ROWN2,ROWN3,BN)  
C
C
CALL STMTRX(EMODE,SMODE,SAMPT,DAMP,PHI,GAMMA,A,B,LAMA,UGVEX,C,  
+ ROWN1,ROWN2,ROWN3,BN)  
C
C
C  *** PRE-LOOP PORTION OF KALMAN FILTER  
JK=SMODE*2-2  
M2=2*MODE  
DO 94 I=1,3  
DO 92 J=1,3,M2  
JL=JK+J  
SUM=0.0  
DO 90 K=1,3  
SUM=SUM+QPN(I,K)*BN(JL,K)  
90  CONTINUE  
TMP1(J,1)=SUM  
92  CONTINUE  
94  CONTINUE
C
C     DO 98 I=1,M2
JL=JK+I
C     DO 97 J=1,M2
SUM=0.0
C     DO 96 K=1,3
SUM=SUM+BN(JL,K)*TMP1(J,K)
C     CONTINUE
BQBT(I,J)=SUM
C     CONTINUE
C     CONTINUE
C
M2=2*MODE
C     DO 100 I=1,M2
DO 99 J=1,M2
C(I,J)=C(I,J)*SF
C     CONTINUE
100 CONTINUE
JL=JK+M2
C     DO 9375 I=1,3
DO 9374 J=1,JL
C(I,J)=C(I,J)*SF
C     CONTINUE
9374 CONTINUE
9375 CONTINUE
C
C***************************************************************
C     *** THIS SECTION COMPUTES THE STATE UPDATE  ***
C***************************************************************
C
ESUM=0.0
COUNT=0
ENERGY=0.0D0
TIME=0.0
CGN=0.0
C
C***************************************************************
C     *** SETS LOOP FOR THE ITERATIONS NECESSARY TO OBSERVE  ***
C***************************************************************
C
LOOP = INT((MIN*60.0)/SAMPT)
PRT=(DBLE(LOOP))/1200.0
PRTA=(DBLE(LOOP))/2400.0
CNTA=0.0
ACHK=1.0D-10
H1=0.0
H2=0.0
H3=0.0
H4=0.0
H5=0.0
H6=0.0
TCHK=MIN*60.0
9991 CONTINUE
C
TIME = TIME+ SAMPT
C
CGN=CGN+1.0
C
CNTA=CNTA+1.0
*** START OF KALMAN FILTER ***
M2=2*MODE

*** COMPUTATION OF PK*AT ***
JK=2*SMODE-2
DO 175 I=1,M2
    DO 170 J=1,M2
        JL=JK+J
        SUM=0.0
        DO 165 K=1,M2
            JM=JK+K
            SUM =SUM+PK(I,K)*A(JL,JM)
        CONTINUE
        TMP3(I,J)=SUM
    CONTINUE
175 CONTINUE

*** COMPUTATION OF A*(PK*AT)+ BQBT = PK1 ***
DO 190 I=1,M2
    JL=JK+I
    DO 185 J=I,M2
        SUM=0.0
        DO 180 K=1,M2
            JM=JK+K
            SUM=SUM+A(JL,JM)*TMP3(K,J)
        CONTINUE
        PK1(I,J)=SUM+BQBT(I,J)
    CONTINUE
190 CONTINUE

*** COMPUTE PK1*CT ****
DO 205 I=1,M2
    DO 200 J=1,3
        SUM=0.0
        DO 195 K=I,M2
            JM=JK+K
            SUM=SUM+PK1(I,K)*C(J,JM)
        CONTINUE
        TMP1(I,J)=SUM
    CONTINUE
200 CONTINUE
205 CONTINUE

*** COMPUTE C*(PK1*CT)+RMN ***
DO 220 I=1,3
    DO 215 J=1,3
        SUM=0.0
        DO 210 K=1,M2
            JM=JK+K
GMA03290
GMA03300
GMA03310
GMA03320
GMA03330
GMA03340
GMA03350
GMA03360
GMA03370
GMA03380
GMA03390
GMA03400
GMA03410
GMA03420
GMA03430
GMA03440
GMA03450
GMA03460
GMA03470
GMA03480
GMA03490
GMA03500
GMA03510
GMA03520
GMA03530
GMA03540
GMA03550
GMA03560
GMA03570
GMA03580
GMA03590
GMA03600
GMA03610
GMA03620
GMA03630
GMA03640
GMA03650
GMA03660
GMA03670
GMA03680
GMA03690
GMA03700
GMA03710
GMA03720
GMA03730
GMA03740
GMA03750
GMA03760
GMA03770
GMA03780
GMA03790
GMA03800
GMA03810
GMA03820
GMA03830
SUM=SUM+C(I,JM)*TMP1(K,J)
210 CONTINUE
TMP2(I,J)=SUM+RMN(I,J)
215 CONTINUE
220 CONTINUE

*** COMPUTATION OF THE INVERSE OF C*PK1*CT + R

CALL DLINRG (3,TMP2,3,TMP2,3)

*** COMPUTE CT*INV(C*PK1*CT+R)

DO 245 I=1,M2
   JL=JK+I
      DO 240 J=1,3
         SUM=0.0
            DO 235 K=1,3
               SUM=SUM+C(K,JL)*TMP2(K,J)
            235 CONTINUE
         TMPI(I,J)=SUM
      240 CONTINUE
245 CONTINUE

*** COMPUTE PK1*C*INV(C*PK1*CT+R) = G ******

DO 260 I=1,M2
   DO 255 J=1,3
      SUM=0.0
         DO 250 K=1,M2
            SUM=SUM+PK1(I,K)*TMP1(K,J)
         250 CONTINUE
      G(I,J)=SUM
   255 CONTINUE
260 CONTINUE

N9=DABS((G(1,1)-H1)/G(1,1))
IF (N9.GT.ACHK) THEN
   GO TO 7377
END IF
N9=DABS((G(1,3)-H2)/G(1,3))
IF (N9.GT.ACHK) THEN
   GO TO 7377
END IF
N9=DABS((G(2,1)-H3)/G(2,1))
IF (N9.GT.ACHK) THEN
   GO TO 7377
END IF
N9=DABS((G(2,3)-H4)/G(2,3))
IF (N9.GT.ACHK) THEN
   GO TO 7377
END IF
N9=DABS((G(3,3)-H5)/G(3,3))
IF (N9.GT.ACHK) THEN
   GO TO 7377
END IF

43
N9=DABS((G(M2,3)-H6)/G(M2,3))
IF (N9.GT.ACHK) THEN
GO TO 7377
END IF
GO TO 400
C
C 7377 CONTINUE
H1=G(1,1)
H2=G(1,3)
H3=G(2,1)
H4=G(2,3)
H5=G(3,3)
H6=G(M2,3)

IF (TCHK.LE.TIME) THEN
GO TO 400
END IF
IF (CGN.GE.PRT) THEN
C WRITE (6,*) 'TIME=',TIME,'SEC.'
C WRITE (6,*) 'N9=',N9
CGN=0.0
END IF
C
C *** COMPUTE IDENT - G*C
C
DO 275 I=1,M2
   DO 270 J=1,M2
      JL=JK+J
      SUM=0.0
         DO 265 K=1,3
            SUM=SUM+G(I,K)*C(K,JL)
            CONTINUE
         265 CONTINUE
      TMP3(I,J)= IDENT(I,J)-SUM
      CONTINUE
   270 CONTINUE
275 CONTINUE

C *** COMPUTE PK= (IDENT - G*C)*PK1
C
DO 290 I=1,M2
   DO 285 J=1,M2
      SUM=0.0
         DO 280 K=1,M2
            SUM=SUM+TMP3(I,K)*PK1(K,J)
            CONTINUE
         280 CONTINUE
      PK(I,J)=SUM
      CONTINUE
285 CONTINUE
290 CONTINUE
C
C
C END OF KALMAN FILTER PART 1 - START OF PART 2 ****

GO TO 9991

400 CONTINUE

WRITE (20,1008)
WRITE (20,*) 'TIME= ',TIME
DO 384 I=1,M2
WRITE (20,5350) G(I,1),G(I,2),G(I,3)
384 CONTINUE

5350 FORMAT (',5X,D15.8,5X,D15.8,5X,D15.8)
WRITE (20,*), 'N9=',N9

C *** COMPUTE AGC = A - G*C
C M2=2*MODE
JK=2*SMODE-2
C DO 7155 I=1,M2
JL=JK+I
DO 7154 J=1,M2
JM=JK+J
SUM=0.0
   DO 7153 K=1,3
      SUM=SUM+G(I,K)*C(K,JM)
   7153 CONTINUE
AGC(I,J)=A(JL,JM)-SUM
7154 CONTINUE
7155 CONTINUE

C
C *** COMPUTE THE EIGENVALUES OF AGC
C CALL DEVCRG (M2, AGC, 100, EVAL, EVEC, 100)
C
C ***** PRINT EVAL (EIGENVALUE) MATRIX
C DO 7157 I=1,M2
WRITE (20,*) 'I= ',I,' EIG= ',EVAL(I)
7157 CONTINUE

599 STOP
END

C
C********************************************************************************************
C THIS SUBROUTINE COMPUTES THE STATE TRANSITION MATRIX FOR EACH
C OF THE 100 MODES

******************************************************************************

SUBROUTINE STMTRX(EMODE,SMODE,T,D,PHI,GAMMA,A,B,LAMA,UGVEX,C,
+ ROW1,ROWN2,ROWN3,BN)

REAL*8 WN,GMA,PHI(2,2,100),GAMMA(2,100),EGT,T,COSW1T,SINW1T
REAL*8 W1,D,A(200,200),B(200,3),C(6,200),BN(200,3)
REAL LANAC 100) ,UGVEX(684,100)
INTEGER SMODE,R,EMODE,JJ,KK,ROWN1,ROWN2,ROWN3

DO 600 I = 1 ,100
WN = DBLE(SQRT(LAMA(I)))
GMA = D*WN/2.0
EGT = DEXP(-GMA*T)
W1 = DSQRT((WN**2)-(GMA**2))
COSW1T = DCOS(W1*T)
SINW1T = DSIN(W1*T)

IF(WN. EQ. 0)THEN
PHI(1,1,I) = EGT*COSW1T
PHI(1,2,I) = T
PHI(2,1,I) = 0
PHI(2,2,I) = EGT*COSW1T
ELSE

GAMMA(1,I) = 0
GAMMA(2,I) = 0

ENDIF

PHI(1,1,I) = EGT*(COSW1T + (GMA*(W1**(-1)))*SINW1T)
PHI(1,2,I) = (W1**(-1))*EGT*SINW1T
PHI(2,1,I) = -(WN**2)*(W1**(-1))*EGT*SINW1T
PHI(2,2,I) = EGT*(COSW1T - (GMA*(W1**(-1)))*SINW1T)

GAMMA(1,I)=(WN**(-2))*(1.DO-EGT*COSW1T-EGT*(GMA/W1)*SINW1T)

46
GAMMA(2,I) = (W1**(-1))*EGT*SINW1T

C
C
C
ENDF

C
C
C
600 CONTINUE

C
C
C
R = 1

C
DO 610 K = 1,100

C
A(R,R) = PHI(1,1,K)
A(R,R+1) = PHI(1,2,K)
A(R+1,R) = PHI(2,1,K)
A(R+1,R+1) = PHI(2,2,K)

C
C
C
*** B MATRIX FOR MULTIPLYING CONTROL TORQUES

B(R,1) = GAMMA(1,K)*DBLE(UGVEX(412,K))
B(R,2) = GAMMA(1,K)*DBLE(UGVEX(413,K))
B(R,3) = GAMMA(1,K)*DBLE(UGVEX(414,K))
B(R+1,1) = GAMMA(2,K)*DBLE(UGVEX(412,K))
B(R+1,2) = GAMMA(2,K)*DBLE(UGVEX(413,K))
B(R+1,3) = GAMMA(2,K)*DBLE(UGVEX(414,K))

C
C
C
*** BN MATRIX FOR MULTIPLYING THE NOISE DISTURBANCES

BN(R,1) = GAMMA(1,K)*DBLE(UGVEX(ROWN1,K))
BN(R,2) = GAMMA(1,K)*DBLE(UGVEX(ROWN2,K))
BN(R,3) = GAMMA(1,K)*DBLE(UGVEX(ROWN3,K))
BN(R+1,1) = GAMMA(2,K)*DBLE(UGVEX(ROWN1,K))
BN(R+1,2) = GAMMA(2,K)*DBLE(UGVEX(ROWN2,K))

47
BN(R+1,3) = GAMMA(2,K) * DBLE(UGVEX(ROWN3,K))

R = R+2

610 CONTINUE

*** C MATRIX PRODUCTION ***

JJ=-1
DO 640 I=1,100
J=JJ+1
KK=I+JJ

C(1, KK) = DBLE(UGVEX(418, I))
C(2, KK) = DBLE(UGVEX(419, I))
C(3, KK) = DBLE(UGVEX(420, I))

KK=KK+1

C(1, KK)=0.0
C(2, KK)=0.0
C(3, KK)=0.0

640 CONTINUE

RETURN
END
APPENDIX  B. KALMAN OBSERVER AND PLANT SIMULATION

****** SIMRUN

****** ADAPTED TO READ KALMAN FILETER G MATRICE

****** THEN RUN ALL N MODES OF THE PLANT WHILE

****** USING A KALMAN FILTER TO OBSERVE M

****** NUMBER OF STATES

*******************************************************************************

****** VARIABLE DECLARATIONS

*******************************************************************************

EXTERNAL STMTRX,EXCMS
CHARACTER*6 NAM
CHARACTER*1 AGAIN,CORECT,RAGAIN
INTEGER ROWN1,ROWN2,ROWN3,COUNT,MODE,KQ,EMODE,SMODE,R2M,C2M
INTEGER CT,CF,KADJ,CFADJ,LOOP,PRNT,J1,JK,N1,JR,KR,MR,ISEED,M2
INTEGER CT,CF,IPVT(100), NS, NF, SN, FN
INTEGER JL,J1,JM,JP,JQ,KA,KB,KC,KD,KE,KF,KG
REAL LAMA(100), UGVEX(684,100),RNODE,RNODE,MIN
REAL*8 PHI(2,2,100),GAMMA(2,100),EGT,GMA,WN,W1,X1T,X2T,TIME
REAL*8 A(200,200),B(200,3),F(3, 50),IMPLSE,ENERGY
REAL*8 COSW1T,SINW1T,X(200)
REAL*8 TCX,TCY,TCZ,DAMP,SAMPT,PI,SAMPTM,SUM1,SUM2,SUMC
REAL*8 C(3,200), IDENT( 50, 50), QPN(3,3)
REAL*8 Y(3), RN(200,3)
REAL*8 PNVARX, PNVARY, PNVARZ
REAL*8 MNVARX, MNVARY, MNVARZ
REAL*8 SUM, RNDM(6), RND1, RND2
REAL*8 XH( 50) ,BQBT( 50, 50)
REAL*8 SF1
REAL*8 TMP1( 50,3), TM2(3,3), TEMP( 50, 50)
REAL*8 G( 50,3)
REAL*8 XH1( 50) ,DY(3) , ES,ED,ESUM,CGN,PRT
REAL*8 WT , WD(3), BNVD(200)
REAL*8 AX(200) , V(3), SF , TO, CTT, ESS
REAL*8 CTG, XDEL, E2(100), XDEL1, ERS, PRT1, E3(100), XS(100)

*******************************************************************************

****** VARIABLE DEFINITIONS

*******************************************************************************

STMTRX = SUBROUTINE ESTABLISHES STATE TRANSITION MATRICIES
LAMA = VECTOR OF THE SQUARE OF THE NATURAL FREQUENCIES
UGVEX = MODE POSITIONS AND SLOPES OF THE NODAL POINTS
PHI = STATE TRANSITION MATRICIES FOR EACH MODE
C GAMMA = INPUT TRANSITION MATRIX
C A = DIAGONAL MATRIX CONSISTING OF PHI
C B = INPUT MATRIX OF GAMMA AND CONTROL SLOPES
C DAMP = DAMPING FACTOR
C SAMPT = SAMPLING TIME
C TCX, TCY, TCZ = CONTROL TORQUE VALUES
C ENERGY = TOTAL SYSTEM ENERGY
C IMPLSE = IMPULSE INPUT FUNCTION
C MIN = NUMBER OF MINUTES SYSTEM WILL BE OBSERVED
C SMODE = NUMBER OF STARTING MODE (INT)
C MODE = NUMBER OF MOCDES (INT)
C EMODE = NUMBER OF THE LAST MODE (INT)
C NODE = NUMBER OF THE NOISE INPUT MODE (INT)
C ROWN1 = X-SLOPE LOCATION
C ROWN2 = Y-SLOPE LOCATION
C ROWN3 = Z-SLOPE LOCATION
C C = OUTPUT MATRIX FOR Y
C IDENT = IDENTITY MATRIX
C RNMI = MEASUREMENT NOISE COVARIANCE MATRIX
C QPN = PLANT NOISE COVARIANCE MATRIX
C PNVARX = PLANT NOISE X-SLOPE VARIANCE
C PNVARY = PLANT NOISE Y-SLOPE VARIANCE
C PNVARZ = PLANT NOISE Z-SLOPE VARIANCE
C MNVARX = MEASUREMENT NOISE X-SLOPE VARIANCE
C MNVARY = MEASUREMENT NOISE Y-SLOPE VARIANCE
C MNVARZ = MEASUREMENT NOISE Z-SLOPE VARIANCE
C ISEED = INITIALIZATION FOR RANDOM NUMBER GENERATOR
C XKAL = X MATRIX
C Y = OUTPUT MATRIX
C RNDM = RANDOM NUMBERS USED FOR WHITE NOISE IN MEASUREMENTS AND
C IN PLANT FORCES
C BN = B MATRIX TO MULTIPLY NOISE DISTURBANCES
C TNX,TNY,TNZ = NOISE TORQUES X,Y,Z SLOPES
C M2=2*MODE
C

*************** SAMPLE OF SPACE EXEC FILE ***************
C
C THIS FILE MUST BEGIN IN COLUMN 1 AND RUN WITH THE FOLLOWING
C SEQUENCE FOR THE INITIAL RUN OF THE PROGRAM:
C
C FORTVS SPACE (COMPiles PROGRAM)
C SPACE (EXECutes EXEC FILE)
C LOAD SPACE (START) (LOADS AND EXECutes PROGRAM)
C
C SUBSEQUENT PROGRAM RUNS CAN ELIMINATE "FORTVS SPACE" IF NO
C CHANGES HAVE BEEN MADE TO THE PROGRAM, AND CAN ELIMINATE
C RUNNING THE EXEC FILE.
C
C FI 4 DISK THESIS INPUT B (PERM)
C FI 8 DISK UTILITY DATA (RECFM VS BLOCK 133 PERM)
C FI 11 DISK CNTRL OUTPUT (RECFM F BLOCK 80 LRECL 80 PERM)
C FI 13 DISK GAMMA OUTPUT (RECFM VS BLOCK 133 PERM)
C FI 14 DISK MODE OUTPUT (RECFM F BLOCK 80 LRECL 80 PERM)
C FI 16 DISK COST OUTPUT (RECFM F BLOCK 80 LRECL 80 PERM S1M01080
C FI 17 DISK PRT OUTPUT (RECFM F BLOCK 80 LRECL 80 PERM S1M01090
C FI 18 DISK ERROR DATA (RECFM F BLOCK 80 LRECL 80 PERM S1M01100
C FI 19 DISK END FILE (RECFM F BLOCK 80 LRECL 80 PERM S1M01110
C FI 20 DISK GMAT FILE (RECFM F BLOCK 80 LRECL 80 PERM S1M01120
C
**********************************************************************
C PARAMETER (JR=5243, KR=5397, MR=262139)
C
C MIN =1.00
C
C WT=1.0D00
C PI = 4.0D0 * ATAN(1.0D0)
C
**********************************************************************
C ***** READ LAMA AND UGVEX MATRICIES *****
C
C CALL EXCMS ( 'CLRSCRN' )
C WRITE(6,1008) S1M01280
C WRITE(6,*) ' READING LAMA AND UGVEX MATRICIES' S1M01290
C WRITE(6,*) ' ' S1M01300
C THIS SECTION READS THE LAMA VECTOR AND THE UGVEX S1M01310
C MATRIX AND STORES THEM IN MEMORY FOR FURTHER RECALL OF S1M01320
C DESIRED LOCATION DATA.
C
C READ(4,1001) NAM S1M01330
C READ(4,1002)(LAMA(I),1=1,100) S1M01340
C DO 5 J = 1,100 S1M01350
C READ(4,1002)(UGVEX(I,J),I=1,684) S1M01360
C CONTINUE S1M01370
C
C 1001 FORMAT(1X,A6) S1M01380
C 1002 FORMAT(1X,8EI5.8) S1M01390
C 1008 FORMAT (1X,///) S1M01400
C
C 500 CALL EXCMS ( 'CLRSCRN' ) S1M01410
C
C ************* STARTING MODE NUMBER ************* S1M01420
C ** SMODE 7 TO 100 (INTEGER) **** S1M01430
C SMODE= 7 S1M01440
C
C WRITE (16,700) SMODE S1M01450
C 700 FORMAT ( ' ', 'STARTING MODE NUMBER: ',I2) S1M01460
C
C ************* NUMBER OF MODES TO SCAN ************* S1M01470
C ** MODE 1 TO 93 (INTEGER) ** S1M01480
C
C MODE=20 S1M01490
C
C EMODE = SMODE + MODE - 1 S1M01500
C
WRITE (16,701) NODE
701 FORMAT (',', 'NUMBER OF MODES SCANNED: ',I2)
C
C ******************** NOISE INPUT POSITION ********************
C ** NODE 1 TO 114 (INTEGER) (IF 0 THEN NO NOISE INPUT)
C NODE= 8
C
C WRITE (16,702) NODE
702 FORMAT (',', 'NOISE NODE LOCATION: ',I5)
C
C ******************** START AND STOP FOR PLANT
SN=7
FN=20
NS=SN*2-1
NF=SN*2+2*FN-2
WRITE (16,899) SN,FN
899 FORMAT (',', 'PLANT -- SN= ',I5,' FN= ',I5)
C ******************** SAMPLING TIME ********************
C ** SAMPT MUST BE LESS THAN OR EQUAL TO SAMPTM **
C SAMPT = 0.05
C SAMPTM = ((2.0D0*PI)/SQRT(LAMA(EMODE)))/1.0D0
C IF (SAMPT.GE.SAMPTM) THEN
C SAMPT=SAMPTM
C ENDIF
C
C WRITE (16,900) MIN
900 FORMAT (',', 'MIN: ',F8.3)
C
C WRITE (16,703) SAMPT, SAMPTM
703 FORMAT (',', 'SAMPLING TIME: ',D12.4,2X,'SAMPTM= ',D15.8)
C
C ******************** DAMPING FACTOR ********************
C ** DAMP 0.0 TO 1.0 (REAL*8)
DAMP=.01
C
C WRITE (16,704) DAMP
704 FORMAT (',', 'DAMPING FACTOR: ',D12.4)
C
C ******************** PLANT NOISE VARIANCE ********************
C ** PNVARX, PNVARY, PNVARZ GT 0.0
SF1=2.5D06
SF=1.0D00
C PNVARX=1.0D00*SF1
PNVARY=1.0D00*SF1
PNVARZ=1.0D00*SF1
C
C ******************** MEASUREMENT NOISE VARIANCE ********************
C ** MNVARX, MNVARY, MNVARZ GT 0.0
MNVARX=1.0D-03 *SF
MNVARY=1.0D-03 *SF
MNVARZ=1.0D-03 *SF
MNVARZ=1.0D-03 *SF

WRITE (16,711)
711 FORMAT('PLANT NOISE VARIANCE: ', F15.8)
WRITE (16,712)
712 FORMAT(14X, 'PNVARX', 13X, 'PNVARY', 13X, 'PNVARZ')
WRITE (16,713) PNVARX, PNVARY, PNVARZ
WRITE(16,714)
714 FORMAT('MEASUREMENT NOISE: ') WRITE(16,715)
715 FORMAT(14X, 'MNVARX', 13X, 'MNVARY', 13X, 'MNVARZ')
WRITE(16,713) MNVARX, MNVARY, MNVARZ
CALL EXCMS ('CLRSCRN')
WRITE (6,1008)
WRITE (6,*) 'PROGRAM RUNNING'

************ NOISE INPUT LOCATION ************

ROWN3 = NODE*6
ROWN2 = (NODE*6) - 1
ROWN1 = (NODE*6) - 2
COUNT = 0

************ INITIALIZE MATRICES ************

DO 48 K=1,50
   IDENT(K,K)=1.0
48 CONTINUE

DO 54 K = 1, 200
   X(K) = 0.0
54 CONTINUE

WRITE(6,1008)
WRITE (6,*) 'INITIALIZE RMN AND QPN MATRICES'

*** INITIALIZE RMN AND QPN MATRICES ***

DO 60 I=1,3
   DO 58 J=1,3
      RMN(I,J)=0.0
      QPN(I,J)=0.0
58 CONTINUE
60 CONTINUE

RMN(1,1)=MNVARX**2
RMN(2,2)=MNVARY**2
RMN(3,3)=MNVARZ**2
QPN(1,1)=PNVARX**2
QPN(2,2)=PNVARY**2
QPN(3,3)=PNVARZ**2.0
WRITE(6,1008)
WRITE(6,*), 'ENTER STMTRX'
CALL STMTRX(EMODE, SMODE, SAMPT, DAMP, PHI, GAMMA, A, B, LAMA, UGTEX, C, + ROWN1, ROWN2, ROWN3, BN)
WRITE (16,1008)
DO 11000 I=1,200
   DO 10900 J=1,3
      C(J,I)=C(J,I)*SF
   10900 CONTINUE
11000 CONTINUE

C 4,* PRE-LOOP PORTION OF KALMAN FILTER
M2=2*MODE
JP=2*SMODE-1
JQ=2*EMODE
DO 90 I=1,50
   XH(I)=0.0
90 CONTINUE

C THIS SECTION COMPUTES THE STATE UPDATE
DO 9771 I=1,100
   E2(I)=0.0
   E3(I)=0.0
   XS(I)=0.0
9771 CONTINUE

DO 384 I=1,M2
   WRITE (14,1008) G(I,1), G(I,2), G(I,3)
384 CONTINUE
WRITE (14,5350) G(I,1), G(I,2), G(I,3)
5350 FORMAT (' ', 2X, D15.8, 2X, D15.8, 2X, D15.8)
CTG=0.0
C ***** SETS LOOP FOR THE ITERATIONS NECESSARY TO OBSERVE *****
C ***** THE SYSTEM FOR THE NUMBER OF MINUTES SPECIFIED *****
WRITE (6,1008) SIM03290
WRITE (6,*)' START STATE UPDATE', SIM03290
LOOP = INT((MIN*60.0)/SAMPT)
PRT= (DBLE(LOOP))/30.0
CTT=0.0
C
DO 400 L = 0, LOOP
   TIME = DBLE(L)*SAMPT
C
   IF(L.EQ.0)THEN
      IMPLSE =(1.0D0*SF1)/(DSQRT(SAMPT))
   ELSE
      IMPLSE = 0.0D0
   ENDIF
C
   TO=0.0
C
*** RANDOM NUMBER GENERATOR ***
C
DO 101 I=1,6
   ISEED=MOD(ISEED*JR+KR,MR)
   RND1=(DBLE(ISEED)+0.5D00)/DBLE(MR)
   ISEED=MOD(ISEED*JR+KR,MR)
   RND2=(DBLE(ISEED)+0.5D00)/DBLE(MR)
   RNDM(I)=DSQRT(-2.0*DLOG(RND1))*DCOS(6.2831853D00*RND2)
101 CONTINUE
C
*** START OF STATE UPDATE ***
C
*** COMPUTE AX°200 = A°200 X 200 * X°200
C
JK=SMODE+2-2
JP=JK+1
JQ=2*EMODE
C
DO 5015 I=NS,NF
   SUM=0.0
   DO 5010 K=NS,NF
      SUM=SUM+A(I,K)*X(K)
   5010 CONTINUE
   AX(I)=SUM
5015 CONTINUE
C
*** COMPUTE WD°3
C
   WD(1)=PNVARX*RNDM(1)*TO+IMPLSE
   WD(2)=PNVARY*RNDM(2)*TO
   WD(3)=PNVARZ*RNDM(3)*TO
C
**C**

*** COMPUTE BNWD^0 200 = BN^0 200 X 3 * WD^3

DO 5035 I=NS,NF  
    SUM=0.0  
    DO 5030 K=1,3  
        SUM=SUM+BN(I,K)*WD(K)  
    CONTINUE  
BNWD(I)=SUM  
CONTINUE  

5035 CONTINUE

C

*** COMPUTE X^0 200 = AX^0 200 + BNWD^0 200

DO 5040 I=NS,NF  
    X(I)= AX(I) + BNWD(I)  
    IF (DABS(X(I)).LT.1.0D-60) THEN  
        X(I)=1.0D-60  
    END IF  

C

C

5040 CONTINUE

C

*** COMPUTE V^0 3
V(1)=MNVARX*RNDM(4)  
V(2)=MNVARY*RNDM(5)  
V(3)=MNVARZ*RNDM(6)  

C

*** COMPUTE Y^0 3 = C^0 3 X 200 * X^0 200 + V^0 3

DO 5050 I=1,3  
    SUM=0.0  
    DO 5045 K=NS,NF  
        SUM=SUM+C(I,K)*X(K)  
    CONTINUE  
Y(I)=SUM+V(I)  
CONTINUE  

5050 CONTINUE

C

***************

C

*** START OF KALMAN FILTER ***

M2=2*MODE

C

*** COMPUTE XH1 = A*XH

DO 300 I=JP,JQ  
    SUM=0.0  
    DO 295 K=JP,JQ  
        SUM=SUM+A(I,K) * XH(K)  
    CONTINUE  
XH1(I)=SUM  
CONTINUE  

300 CONTINUE

C

***********************
**COMPUTE \( \Delta Y = Y - C \cdot XH \)**

DO 315 I=1,3
    SUM=0.0
    DO 310 K=JP,JQ
        SUM=SUM+C(I,K)*XH(K)
    CONTINUE
    310 CONTINUE
    \( \Delta Y(I) = Y(I) - SUM \)
315 CONTINUE

**COMPUTE \( XH = XH1 + G \cdot \Delta Y \)**

DO 325 I=1,M2
    J1=JP-1+I
    SUM=0.0
    DO 320 K=1,3
        SUMI=SUM+G(I,K)*\( \Delta Y(K) \)
    CONTINUE
    XH(J1)=XH1(J1)+SUM
    IF (DABS(XH(J1)).LT.1.0D-60) THEN
        XH(J1)=1.0*D-60
    END IF
325 CONTINUE

****** END OF KALMAN ROUTINES ******

**COMPUTATION OF ESUM ***

DO 340 I=JP,JQ
    XDEL= X(I)-XH(I)
    XDEL1=XDEL*XDEL*SAMPT
    E2(I)=E2(I)+XDELI
    XS(I)=XS(I)+X(I)*X(I)*SAMPT
    E3(I)=E2(I)/XS(I)
340 CONTINUE

CGN=CGN+1.0
IF (CTT.EQ.1.0.OR.CGHN.GT.PRT) THEN
    WRITE (6,*),'TIME= ', TIME, ' SEC.'
WRITE (6,*) 'TIME= ', TIME, ' SEC.'
WRITE (17,1008) TIME
WRITE (16,4500) I,X(I),I,XH(I)
WRITE (17,21000) TIME
2100 FORMAT(' ', 'TIME= ', F9.3)
DO 380 I=JP, JQ
    WRITE (16,4500) I,X(I),I,XH(I)
380 CONTINUE

57
**THIS SUBROUTINE COMPUTES THE STATE TRANSITION MATRIX FOR EACH OF THE 100 MODES**

```
C SUBROUTINE STMTRX(EMODE,SMODE,T,D,PHI,GAMMA,A,B,LAMA,UGVEX,C, + ROWN1,ROWN2,ROWN3,BN)
C
REAL*8 WN,GMA,PHI(2,2,100),GAMMA(2,100),EGT,T,COSW1T,SINW1T
REAL*8 WI,D,A(200,200),B(200,3),C(3,200),BN(200,3)
INTEGER SMODE,R,EMODE,JJ,KK,ROWN1,ROWN2,ROWN3
C
DO 600 I = 1 ,100
  WN = DBLE(SQRT(LAMA(I)))
  GMA = D*WN/2.0
  EGT = DEXP(-GMA*T)
  WI = DSQRT((WN**2)-(GMA**2))
  COSW1T = DCOS(W1*T)
  SINW1T = DSIN(W1*T)
  IF(WN.EQ.0)THEN
    PHI(1,1,I) = EGT*COSW1T
    PHI(1,2,I) = T
    PHI(2,1,I) = 0
    PHI(2,2,I) = EGT*COSW1T
  ELSE
    PHI(1,1,I) = EGT*COSW1T
    PHI(1,2,I) = T
    PHI(2,1,I) = 0
    PHI(2,2,I) = EGT*COSW1T
  END IF
400 CONTINUE
```

58
**GAMMA(I,I)** = 0  
**GAMMA(2,I)** = 0

**ELSE**

**PHI**(1,1,I) = **EGT***(**COSW1T** + (**GMA***(**W1**(-1)))***SINW1T**)  
**PHI**(1,2,I) = (**W1**(-1))***EGT**'***SINW1T**  
**PHI**(2,1,I) = -**(W1**(-2))*(**W1**(-1))***EGT*****SINW1T**  
**PHI**(2,2,I) = **EGT***(**COSW1T** - (**GMA***(**W1**(-1)))***SINW1T**)  

**GAMMA**(1,I) = (**W1**(-2))*((1.0D0-**EGT*****COSW1T** -**EGT***(**GMA**/**W1**)***SINW1T**))  
**GAMMA**(2,I) = (**W1**(-1))***EGT*****SINW1T**

**ENDIF**

600 CONTINUE

**R** = 1

DO 610 **K** = 1,100

**A**(R,R) = **PHI**(1,1,**K**)  
**A**(R,R+1) = **PHI**(1,2,**K**)  
**A**(R+1,R) = **PHI**(2,1,**K**)  
**A**(R+1,R+1) = **PHI**(2,2,**K**)  

*** B MATRIX FOR MULTIPLYING CONTROL TORQUES

**B**(R,1) = **GAMMA**(1,**K**) * **DBL**E(UGVEX(412,**K**))  
**B**(R,2) = **GAMMA**(1,**K**) * **DBL**E(UGVEX(413,**K**))  
**B**(R,3) = **GAMMA**(1,**K**) * **DBL**E(UGVEX(414,**K**))  
**B**(R+1,1) = **GAMMA**(2,**K**) * **DBL**E(UGVEX(412,**K**))  

59
$B(R+1,2) = \Gamma(2,K) \cdot \text{DBLE}(UGVEX(413,K))$

$B(R+1,3) = \Gamma(2,K) \cdot \text{DBLE}(UGVEX(414,K))$

*** BN MATRIX FOR MULTIPLYING THE NOISE DISTURBANCES ***

$BN(R,1) = \Gamma(1,K) \cdot \text{DBLE}(UGVEX(ROWN1,K))$

$BN(R,2) = \Gamma(1,K) \cdot \text{DBLE}(UGVEX(ROWN2,K))$

$BN(R,3) = \Gamma(1,K) \cdot \text{DBLE}(UGVEX(ROWN3,K))$

$BN(R+1,1) = \Gamma(2,K) \cdot \text{DBLE}(UGVEX(ROWNI,K))$

$BN(R+1,2) = \Gamma(2,K) \cdot \text{DBLE}(UGVEX(ROWN2,K))$

$BN(R+1,3) = \Gamma(2,K) \cdot \text{DBLE}(UGVEX(ROWN3,K))$

R = R+2

610 CONTINUE

*** C MATRIX PRODUCTION ***

J=J+1

DO 640 I=1,100

$C(I,K) = \text{DBLE}(UGVEX(418,I))$

$C(2,K) = \text{DBLE}(UGVEX(419,I))$

$C(3,K) = \text{DBLE}(UGVEX(420,I))$

KK=KK+1

C(1,K)=0.0

C(2,K)=0.0

C(3,K)=0.0

640 CONTINUE
C
C
RETURN
END
APPENDIX  C. PROGRAM TO ESTIMATE NOISE IN KALMAN FILTER FROM UNOBSERVED MODES

C

************************************************************************************************
C  SPAC 24
C  ADAPTED TO RUN N MODES OF THE PLANT AND
C  COMPUTE THE NOISE IN THE KALMAN FILTER
C  FROM THE UNOBSERVED MODES
C
C************************************************************************************************

******* VARIABLE DECLARATIONS  ******

REAL LAMA(100), UGVEX(684,100), RNODE, RMODE, MIN
REAL*8 PHI(2,2,100), GAMMA(2,100), EGT, GMA, WN, W1, X1T, X2T, TIME
REAL*8 A(200,200), B(200,3), P(3, 50), IMPLSE, ENERGY
REAL*8 COSW1T, SINW1T, X(200)
REAL*8 DAMP, SAMPT, PI, SAMPTM, SUM1, SUM2, SUM3, SUMC
REAL*8 C(9,200), RMN(3,3), QPN(3,3)
REAL*8 RN(200,3)
REAL*8 PNVARX, PNVARY, PNVARZ
REAL*8 MNVARX, MNVARY, MNVARZ
REAL*8 SUM, RNDM(6), RND1, RND2
REAL*8 ES, ED, ESUM, CGN, PRT
REAL*8 WT, WD(3), BW(200), EX1(9)
REAL*8 EX(9), AX(200), S2, T0, CTT, ESS
REAL*8 CTG, XDEL, XDEL1, ERS, PRT1
REAL*8 SF1

C

************************************************************************************************
C  VARIABLE DEFINITIONS  ******

C

EXTERNAL STMTRX, EXCMS
CHARACTER*6 NAM
CHARACTER*1 AGAIN, CORRECT, RAGAIN
INTEGER ROWN1, ROWN2, ROWN3, COUNT, NODE, MODE, KQ, EMODE, SMODE, R2M, C2M
INTEGER CT, CF, RADJ, CFAJ, LOOP, PRNT, JJ, JK, N1, JR, KR, MR, ISEED, M2
INTEGER NO, NS, N, SN, FN, JL, J1, JM, JP, JQ, IA, IB, IC, ID, IE, IF
REAL LAMA(100), UGVEX(684,100), RNODE, RMODE, MIN
REAL*8 PHI(2,2,100), GAMMA(2,100), EGT, GMA, WN, W1, X1T, X2T, TIME
REAL*8 A(200,200), B(200,3), P(3, 50), IMPLSE, ENERGY
REAL*8 COSW1T, SINW1T, X(200)
REAL*8 DAMP, SAMPT, PI, SAMPTM, SUM1, SUM2, SUM3, SUMC
REAL*8 C(9,200), RMN(3,3), QPN(3,3)
REAL*8 RN(200,3)
REAL*8 PNVARX, PNVARY, PNVARZ
REAL*8 MNVARX, MNVARY, MNVARZ
REAL*8 SUM, RNDM(6), RND1, RND2
REAL*8 ES, ED, ESUM, CGN, PRT
REAL*8 WT, WD(3), BW(200), EX1(9)
REAL*8 EX(9), AX(200), S2, T0, CTT, ESS
REAL*8 CTG, XDEL, XDEL1, ERS, PRT1
REAL*8 SF1
B = INPUT MATRIX OF GAMMA AND CONTROL SLOPES
C DAMP = DAMPING FACTOR
C SAMPT = SAMPLING TIME
C IMPLSE = IMPULSE INPUT FUNCTION
C MIN = NUMBER OF MINUTES SYSTEM WILL BE OBSERVED
C SMODE = NUMBER OF STARTING MODE (INT)
C MODE = NUMBER OF MODES (INT)
C EMODE = NUMBER OF THE LAST MODE (INT)
C NODE = NUMBER OF THE NOISE INPUT MODE (INT)
C *** NOISE SLOPE LOCATIONS IN DATA MATRIX ***
C ROWN1 = X-SLOPE LOCATION
C ROWN2 = Y-SLOPE LOCATION
C ROWN3 = Z-SLOPE LOCATION
C C = OUTPUT MATRIX FOR Y
C IDENT = IDENTITY MATRIX
C RMN = MEASUREMENT NOISE COVARIANCE MATRIX
C QPN = PLANT NOISE COVARIANCE MATRIX
C PNVARX = PLANT NOISE X-SLOPE VARIANCE
C PNVARY = PLANT NOISE Y-SLOPE VARIANCE
C PNVARZ = PLANT NOISE Z-SLOPE VARIANCE
C MNVARX = MEASUREMENT NOISE X-SLOPE VARIANCE
C MNVARY = MEASUREMENT NOISE Y-SLOPE VARIANCE
C MNVARZ = MEASUREMENT NOISE Z-SLOPE VARIANCE
C ISEED = INITIALIZATION FOR RANDOM NUMBER GENERATOR
C RNDM = RANDOM NUMBERS USED FOR WHITE NOISE IN MEASUREMENTS AND
C IN PLANT FORCES
C BN = B MATRIX TO MULTIPLY NOISE DISTURBANCES

************* SAMPLE OF SPACE EXEC FILE **************

THIS FILE MUST BEGIN IN COLUMN 1 AND RUN WITH THE FOLLOWING
SEQUENCE FOR THE INITIAL RUN OF THE PROGRAM:
FORTVS SPACE  (COMPILES PROGRAM)
SPACE  (EXECUTES EXEC FILE)
LOAD SPACE  (START (LOADS AND EXECUTES PROGRAM)

SUBSEQUENT PROGRAM RUNS CAN ELIMINATE "FORTVS SPACE" IF NO
CHANGES HAVE BEEN MADE TO THE PROGRAM, AND CAN ELIMINATE
RUNNING THE EXEC FILE.

FI 4 DISK THESIS INPUT (PERM
FI 8 DISK UTILITY DATA (RECFM VS BLOCK 133 PERM
FI 11 DISK CNTRL OUTPUT (RECFM F BLOCK 80 LRECL 80 PERM
FI 13 DISK GAMMA OUTPUT (RECFM VS BLOCK 133 PERM
FI 14 DISK MODE OUTPUT (RECFM F BLOCK 80 LRECL 80 PERM
FI 16 DISK COST OUTPUT (RECFM F BLOCK 80 LRECL 80 PERM
FI 17 DISK PRT OUTPUT (RECFM F BLOCK 80 LRECL 80 PERM
FI 18 DISK ERROR DATA (RECFM F BLOCK 80 LRECL 80 PERM
FI 19 DISK END FILE (RECFM F BLOCK 80 LRECL 80 PERM
FI 20 DISK GMAT FILE (RECFM F BLOCK 80 LRECL 80 PERM

************** SAMPLE OF SPACE EXEC FILE **************
PARAMETER (JR=5243, KR=5397, MR=262139)
MIN = 15.0
WT=1.0D0
PI = 4.0D0 * ATAN(1.0D0)

CALL EXCMS ('CLRSCRN')
WRITE(6,1008)
WRITE(6,*), 'READING LAMA AND UGVEX MATRICIES'
WRITE(6,*),
THIS SECTION READS THE LAMA VECTOR AND THE UGVEX
MATRIX AND STORES THEM IN MEMORY FOR FURTHER RECALL OF
DESIRED LOCATION DATA.
READ(4,1001) NAM
READ(4,1002)(LAMA(I),I=1,100)
READ(4,1001)
DO 5 J = 1,100
READ(4,1002)(UGVEX(I,J),I=1,684)
CONTINUE
500 CALL EXCMS ('CLRSCRN')
1001 FORMAT(1X,A6)
1002 FORMAT(1X,8E5.8)
1008 FORMAT(1X,///)
500 CALL EXCMS ('CLRSCRN')

*************** STARTING MODE NUMBER ***************
** SMODE 7 TO 100 (INTEGER) ****
SMODE= 17
WRITE (16,700) SMODE
700 FORMAT (' ', 'STARTING MODE NUMBER: ',12)

*************** NUMBER OF MODES TO SCAN ***************
** MODE 1 TO 93 (INTEGER)
MODE=3
EMODE = SMODE + MODE - 1
WRITE (16,701) MODE
701 FORMAT (' ', 'NUMBER OF MODES SCANNED: ',12)

*************** NOISE INPUT POSITION ***************
** NODE 1 TO 114 (INTEGER) (IF 0 THEN NO NOISE INPUT)
NODE= 8
WRITE (16,702) NODE
702 FORMAT (' ', 'NOISE NODE LOCATION: ',15)

64
C *********** START AND STOP FOR PLANT
SN=17
FN=4
NS=SN*2-1
NF=SN*2+2*FN-2
WRITE (16,899) SN, FN
FORMAT (' PLANT -- SN= ',15,' FN= ',15)
C ********** SAMPT MUST BE LESS THAN OR EQUAL TO SAMPTM **********
SAMPT = 0.05
SAMPTM = (2.0D0*PI)/SQRT(LAMA(EMODE))/1.0D0
IF (SAMPT.GE.SAMPTM) THEN
SAMPT=SAMPTM
ENDIF
C
WRITE (16,900) MIN
FORMAT (' MIN: ',F8.3)
C
WRITE (16,703) SAMPT, SAMPTM
FORMAT (' SAMPLING TIME: ',D12.4,'SAMPT= ',D15.8)
C *************** DAMPING FACTOR ***************
C ** DAMP 0.0 TO 1.0 (REAL*8) **
DAMP=.01
C
WRITE (16,704) DAMP
FORMAT (' DAMPING FACTOR: ',D12.4)
C
NO=3
C *** PLANT NOISE VARIANCE ***
C ** PNVARX, PNVARY, PNVARZ GT 0.0 **
SF=1.0D0
SF1=2.5D06
PNVARX=1.0D0*SF1
PNVARY=1.0D0*SF1
PNVARZ=1.0D0*SF1
C
C *** MEASUREMENT NOISE VARIANCE ***
C ** MNVARX, MNVARY, MNVARZ GT 0.0 **
MNVARX=1.0D-03*SF
MNVARY=1.0D-03*SF
MNVARZ=1.0D-03*SF
C
WRITE (16,711)
FORMAT( ' PLANT NOISE VARIANCE: ')
WRITE (16,712)
FORMAT( '6X, 'PNVARX',13X,'PNVARY',13X,'PNVARZ')
WRITE (16,713) PNVARX, PNVARY, PNVARZ
WRITE (16,714)
FORMAT( '2X,E15.8,2X,E15.8,2X,E15.8)
WRITE(16,715) 
715 FORMAT(' ',6X,'MNVARX',13X,'MNVARY',13X,'MNVARZ') 
WRITE(16,713) MNVARX,MNVARY,MNVARZ 
C 
510 CALL EXCMS ('CLRSCRN') 
WRITE (6,1008) 
WRITE (6,*) ' PROGRAM RUNNING' 
C 
C ************ NOISE INPUT LOCATION ************ 
C 
ROWN3 = NODE*6 
ROWN2 = (NODE*6) - 1 
ROWN1 = (NODE*6) - 2 
COUNT = 0 
C 
C ************ INITIALIZE MATRICIES ************ 
C 
DO 54 K = 1, 200 
X(K) = 0.0 
54 CONTINUE 
C 
DO 60 I=1,3 
DO 58 J=1,3 
RMN(I,J)=0.0 
QPN(I,J)=0.0 
58 CONTINUE 
60 CONTINUE 
C 
RMN(1,1)=MNVARX**2.0 
RMN(2,2)=MNVARY**2.0 
RMN(3,3)=MNVARZ**2.0 
QPN(1,1)=PNVARX**2.0 
QPN(2,2)=PNVARY**2.0 
QPN(3,3)=PNVARZ**2.0 
C 
C ******************* BEGIN MAIN PROGRAM ******************* 
C 
CALL STMTRX(EMODE,SMODE,SAMPT,DAP,PHI,GAMMA,A,B,LAMA,UGVEX,C, 
+ ROWN1,ROWN2,ROWN3,BN) 
C 
WRITE (6,1008) 
WRITE(6,*), 'EXIT STMTRX -- -- PRE-LOOP KALMAN' 
C 
C 
WRITE (6,*) ' COMPUTING C TIMES SF FOR NEW C' 
C 
WRITE (16,1008) 
DO 11000 I=1,200 
DO 10900 J=1,NO 
C(J,I)= C( J,I)*SF 
10900 CONTINUE 
11000 CONTINUE 
C
*** PRE-LOOP PORTION OF KALMAN FILTER

JK=SMODE*2-2
M2=2*MODE

************

M2=2*MODE
JP=2*SMODE-1
JQ=2*EMODE

DO 8813 I=1,3
EX(I)=0.0
8813 CONTINUE

************

***** SETTINGS LOOP FOR THE ITERATIONS NECESSARY TO OBSERVE *****

WRITE (6,1008) 'START STATE UPDATE', LOOP

DO 400 L = 0, LOOP
TIME = DBLE(L)*SAMPT

IF(L.EQ.0)THEN
    IMPLSE = (1.0D06*SF1)/(DSQRT(SAMPT))
ELSE
    IMPLSE = 0.0D0
END IF

TO=0.0

***** RANDOM NUMBER GENERATOR *****

DO 101 I=1,6
ISEED=MOD(ISEED*JR+KR,MR)
RND1=(DBLE(ISEED)+0.5D0)/DBLE(MR)
ISEED=MOD(ISEED*JR+KR,MR)
RND2=(DBLE(ISEED)+0.5D0)/DBLE(MR)
RNDM(I) = DSQRT(-2.0*DLOG(RND1))*DCOS(6.2831853D00*RND2)

101 CONTINUE

C

C **********************
C
C **** START OF STATE UPDATE ****
C
C *** COMPUTE AX^200 = A^200 X 200 * X^200
C
C *** COMPUTE AXH = A*XH
C
JK=SMODE*2-2
JP=JK+1
JQ=2*EMODE
C
DO 5015 I=NS,NF
   SUM=0.0
   DO 5010 K=NS,NF
      SUM=SUM+A(I,K)*X(K)
   CONTINUE
   AX(I)=SUM
5015 CONTINUE

C

C *** COMPUTE WD^3
C
WD(1)=PNVARY*RNDM(1)*TO+IMPLSE
WD(2)=PNVARY*RNDM(2)*TO
WD(3)=PNVARZ*RNDM(3)*TO
C
C *** COMPUTE BNWD^200 = BN^200 X 3 * WD^3
C
DO 5035 I=NS,NF
   SUM=0.0
   DO 5030 K=1,3
      SUM=SUM+BN(I,K)*WD(K)
   CONTINUE
   BNWD(I)=SUM
5035 CONTINUE

C

C *** COMPUTE X^200 = AX^200 + BNWD^200
C
DO 5040 I=NS,NF
   X(I) = AX(I) + BNWD(I)
   IF (DABS(X(I)).LT.1.0D-60) THEN
      X(I)=1.0D-60
   END IF
5040 CONTINUE

C

C **********************
C
C *** START OF KALMAN FILTER ***
C
JK=SMODE*2-2
JP=JK+1
JQ=2*EMODE
M2=2*MODE

C

JL=JQ+1
DO 8888 I=1,NO
SUM=0.0
    DO 8887 K=JL,NF
        SUM=SUM+C(I,K)*X(K)
8887 CONTINUE
EX(I)=SUM*SUM*SAMPT+EX(I)
8888 CONTINUE
C

CGN=CGN+1.0
IF (CTT.EQ.1.0.OR.CGNN GT PRT) THEN
C
WRITE (16,1008)
WRITE (16,*) 'TIME = ', TIME
C
DO 380 I=JP, JQ
WRITE (16,4500) I,X(I)
380 CONTINUE
4500 FORMAT (' ',2X,'X(',I4,')= ',D15.8)
C
CGN=0.0
END IF
C
C
400 CONTINUE
WRITE (11,*) 'SMODE = ', SMODE
WRITE (11,*) 'EMODE = ', EMODE
WRITE (11,*) 'SN = ', SN
WRITE (11,*) 'FN = ', FN
C
JL=JQ+1
DO 9499 I=1,NO
WRITE (11,*) 'EX ',I ', EX(I)
9499 CONTINUE
C
C
CALL EXCMS ('CLRSCRN')
WRITE (6,1008)
C
STOP
END
C
C
C
C
**THIS SUBROUTINE COMPUTES THE STATE TRANSITION MATRIX FOR EACH OF THE 100 MODES**

SUBROUTINE STMTRX(EMODE,SMODE,T,D,PHI,GAMMA,A,B,LAMA,UGVEX,C,ROWN1,ROWN2,ROWN3, BN)

REAL*8 WN,GMA,PHI(2,2,100),GAMMA(2,100),EGT,T,COSWIT,SINWIT
REAL*8 WI,D,A(200,200),B(200,3),C(9,200),BN(200,3)
INTEGER SMODE,R,EMODE,JJ,KK,ROWN1,ROWN2,ROWN3,NN(9), N9, NO

WRITE (6,*) 'INSIDE STMTRX - - COMPUTE WN, GMA, EFT, W1'

DO 600 1 = 1,100
WN = DBLE(SQRT(LAMA(I)))
GMA = D*WN/2.0
EGT = DEXP(-GMA*T)
WI = DSQRT((WN**2)-(GMA**2))
COSWIT = DCOS(W1*T)
SINWIT = DSIN(WI*T)

IF(WN. EQ.0)THEN
PHI(1,1,I) = EGT*COSWIT
PHI(1,2,I) = T
PHI(2,1,I) = 0
PHI(2,2,I) = EGT*COSWIT
GAMMA(1,I) = 0
GAMMA(2,I) = 0
ELSE
PHI(1,1,I) = EGT*(COSWIT + (GMA*(W1**(-1)))*SINWIT)
PHI(1,2,I) = (W1**(-1))*EGT*SINWIT
PHI(2,1,I) = -(WN**2)*((W1**(-1))*EGT*SINWIT
PHI(2,2,I) = EGT*(COSWIT - (GMA*(W1**(-1))))*SINWIT)
GAMMA(1,I)=-(WN**(-2))*(1.0D0-EGT*COSWIT -EGT*(GMA/W1)*SINWIT)

70
GAMMA(2,1) = (W1**(-1))*EGT*SINW1T

ENDIF

600 CONTINUE

WRITE (6,*) 'PHI AND GAMMA COMPUTED'
WRITE (6,*) 'COMPUTING A, B, BN'

R = 1

DO 610 K = 1,100

A(R,R) = PHI(1,1,K)
A(R,R+1) = PHI(1,2,K)
A(R+1,R) = PHI(2,1,K)
A(R+1,R+1) = PHI(2,2,K)

*** B MATRIX FOR MULTIPLYING CONTROL TORQUES

B(R,1) = GAMMA(1,K)*DBLE(UGVEX(412,K))
B(R,2) = GAMMA(1,K)*DBLE(UGVEX(413,K))
B(R,3) = GAMMA(1,K)*DBLE(UGVEX(414,K))
B(R+1,1) = GAMMA(2,K)*DBLE(UGVEX(412,K))
B(R+1,2) = GAMMA(2,K)*DBLE(UGVEX(413,K))
B(R+1,3) = GAMMA(2,K)*DBLE(UGVEX(414,K))

*** BN MATRIX FOR MULTIPLYING THE NOISE DISTURBANCES

BN(R,1)=GAMMA(1,K)*DBLE(UGVEX(ROWN1,K))
BN(R,2)=GAMMA(1,K)*DBLE(UGVEX(ROWN2,K))
BN(R,3)=GAMMA(1,K)*DBLE(UGVEX(ROWN3,K))
BN(R+1,1)=GAMMA(2,K)*DBLE(UGVEX(ROWN1,K))
BN(R+1,2)=GAMMA(2,K)*DBLE(UGVEX(ROWN2,K))
\[ \text{BN}(R+1,3) = \Gamma(2,K) \times \text{DBLE}(\text{UGVEX}(\text{ROWN}3,K)) \]

\[ R = R+2 \]

\[ \text{WRITE (6,*) 'A, B, BN COMPUTED'} \]

\[ \text{WRITE (6,*) 'COMPUTING C'} \]

*** C MATRIX PRODUCTION ***

NO=3
NN(1)=418
NN(2)=419
NN(3)=420

JJ=-1
DO 640 I=1,100
JJ=JJ+1

DO 9127 K=1,NO
KK=I+JJ
N9=NN(K)
C(K,KK) = \text{DBLE}(\text{UGVEX}(N9,I))

KK=KK+1
C(K,KK)=0.0
9127 CONTINUE

640 CONTINUE
RETURN
END
LIST OF REFERENCES


<table>
<thead>
<tr>
<th>No.</th>
<th>Initial Distribution List</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Defense Technical Information Center</td>
</tr>
<tr>
<td></td>
<td>Cameron Station</td>
</tr>
<tr>
<td></td>
<td>Alexandria, VA 22304-6145</td>
</tr>
<tr>
<td>2.</td>
<td>Library, Code 0142</td>
</tr>
<tr>
<td></td>
<td>Naval Postgraduate School</td>
</tr>
<tr>
<td></td>
<td>Monterey, CA 93943-5002</td>
</tr>
<tr>
<td>3.</td>
<td>Director</td>
</tr>
<tr>
<td></td>
<td>Naval Research Laboratory</td>
</tr>
<tr>
<td></td>
<td>Washington, D.C. 20375</td>
</tr>
<tr>
<td>4.</td>
<td>Commander</td>
</tr>
<tr>
<td></td>
<td>Naval Space Command</td>
</tr>
<tr>
<td></td>
<td>Attn: Code N3</td>
</tr>
<tr>
<td></td>
<td>Dahlgren, VA 22448-5170</td>
</tr>
<tr>
<td>5.</td>
<td>Chairman, Code 62</td>
</tr>
<tr>
<td></td>
<td>Department of Electrical and Computer Engineering</td>
</tr>
<tr>
<td></td>
<td>Naval Postgraduate School</td>
</tr>
<tr>
<td></td>
<td>Monterey, CA 93943</td>
</tr>
<tr>
<td>6.</td>
<td>Prof. Jeff B. Burl, Code 62BL</td>
</tr>
<tr>
<td></td>
<td>Department of Electrical and Computer Engineering</td>
</tr>
<tr>
<td></td>
<td>Naval Postgraduate School</td>
</tr>
<tr>
<td></td>
<td>Monterey, CA 93943</td>
</tr>
<tr>
<td>7.</td>
<td>Prof. H. A. Titus, Code 62TS</td>
</tr>
<tr>
<td></td>
<td>Department of Electrical and Computer Engineering</td>
</tr>
<tr>
<td></td>
<td>Naval Postgraduate School</td>
</tr>
<tr>
<td></td>
<td>Monterey, CA 93943</td>
</tr>
<tr>
<td>8.</td>
<td>Major W. J. Preston</td>
</tr>
<tr>
<td></td>
<td>7 Cary Place</td>
</tr>
<tr>
<td></td>
<td>Fredricksburg, VA 22405</td>
</tr>
<tr>
<td>9.</td>
<td>Lcdr. Bruce M. Jackson</td>
</tr>
<tr>
<td></td>
<td>c/o Col. Jose A. Aybar (USA-Ret)</td>
</tr>
<tr>
<td></td>
<td>22577 Veronica Drive</td>
</tr>
<tr>
<td></td>
<td>Salinas, CA 93943</td>
</tr>
</tbody>
</table>