AN ABSTRACT OF THE THESIS OF

SHOU YUAN WEI for the degree of DOCTOR OF PHILOSOPHY in Mechanical Engineering presented on August 11, 1980.

Title: APPLICATION OF LINEAR QUADRATIC CONTROL DESIGN IN REDUCTION OF AERODYNAMIC FORCES ON AIRCRAFT

Redacted for Privacy

Abstract approved: ______________________
Dr. William E. Holley

A problem found in high speed transport aircraft is excessive tail loading when flying at cruise speeds through turbulence. Attempts to reduce these areodynamic forces on the tail may result in unstable aircraft motions. Using a linear quadratic regulator in conjunction with a Kalman filter, the feasibility of designing an autopilot utilizing the ailerons and rudder is studied in terms of minimal combined lateral ride motion and aerodynamic tail force.

A new numerical algorithm for solving general quadratic regulator and state estimation problems is developed, and is presented with a computer program to solve the complicated matrix equations involved. A systematic procedure is also developed for choosing a reduced combination of sensors which gives near optimal performance. With this approach, aerodynamic tail forces can be theoretically reduced by 20 to 50 percent.

# Application of Linear Quadratic Control in Reduction of Aerodynamic Forces on Aircraft

by

SHOU YUAN WEI

A THESIS

submitted to

Oregon State University

in partial fulfillment of the requirements for the degree of

Doctoral of Philosophy

June, 1981

APPROVED:

Redacted for Privacy

Professor of Mechanical Engineering
in charge of major

Redacted for Privacy

Head of Department of Mechanical Engineering

Redacted for Privacy

Dean of Graduate School

Date thesis is presented August 11, 1980

Typed by J. Togger for SHOU YUAN WEI

## ACKNOWLEDGMENTS

First and foremost, I would like to express my appreciation to Dr. William E. Holley without whose help this paper would not have been possible. His knowledge, ingenuity, and expert direction guided me to the completion and presentation of this paper.

To my wife, Jane, I owe much. Without her encouragement and patience, this paper might never have been completed.

I also would like to take this opportunity to thank my parents for their support and to thank my brother-in-law, Dr. M.H. Chen, for his encouragement.

Financial assistance in the form of computer time was provided for me by the O.S.U. Computer Center Unsponsored Research Fund.

I wish to thank the people at Litton Guidance and Control for their gracious help in getting my thesis typed.

TABLE OF CONTENTS

TABLE OF CONTENTS (cont)

TABLE OF CONTENTS (cont)

## LIST OF TABLES

## NOMENCLATURE

| | |
|---|---|
| $A$ | system dynamics matrix |
| $a_y$ | side acceleration of center of mass per unit gravitational acceleration |
| $a_{ty}$ | side acceleration of tail per unit gravitational acceleration |
| $a_{ny}$ | side acceleration of nose per unit gravitational acceleration |
| $\delta a$ | aileron deflection |
| $B$ | control distribution matrix |
| $C_1$ | output scaling matrix for state |
| $C_2$ | measurement scaling matrix for state |
| $D_1$ | output scaling matrix for control |
| $D_2$ | measurement scaling matrix for control |
| $f_t$ | tail force |
| $G$ | filter gain |
| $H$ | Hamiltonian matrix |
| $I$ | expected value of the integrand of the quadratic performance criteria |
| $J$ | performance criteria |
| $K$ | control gain |
| $\ell$ | fuselage length |
| $n$ | order of the system |
| OP | operations count |
| $P$ | solution of the filter Riccati equation |
| $p$ | roll rate |
| $Q$ | similarity transformation matrix |

NOMENCLATURE (cont)

| | |
|---|---|
| $Q_1$ | weighting matrix of the output in performance criteria |
| $Q_2$ | power spectral density of process noise |
| $R_1$ | weighting matrix of the control in performance criteria |
| $R_2$ | power spectral density of measurement noise |
| $r$ | yaw rate |
| $\dot{r}$ | yaw acceleration |
| $\delta r$ | rudder deflection |
| $S$ | solution of the regulator Riccati equation |
| $U$ | control covariance matrix |
| $\tilde{U}$ | control error covariance |
| $u$ | the control variable |
| $u_y$ | lateral gradient of the longitudinal velocity |
| $\tilde{u}$ | control error |
| $v$ | lateral aircraft velocity |
| $v_o$ | lateral gust velocity |
| $v_x$ | longitudinal gradient of the lateral velocity |
| $w$ | process noise |
| $w_y$ | lateral gradient of normal gust velocity |
| $X$ | the state covariance matrix |
| $X*$ | the estimated state covariance matrix |
| $\tilde{X}$ | the state estimation error covariance matrix |
| $x$ | the system state |
| $x*$ | the estimated state |
| $\tilde{x}$ | the state estimation error |

## NOMENCLATURE (cont)

| | |
|---|---|
| $Y$ | the output covariance matrix |
| $\tilde{Y}$ | output error covariance |
| $y$ | the output |
| $\tilde{y}$ | output error |
| $z$ | the measurement |
| $\Gamma$ | process noise distribution matrix |
| $\Theta$ | measurement coupling matrix for process noise |
| $\lambda$ | eigenvalue |
| $\nu$ | measurement noise |
| $\tau$ | the correlation time factor of the measurement noise power spectral density |
| $\phi$ | roll angle |

APPLICATION OF LINEAR QUADRATIC CONTROL DESIGN IN REDUCTION OF AERODYNAMIC FORCES ON AIRCRAFT

# I. INTRODUCTION

In order to improve the stability and handling qualities for flight at low speed, the vertical tail of an aircraft is often designed to be larger than required for flight at higher cruise speeds. Because of this larger tail area, excessive gust loading can occur at higher flight speeds. The purpose of this report is to show the feasibility of designing an autopilot which utilizes the aileron and rudder in such a way as to reduce the gust loading on the vertical tail while maintaining acceptable levels of the lateral ride motion. The technique used involves a linear quadratic regulator accomplished with a stationary Kalman filter. The aircraft model is chosen to be a typical business jet for which the equation of motion of the aircraft was derived by the author in a previous report[1]. The equation of motion was linearized for the nominal flight condition of 450 knots (231 m/s) at 20,000 feet (6.1 km) altitude.

The required theoretical background of the linear quadratic regulator and the Kalman filter is presented in Chapter II. In order to solve the regulator and the filter problem, a Riccati equation has to be solved. An algorithm is developed for solving this equation and is described in Chapter III. Some of the numerical properties of the algorithm are also discussed in this chapter. A computer program implementing the proposed algorithm was written and a user's manual is presented in Appendix E. Also, the program listing is given in Appendix F.

Measurements are required in order to estimate the system state. Due to the fact that the sensors for the measurements are costly, it would be desirable to minimize the number of sensors required while maintaining acceptable information for state estimation. A new procedure for measurement elimination is developed and presented in Chapter IV. Also, a computer program is coded utilizing the measurement elimination procedure and is presented in Appendix G.

# II. THEORETICAL BACKGROUND

## II.1 Introduction

Optimal control is a technique used to determine the minimum or maximum of some performance criteria related to the performance of a dynamic system. A particular problem may concern minimizing the pertubation from a nominal trajectory of an aircraft, maximizing the flight range of a rocket, minimizing the fuel consumption of a vehicle, maximizing the profit in a business, or any of a vast variety of similar problems.

The fundamental problem of optimal control theory may be divided into four interrelated parts:

1. Definition of the desired goal.
2. Knowledge of our position with respect to the desired goal.
3. Knowledge of all environmental factors influencing the past, present, and future.
4. Determination of the optimal policy to achieve the desired goal from the knowledge stated in (2) and (3).

The desired goal is defined as the performance criteria which is to be minimized or maximized. For instance, the pertubation from the nominal trajectory of an aircraft is the performance criteria for one of the problems stated earlier. To solve an optimal control problem, the knowledge of the system and environmental factors influencing the system is translated into mathematical terms. This is called the system model. Realistically speaking, most physical systems are nonlinear. However, since it is rarely feasible to solve the optimal control problem for a nonlinear system of any practical significance, the development of explicit feedback control schemes for nonlinear systems is usually out of reach. In many cases, it is feasible to analyze small pertubations away from the nominal trajectory. When characteristics of the system do not significantly change with time, calculations for the linear time-invariant system can be applied. In this thesis, the linear time-invariant system model will be assumed.

A linear time-invariant system can be described by a set of first order differential equations

$$\begin{aligned} \dot{x}(t) &= A\, x(t) + B\, u(t) + \Gamma\, w(t) \\ y(t) &= C_1\, x(t) + D_1\, u(t) \qquad\qquad (2.1.1) \\ z(t) &= C_2\, x(t) + D_2\, u(t) + \nu(t) + \Theta\, w(t), \end{aligned}$$

where

$x(t)$ is the system state, $u(t)$ is the control, $w(t)$ is the process noise, $y(t)$ is the output, $z(t)$ is the measurement, $\nu(t)$ is the measurement noise. Note, all these quantities are vectors.

Equation (2.1.1) is called the state-variable description of the system.

To effectively control the behavior of a system, knowledge of the system state should be available. If perfect knowledge of the system state is at hand, and, if the performance criteria is chosen to be in quadratic form, a linear quadratic regulator can be designed to solve the state feedback control problem. In cases where perfect knowledge of the system state is not available, a Kalman filter can be used. The linear quadratic regulator combined with the Kalman filter form a stochastic control problem.

The focus of this chapter is classified into three problems:

1. The linear quadratic regulator,
2. The Kalman Filter,
3. The covariance properties.

These will be discussed sequentially in sections 2, 3, and 4 in this chapter.

## II.2 The Linear Quadratic Regulator

Consider a linear time-invariant system described by

$$\dot{x} = Ax + Bu + \Gamma w$$
$$y = C_1 x + D_1 u \tag{2.2.1}$$

where state $x(t)$ is an n dimensional vector, the control $u(t)$ is an $\ell$ dimensional vector, the process noise $w(t)$ is a p dimensional vector, and the output $y(t)$ is an m dimensional vector. Assume that the process noise $w(t)$ is zero-mean Gaussian white noise with non-negative definite power spectral density matrix $Q_2$, and assume that the initial conditions $x(t_o)$ and the process noise $w(t)$ are independent.

To design a linear quadratic regulator for the system described by equation (2.2.1), the performance criteria is chosen as the ensemble average of the quadratic form

$$J = E\left[\lim_{t_f \to \infty} \frac{1}{2} \int_{t_o}^{t_f} (y^T Q_1 y + u^T R_1 u)\, dt\right] \tag{2.2.2}$$

where $Q_1$, $R_1$ are positive-definite, constant matrices. The linear quadratic regulation problem is solved by minimizing the performance criteria, equation (2.2.2), subject to the constraints of the system equation described by equation (2.2.1).

It is shown by Åström (1970)[2] that the solution of the above linear quadratic regulator problem is the same as in the deterministic case when there is no process noise,

$$\min_{u} J = \lim_{t_f \to \infty} \frac{1}{2} \int_{t_o}^{t_f} (y^T Q_1 y + u^T R_1 u)\, dt \tag{2.2.3}$$

subject to the constraints of the system and the output equations

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= C_1 x + D_1 u. \end{aligned} \tag{2.2.4}$$

It is shown in Appendix A-1 that, by using Pontryagin's maximum principle[3], the solution of the deterministic linear quadratic regulator problem defined by equations (2.2.3) and (2.2.4) is given by

The state feedback control law

$$u = Kx \tag{2.2.5}$$

where the control gain K is given by

$$K = C_* - R_*^{-1} B^T S \tag{2.2.6}$$

and S satisfies the algebraic Riccati equation

$$0 = -SA_* - A_*^T S + SBR_*^{-1} B^T S - Q_*. \tag{2.2.7}$$

The matrices in equations (2.2.6) and (2.2.7) are given by the following equations

$$\begin{aligned}
R_* &= D_1^T Q_1 D_1 + R_1 \\
C_* &= -R_*^{-1} D_1^T Q_1 C_1 \\
A_* &= A + BC_* \\
Q_* &= C_1^T Q_1 C_1 - C_*^T R_* C_* .
\end{aligned} \tag{2.2.8}$$

Combining equation (2.2.1) and (2.2.5), the closed loop dynamic equation is

$$\dot{x} = (A + BK)\, x + \Gamma w \ . \tag{2.2.9}$$

## II.3 The Stationary Kalman Filter

Consider a linear time-invariant system described by

$$\dot{x} = Ax + Bu + \Gamma w$$

$$z = C_2 x + D_2 u + \nu + \Theta w \qquad (2.3.1)$$

where the state $x(t)$ is an n dimensional vector, the control $u(t)$ is a $\ell$-vector, the process noise $w(t)$ is a p-vector, and the measurement $z(t)$ is a q-vector. Assume that the process noise $w(t)$ and the measurement noise $\nu(t)$ are zero-mean Gaussian white noise with power spectral density matrices $Q_2$ and $R_2$ respectively. Furthermore, assume that the initial condition $x(t_o)$, the process noise $w(t)$ and the measurement noise $\nu(t)$ are independent.

To estimate the state, it is desired to maximize the probability of the state given the measurement (i.e. choose the most probable state with knowledge of the measurement). In this case, the performance criteria is given by

$$\underset{x(t)}{\mathrm{Max}} \; J_* = P[x(t) \mid z(\tau), \; \tau \leq t] \qquad (2.3.2)$$

When the probability density functions are Gaussian, maximizing the conditional probability of equation (2.3.2) is equivalent to minimizing the following criteria[4].

$$\underset{w,\nu}{\text{Min}}\ J = E\left[\lim_{t_o \to -\infty} \frac{1}{2}\int_{t_o}^{t_f} (w^T Q_2^{-1} w + \nu^T R_2^{-1} \nu)\,dt\right] \tag{2.3.3}$$

subject to the system dynamic constraints and that w and ν be causally related to z.

Many authors[4][5][6] have solved the problem defined by equations (2.3.1) and (2.3.3) when B, $D_2$, Θ are zero matrices. The case of non-zero, B, $D_2$ and Θ is discussed in Appendix A-2, where the solution of the stationary Kalman filter described by equation (2.3.1) and (2.3.3) is given.

The equation of the state estimate $x_*$ is

$$\dot{x}_* = Ax_* + Bu + G[-z + D_2 u + C_2 x_*] \quad . \tag{2.3.4}$$

The filter gain G is given by

$$G = -P\, C_2^T\, R_{2*}^{-1} - \Gamma Q_2\, \Theta^T\, R_{2*}^{-1} \tag{2.3.5}$$

where P satisfies the algebraic Riccati equation

$$0 = A_{**}P + PA_{**}^T + \Gamma Q_{2*}\Gamma^T - PC_2^T R_{2*}^{-1} C_2 P \,. \qquad (2.3.6)$$

The matrices in equations (2.3.5) and (2.3.6) are given by

$$R_{2*} = R_2 + \Theta\, Q_2\, \Theta^T$$

$$A_{**} = A - \Gamma Q_2\, \Theta^T\, R_{2*}^{-1}\, C_2 \qquad (2.3.7)$$

$$Q_{2*} = Q_2 - Q_2\, \Theta^T\, R_{2*}^{-1}\, \Theta Q_2 \,.$$

The equation of the estimated state $x_*$ can be rewritten as

$$\dot{x}_* = (A + GC_2)x_* + (B + GD_2)u - Gz \,. \qquad (2.3.8)$$

## II.4 Discussion of the Covariance Properties

The description of the covariance properties can be divided into four parts which are considered in the following sections.

1. The system with process noise and zero control

In this case, the system is described by

$$\dot{x} = Ax + \Gamma w \tag{2.4.1}$$

$$y = C_1 x \tag{2.4.2}$$

The state covariance matrix is defined by

$$X = E[x(t)\ x^T(t)] \ . \tag{2.4.3}$$

A well known result from stochastic control theory and described in Bryson and Ho (1969)[7] shows that X satisfies the following Lyapunov equation for the stationary case.

$$0 = AX + XA^T + \Gamma Q_2 \Gamma^T \ . \tag{2.4.4}$$

The rms state is the square root of each of the diagonal elements of X.

The output covariance matrix is defined by

$$Y = E[y(t)\ y^T(t)] \ . \qquad (2.4.5)$$

From equation (2.4.2), Y is given by

$$Y = E[C_1\ xx^T\ C_1{}^T]$$

$$= C_1\ E[xx^T]\ C_1{}^T$$

$$= C_1\ X\ C_1{}^T \ . \qquad (2.4.6)$$

2. The linear quadratic regulator problem with process noise.

The closed loop dynamic equation of a system with the state feedback control law, $u = Kx$, is given by

$$\dot{x} = (A + BK)\ x + \Gamma w \qquad (2.4.7)$$

$$y = (C_1 + D_1K)\ x \qquad (2.4.8)$$

where K is given by equation (2.2.6).

Comparing equations (2.4.7) and (2.4.8) with equation (2.4.1) and (2.4.2), we find:

a. The state covariance matrix X satisfies the following Lyapunov equation

$$0 = (A + BK)X + X(A + BK)^T + \Gamma Q_2 \Gamma^T . \qquad (2.4.9)$$

b. The output covariance matrix Y is given by

$$Y = (C_1 + D_1 K) X (C_1 + D_1 K)^T . \qquad (2.4.10)$$

c. The control covariance matrix, which is defined as $U = E[u(t)\ u^T(t)]$, is given by

$$\begin{aligned} U &= E[uu^T] \\ &= E[Kx\ x^T K^T] \\ &= K\ E[x\ x^T]K^T \\ &= K\ X\ K^T . \end{aligned} \qquad (2.4.11)$$

3. The stationary Kalman filter problem

The solution of this problem is described in section (II.3). The state estimation error is defined by

$$\tilde{x} = x - x_* \ , \qquad (2.4.12)$$

which is the difference between the actual state x and the estimated state $x_*$. The equation for the state estimation error can be obtained by subtracting equation (2.3.4) from (2.3.1). The resulting equation is given by

$$\dot{\tilde{x}} = A\tilde{x} + \Gamma w - G[-z + D_2u + C_2x_*] \ . \qquad (2.4.13)$$

Using the measurement equation

$$z = C_2x + D_2u + \nu + \Theta w \ ,$$

and the equation for $\tilde{x}$, the quantity $-z + D_2u + C_2x_*$ is given by

$$-z + D_2u + C_2x_* = -C_2\tilde{x} - \nu - \Theta w \ . \qquad (2.4.14)$$

Substituting equation (2.4.14) into (2.4.13), it yields

$$\dot{\tilde{x}} = (A + GC_2)\tilde{x} + \Gamma w + G\Theta w + G\nu \ . \qquad (2.4.15)$$

Using the equation of the filter gain G, the quantity $\Gamma w + G\,\Theta w + G\nu$ can be rearranged as

$$\Gamma w + G\,\Theta w + G\nu = \Gamma w_* + G_* \nu_* \tag{2.4.16}$$

where

$$w_* = w - Q_2\,\Theta^T R_{2*}^{-1} \nu_*$$

$$G_* = -P\,C_2^T R_{2*}^{-1} \tag{2.4.17}$$

$$\nu_* = \nu + \Theta w\ ,$$

the equation (2.4.15) of the state estimation error can then be written as

$$\dot{\tilde{x}} = (A + G\,C_2)\tilde{x} + \Gamma w_* + G_*\,\nu_*\ . \tag{2.4.18}$$

It is shown in Appendix A-2 that the power spectral densities of $w_*$ and $\nu_*$ are given by $Q_{2*}$ and $R_{2*}$ which are defined in equation (2.3.7). In addition, the assumption that $x(t_o)$, $w(t)$ and $\nu(t)$ are uncorrelated insures $x(t_o)$, $w_*(t)$ and $\nu_*(t)$ are also uncorrelated. Comparing equation (2.4.18) with (2.4.1) and using the uncorrelated property of $x(t_o)$, $w_*(t)$ and $\nu_*(t)$, the covariance matrix of the stationary state estimation error $\tilde{X}$ satisfies the following Lyapunov equation

$$0 = (A + G\,C_2)\,\tilde{X} + \tilde{X}(A + G\,C_2)^T + \Gamma Q_{2*}\,\Gamma^T + G_*\,R_{2*}\,G_*^T \tag{2.4.19}$$

where

$$Q_{2*} = Q_2 - Q_2\,\Theta^T\,R_{2*}^{-1}\,\Theta\,Q_2$$

$$G_* = -P\,C_2^T\,R_{2*}^{-1} \tag{2.4.20}$$

$$R_{2*} = R_2 + \Theta\,Q_2\,\Theta^T\,.$$

To prove $\tilde{X}$ is equal to P, the quantity E is first defined:

$$E = P - \tilde{X} \tag{2.4.21}$$

where P is the solution of the algebraic Riccati equation (2.3.6). Subtracting equation (2.4.19) from (2.3.6), making use of equations (2.4.20) and

(2.4.20) and (2.4.21), the quantity E satisfies the following Lyapunov equation

$$0 = (A + G\,C_2)E + E(A + G\,C_2)^T \ . \qquad (2.4.22)$$

It was shown by Kalman[8] in 1960 that, if the system is controllable and observable, the eigenvalues of $A + G\,C_2$ are all in the open left-half plane. With this eigenvalue property, it was shown by Rutherford[9] in 1932 that the unique solution E of equation (2.4.22) is equal to zero. Thus, the covariance of the state estimation error is equal to the solution of the algebraic Riccati equation (2.3.7), i.e.,

$$\widetilde{X} = P \ . \qquad (2.4.23)$$

## 4. The stochastic control problem

In cases when perfect knowledge of the state is not available, a Kalman filter can be used to estimate the state. This estimated state is then used in the state feedback control law for the regulator. Wonham[10] shows that the problem of filtering and control can be treated independently in some cases. The result is called the separation theorem. According to this theorem, the solution of the stochastic control problem is given by the solution of the linear regulator and the solution of the Kalman filter with the state feedback control law.

$$u = K x_* \tag{2.4.24}$$

where $x_*$ is the estimated state.

These solutions for the linear regulator and the Kalman filter were given in the previous two sections.

The covariance of the estimated state is defined by

$$X_* = E[x_* \, x_*^T] \ . \tag{2.4.25}$$

It is shown in Appendix A-2 that $x_*$ and $\tilde{x}$ are uncorrelated, i.e.,

$$E[x_* \, \tilde{x}^T] = 0 \ . \tag{2.4.26}$$

Using equations (2.4.26) and (2.3.4) and the previous steady covariance results, the estimated state covariance satisfies the following Lyapunov equation

$$0 = (A + BK)\, X_* + X_*(A + BK)^T + GR_{2*}\, G^T \tag{2.4.27}$$

where $R_{2*} = R_2 + \Theta Q_2\, \Theta^T$ .

The covariance of the actual system state is defined by

$$X = E[x\, x^T] \ . \tag{2.4.28}$$

Using equations (2.4.12), (2.4.23), (2.4.26), the covariance of the state is given by the sum of the covariance of the estimated state and the covariance of the state estimation error

$$X = X_* + P \tag{2.4.29}$$

where $X_*$ satisfies equation (2.4.27) and
$P$ satisfies equation (2.3.7).

Using the equation (2.4.24) of the control, the control covariance matrix is given by

$$U = K\, X_*\, K^T \ . \tag{2.4.30}$$

Substituting the equation (2.4.24), into the output equation yields

$$y = (C_1 + D_1K)\ x_* + C_1\ \tilde{x}\ . \qquad (2.4.31)$$

Since $x_*$ and $\tilde{x}$ are uncorrelated, the output covariance matrix is given by

$$Y = (C_1 + D_1K)\ X_*\ (C_1 + D_1K)^T + C_1PC_1^T\ . \qquad (2.4.32)$$

# III AN IMPROVED ALGORITHM FOR SOLVING THE ALGEBRAIC RICCATI EQUATION

## III.1 Introduction

From the discussion in Chapter II, it can be seen that, in order to solve the linear quadratic regulator and the stationary Kalman filter problem, the algebraic Riccati equations (2.2.7) and (2.3.6) have to be solved. Many authors have suggested methods[11][12][13] to solve these equations. One of the methods that has been most successful is the eigenvector decomposition method proposed by MacFarlane[11] in 1963 and by Potter[12] in 1966. In this method, the eigenvalues and the corresponding eigenvectors of the Hamiltonian matrix are determined. The Hamiltonian matrix is the coefficient matrix of the Euler-Lagrange system discussed in Appendices A-1 and A-2. The eigenvectors of the Hamiltonian matrix associated with eigenvalues whose real parts are all of the same sign are partitioned into the form of

$$T = \begin{bmatrix} X_+ & X_- \\ \Lambda_+ & \Lambda_- \end{bmatrix} = \begin{bmatrix} T_+ & T_- \end{bmatrix} \tag{3.1.1}$$

where

$$T_+ = \begin{bmatrix} X_+ \\ \Lambda_+ \end{bmatrix} \quad \text{and} \quad T_- = \begin{bmatrix} X_- \\ \Lambda_- \end{bmatrix}$$

are eigenvectors associated with eigenvalues with positive real parts and negative real parts respectively. In the case that the system is not controllable or not observable, then eigenvalues with zero real parts may exist, and the algorithm can fail. Using two of the four submatrices in equation (3.1.1), one forms a set of linear equations whose solution yields the solution of the corresponding algebraic Riccati equation. The choice of which two of the four submatrices depends solely on which Riccati equation is to be solved.

The success of this method requires that the partitioned eigenvector matrices be non-singular. However, as was discussed by Holley[14] and Wei in 1979, the resulting matrices may be singular when one or more of the eigenvalues are repeated. This difficulty can be overcome by using the generalized eigenvectors, which, however, is not an entirely satistactory method. In cases when two eigenvalues are nearly equal, the partitioned eigenvector matrices, while not singular, remain ill conditioned. This can lead to errors in the computed solution. Small perturbations in the system matrix elements can lead to drastic changes in the partitioned eigenvector matrices, which also causes poor numerical stability.

An improved algorithm for solving the algebraic Riccati equation[14] is presented in this chapter. In this algorithm, the Hamiltonian matrix is transformed into quasi-upper triangular form by an orthogonal similarity transformations. This transformation is chosen so that the resulting quasi-upper triangular matrix has all the eigenvalues with positive real parts in the lower right hand corner when the regulator problem is solved, and all the eigenvalues with negative real parts in this corner when the filter problem is solved. The quasi-upper triangularization is accomplished by using the stable QR algorithm with implicit double shifts[15][16]. As in the eigenvector decomposition method mentioned above, the resulting orthogonal similarity transformation matrix is partitioned into four n by n matrices. Using two of the four submatrices, one forms a set of linear equations whose solution yields the solution of the corresponding algebraic Riccati equation.

In Section III.2, the proposed algorithm and its theoretical basis are presented in more detail. An error bound for the orthogonal similarity transformation to quasi-upper triangular form and the operations count for solving the Riccati equation are discussed in Section III.3. A computer program which solves the optimal control problem presented in Chapter II by using the proposed algorithm was coded and is listed in Appendix F. The user's manual for the program is presented in Appendix E.

## III.2 The Algorithm and its Theoretical Basis

The proposed algorithm for solving the algebraic Riccati equation has three major steps:

1. Setting up the Hamiltonian matrix of the corresponding algebraic Riccati equation,
2. From the Hamiltonian matrix, determine an orthogonal basis of the invariant subspace(17) associated with the desired eigenvalues, and
3. The symmetric, non-negative definite solution of the Riccati equation is found by solving a set of linear equations involving the orthogonal basis vectors of the invariant subspace.

These steps will be described sequentially in this section.

As discussued in Chapter II, the algebraic Riccati equation of the linear quadratic regulator is given by equation (2.2.7) which is rewritten here

$$0 = -SA_* - A_*^T S + S\, BR_*^{-1}\, B^T\, S - Q_* \ . \qquad (3.2.1)$$

As described in Appendix A-1, the corresponding Euler-Lagrange system has the form

$$\begin{bmatrix} \dot{x} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} A_* & -BR_*^{-1} B^T \\ -Q_* & -A_*^T \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} \tag{3.2.2}$$

where $x$ is the system state and $\lambda$ is the Lagrange multiplier resulting from the application of the Pontryagin maximum principle

The Hamiltonian matrix is the coefficient matrix of the Euler-Lagrange system and, in this case, is given by

$$H = \begin{bmatrix} A_* & -BR_*^{-1} B^T \\ -Q_* & -A_*^T \end{bmatrix} . \tag{3.2.3}$$

Matrices $A_*$, $R_*$, $Q_*$ etc. are defined in Chapter II.

The Hamiltonian matrix for the stationary Kalman filter is similarly defined. The Riccati equation of the Kalman filter is given by equation (2.3.6), rewritten here as

$$0 = A_{**} P + PA_{**}^T + \Gamma Q_{2*} \Gamma^T - P C_2^T R_{2*}^{-1} C_2 P . \tag{3.2.4}$$

It is shown in Appendix A-2 that the corresponding Euler-Lagrange system is given by

$$\begin{bmatrix} \dot{x} \\ \\ -\dot{\lambda} \end{bmatrix} = \begin{bmatrix} A_{**} & \Gamma Q_{2*}\Gamma^T \\ \\ C_2{}^T R_{2*}{}^{-1} C_2 & -A_{**}{}^T \end{bmatrix} \begin{bmatrix} x \\ \\ -\lambda \end{bmatrix} + \begin{bmatrix} B_* & L \\ \\ C_2{}^T R_{2*}{}^{-1} D_2 & -C_2{}^T R_{2*}{}^{-1} \end{bmatrix} \begin{bmatrix} u \\ \\ z \end{bmatrix} \tag{3.2.5}$$

where $A_{**}$, $Q_{2*}$, $R_{2*}$ are defined in Chapter II and $B_*$, L are defined in Appendix A-2.

The Hamiltonian matrix in this case is given by

$$H = \begin{bmatrix} A_{**} & \Gamma Q_{2*}\ \Gamma^T \\ \\ C_2{}^T R_{2*}{}^{-1} C_2 & -A_{**}{}^T \end{bmatrix} . \tag{3.2.6}$$

It is shown in Appendix B-1 that if $Q_*$ and $R_*$ are symmetric, the eigenvalues of the Hamiltonian matrix defined in equation (3.2.3) are symmetric with respect to the imaginary axis in the eigenvalue plane, (i.e., if s is an eigenvalue of H in equation (3.2.3), then, -s is also an eigenvalue). A similar result follows for the Hamiltonian matrix defined in equation (3.2.6) if $Q_{2*}$ and $R_{2*}$ are symmetric matrices.

It is also shown in Appendix B-1 that, for a controllable and observable system, the solution of the control Riccati equation (3.2.1) satisfies the relation

$$Q_{11}\, S = Q_{12} \quad , \tag{3.2.7}$$

where $Q_{11}$ and $Q_{12}$ are submatrices of the orthogonal similarity transformation matrix

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \tag{3.2.8}$$

which transforms the Hamiltonian matrix of equation (3.2.3) into the form

$$\begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \begin{bmatrix} A_* & -BR_*^{-1}B^T \\ -Q_* & -A_*^T \end{bmatrix} \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}^T \tag{3.2.9}$$

with $U_{22}$ having all eigenvalues in the open right half plane.

The solution of the filter Riccati equation (3.2.4) satisfies the following equation

$$Q_{12}\, P = Q_{11} \tag{3.2.10}$$

where $Q_{11}$ and $Q_{12}$ are also submatrices of the orthogonal similarity transformation matrix

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \tag{3.2.11}$$

which transforms the Hamiltonian matrix of equation (3.2.6) into the form

$$\begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \begin{bmatrix} A_{**} & \Gamma Q_* \Gamma^T \\ C_2{}^T R_{2*}{}^{-1} C_2 & -A_{**}{}^T \end{bmatrix} \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}^T \tag{3.2.12}$$

With $U_{22}$ having all eigenvalues in the open left half plane.

The similarity transformation of equations (3.2.9) and (3.2.12) has the following form

$$\begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} H \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}^T . \tag{3.2.13}$$

Since Q is orthogonal, premultiplication by $Q^T$ on both sides of equation (3.2.13) yields

$$H \begin{bmatrix} Q_{11}^T & Q_{21}^T \\ Q_{12}^T & Q_{22}^T \end{bmatrix} = \begin{bmatrix} Q_{11}^T & Q_{21}^T \\ Q_{12}^T & Q_{22}^T \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} . \tag{3.2.14}$$

The following equation results

$$H \begin{bmatrix} Q_{11}^T \\ Q_{12}^T \end{bmatrix} = \begin{bmatrix} Q_{11}^T & U_{11} \\ Q_{12}^T & U_{11} \end{bmatrix} = \begin{bmatrix} Q_{11}^T \\ Q_{12}^T \end{bmatrix} U_{11} . \tag{3.2.15}$$

Equation (3.2.15) shows that the linear operator, represented by the matrix H, transforms the first n rows of Q into linear combinations of themselves. This shows that the first n rows of Q

$$[Q_{11} \quad Q_{12}] \tag{3.2.16}$$

form a set of orthogonal basis vectors for the invariant subspace, which is associated with n eigenvalues of submatrix $U_{11}$ in equations (3.2.9) and (3.2.12).

The highly stable QR algorithm with implicit double shifts of origin is employed to determine the orthogonal similarity transformation matrix Q. Wilkinson and Reinsch(15)(16) point out that for the QR algorithm, the volume of work is greatly reduced if the matrix H is first transformed to upper-Hessenberg form (i.e., to a matrix H' such that $h'_{ij} = 0$ for $i > j+1$). The transformation may be accomplished in a stable manner by the use of Householder type orthogonal similarity transformation matrices(15)(16). The transformation by orthogonal matrices takes place in n-2 major steps. Immediately before the r-th step H has been reduced to $H_r$ which is of upper-Hessenberg form in its first r-1 columns. The matrix $H_{r+1}$ is derived from $H_r$ via the relation

$$H_{r+1} = Q_r \, H_r \, Q_r \quad . \tag{3.2.17}$$

The orthogonal matrix $Q_r$ is of the form

$$Q_r = I - u_r u_r^T / \beta_r \tag{3.2.18}$$

where

$$u_r^T = [0, ---, 0, \; h^{(r)}_{r+1,r} \pm \sigma_r^{1/2} \; , \; h^{(r)}_{r+2,r}, \ldots h^{(r)}_{n,r}]$$

$$\sigma_r = \sum_{i=1}^{n} (h^{(r)}_{r+i,r})^2 \tag{3.2.19}$$

$$\beta_r = \sigma_r \pm h^{(r)}_{r+1,r} \, \sigma_r^{1/2} \quad .$$

This transformation leaves the zeros already produced in the first r-1 columns of $H_r$ unaffected.

Given the upper-Hessenberg matrix H', the QR algorithm is used to accomplish the quasi-triangularization which was described in equations (3.2.9) and (3.2.12). For iteration s of the QR algorithm, we have

$$H_{s+2} = Q_{s+1}\, Q_s\, H_s\, Q_s^T\, Q_{s+1}^T \tag{3.2.20}$$

giving

$$H_s\, (Q_s^T\, Q_{s+1}^T) = (Q_s^T\, Q_{s+1}^T) H_{s+2} \tag{3.2.21}$$

and

$$(Q_s^T\, Q_{s+1}^T)(R_{s+1}\, R_s) = (H_s - k_s I)(H_s - k_{s+1} I). \tag{3.2.22}$$

writing

$$Q_{s+1}\, Q_s = T \quad , \quad R_{s+1}\, R_s = N,$$

$$\text{and} \qquad (H_s - k_s I)(H_s - k_{s+1}\, I) = M\,, \tag{3.2.23}$$

we have

$$H_s\, T^T = T^T\, H_{s+2}\,,\ N = TM, \tag{3.2.24}$$

where

$k_s$ and $k_{s+1}$ are shifts of origin and

N is an upper triangular matrix (since $R_s$ and $R_{s+1}$ are upper triangular).

Householder[18] demonstrated in 1958 that any real matrix can be triangularized by successive premultiplication with $Q_1, Q_2, \ldots Q_{n-1}$, where $Q_i$ are of the form

$$Q_i = I - 2\, w_i\, w_i^T \tag{3.2.25}$$

and $w_i$ is a unit vector with zeros for its first i-1 components.

Since the first row of the matrix $Q_{n-1}\, Q_{n-2} \ldots Q_2 Q_1$ is the first row of $Q_1$ itself, it was pointed out by Francis[19] in 1961 that in triangularizing any matrix the first factor, $Q_1$, is determined only by the first column of that matrix. The first column of M is of the form

$$(p_1,\ q_1,\ r_1,\ 0,\ \ldots\ ,0)^T \tag{3.2.26}$$

where

$$\begin{aligned} p_1 &= h_{11}^2 - h_{11}(k_s + k_{s+1}) + k_s\, k_{s+1} + h_{12}\, h_{21} \\ q_1 &= h_{21}(h_{11} + h_{22} - k_s - k_{s+1}) \qquad (3.2.27) \\ r_1 &= h_{32}\, h_{21}\ , \end{aligned}$$

$h_{ij}$ is the (i,j) element of the matrix $H_s$ and
$k_s$, $k_{s+1}$ are either real or are complex conjugates giving real values for $p_1$, $q_1$, and $r_1$.

Because of these properties, the vector $w_1$ associated with $Q_1$ has only three non-zero elements, giving the matrix $Q_1 H_s Q_1^T$ a maximum of three non-zero elements below the diagonal element. The matrix with three non-zero elements below the diagonal is then transformed back to an upper Hessenberg matrix by using the algorithm for initial Hessenberg reduction, which is described in equations (3.2.17) to (3.2.19). Francis[19] shows that this matrix is the same as $H_{s+2}$ in the QR algorithm. A subroutine called HESS has been coded for the upper Hessenberg reduction. Since the subroutine HESS is necessary for the first Hessenberg reduction, the transformation $Q_1 H_s Q_1^T$ described above is coded as a separate subroutine called SHIFT2. The resulting matrix from subroutine SHIFT2, which has three non-zero elements below the diagonal, is transformed to the Hessenberg matrix by again using the subroutine HESS. This subroutine was designed to take advantage of zero subdiagonal elements. Subroutines HESS and SHIFT2 are described in Appendix E and listed in Appendix F.

In order to isolate eigenvalues with proper signs in the $U_{22}$ matrix in equations (3.2.9) and (3.2.12), several possible shifting strategies for choosing $k_s$ and $k_{s+1}$ were tried. Experiments indicated that choosing $k_s$ and $k_{s+1}$ to be the desired eigenvalues resulted in a stable and efficient method. This results from the fact that the QR algorithm isolates zero eigenvalues in one step[15]. The eigenvalues of proper sign are chosen from the full set of eigenvalues determined by a complete quasi-triangularization of H with arbitrary ordering. This results in a two step process: first, the eigenvalues of H are computed by a full quasi-triangularization, and second, this knowledge is used to isolate eigenvalues with proper sign which involves only a partial quasi-triangularization.

It may happen during the course of the iteration that a matrix $H_s$ has one or more sufficiently small (to be regarded as zero) sub-diagonal elements in an undesirable position. This split of $H_s$ may cause the isolation of the desirable eigenvalue to be impossible. In this case, an arbitrary Householder type similarity transformation is performed to remove the undesired zero on the sub-diagonal elements. The Householder type matrix is chosen to be in the form

$$Q = \begin{bmatrix} Q_1 & 0 \\ 0 & I \end{bmatrix}$$

where

$$Q_1 = I - \frac{uu^T}{\beta} \quad , \quad u^T = [1,\ 1,\ ---,1], \text{ and } \beta = \frac{u^T u}{2} .$$

The similarity transformation for removing the undesirable zero on the sub-diagonal is performed according to the relation

$$H_s' = Q\ H_s\ Q$$

$$= \begin{bmatrix} Q_1 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} H_1 & H_2 \\ 0 & H_3 \end{bmatrix} \begin{bmatrix} Q_1 & 0 \\ 0 & I \end{bmatrix}$$

$$= \begin{bmatrix} Q_1\ H_1\ Q_1 & Q_1\ H_2 \\ 0 & H_3 \end{bmatrix}$$

where $H_3$ is the submatrix with desired eigenvalues previously isolated, and $H_1$ is the submatrix with the undesirable zero on the sub-diagonal. The submatrix $Q_1\ H_1\ Q_1$ of the resulting matrix $H_s'$ is no longer in upper Hessenberg form, allowing use of subroutine HESS to reduce it to an upper Hessenberg matrix. With the resulting matrix, the QR algorithm is then used to continue the isolation of the desired eigenvalues.

The Householder type reduction is also employed to solve the linear equations given in (3.2.7) and (3.2.10). In general, the method can be described as follows:

Given a linear system

$$AX = B, \tag{3.2.28}$$

a sequence of Householder-type matrices is found such that the product Q satisfies

$$QAX = QB \tag{3.2.29}$$

where $T = QA$ is an upper triangular matrix. Back substitution is performed to solve the equation

$$TX = Y \tag{3.2.30}$$

where $Y = QB$.

Once the solution of the Riccati equation found, the control gains, the closed loop dynamics matrix etc. can be found by simple matrix operations.

In order to solve the covariance equations discussed in Chapter II, a Lyapunov type equation

$$AX + XA^T = C \tag{3.2.31}$$

has to be solved. Bartels and Stewart[20] proposed a method for solving the above Lyapunov equation in 1972. In this method, the matrix A is first transformed to a quasi-upper triangular matrix by using the QR algorithm stated above. The resulting matrix satisfies the following equation

$$UY + YU^T = T \tag{3.2.32}$$

where

$U = QAQ^T$ is a quasi-upper triangular matrix,
$Y = QXQ^T$, and $T = QCQ^T$.

Equation (3.2.32) can be written as

$$\begin{bmatrix} U_1 & U_2 \\ 0 & U_3 \end{bmatrix} \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix} + \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix} \begin{bmatrix} U_1^T & 0 \\ U_2^T & U_3^T \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} \\ T_{12}^T & T_{22} \end{bmatrix} \tag{3.2.33}$$

where $U_3$ is a one-by-one or a two-by-two block.

Multiplying the partitioned matrices yields

$$U_1 Y_{11} + Y_{11} U_1^T + U_2 Y_{12}^T + Y_{12} U_2^T = T_{11} \tag{3.2.34}$$

$$U_1 Y_{12} + U_2 Y_{22} + Y_{12} U_3^T = T_{12} \tag{3.2.35}$$

$$U_3\, Y_{22} + Y_{22}\, U_3^{T} = T_{22} \quad . \tag{3.2.36}$$

Since U is in quasi-upper triangular form, the submatrix $U_3$ is either a one-by-one or a two-by-two matrix. If $U_3$ is a one-by-one matrix, equation (3.2.36) is a scalar equation which can be solved immediately. If $U_3$ is a two-by-two matrix, equation (3.2.36) can be transformed into a standard linear system of equations of order four or less in the form of equation (3.2.28). The linear equation can then be solved by using the Householder type reduction which is described by equations (3.2.28) to (3.2.30). The solution $Y_{22}$ obtained from equation (3.2.36) is back substituted into equation (3.2.35). Since $U_1$ is also in the quasi-upper triangular form, equation (3.2.35) can be broken down into sub-equations similar to equations (3.2.34), (3.2.35), and (3.2.36). The back substitution method of matrix blocks described above is again employed to solve equation (3.2.35). The solution $Y_{12}$ obtained from equation (3.2.36) is then back substituted into equation (3.2.34). Now, equation (3.2.34) is a reduced order Lyapunov equation with $U_1$ in the quasi-upper triangular form, the back substitution method of matrix blocks can then be used to solve equation (3.2.34). The solution of the Lyapunov equation (3.2.31) is then obtained by the relation

$$X = Q^T Y Q \qquad (3.2.37)$$

where Y is the solution of equation (3.2.32).

It is shown in Appendix B-1 that, if the eigenvalues of A are all in the open left half plane, and if the C matrix is negative definite, the solution of Lyapunov equation (3.2.31) is a positive definite matrix.

## III.3 Numerical Properites of the Quasi-triangularization and the Operations Count for Solving the Riccati Equation

The quasi-upper triangularization described in the previous section can, in general, be written in the form

$$H_s = G_1 \; H \; G_1^T \tag{3.3.1}$$

where $G_1$ is the product of a sequence of Householder type orthogonal similarity transformation matrices, H is the original Hamiltonian matrix, $H_s$ is the transformed matrix which is of the form

$$H_s = \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} . \tag{3.3.2}$$

It is shown in Appendix C-1 that the computed matrix $\overline{H}_s$ satisfies the following equation

$$\overline{H}_s = G_1 \; (E + H) \; G_1^T \tag{3.3.3}$$

where $\overline{H}_s$ is the computed $H_s$ and E is a perturbation in H.

It is also shown in Appendix C-1 that the perturbation E, for a machine with a t-digit mantissa, is bounded by

$$||E||_2 \leq 2^{-t} \; ||H||_2 \; \{ \; 8 \; k \; n^3 + (3.82 + 4 \; k)n^2 + 44.5 \; n$$

$$+ 50.9 + (4 \; k \; n^2 + 50.9)s \; \} \qquad (3.3.4)$$

where $k = 2.12$, $||\cdot||_2$ is the $L_2$ norm of the matrix and s is, in most cases, between $3/2 \; n^2$ and $(3n^2 - n)$.

It should be pointed out that, since the transformation removing undesired zeros on the sub-diagonal is unnecessary in most cases, error caused by this transformation was not included in equation (3.3.4). It can be seen from equation (3.3.4) that an $n^4$ term dominates the bound of the perturbation E. In order to have a quantitative feeling for the above result, assume the order of the system, n, is equal to 100 and the factors k and s to be the larger values

$$k = 2.12$$
$$s = 3 \; n^2 - n \; . \qquad (3.3.5)$$

For the CDC CYBER machine, $t = 48$, and the bound given in equation (3.3.4) is

$$||E||_2 \leq (.91 \times 10^{-5}) \; ||H||_2 \; . \qquad (3.3.6)$$

Thus, the computation is reasonably accurate even for such a large-scale system.

An error bound for the computed similarity transformation, $\bar{G}_1$, can be derived as follows. First of all, a supporting lemma for a derivation of the error bound, which is proved in Appendix C, is stated as:

Lemma 3.1.

The floating point computation, fl[ ], of matrix multiplication is given by

$$fl(A_1 A_2 \;....\; A_s) = A_1 A_2 \;....\; A_s + F \qquad (3.3.7)$$

where

$$||F||_2 \leq 1.06\ (s-1)n^2\ 2^{-t} \left[ \prod_{j=1}^{s} ||A_j||_2 \right] \qquad (3.3.8)$$

With this lemma, the error bound of $\bar{G}_1$ can then be derived. $G_1$ has the form

$$G_1 = Q_s \;....\; Q_1 \qquad (3.3.9)$$

where $Q_i$ are Householder type matrices and s is the number of orthogonal matrices required to accomplish the quasi-triangularization described in equation (3.3.1).

The computed $G_1$ can be written as

$$\overline{G}_1 = \overline{Q}_s \;....\; \overline{Q}_1 \tag{3.3.10}$$

where $\overline{Q}_i = Q_i + e_i$ is $Q_i$ with perturbation $e_i$.

With floating-point arithmetic, the computed matrix $\overline{G}_1$ is given by

$$\overline{G}_1 = fl(\overline{Q}_s \;....\; \overline{Q}_1)$$

$$= Q_s \;....\; Q_1 + e_{s+1}$$

$$= (Q_s + e_s)(Q_{s-1} + e_{s-1}) \;....\; (Q_1 + e_1) + e_{s+1}$$

$$= Q_s \;...\; Q_1 + \sum_{m=1}^{s} \left\{ \sum_{1 \le i_1 < ... < i_m \le s} (e_{i_m} \cdots e_{i_1}) \prod_{j \notin \{i_1 ... i_m\}} Q_j \right\} + e_{s+1} \tag{3.3.11}$$

where $e_{s+1}$ is the error in matrix multiplication.

Using equations (3.3.7) and (3.3.8), the bound of $e_{s+1}$ in equation (3.3.11) is given by

$$||e_{s+1}||_2 \leq (1.06)(s-1)\ n^2\ 2^{-t} \left[ \prod_{i=1}^{s} ||\overline{Q_i}||_2 \right]. \qquad (3.3.12)$$

Substituting the equation

$$||\overline{Q}_i||_2 \leq ||Q_i||_2 + ||e||_2 = 1 + ||e||_2 \quad ,$$

where the fact $||Q_i||_2 = 1$ is used.

Since $Q_i$ is an orthogonal matrix, equation (3.3.12) becomes

$$||e_{s+1}||_2 \leq (1.06)(s-1)n^2\ 2^{-t} \left[ \prod_{i=1}^{s} (1 + ||e_i||_2) \right] . \qquad (3.3.13)$$

It is shown in appendix C-1 that $e_i$ satisfies the inequality

$$||e_i||_2 \leq ||e||_2 \leq (4.8\ n + 11.2)\ 2^{-t} . \qquad (3.3.14)$$

Using equation (3.3.14) and the relation

$$\begin{aligned} \prod_{i=1}^{s} (1 + ||e_i||_2) &\leq (1 + ||e||_2)^s \\ &\leq \exp\ [||e||_2 s] \\ &\leq 1 + 1.01\ s\ ||e||_2\ , \end{aligned}$$

if $s||e||_2 < .01$, equation (3.3.13) becomes

$$||e_{s+1}||_2 \le (s-1)n^2\ (1.06)2^{-t}\ (1 + 1.01\ s||e||_2) \quad (3.3.15)$$

where the higher order term is neglected and

$$||e||_2 \le (4.8\ n + 11.2)2^{-t} \quad .$$

The $L_2$ norm of the summation term in equation (3.3.11) can be simplified as follows:

Using equation (3.3.14) and $||q_i|| = 1$, the quantity

$$\left|\left|\left\{\sum_{m=1}^{s} \sum_{1\le i_1<\ldots<i_m\le s} (e_{i_m}\cdots e_{i_1}) \prod_{j\notin\{i_1\ldots i_m\}} q_i\right\}\right|\right|$$

$$\le \sum_{m=1}^{s}\binom{s}{m} ||e||_2{}^{m}$$

$$= -1 + \sum_{m=0}^{s}\binom{s}{m} ||e||_2^{m}$$

$$= -1 + (1 + ||e||_2)^{s}$$

$$\le -1 + 1 + 1.01\ s||e||_2\ ,\ \text{if}\ s||e||_2 \le .01, \qquad (3.3.17)$$

where

$$\binom{s}{m} = \frac{s!}{m!\ (s-m)!} \quad . \tag{3.3.18}$$

Substituting equations (3.3.14), (3.3.15) and (3.3.17) into (3.3.11), the resulting equation is given by

$$\begin{aligned} ||e_G||_2 &= ||\bar{G}_1 - G_1||_2 \\ &\leq 1.01\ s||e||_2 \quad ||e_{s+1}||_2 \\ &\leq (s-1)n^2\ (1.06)2^{-t} \\ &\quad + 1.01[s(s-1)n^2\ (1.06)2^{-t} + s]\ (4.8\ n \\ &\quad + 11.2)\ 2^{-t} \quad . \end{aligned} \tag{3.3.19}$$

If $s = 3n^2 - n$, $n = 100$, and $t = 48$, the above bound is

$$||e_G||_2 \leq 1.2 \times 10^{-6}. \tag{3.3.20}$$

The accuracy of the computed orthogonal matrix is acceptable even for a large scale system with n = 100.

One way to express the efficiency of an algorithm is by an operations count; the number of arithmetic operations required for the algorithm. Since the computational time for multiplication and division is usually much larger than for addition and substraction, it is customary to count only the number of multiplications and divisions. The operations count of multiplications and divisions for solving the algebraic Riccati equation is now determined. This operations count has three major parts: (1) OP for finding the eigenvalues of the Hamiltonian matrix, (2) OP for isolating the n desired eigenvalues on the lower right hand corner of the transformed Hamiltonian matrix, and (3) OP for solving the linear equations formed by the partitioned matrix of the orthogonal similarity transformation matrix. It is shown in Appendix C-2 that the total OP required for solving the algebraic Riccati equation using the algorithm described in the previous section is usually between limits

$$OP_{max} = 186.2\ n^3 + 81\ n^2 - 56.2\ n - 67$$

and (3.3.21)

$$OP_{min} = 186.2\ n^3 + 57\ n^2 - 83.2\ n - 41$$

where n is the order of the system.

In the case n = 100,

$$OP_{max} = 1.870 \times 10^8$$

$$OP_{min} = 1.868 \times 10^8 .$$

# IV ELIMINATION OF UNNECESSARY MEASUREMENTS

## IV.1 Introduction

State estimation accuracy is highly dependent on which measurements are used. When more measurements are available, the estimation of the state will be more accurate. These measurements of the system outputs are gathered by sensors. However, due to sensor cost, it is desirable to minimize the number of sensors required.

In the following section, possible criteria for measurement elimination are discussed, and an elimination procedure is described. At the end of this chapter, the proposed procedure is applied to a small jet aircraft control problem. A computer program was developed for the application of the proposed general procedure and is presented in Appendix G.

## IV.2 The Measurement Elimination Criteria

The system model is assumed to be linear and time-invariant with the state equation

$$\dot{x} = Ax + Bu + \Gamma w \;. \tag{4.2.1}$$

The controlled outputs and measurements satisfy

$$y = C_1 x + D_1 u \tag{4.2.2}$$

$$z = C_2 x + D_2 u + \nu + \Theta w \;, \tag{4.2.3}$$

where z represents the vector of sensor measurements.

The criteria for measurement elimination depends on the purpose of the state estimation. For instance, when state estimation alone is desired, the accuracy of the estimated state is the obvious criterion. However, if the estimated state is used in state feedback control, the accuracy of the control variable is more important. When the objective of the controller is to minimize the quadratic performance

$$J = \frac{1}{2} E\left[ \int_0^\infty (y^T Q_1 y + u^T R_1 u)\,dt \right], \tag{4.2.4}$$

the stationary expected value of the integrand

$$E[y^TQ_1y + u^TR_1u] \tag{4.2.5}$$

is another possible criterion. If the output y is the variable of interest, the output could also be a criterion. Therefore, four criteria for measurement elimination are discussed.

a. The rms state estimation error

As discussed in Chapter II, the state estimation error satisfies equation (2.4.15) which is rewritten here

$$\dot{\tilde{x}} = (A + GC_2)\tilde{x} + \Gamma w + G\Theta w + G\nu \tag{4.2.6}$$

where G is the filter gain and is given in equation (2.3.5). Also shown in Chapter II is the covariance of the state estimation error, given by

$$\begin{aligned} \tilde{X} &= E[\tilde{x}\,\tilde{x}^T] \\ &= P \end{aligned} \tag{4.2.7}$$

where P satisfies the algebraic Riccati equation

$$0 = A_{**}P + PA_{**}^T + \Gamma Q_{2*}\Gamma^T - PC_2^TR_{2*}^{-1}C_2P \ , \tag{4.2.8}$$

and $A_{**}$, $Q_{2*}$, $R_{2*}$ are defined in equation (2.3.7).

The rms state estimation error, which is defined as

$$\text{rms } \tilde{x}_i = \sqrt{E[\tilde{x}_i^2]} \quad , \tag{4.2.9}$$

is equal to the square root of the diagonal elements of the state estimation error covariance $\tilde{X}$.

b. The rms control error

The control error is defined as

$$\tilde{u} = Kx - Kx_*$$

$$= K\tilde{x} \ . \tag{4.2.10}$$

The control error covariance is given by

$$\tilde{U} = E[\tilde{u}\ \tilde{u}^T]$$

$$= K\ P\ K^T \ . \tag{4.2.11}$$

The rms control error

$$\text{rms } \tilde{u}_i = \sqrt{E[\ \tilde{u}_i^2\ ]} \tag{4.2.12}$$

is given by the square root of the diagonal elements of $\widetilde{U}$.

c. The rms output error

The output error is defined as

$$\widetilde{y} = (C_1 + D_1K)(x - x_*)$$

$$= (C_1 + D_1K)\widetilde{x} \quad . \tag{4.2.13}$$

The rms output error

$$\text{rms } \widetilde{y}_i = \sqrt{E[\ \widetilde{y}_i^{\ 2}\ ]} \tag{4.1.14}$$

is given by the square root of the diagonal elements of the output error covariance

$$\widetilde{Y} = E[\widetilde{y}\ \widetilde{y}^T]$$

$$= (C_1 + D_1K)\ P\ (C_1 + D_1K)^T \ . \tag{4.2.15}$$

d. The expected integrand of the control performance criterion

It is shown in Appendix D that the expectation of the integrand is given by

$$I = E[y^TQ_1y + u^TR_1u]$$

$$= tr[S\Gamma Q_2\Gamma^T + K^T(R_1 + D_1^TQ_1D_1)\ KP]\ , \qquad (4.2.16)$$

where tr[A] is the trace of the matrix A, S is the solution of the regulator Riccati equation (2.2.7), and P is the solution of the filter Riccati equation (4.2.8).

It is also shown in Appendix D that the rms values of $\tilde{x}_i$, $\tilde{u}_i$, $\tilde{y}_i$, and I are non-decreasing as the number of measurements decreases. The properties of non-decreasing error are stated in the following theorems.

Theorem 4.1.

Let $P_i$, $P_{ik}$ be the covariance of the state estimation error without $i^{th}$ and without $i^{th}$ and $k^{th}$ measurements respectively. These covariances satisfy the following algebraic Riccati equations

$$A_{**}P_i + P_iA_{**}^T + \Gamma Q_{2*}\Gamma^T - P_iR_iP_i = 0 \qquad (4.2.17)$$

$$A_{**}P_{ik} + P_{ik}A_{**}^T + \Gamma Q_{2*}\Gamma^T - P_{ik}R_{ik}P_{ik} = 0. \qquad (4.2.18)$$

The matrices in the above equations are defined in equations (2.3.7) of Chapter II. $R_i$ and $R_{ik}$ are the term $C_2{}^T R_{2*}{}^{-1} C_2$ with the $i^{th}$ and with the $i^{th}$ and $k^{th}$ measurements eliminated. Assuming that the estimation error dynamic equation is asymptotically stable (i.e., the eigenvalues of $A_{**} - P_{ik}R_{ik}$ and $A_{**} - P_iR_i$ are all in the open left half eigenvalue plane), then,

$$\Delta P = P_{ik} - P_i \geq 0 \tag{4.2.19}$$

if

$$\Delta R = R_i - R_{ik} \geq 0 \tag{4.2.20}$$

Also,

$$(\text{rms } \tilde{x}_j)_{ik} - (\text{rms } \tilde{x}_j)_i \geq 0 \tag{4.2.21}$$

if

$$\Delta R = R_i - R_{ik} \geq 0 \quad , \tag{4.2.22}$$

where subscripts j means $j^{th}$ element of the vector $\tilde{x}$ and subscripts ik and i indicate the elimination of the $i^{th}$ and $k^{th}$ and the $i^{th}$ measurements.

Theorem 4.2.

Under the same condition as in theorem 4.1, the following results are concluded.

If $$\Delta R = R_i - R_{ik} \geq 0 \quad , \tag{4.2.23}$$

then

a. $$\tilde{U}_{ik} - \tilde{U}_i \geq 0 \tag{4.2.24}$$

and

$$(\text{rms } \tilde{u}_j)_{ik} - (\text{rms } \tilde{u}_j)_i \geq 0 \quad , \tag{4.2.25}$$

b. $$\tilde{Y}_{ik} - \tilde{Y}_i \geq 0 \tag{4.2.26}$$

and

$$(\text{rms } \tilde{y}_j)_{ik} - (\text{rms } \tilde{y}_j)_i \geq 0 \ , \tag{4.2.27}$$

c. $$I_{ik} - I_i \geq 0, \tag{4.2.28}$$

where subscripts j, ik, and i have the same meaning as in the above theorem.

The proof of theorems 4.1 and 4.2 is given in Appendix D. If the matrix $R_{2*} = R_2 + \Theta Q_2 \Theta^T$ is a diagonal matrix, $\Delta R$ in equations (4.2.20), (4.2.22), and (4.2.23) is given by

$$\Delta R = c_{2k}^T \, r_{2*k}^{-1} \, c_{2k}$$

$$= (c_{2k}^T \, c_{2k})/r_{2*k} \quad , \qquad (4.2.29)$$

where $c_{2k}$ is the $k^{th}$ row of the $C_2$ matrix and $r_{2*k}$ is the $k^{th}$ diagonal element of $R_{2*}$ matrix.

It is understood that $\Delta R$ in equation (4.2.29) is a rank 1 matrix, provided $c_{2k}$ is not a zero row vector, and the non-zero eigenvalue is given by $(c_{2k}c_{2k}^T)/r_{2*k}$ which is a positive real number. This shows that if $R_{2*}$ is a diagonal matrix, $\Delta R$ is theorem 4.1 and 4.2 is always a non-negative definite matrix.

From theorem 4.1 and 4.2, a measurement elimination procedure is developed and is described as follows.

Step 1.

Decide the desirable criteria and an acceptable percentage of the relative difference. The relative difference is defined as

$$RD_i = \frac{(\mathrm{rms}\ \tilde{x}_i)_k - \mathrm{rms}\ \tilde{x}_i}{\mathrm{rms}\ \tilde{x}_i} \qquad (4.2.30)$$

where rms $\tilde{x}_i$ is chosen for the criterion example, the subscript k refers to the elimination of the $k^{th}$ measurement, and rms $\tilde{x}_i$ is the value with all the available measurements.

For discussion purposes, the criterion is chosen to be rms $\tilde{x}_i$ and 10 percent is chosen as the acceptable percentage.

Step 2.

Calculate rms $\tilde{x}_i$ with all the q measurements.

Step 3.

Take out one measurement at a time and calculate the maximum relative difference

$$MRD_k = \max_i RD_i \ . \qquad (4.2.31)$$

Again, subscript k refers to the elimination of the $k^{th}$ measurement.

Step 4.

Store all the indices k such that

$$MRD_k \leq 0.1 \quad (10\%) \tag{4.2.32}$$

and eliminate the $j^{th}$ measurement

where

$$MRD_j = \min_k \{ MRD_k \} . \tag{4.2.33}$$

Step 5.

If there are none or only one k satisfying equation (4.2.32), we are done. Otherwise, go back to step 3 and eliminate measurements among the remaining.

Step 2 and Step 3 require the calculation of the Riccati equation solution, which dominates the computation required. The maximum number of times Step 3 will be carried out is given by

$$q + (q - 1) + \ldots\ldots + 2 + 1 = \frac{q(q + 1)}{2} \tag{4.2.34}$$

where q is the number of measurements to start with.

However, since it is impossible to eliminate all the measurements, in most cases, the number of times Step 3 is carried out is usually much less than $\frac{q(q + 1)}{2}$. The minimum number of times Step 3 is carried out will be q, the number of original measurements.

The proposed measurement elimination procedure is applied to the small jet aircraft problem described in Chapter I and the results presented in the next section.

## IV.3 Application of the Measurement Elimination Procedure

The measurement elimination procedure proposed in the previous section is applied to a small jet aircraft. The aircraft model is a business transport flying at a constant altitude of 20,000 feet (6.1 km) and a nominal cruise speed of 450 knots (231 m/s). The stability derivative data was supplied by G.D. Park[22] and resulted in a standard fourth order lateral aircraft model previously developed by Wei[1]. The model was linearized about the nominal flight condition.

The aircraft model is augmented with a fourth order wind gust model previously developed by Holley and Bryson[23]. The resulting eighth order linear system forced by white noise has two controls, aileron and rudder, and can be described as

$$\dot{x} = Ax + Bu + \Gamma w \qquad (4.3.1)$$

where

$x^T = [p,\ r,\ v,\ \phi,\ v_o,\ v_x,\ u_y,\ w_y]$

$u^T = [\delta a,\ \delta r]$

$p$ = roll rate (rad/sec)

$r$ = yaw rate (rad/sec)

$v$ = lateral aircraft velocity (ft/sec)

$\phi$ = roll angle (rad)

$v_o$ = lateral gust velocity (ft/sec)

$v_x$ = longitudinal gradient of the lateral velocity ($sec^{-1}$)

$u_y$ = lateral gradient of the longitudinal velocity ($sec^{-1}$)

$w_y$ = lateral gradient of normal gust velocity ($sec^{-1}$)

$\delta a$ = aileron deflection (rad)

$\delta r$ = rudder deflection (rad)

The optimization criterion chosen for this study is given by

$$\underset{\delta a, \delta r}{\text{Min}} \; J = \frac{1}{2} E \left\{ \int_0^{\infty} \left[ \left(\frac{a_y}{a_{yo}}\right)^2 + \left(\frac{\dot{r}}{\dot{r}_o}\right)^2 + \left(\frac{f_t}{f_{to}}\right)^2 \right] dt \right\} \tag{4.3.2}$$

where

$a_y$ = lateral acceleration per unit gravitational acceleration

$\dot{r}$ = yaw acceleration (rad/sec$^2$)

$f_t$ = side force on vertical tail per unit weight of the aircraft.

The criterion of equation (4.3.2) seems reasonable since it minimizes the expected deviations in the lateral and yaw acceleration (ride performance) and in the tail side force. The control deviations are not explicitly weighted in the performance criterion because they are already indirectly included in the response variables chosen.

The criterion above is of the general form

$$\underset{u}{\text{Min}}\ J = \frac{1}{2} E[\int_{0}^{\infty} (y^T Q_1 y + u^T R_1 u) dt] \qquad (4.3.3)$$

subject to the constrains

$$\dot{x} = Ax + Bu + \Gamma w$$
$$(4.3.4)$$
$$y = C_1 x + D_1 u$$

where $y^T = [a_y, \dot{r}, f_t]$ is the output of the system and w is the white noise disturbance vector with power spectral density $Q_2$.

In the case when perfect state information is not available, additional measurement and causality constraints are added.

$$z = C_2 x + D_2 u + \nu \qquad (4.3.5)$$

where z is the measurement vector and $\nu$ is the white measurement noise with power spectral density $R_2$ and uncorrelated with w.

Equations (4.3.4) and (4.3.5) are discussed in detail in Appendix D.

Application of the measurement elimination procedure and solution of the control synthesis problem requires a reasonable choice of weighting factors $a_{y_o}$, $\dot{r}_o$, and $f_{t_o}$ appearing in the performance criteria (4.3.2) as well as a choice of the measurement vector z. In order to provide reasonable aircraft ride performance, it is desirable to keep lateral acceleration small all along the fuselage. This can be achieved with a rigid aircraft if the nose and tail accelerations are kept small. Averaging the squares of these quantities yields

$$\frac{1}{2}\,(a_{ny}{}^2 + a_{ty}{}^2) = \frac{1}{2}\,(a_y + \frac{\ell}{2g}\,\dot{r})^2 + \frac{1}{2}\,(a_y - \frac{\ell}{2g}\,\dot{r})^2$$

$$= a_y{}^2 + (\frac{\ell}{2g}\,\dot{r})^2 \qquad (4.3.6)$$

where $\ell$ = fuselage length

$a_{ny}$ = non-dimensional nose side acceleration

$a_{ty}$ = non-dimensional tail side acceleration.

Thus, the weighting factors $a_o = 1$ and $\dot{r}_o = \frac{2g}{\ell} = 1.323$ are chosen. It is pointed out by Holley and Wei[24] that a reasonable control implementation occurs when $f_{to}$ is designated as 0.3.

The measurement vector is chosen to be

$$z^T = [p,\ r,\ \phi,\ a_{ty},\ a_y,\ f_t,\ v,\ v - v_o] \qquad (4.3.7)$$

where $v - v_o$ is the relative velocity of the aircraft with respect to the surrounding air.

The sensors used for the measurement (4.3.7) are described as follows. The roll and yaw rates, p, r, can be measured by mounting rate gyros[25] on the body of the aircraft. A two-degree-of-freedom vertical Gyro[25], whose spin axis is vertical, can be placed on the body of the aircraft to measure the roll angle $\phi$ . The side acceleration of the center of mass and of the vertical tail, $a_y$ and $a_{ty}$, can be obtained by placing accelerometers[25] near the mass center and on the vertical tail of the aircraft respectively. Strain gauges can be used to measure the strain in the tail. The strain gauge can be mounted on the tail near the main fuselage structure to prevent a possible large amplification of noise. The tail strain can then be converted to the tail loading, $f_t$ , if linear elastic properties of the tail are assumed. The side velocity can be calculated from the outputs of the horizontal accelerometers[25]. However, since outputs of accelerometers may have bias, the integrated signal is potentially unstable. To overcome this difficulty, a doppler radar system[25] can be used to measure the velocity. A low pass filter is applied on the velocity signal from the doppler radar and a high pass filter is applied on the velocity integrated from the accelerometers outputs. The combined signals from the low-pass and high-pass filters give a good estimation of the side velocity. To measure the relative velocity of the aircraft with respect to air $v-v_o$, a pitot-static tube[25] can be used. A pitot-static tube consists of a static-pressure port which measures the static pressure and a pitot tube which measures the stagnation pressure. A pitot-static tube is mounted in such a way that it aims directly into the relative wind component to be measured (in this case laterally). The precise location of a pitot-static tube is selected by wind-tunnel tests and by tests at numerous locations on the actual aircraft, in order to be as free from error as possible at all flight speeds and attitudes. The static and stagnation pressures can then be converted to the relative side velocity $v-v_o$.

The power spectral densities for the measurement noise were chosen on the following basis. The two rate gyros for p and r being similar, should have the same error characteristics. The two accelerometers for $a_{ty}$ and $a_y$, also, should have the same error characteristic. Thus, the following form for the measurement noise power spectral density was assumed.

$$R_2 = \text{diagonal } [r_1, r_2, r_3, \ldots., r_8] \tag{4.3.8}$$

where

$$r_1 = r_2 = \tau_1 * [\frac{(\text{rms } p)^2 + (\text{rms } r^2)}{2}]$$

$$r_3 = \tau_3 * (\text{rms } \phi)^2$$

$$r_4 = r_5 = \tau_4 * (\text{rms } a_y)^2$$

$$r_6 = \tau_6 * (\text{rms } f_t)^2$$

$$r_7 = \tau_7 * (\text{rms } v)^2$$

$$r_8 = \tau_8 * [\text{rms } v)^2 + [\text{rms } v_o)^2]$$

and the rms values are the controlled responses assuming perfect state knowledge.

The correlation time constants $\tau_1$ to $\tau_8$ were varied to achieve reasonable eigenvalues for the state estimation error dynamics. The result turns out to be

$$r_1 = r_2 = 0.9387082 * 10^{-4}, \quad \tau_1 = 1 \text{ sec}$$

$$r_3 = 0.14625 * 10^{-5} \quad , \quad \tau_3 = 0.05 \text{ sec}$$

$$r_4 = r_5 = 0.1901376 * 10^{-4}, \quad \tau_4 = 0.3 \text{ sec}$$

$$r_6 = 0.1123035 * 10^{-5} \, , \, \tau_6 = 0.3 \text{ sec}$$

$$r_7 = 0.1270288 * 10^{-2}, \quad \tau_7 = 0.002 \text{ sec}$$

$$r_8 = 0.33143 \quad , \quad \tau_8 = 0.2 \text{ sec} \quad .$$

The resulting eigenvalues of the controller and the estimator are given by

| The controller ($sec^{-1}$) | The estimator ($sec^{-1}$) |
|---|---|
| -1.0227 | -2.5493 ±0.97726j |
| -2.1709 ±2.8732j | -5.6246 ±6.8560j |
| -3.9502 ±0.33407j | -13.259 |
| -16.788 | -16.606 |
| -25.181 | -25.502 |
| -89.863 | -90.712 |

The measurement elimination procedure is now applied to the system model described above. Since the purpose of the Kalman filter in this case is to provide state estimates for the controller, the rms control error, which is described in equations (4.2.10) to (4.2.12), is used as the criterion of the elimination procedure. The acceptable percentage of the relative difference, which is described by equation (4.2.30), is chosen to be 10 percent for engineering purposes.

The results of the measurement elimination procedure can be described as follows. First, application of Step 1 through Step 5 of the elimination procedure will eliminate the measurement r and will recognize the measurements p, $\phi$, $a_y$, $f_t$, and $v-v_o$ are potential candidates for further elimination. Step 3 through Step 5 of the procedure is then applied to those potential candidates which yield the elimination of the measurement $v-v_o$, which leaves p, $\phi$, $a_y$ and $f_t$. Again, these potential candidates are used for further elimination. This yields the elimination of p followed by $a_y$ and $f_t$. The resulting measurement is given by

$$z^T = [\phi, a_{ty}, v] \quad . \tag{4.3.9}$$

This procedure resulted in 22 computations of Step 3 requiring approximately 90 cpu seconds on the CDC CYBER 73 system.

# V RESULTS AND DISCUSSION

## V.1 Results of Tail Force Reduction

As discussed before, the aircraft used for the tail force reduction is a small business jet flying at a constant altitude of 20,000 feet (6.1 km) and a nominal cruise speed of 450 knots (231 m/s). This system model is described in more detail in Chapter I, Chapter IV, and Appendix D. Equations (4.3.1) through (4.3.5) give the system equations as well as the performance criterion. The measurement for state estimation described in equation (4.3.5) is given by

$$z^T = [\phi, a_{ty}, v] \tag{5.1.1}$$

which results from the measurement elimination procedure discussed in the previous chapter.

The computations were carried out on a CDC Cyber 73 computer system using the program listed in Appendix F. Tables 1 and 2 give the resulting control gain and filter gain. The closed loop eigenvalues of the controller and the estimation eigenvalues are given in table 3.

TABLE 1. THE LATERAL FEEDBACK CONTROL GAIN

| | $p$ | $r$ | $v$ | $\phi$ |
|---|---|---|---|---|
| $\delta a$ | $.59479 \times 10^{-2}$ | .32849 | $.97467 \times 10^{-3}$ | -.10210 |
| $\delta r$ | $.54081 \times 10^{-1}$ | .64000 | $.63828 \times 10^{-3}$ | .12039 |

| | $v_o$ | $v_x$ | $u_y$ | $u_y$ |
|---|---|---|---|---|
| $\delta a$ | $-.12611 \times 10^{-2}$ | $-.27779 \times 10^{-1}$ | $-.41981 \times 10^{-1}$ | $-.36929 \times 10^{-1}$ |
| $\delta r$ | $-.13524 \times 10^{-2}$ | $.72326 \times 10^{-1}$ | $-.17224 \times 10^{-1}$ | $-.16450 \times 10^{-1}$ |

TABLE 2. THE FILTER GAIN

| | $\phi$ | $a_{ty}$ | $v$ |
|---|---|---|---|
| $p$ | -5.0915 | -3.6635 | -.46825 |
| $r$ | .50090 | .35463 | $.75626 \times 10^{-1}$ |
| $v$ | -64.884 | -11.320 | -10.937 |
| $\phi$ | -2.1789 | -.21271 | $-.74704 \times 10^{-1}$ |
| $v_o$ | -129.33 | -251.98 | -18.459 |
| $v_x$ | .14908 | .55355 | $.23344 \times 10^{-1}$ |
| $u_y$ | $.42505 \times 10^{-1}$ | -.22464 | $-.25350 \times 10^{-1}$ |
| $w_y$ | -.22901 | -.65860 | $-.41956 \times 10^{-2}$ |

TABLE 3. THE CLOSED LOOP EIGENVALUES OF CONTROLLER AND FILTER

| Controller | Filter |
|---|---|
| -.21709 ±2.8732j | -2.4759 ±1.0159j |
| -3.9502 ±.33407j | -5.3053 ±7.5734j |
| -1.0227 | -12.094 |
| -89.863 | -90.001 |
| -16.788 | -16.731 |
| -25.181 | -25.516 |

In state estimation, obviously the best estimated state can be achieved when the measurement is equal to the state itself. However, since the wind states $v_o$, $v_x$, $u_y$, $w_y$ are difficult to measure, the case z = x is not of interest in practice. However, in order to see the effect of the measurement, comparison between four cases: (1) z = x, (2) $z^T = [p, r, \phi, a_{ty}, a_y, f_t, v, v-v_o]$, (3) $z^T = [\phi, a_{ty}, v]$, and (4) open loop, is of interest. The power spectral density of the measurement noise for the case z = x is assumed to be

$$R_2 = \text{diagonal } [r_1, r_2, r_3, \ldots., r_8] \tag{5.1.2}$$

where $r_1$, $r_2$, $r_3$, $r_4$ are for the variables p, r, v, $\phi$ and are assumed to have the same values as in the case $z^T = [p, r, \phi, a_{ty}, a_y, f_t, v, v-v_o]$. The remaining terms

$r_5$, $r_6$, $r_7$, and $r_8$ are for the variables $v_o$, $v_x$, $u_y$, $w_y$ which are assumed to be of the form (choosing $v_o$ as an example)

$$r_5 = \tau * (\text{rms } v_o)^2 \tag{5.1.3}$$

where the correlation time constant $\tau$ is chosen to be 0.2 for all cases. The resulting noise power spectral densities are given by

$$r_1 = r_2 = .93871 \times 10^{-4}$$

$$r_3 = .12703 \times 10^{-2}$$

$$r_4 = .14625 \times 10^{-5}$$

$$r_5 = .20438$$

$$r_6 = .30812 \times 10^{-4}$$

$$r_7 = .15158 \times 10^{-4}$$

$$r_8 = .15842 \times 10^{-4} \quad .$$

The rms state for those four cases is presented in table 4 and the rms control and output is given in table 5.

TABLE 4. THE RMS VALUES OF AIRCRAFT AND WIND STATES

| Cases | $p$ | $r$ | $v$ | $\phi$ |
|---|---|---|---|---|
| 1 | $.17654\times10^{-1}$ | $.21546\times10^{-2}$ | .92196 | $.59229\times10^{-2}$ |
| 2 | $.16461\times10^{-1}$ | $.18770\times10^{-2}$ | .87406 | $.60863\times10^{-2}$ |
| 3 | $.16924\times10^{-1}$ | $.19824\times10^{-2}$ | .89189 | $.60645\times10^{-2}$ |
| 4 | $.28081\times10^{-1}$ | $.37140\times10^{-2}$ | 1.2102 | $.12334\times10^{-1}$ |

| Cases | $v_o$ | $v_x$ | $u_y$ | $w_y$ |
|---|---|---|---|---|
| 1 | 1.0109 | $.12413\times10^{-1}$ | $.87058\times10^{-2}$ | $.89001\times10^{-2}$ |
| 2 | 1.0109 | $.12413\times10^{-1}$ | $.87058\times10^{-2}$ | $.89001\times10^{-2}$ |
| 3 | 1.0109 | $.12413\times10^{-1}$ | $.87058\times10^{-2}$ | $.89001\times10^{-2}$ |
| 4 | 1.0109 | $.12413\times10^{-1}$ | $.87058\times10^{-2}$ | $.89001\times10^{-2}$ |

TABLE 5. THE RMS VALUES OF CONTROL AND OUTPUT

| Cases | $\delta a$ | $\delta r$ | $a_y$ |
|---|---|---|---|
| 1 | $.12177\times10^{-2}$ | $.66513\times10^{-3}$ | $.77019\times10^{-2}$ |
| 2 | $.13342\times10^{-2}$ | $.71589\times10^{-3}$ | $.82503\times10^{-2}$ |
| 3 | $.12971\times10^{-2}$ | $.69302\times10^{-3}$ | $.81018\times10^{-2}$ |
| 4 | | | $.12042\times10^{-1}$ |

| Cases | $r$ | $f_t$ |
|---|---|---|
| 1 | $.13958\times10^{-1}$ | $.23506\times10^{-2}$ |
| 2 | $.11915\times10^{-1}$ | $.22995\times10^{-2}$ |
| 3 | $.12645\times10^{-1}$ | $.23267\times10^{-2}$ |
| 4 | $.17959\times10^{-1}$ | $.33337\times10^{-2}$ |

From table 5, the reduction of $a_y$, $\dot{r}$, and $f_t$ with respect to the open loop can be calculated and is presented in table 6.

TABLE 6. THE OUTPUT REDUCTION ACCOMPLISHED WITH STATE ESTIMATION

| Reduction / Cases | $a_y$ | $\dot{r}$ | $f_t$ |
|---|---|---|---|
| 1 | 36.04% | 22.28% | 29.49% |
| 2 | 32.49% | 33.65% | 31.02% |
| 3 | 32.72% | 29.59% | 30.21% |

It is also interesting to see the percentage change in the rms state, control, and output in relation to the case 1 where of $z = x$. It can be calculated from table 4 and 5 and is presented in table 7, 8. Due to the fact that the wind state variables, $v_o$, $v_x$, $u_y$, and $w_y$ are not controllable, the rms values of those state variables are not affected by the controller. The unchanged property of the wind state variables can be seen in table 4. Because of this property, only the aircraft state variables are presented in table 7.

TABLE 7. THE PERCENTAGE CHANGE IN THE RMS AIRCRAFT STATE IN RELATION TO THE CASE $z = x$

| Cases | $p$ | $r$ | $v$ | $\phi$ |
|---|---|---|---|---|
| 2 | -6.76% | -12.88% | -5.19% | -2.76% |
| 3 | -4.13% | -7.99% | -3.26% | -2.39% |

TABLE 8. THE PERCENTAGE CHANGE IN THE RMS CONTROL AND OUTPUT IN RELATION TO THE CASE $z = x$

| Cases | $\delta a$ | $\delta r$ | $a_y$ | $\dot{r}$ | $f_t$ |
|---|---|---|---|---|---|
| 2 | 9.57% | 7.63% | 7.12% | -14.64% | -2.17% |
| 3 | 6.52% | 4.19% | 5.19% | -9.41% | -1.02% |

As shown in table 6 that, with the measurement $z^T = [\phi, a_{ty}, v]$, a reduction of 30 percent in the open loop response of the aerodynamic tail forces appears to be feasible for the aircraft flying in turbulence. Also, a reduction of 30 percent of the open loop response in the side acceleration $a_y$ and the yaw acceleration $\dot{r}$ appear to be feasible in the case of $z^T = [\phi, a_{ty}, v]$. This shows that the aircraft ride performance is improved while reducing the tail loading.

It is evident that, for state estimation, the best filter can be achieved with the measurement $z = x$. However, from tables 7 and 8, the percentage change in the rms responses of the state output, and control are within 15 percent of the case when $z = x$. The filter with measurement $z^T = [\phi, a_{ty}, v]$ is evidently feasible, and gives nearly optimal results.

## V.2 Discussion

It is pointed out in Chapter IV, that the correlation time factor $\tau$ of the power spectral density $R_2$ is varied to achieve reasonable eigenvalues for the state estimation error dynamics. Nine cases were tested to determine appropriate values of $\tau$ . The values tested are listed in table 9.

TABLE 9. CASES TESTED FOR DIFFERENT VALUES OF THE CORRELATION TIME FACTOR $\tau$ IN $R_2$

| $\tau$ of case / measurement | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| p, r | 1. | 1. | 1. | 1. | 1. | 1. | 1. | .1 | .1 |
| $\phi$ | .01 | .1 | .01 | .01 | .01 | .01 | .01 | .01 | .05 |
| $a_{ty}$, $a_y$ | .3 | .3 | 3. | .3 | .3 | .3 | .3 | .3 | .3 |
| $f_t$ | .3 | .3 | .3 | .3 | .3 | .3 | .3 | .3 | .3 |
| v | .2 | .2 | .2 | .4 | .02 | .02 | .002 | .002 | .002 |
| $v - v_o$ | 2. | 2. | 2. | 2. | 2. | .02 | 2. | .2 | .2 |

It is useful to see how the eigenvalues of the state estimation error dynamics, and the rms response of the state, control, and output change as $\tau$ changes. The eigenvalues are listed in table 10 for the nine cases.

TABLE 10. THE EIGENVALUES OF THE STATE ESTIMATION ERROR DYNAMICS FOR DIFFERENT VALUES OF $\tau$ IN $R_2$

| Case 1 | Case 2 | Case 3 |
|---|---|---|
| -1.1231 ±.90455j | -1.2169 ±.70785j | -1.1394 ±.91480j |
| -5.4053 ±5.9140j | -3.8116 ±3.3171j | -5.5202 ±7.2646j |
| -12.646 | -11.863 | -9.3924 |
| -16.626 | -16.649 | -16.779 |
| -25.512 | -25.488 | -25.204 |
| -90.743 | -90.743 | -90.636 |

| Case 4 | Case 5 | Case 6 |
|---|---|---|
| -.98103 ±.74612j | -1.8842 ±1.5805j | -1.8825 ±1.5801j |
| -5.4018 ±5.9019j | -5.4699 ±6.1216j | -5.4925 ±6.0562j |
| -12.640 | -12.740 | -12.901 |
| -16.626 | -16.624 | -16.609 |
| -25.512 | -25.513 | -25.514 |
| -90.743 | -90.743 | -90.711 |

| Case 7 | Case 8 | Case 9 |
|---|---|---|
| -2.9015 ±2.1613j | -2.9009 ±2.16217j | -2.5493 ±.97726j |
| -6.0415 ±7.5074j | -6.0631 ±7.4448j | -5.6246 ±6.8560j |
| -13.484 | -13.631 | -13.259 |
| -16.600 | -16.580 | -16.606 |
| -25.523 | -25.524 | -25.502 |
| -90.743 | -90.711 | -90.712 |

As shown in table 10, different eigenvalues are changed when different values of $\tau$ are changed. However, when the $\tau$ associated with the measurements p, r, and $v-v_o$ is changed, the eigenvalues do not change very much. Thus, it is predicted that the filter is not very sensitive to the elements of the $R_2$ matrix corresponding to the measurements p, r, $v-v_o$.

The rms response of the aircraft state, the control, and the output for the nine cases is listed in table 11. Again, the reason that the rms wind state are not listed is because those states are uncontrolled and are not effected by the controller or filter. From table 11, it can be seen that the maximum percentage change in the state is 17 percent, 18 percent in the control, and 14 percent in the output. Since the accuracy of the estimated state depends on the measurement noise, the power spectral density $R_2$ plays an important role in the state estimation. As the $R_2$ matrix is not determined by real sensor tests, it is necessary to study the sensitivity of the $R_2$ with respect to the rms responses. This sensitivity study will show the effect of the $R_2$ matrix in this autopilot design.

TABLE 11. RMS RESPONSES OF THE AIRCRAFT STATE, THE CONTROL, AND THE OUTPUT FOR DIFFERENT VALUES OF $\tau$ IN $R_2$

| rms of / case | p | r | v |
|---|---|---|---|
| 1 | $.17526 \times 10^{-1}$ | $.20229 \times 10^{-2}$ | .93212 |
| 2 | $.17958 \times 10^{-1}$ | $.20987 \times 10^{-2}$ | .94845 |
| 3 | $.18481 \times 10^{-1}$ | $.22202 \times 10^{-2}$ | .96243 |
| 4 | $.17724 \times 10^{-1}$ | $.20424 \times 10^{-2}$ | .94871 |
| 5 | $.16986 \times 10^{-1}$ | $.19567 \times 10^{-2}$ | .89555 |
| 6 | $.16943 \times 10^{-1}$ | $.19457 \times 10^{-2}$ | .89364 |
| 7 | $.16449 \times 10^{-1}$ | $.18706 \times 10^{-2}$ | .87369 |
| 8 | $.16417 \times 10^{-1}$ | $.18612 \times 10^{-2}$ | .87204 |
| 9 | $.16461 \times 10^{-1}$ | $.18770 \times 10^{-2}$ | .87406 |

| rms of / case | $\phi$ | $\delta a$ | $\delta r$ |
|---|---|---|---|
| 1 | $.63770 \times 10^{-2}$ | $.13800 \times 10^{-2}$ | $.64370 \times 10^{-3}$ |
| 2 | $.65659 \times 10^{-2}$ | $.13650 \times 10^{-2}$ | $.63387 \times 10^{-3}$ |
| 3 | $.63347 \times 10^{-2}$ | $.12930 \times 10^{-2}$ | $.59499 \times 10^{-3}$ |
| 4 | $.64844 \times 10^{-2}$ | $.13977 \times 10^{-2}$ | $.62330 \times 10^{-3}$ |
| 5 | $.61143 \times 10^{-2}$ | $.13412 \times 10^{-2}$ | $.69037 \times 10^{-3}$ |
| 6 | $.61184 \times 10^{-2}$ | $.13451 \times 10^{-2}$ | $.69239 \times 10^{-3}$ |
| 7 | $.59645 \times 10^{-2}$ | $.13316 \times 10^{-2}$ | $.71811 \times 10^{-3}$ |
| 8 | $.59706 \times 10^{-2}$ | $.13352 \times 10^{-2}$ | $.71970 \times 10^{-3}$ |
| 9 | $.60863 \times 10^{-2}$ | $.13342 \times 10^{-2}$ | $.71589 \times 10^{-3}$ |

TABLE 11. RMS RESPONSES OF THE AIRCRAFT STATE, THE CONTROL, AND THE OUTPUT FOR DIFFERENT VALUES OF $\tau$ IN $R_2$ (cont)

| case \ rms of | $a_y$ | $r$ | $f_t$ |
|---|---|---|---|
| 1 | $.87087 \times 10^{-2}$ | $.12128 \times 10^{-1}$ | $.24385 \times 10^{-2}$ |
| 2 | $.87952 \times 10^{-2}$ | $.12210 \times 10^{-1}$ | $.24955 \times 10^{-2}$ |
| 3 | $.83987 \times 10^{-2}$ | $.13636 \times 10^{-1}$ | $.24677 \times 10^{-2}$ |
| 4 | $.88696 \times 10^{-2}$ | $.12147 \times 10^{-1}$ | $.24691 \times 10^{-2}$ |
| 5 | $.83231 \times 10^{-2}$ | $.12065 \times 10^{-1}$ | $.23558 \times 10^{-2}$ |
| 6 | $.83398 \times 10^{-2}$ | $.11981 \times 10^{-1}$ | $.23538 \times 10^{-2}$ |
| 7 | $.81437 \times 10^{-2}$ | $.11932 \times 10^{-1}$ | $.22879 \times 10^{-2}$ |
| 8 | $.81613 \times 10^{-2}$ | $.11853 \times 10^{-1}$ | $.22867 \times 10^{-2}$ |
| 9 | $.82503 \times 10^{-2}$ | $.11915 \times 10^{-1}$ | $.22995 \times 10^{-2}$ |

From table 11, it is expected that changes in the $R_2$ matrix will not cause drastic changes in rms responses. However, since the change in $\tau$ for these nine cases is only a factor of 10 to 100, this is not a sufficient evidence for complete insensitivity of the rms responses to all possible changes in the $R_2$ matrix.

One reasonable way to study the sensitivity of the rms responses with respect to the $R_2$ matrix is to set $R_2 = 0$ when the rms response is calculated. Since the state covariance X is the sum of the estimated state covariance $X_*$ and the estimation error covariance P, which is shown in equation (2.4.29), the effect of $R_2$ is through $X*$ and P.

If $R_2$ is set to zero when the rms response is calculated, then $R_2$ will not effect the calculation of the covariance properties. In this case, the only effect of $R_2$ with respect to the rms response is through the filter which can not be avoided. However, when $R_2 = 0$, the filter is no longer optimal, and the expectation $E[x_* \tilde{x}^T]$ is no longer equal to zero. In this case, a 2n'th order system is set up to calculate the rms reponse. The 2n'th order state equation is given by

$$\begin{bmatrix} \dot{x} \\ \\ \dot{x}_* \end{bmatrix} = \begin{bmatrix} A & BK \\ \\ -GC_2 & A + BK + GC_2 \end{bmatrix} \begin{bmatrix} x \\ \\ x_* \end{bmatrix} + \begin{bmatrix} \Gamma \\ \\ 0 \end{bmatrix} w \qquad (5.2.1)$$

where x is the actual state, $x_*$ is the estimated state, w is the process noise.

The covariance matrix satisfies the usual Lyapunov equation for a linear system forced by white noise. The resulting rms response are listed in table 12.

TABLE 12. THE RMS RESPONSE CALCULATION WITH $R_2 = 0$

| case \ rms | p | r | v |
|---|---|---|---|
| z = x | $.17143\text{x}10^{-1}$ | $.20821\text{x}10^{-2}$ | .91620 |
| z=8 measurements | $.15943\text{x}10^{-1}$ | $.17893\text{x}10^{-2}$ | .86647 |
| z=3 measurements | $.16336\text{x}10^{-1}$ | $.18865\text{x}10^{-2}$ | .88355 |

TABLE 12. THE RMS RESPONSE CALCULATIONS WITH $R_2$ = 0 (cont)

| case \ rms | $\phi$ | $\delta a$ | $\delta r$ |
|---|---|---|---|
| z=x | $.57535\text{x}10^{-2}$ | $.10670\text{x}10^{-2}$ | $.51991\text{x}10^{-3}$ |
| z=8 measurements | $.59402\text{x}10^{-2}$ | $.12243\text{x}10^{-2}$ | $.60378\text{x}10^{-3}$ |
| z=3 measurements | $.59034\text{x}10^{-2}$ | $.11629\text{x}10^{-2}$ | $.56600\text{x}10^{-3}$ |

| case \ rms | $a_y$ | $\dot{r}$ | $f_t$ |
|---|---|---|---|
| z=x | $.75261\text{x}10^{-2}$ | $.12749\text{x}10^{-1}$ | $.22571\text{x}10^{-2}$ |
| z=8 measurements | $.80792\text{x}10^{-2}$ | $.10727\text{x}10^{-1}$ | $.21945\text{x}10^{-3}$ |
| z=3 measurements | $.79136\text{x}10^{-2}$ | $.11358\text{x}10^{-1}$ | $.22148\text{x}10^{-3}$ |

The percentage change for the case $R_2$ = 0 relative to the nominal $R_2$ case and the open loop can be calculated from tables 4, 5, and 12. The results are listed in tables 13, 14, and 15.

TABLE 13. THE RELATIVE $R_2$ SENSITIVITY FOR z=x

| | p % | r % | v % | $\phi$ % | $\delta a$ % | $\delta r$ % | $a_y$ % | $\dot{r}$ % | $f_t$ % |
|---|---|---|---|---|---|---|---|---|---|
| $R_2$ | 2.89 | 3.36 | .62 | 2.86 | 12.38 | 21.83 | 2.28 | 8.66 | 3.98 |
| Open loop | 38.95 | 43.94 | 24.29 | 53.35 | | | 37.50 | 29.01 | 32.29 |

TABLE 14. THE RELATIVE $R_2$ SENSITIVITY FOR z = 8 SENSORS

| | p % | r % | v % | $\phi$ % | $\delta a$ % | $\delta r$ % | $a_y$ % | $\dot{r}$ % | $f_t$ % |
|---|---|---|---|---|---|---|---|---|---|
| $R_2$ | 3.15 | 4.67 | .87 | 2.40 | 8.24 | 15.66 | 2.07 | 9.97 | 4.57 |
| Open loop | 43.22 | 51.82 | 28.40 | 51.84 | | | 32.91 | 40.27 | 34.17 |

TABLE 15. THE RELATIVE $R_2$ SENSITIVITY FOR z = 3 SENSORS

| | p % | r % | v % | $\phi$ % | $\delta a$ % | $\delta r$ % | $a_y$ % | $\dot{r}$ % | $f_t$ % |
|---|---|---|---|---|---|---|---|---|---|
| $R_2$ | 3.47 | 4.84 | .93 | 2.66 | 10.35 | 18.33 | 2.32 | 10.18 | 4.81 |
| Open loop | 41.83 | 49.21 | 26.99 | 52.14 | | | 34.28 | 36.76 | 33.56 |

From tables 13, 14, and 15, it can be seen that the effect of the zero $R_2$ matrix is less than five percent in the rms aircraft state, is less than 11 percent in the rms output, and is less than 22 percent in the rms control. Again, since the wind states are uncontrollable, the $R_2$ matrix will not effect them.

The effect of the measurement z in the control u can be better understood by examining the filter transfer function of z to u. From the filter implementation equation

$$\dot{x}_* = \tilde{A}_* x_* - Gz \tag{5.2.2}$$

and the optimal control law

$$u = K x_* , \tag{5.2.3}$$

the transfer function of z to u is given by

$$\begin{aligned} T_{zu} &= \frac{L\{u(t)\}}{L\{z(t)\}} \\ &= K (\tilde{A}_* - Is) G \quad . \end{aligned} \tag{5.2.4}$$

The notation L{u(t)} in equation (5.2.4) is the Laplace transformation of u(t). If the eigenvalues of the matrix $A_*$ are distinct (as in this case) the transfer function, equation (5.2.4), can be represented as

$$T_{zu} = \sum_{i=1}^{n} H_i/(s - \lambda_i) \qquad (5.2.5)$$

where n is the order of the matrix $\tilde{A}_*$, $\lambda_i$ is the eigenvalue of $\tilde{A}_*$, and $H_i$ is the matrix of residues of the transfer function associated with the eigenvalue $\lambda_i$.

Since the value of $H_i/(s - \lambda_i)$ is small at high frequency (which means the effect of z in u is small at high frequency), the following transfer function discussion is focussed on the case of low frequency (i.e., when s is small compared with $\lambda_i$) where the transfer function is approximately given by

$$T_{zu} \approx \sum_{i=1}^{n} (- \frac{1}{\lambda_i} H_i) \ . \qquad (5.2.6)$$

The residues $H_i$ of the transfer function $T_{zu}$ can be found as follows. Since the eigenvector matrix of $\tilde{A}_*$ diagonalizes the matrix $\tilde{A}_*$, equation (5.2.4) can be transformed to

$$T_{zu} = KE\ (sI - D_s)^{-1}\ E^{-1}\ G, \tag{5.2.7}$$

where E is the eigenvector matrix and $D_s$ is in the diagonal form.

The coefficient matrix $H_i$ is obtained by multiplying the $i^{th}$ row of the KE matrix and the $i^{th}$ column of the $E^{-1}$ G matrix. The resulting coefficient $H_i/\lambda_i$ for the case $z^T = [\phi,\ a_{ty},\ v]$ is listed as follows

for $\lambda_1 = -89.860$

$$-\frac{H_1}{\lambda_1} = \begin{bmatrix} -.3086\text{x}10^{-5} & .4340\text{x}10^{-4} & -.2882\text{x}10^{-6} \\ .1603\text{x}10^{-4} & -.2255\text{x}10^{-3} & .1497\text{x}10^{-5} \end{bmatrix}$$

for $\lambda_2 = -25.195$

$$-\frac{H_2}{\lambda_2} = \begin{bmatrix} -.4879\text{x}10^{-3} & -.1204\text{x}10^{-2} & .4554\text{x}10^{-5} \\ -3.175\text{x}10^{-3} & -.7833\text{x}10^{-3} & .2964\text{x}10^{-5} \end{bmatrix}$$

for $\lambda_3, \lambda_4 = -7.2636 \pm 7.4072j$

$$-\frac{H_{3,4}}{\lambda_{3,4}} = \begin{bmatrix} -.3401\times10^{-1} & .6401\times10^{-2} & -.5871\times10^{-2} \\ -.2977\times10^{-1} & .2677\times10^{-2} & -.4912\times10^{-2} \end{bmatrix}$$

$$\pm j \begin{bmatrix} -.1463\times10^{-1} & -.1573\times10^{-1} & -.1082\times10^{-2} \\ -.6426\times10^{-2} & -.1372\times10^{-1} & .5616\times10^{-4} \end{bmatrix}$$

for $\lambda_5, \lambda_6 = -3.7736 \pm 3.3764j$

$$-\frac{H_{5,6}}{\lambda_{5,6}} = \begin{bmatrix} -.1819\times10^{-1} & -.1197\times10^{-1} & -.3984\times10^{-3} \\ .5474\times10^{-1} & -.2222\times10^{-1} & -.5171\times10^{-3} \end{bmatrix}$$

$$\pm j \begin{bmatrix} .2225\times10^{-1} & -.4452\times10^{-2} & -.2637\times10^{-3} \\ .7153\times10^{-1} & .3329\times10^{-1} & -.1406\times10^{-2} \end{bmatrix}$$

for $\lambda_7 = -7.4058$

$$-\frac{H_7}{\lambda_7} = \begin{bmatrix} -.1448\times10^{-1} & -.3912\times10^{-1} & .9875\times10^{-3} \\ -.1497\times10^{-1} & -.4044\times10^{-1} & .1021\times10^{-2} \end{bmatrix}$$

for $\lambda_8 = -16.784$

$$- \frac{H_8}{\lambda_8} = \begin{bmatrix} .5710\text{x}10^{-3} & .2887\text{x}10^{-3} & -.1373\text{x}10^{-4} \\ .1785\text{x}10^{-3} & .9023\text{x}10^{-4} & -.4292\text{x}10^{-5} \end{bmatrix}$$

To determine any weakly observable modes of the system, Equation (5.2.2) is transformed into

$$\dot{y} = D_s\, y - E^{-1}\, G\, z \tag{5.2.8}$$

where $\quad y = E^{-1}\, x_* \;.$

The transfer function of z to y is given by

$$T_{zy} = (D_s - Is)^{-1}\, E^{-1}\, G \;. \tag{5.2.9}$$

From equation (5.2.9), if $i^{th}$ row of the matrix $E^{-1}$ G is small compared with the rest of the rows, the measurement z weakly effects the mode $y_i$, which means $y_i$ is weakly observable. The matrix $E^{-1}$ G in equation (5.2.12) is given by

$$E^{-1}G = \begin{bmatrix} .14535 & -2.0440 & .13574\times10^{-1} \\ .32685 & .80641 & -.30512\times10^{-2} \\ -162.95 & -11.784 & -24.819 \\ -22.456 & 74.552 & -9.3690 \\ 75.970 & -83.853 & -.96848\times10^{-1} \\ -225.34 & -64.318 & 3.9530 \\ 58.064 & 156.81 & -3.9586 \\ .28530 & .14424 & -.68614\times10^{-2} \end{bmatrix}$$

From the matrix $E^{-1}$ G, it can be seen that the wind state modes associate with eigenvalues

$$\lambda_1 = -89.860$$

$$\lambda_2 = -25.195$$

$$\lambda_8 = -16.784$$

are weakly observable. However, it is found that the coefficients $H_i/\lambda_i$ corresponding to those wind state modes are relatively small in magnitude. This result shows that the weakly observable modes have a small effect on the control and hence will play a minor roll in the effectiveness of the control design.

# VI CONCLUSIONS

The capability of a linear quadratic regulator for the reduction of aerodynamic tail forces was investigated. The results indicate a 30 percent reduction of the tail force is feasible while maintaining acceptable performance of the aircraft lateral motion. The number of measurements required for state estimation can be reduced to three by the application of the procedure described in Chapter IV. The results also indicate that the tail force reduction is about the same with all eight feasible measurements and with the three measurements resulting from the elimination procedure. However, the elimination of the five measurements will significantly reduce the cost of the resulting control system.

The sensitivity of the system to changes in the noise power spectral density, $R_2$, of the measurements was also investigated. The results show that the estimated state and output are relatively insensitive to $R_2$.

The study of the filter transfer function of z to u indicates that weakly-observable wind state modes will not drastically effect the controller. This result is expected since the wind state modes have large negative real eigenvalues causing the modes to be damped out faster.

Realistically, the system model is not known precisely. Future research on this project should be directed toward studying the sensitivity of the control system to modeling errors.

It is discussed in Chapter II that the solutions of the control and the filter problems are mainly dependent on the solution of the Riccati equation. However, concerning the numerical conditioning of the Riccati equation, almost no analytical results are known. The conditioning of the Riccati equation is also an important subject for future research.

# REFERENCES


1. Wei, S.Y., Optimal Control of the Gate-Learjet Model 35 for Reduction of Tail Forces, Master Project Report, Mechanical Engineering Dept. Oregon State Univ., Corvallis, OR, June, 1977.

2. Åström, K.J., Introduction to Stochastic Control Theory, N.Y. and London, Acadamic Press, 1970, Chpt. 8.

3. Flügge-Lotz, I. and Halkin, H., Pontryagin's Maximum Principle and Optimal Control, Stanford Univ., Stanford, CA, Tech. Rept. 130 (1961).

4. Sage, A.P. and Melsa, J.L., System Identification, N.Y., Acadamic Press, 1971, Chpt. 3.

5. Bryson, A.E. and Ho, Y.-C., Applied Optimal Control, Blaisdell, 1969, Chpt. 12.

6. Gelb, A., Applied Optimal Estimation, The M.I.T. Press, 1974, Chpt. 4.

7. Bryson, A.E. and Ho, Y.-C., op. cit. Chpt. 11 and 14.

8. Kalman, R.E., "Contributions to the Theory of Optimal Control," Bol. de Soc. Math. Mexicana, v. 5, 1960, pp. 102-119.

9. Rutherford, D.E. (1932), cited in Bellman, Introduction to Matrix Analysis, McGraw-Hill, pp. 283.

10. Wonham, W.M., "Random Differential Equations in Control Theory," In Probabilistic Methods in Applied Mathematics. Edited by A.T. Bharucha-Reid, Vol 2, Acadamic Press, N.Y. and London, 1970.

11. MacFarlane, A.G.J., "An Eigenvector Solution to the Optimal Regulator Problem," J. Electronics and Control, v. 14, June 1963, pp. 643-654.

12. Potter, J.E., "Matrix Quadratic Solution," SIAM J. Appl. Math., V. 14, May 1966, pp. 496-501.

13. Repperger, D.W., "A Factorization Solution of the Steady State Matrix Riccati Equation," Proc. 1978 JACC, Oct 18-20, Philadelphia, PA. 1978, V4. pp. 201-213.

REFERENCES (cont)


14. Holley, W.E. and Wei, S.Y., "An Improvement in the MacFarlane-Potter Method for Solving the Algebraic Riccati Equation," Proc. 1979 JACC, Denver, Colorado, June 1979, pp. 921-923.

15. Wilkinson, J.H., The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965, Chpt. 8.

16. Wilkinson, J.H. and Reinsch, C., Linear Algebra, Handbook for Automatic Computation, Vol. II, Springer-Verlag, N.Y., Heidelberg, Berlin, 1971, pp. 359-371.

17. Hohn, F.E., Elementary Matrix Algebra, 2nd Edition, The MacMillan Company, N.Y., 1964, p. 225.

18. Householder, A.S., "Unitary Triangularization of a Non-Symmetric Matrix," J. Assoc. Comput. Mach., 5, 1958, pp. 339-342.

19. Francis, J.C.F., "The QR Transformation - A Unitary Analogue to the LR Transformation," Comput. J., 4, (1961/62) pp. 265-271 and 332-345.

20. Bartels, R.H. and Stewart, G.W., "Algorithm 432 Solution of the Matrix Equation AX+XB=C", COMM ACM, Vol. 15, No. 9, 1972 pp. 820-826.

21. Wilkinson, J.H., (1965) op. cit., Chpt. 3.

22. Park, G.D., Private Communication.

23. Holley, W.E. and Bryson, A.E., "Wind Modeling and Lateral Control for Automatic Landing," AIAA J. of Spacecraft and Rockets, V. 14 (1977), pp. 65-72.

24. Holley, W.E. and Wei, S.Y., "Control Design for Reduc-Reduction of Aerodynamic Tail Forces Using Quadratic Synthesis," Proc. 1979 Asilomar Conference on Circuits, Systems, and Computers, Pacific Grove, CA, November, 1979.

25. Kayton, M. and Fried, W., Avionics Navigation Systems, John Wiley & Son, Inc., N.Y. and London, 1969.

26. Bryson, A.E. and Ho, Y.-C., op. cit., Chpt. 5.

REFERENCES (cont)

27. Wonham, W.M., "On a Matrix Riccati Equation of Stochastic Control," SIAM J. Control, Vol. 6, No. 4, 1968, pp.681-697.

28. Schur, I. (1909) cited in Bellman R., Introduction to Matrix Analysis, McGraw-Hill, 1960, p. 219.

29. Kwakernaak H. and Sivan R., Linear Optimal Control Systems, Wiley - Interscience, New York, 1972, p. 100-103.

# APPENDICES

# APPENDIX A

# APPENDIX A-1

## The Solution of the Linear Quadratic Regulator Problem

Lemma A.1

The problem,

$$\underset{u}{\text{Min}}\; J = \lim_{t_f \to \infty} \frac{1}{2} \int_{t_o}^{t_f} (y^T Q_1 y + u^T R_1 u)\,dt \tag{A.1.1}$$

subject to

$$\dot{x} = Ax + Bu \tag{A.1.2}$$

$$y = C_1 x + Du \tag{A.1.3}$$

is equivalent to [(1)(2)] the problem,

$$\underset{u_*}{\text{Min}}\; J = \lim_{t_f \to \infty} \frac{1}{2} \int_{t_o}^{t_f} (x^T Q_* x + u_*^T R_* u_*)\,dt \tag{A.1.4}$$

subject to

$$\dot{x} = A_* x + Bu_* \tag{A.1.5}$$

where

$$R_* = D_1^T\, Q_1\, D_1 + R_1 \tag{A.1.6}$$

$$C_* = -\, R_*^{-1}\, D_1^T\, Q_1 C_1 \tag{A.1.7}$$

$$A_* = A + B\,C_* \tag{A.1.8}$$

$$Q_* = C_1^T\,Q_1C_1 - C_*^T\,R_*C_* \tag{A.1.9}$$

and

$$u_* = u - C_*\,x \quad . \tag{A.1.10}$$

Proof:

The integrand of the performance criteria (A.1.1) can be expressed in terms of x. Using the output equation (A.1.3), the integrand is given by

$$y^TQ_1\,y + u^T\,R_1\,u$$

$$= (C_1x + D_1u)^T\,Q_1\,(C_1x + D_1u) + u^T\,R_1u$$

$$\begin{aligned} &= x^T\,C_1^T\,Q_1C_1\,x + u^T\,D_1Q_1C_1\,x + x^T\,C_1^T\,Q_1D_1\,u \\ &\quad + u^T(D_1^T\,Q_1D_1 + R_1)u \end{aligned} \tag{A.1.11}$$

Let

$$R_* = D_1^T\,Q_1D_1 + R_1 \tag{A.1.12}$$

$$C_* = -\,R_*^{-1}\,D_1^T\,Q_1C_1 \tag{A.1.13}$$

$$u_* = u - C_*x \tag{A.1.14}$$

$$Q_* = C_1^T\,Q_1C - C_*^T\,R_*\,C_* \quad , \tag{A.1.15}$$

Rearranging equation (A.1.11) results in

$$y^T Q_1 y + u^T R_1 u = x^T Q_* x + u_*^T R_* u_* \quad . \qquad (A.1.16)$$

Combining equation (A.1.14) and A.1.2), the system equation is given by

$$\begin{aligned} \dot{x} &= Ax + Bu \quad . \\ &= Ax + Bu_* + B\,C_*\,x \\ &= (A + B\,C_*)x + Bu_* \\ &= A_* x + Bu_* \end{aligned} \qquad (A.1.17)$$

where

$$A_* = A + B\,C_* \qquad (A.1.18)$$

——— Q.E.D. ———

<u>Lemma A.2</u>

The solution of the problem

$$\underset{u_*}{\mathrm{Min}}\ J = \lim_{t_f \to \infty} \frac{1}{2} \int_{t_o}^{t_f} (x^T Q_* x + u_*^T R_* u_*)dt \qquad (A.1.19)$$

subject to

$$\dot{x} = A_* x + Bu_* \qquad (A.1.20)$$

is given by

$$u_* = - R_*^{-1} B^T S x \tag{A.1.21}$$

where S satisfies the algebraic Riccati equation

$$0 = - SA_* - A_*^T S + SBR_*^{-1} B^T S - Q_* \tag{A.1.22}$$

Proof:

Using the Pontryagin's maximum principle[3], the Hamiltonian of the given system is

$$H = \frac{1}{2}(x^T Q_* x + u_*^T R_* u_*) + \lambda^T (A_* x + Bu_*)$$

where $\lambda$ is the adjoint variable.

The necessary conditions for optimal trajectory are

$$\frac{\partial H}{\partial x} = - \dot{\lambda} = Q_* x + A_*^T \lambda; \ \lambda(t_f) = 0 \tag{A.1.23}$$

$$\frac{\partial H}{\partial u_*} = 0 = R_* u_* + B^T \lambda \ . \tag{A.1.24}$$

From equation (A.1.24), the optimal control is defined as

$$u_* = - R_*^{-1} B^T \lambda \tag{A.1.25}$$

Substituting equation (A.1.25) into (A.1.20) and combining (A.1.20), (A.1.23), the Euler-Lagrange system is given by

$$\begin{bmatrix} \dot{x} \\ \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} A_* & -BR_*^{-1}B^T \\ \\ -Q_* & -A_*^T \end{bmatrix} \begin{bmatrix} x \\ \\ \lambda \end{bmatrix} \quad ; \quad \begin{matrix} x(t_o) \text{ is given} \\ \lambda(t_f) = 0 \end{matrix} \qquad (A.1.26)$$

,

which is a two point boundary value problem.

For state feedback, assume $\lambda$ has the following form

$$\lambda = Sx$$

Substituting the above equation into (A.1.23) results

$$\dot{\lambda} = \dot{S}x + S\dot{x}$$

$$= -A_*^T Sx - Q_*x \quad .$$

Rearrange the above equation and substitute the equation (A.1.20) for x. The following equation results:

$$(\dot{S} + SA_* + A_*^T S - SB R_*^{-1} B^T S + Q_*)x = 0.$$

Since x can not be identically zero, the quantity inside the parenthesis has to be zero, i.e.,

$$\dot{S} = - SA_* - A_*^T S + SBR_*^{-1} B^T S - Q_* \; .$$

It is discussed by Bryson and Ho[26] that, for a stationary system, i.e., A, B are constant matrices, and $Q_*$, $R_*$ are constant matrices, it is possible that a steady state, finite solution exists when $t_f - t_o \to \infty$

$$\dot{S} = 0 = - SA_* - A_*^T S + SB R_*^{-1} B^T S - Q_* .$$

In this case, the optimal control law is given by

$$u_* = - R_*^{-1} B^T S x$$

where S satisfies the above steady state Riccati equation.

———— Q.E.D. ————

From Lemma A.1 and Lemma A.2, the solution of the problem defined by equations (A.1.1), (A.1.2), and (A.1.3) is gven by

$$\begin{aligned} u &= u_* + C_* x \\ &= (C_* - R_*^{-1} B^T S) x \end{aligned} \tag{A.1.27}$$

where S satisfies the algebraic Riccati equation

$$0 = - SA_* - A_*^T S + SB R_*^{-1} B^T S - Q_* . \tag{A.1.28}$$

# APPENDIX A-2

## The Solution of the Stationary Kalman Filter Problem

To solve the stationary Kalman filter defined in Section II.3, one approach[16] is to convert the problem into one with no correlation between noise in the process and in the measurement. To do this, add zero to the right-side of equation (2.3.1), in the form

$$\dot{x} = Ax + Bu + \Gamma w + L(z - C_2 x - D_2 u - \nu - \Theta w)$$

$$= (A - LC_2)x + (B - LD_2)u + Lz + \Gamma w_*$$

where $L = \Gamma Q_2\ \Theta^T(\Theta\ Q_2\ \Theta^T + R_2)^{-1}$ . The filtering problem now under consideration is

$$\dot{x} = A_{**}x + B_*\ u + Lz + \Gamma w_* \qquad \text{(A.2.1)}$$

$$z = C_2\ x + D_2\ u + \nu_* \qquad \text{(A.2.2)}$$

where

$$A_{**} = A - LC_2 \qquad \text{(A.2.3)}$$

$$B_* = B - LD_2 \qquad \text{(A.2.4)}$$

$$L = \Gamma Q_2\ \Theta^T\ R_{2*}^{-2} \qquad \text{(A.2.5)}$$

$$R_{2*} = \Theta Q_2 \ \Theta^T + R_2 \tag{A.2.6}$$

$$w_* = w - Q_2 \ \Theta^T \ R_{2*}{}^{-1} \ \Theta w - Q_2 \ \Theta^T \ R_{2*}{}^{-1} \ \nu \tag{A.2.7}$$

$$\nu_* = \nu + \Theta w \tag{A.2.8}$$

$$Q_{2*} = Q_2 - Q_2 \ \Theta^T \ R_{2*}{}^{-1} \ \Theta Q_2 \tag{A.2.9}$$

With the assumption that $w(t)$ and $\nu(t)$ are uncorrelated, it is not difficult to show that $w_*(t)$ and $\nu_*(t)$ are also uncorrelated. The power spectral densities of $w_*$ and $\nu_*$ can be found as follows. The power spectral density of $\nu_*$ is given by

$$\begin{aligned} F[\nu_*(t+\tau)\nu_*{}^T(t)] &= F[\nu(t+\tau)\nu^T(t)] + F[\nu(t+\tau)w^T(t)\Theta^T] \\ &\quad + F[\Theta \ w(t+\tau) \ \nu^T(t)] \\ &\quad + F[\Theta \ w(t+\tau) \ w^T(t) \ \Theta^T] \ . \end{aligned}$$

where the operator F is defined as the Fourier transform of the expected value.

Since $\nu$ and $w$ are uncorrelated, the above expression is given by

$$\begin{aligned} F[\nu_*(t+\tau)\nu_*{}^T(t)] &= F[\nu(t+\tau) \ \nu^T(t)] + \Theta\{F[w(t+\tau) \ w^T(t)]\}\Theta^T \\ &= R_2 + \Theta \ Q_2 \ \Theta^T \\ &= R_{2*} \ . \end{aligned}$$

Since

$$w_* = w - Q_2 \, \Theta^T \, R_{2*}^{-1} (\Theta \, w + \nu),$$

the same procedure results in

$$F[w_*(t+\tau) w_*^T(t)] = Q_2 - Q_2 \, \Theta^T \, R_{2*}^{-1} \, \Theta \, Q_2$$

$$= Q_{2*} \quad .$$

It is discussed by Sage and Melsa[4] (1972) that the above Kalman filter problem can be formulated in the following lemma.

<u>Lemma A.3</u>

The Kalman filter problem can be formulated as

$$\min_{w_*, \nu_*} J = E \left\{ \lim_{t_o \to -\infty} \frac{1}{2} \int_{t_o}^{t_f} (w_*^T Q_{2*}^{-1} w_* + \nu_*^T R_{2*}^{-1} \nu_*) dt \right\} \quad \text{(A.2.10)}$$

subject to

$$\dot{x} = A_{**} \, x + B_* u + Lz + \Gamma w_* \quad \text{(A.2.11)}$$

$$z = C_2 x + D_2 u + \nu_* \quad \text{(A.2.12)}$$

where the power spectral densities of $w_*$ and $\nu_*$ are given by $Q_{2*}$ and $R_{2*}$ respectively and matrices $A_{**}$, $B_*$, L, $w_*$, $\nu_*$ are defined in equations (A.2.3) through (A.2.9).

The solution is given by:

$$\dot{x}_* = A_{**}\, x_* + B_*\, u + G_*[-z + D_2 u + C_2 x_*] + Lz \qquad \text{(A.2.13)}$$

where the estimated state is $x_*$.

The filter gain $G_*$ is

$$G_* = -\, P\, C_2^{\,T}\, R_{2*}^{\;-1} \qquad \text{(A.2.14)}$$

where P satisfies the algebraic Riccati equation

$$0 = A_{**}\, P + PA_{**}^{\;T} - PC_2^{\,T}\, R_{2*}^{\;-1}\, C_2 P + \Gamma Q_{2*}\, \Gamma^T\, . \qquad \text{(A.2.15)}$$

Proof:

Using the maximum principle[3], the Hamiltonian of the system is

$$H = \frac{1}{2}[w_*^{\,T}\, Q_{2*}^{\;-1}\, w_* + \nu_*^{\,T}\, R_{2*}^{\;-1}\, \nu_*] + \lambda^T[A_{**}x + B_* u$$
$$+ \Gamma w_* + Lz]$$

where $\lambda$ is the adjoint variable.

The necessary conditions for the optimal solution are

$$\frac{\partial H}{\partial x} = - \dot{\lambda} = C_2{}^T R_{2*}{}^{-1} C_2 x + A_{**}{}^T \lambda + C_2{}^T R_{2*}{}^{-1} D_2 u$$
$$-C_2{}^T R_{2*}{}^{-1} z \; ; \; \lambda(t_f) = 0$$

$$\frac{\partial H}{\partial w_*} = 0 = -Q_{2*} \Gamma^T \lambda \qquad \text{(A.2.16)}$$

Subject to

$$\dot{x} = A_{**} x + B_* u + Lz + \Gamma w_*$$

$$z = C_2 x + D_2 u + \nu_* \, .$$

Combining the above equations, the Euler-Lagrange system is given by

$$\begin{bmatrix} \dot{x} \\ -\dot{\lambda} \end{bmatrix} = \begin{bmatrix} A_{**} & \Gamma Q_{2*} \Gamma^T \\ C_2{}^T R_{2*}{}^{-1} C_2 & -A_{**}{}^T \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} + \begin{bmatrix} B_* & L \\ C_2 \; R_{2*}{}^{-1} D_2 & C_2{}^T R_{2*}{}^{-1} \end{bmatrix} \begin{bmatrix} u \\ z \end{bmatrix} .$$

For the filter, assume $\lambda$ has the following form

$$\lambda = - P^{-1}(x - x_*) \qquad \text{(A.2.17)}$$

where $x_*$ is the estimated state with a priori estimate of the state $x_{*o}$ .

Differentiating equation (A.2.17) and combining with equations (A.2.16), the resulting equation turns out to be

$$(-\dot{P}^{-1} - P^{-1}A_{**} - P^{-1}\,\Gamma Q_{2*}\,\Gamma^T P^{-1} + C^{\,T} R_{2*}^{\;-1}\, C_2 - A_{**}^{\;T} P^{-1})$$

$$x + \{(\dot{P}^{-1} + P^{-1}\,\Gamma Q_{2*}\,\Gamma^T P^{-1} + A_{**}\,P^{-1})x_* + P^{-1}\,x_*$$

$$- P^{-1}\,B_*u - P^{-1}\,Lz + C_2^{\;T}\,R_{2*}^{\;-1}\,D_2u - C_2^{\;T}\,R_{2*}^{\;-1}\,z\} = 0$$

Setting the coefficient of the x term to zero, the following equation results after some matrix manipulation

$$\dot{P} = A_{**}P + PA_{**}^{\;T} - PC_2^{\;T}R_{2*}^{\;-1}C_2P + \Gamma Q_{2*}\Gamma^T$$

$$\dot{x}_* = A_{**}x_* + B_*u - PC_2^{\;T}R_2^{\;-1}[-z + D_2u + C_2x_*] + Lz$$

For the stationary case, $\dot{P} \to 0$, and the solution is

$$\dot{x}_* = A_{**}x_* + B_*u + G_*[-z + D_2u + C_2x_*] + Lz$$

The filter gain $G_*$ is

$$G_* = -\,PC_2^{\;T}R_{2*}^{\;-1}$$

where

$$0 = A_{**}P + PA_{**}^{T} - PC_2^{T}R_{2*}^{-1}C_2P + \Gamma Q_{2*}\Gamma^{T}$$

——————— Q.E.D. ———————

From Lemma A.3 and equations (A.2.1) to (A.2.9), the solution of the stationary Kalman filter problem defined in Section II.3 is given by

$$\dot{x}_* = Ax_* + Bu + G[-z + D_2u + C_2x_*]$$

where

$$G = G_* - L$$

$$= -PC_2^{T}R_{2*}^{-1} - \Gamma Q_2\Theta^{T}R_{2*}^{-1}$$

and

$$0 = A_{**}P + PA_{**}^{T} + \Gamma Q_{2*}\Gamma^{T} - PC_2^{T}R_{2*}^{-1}C_2P$$

with

$$R_{2*} = R_2 + \Theta Q_2\Theta^{T}$$

$$A_{**} = A - LC_2$$

$$= A - \Gamma Q_2\,\Theta^{T}\,R_{2*}^{-1}\,C_2$$

$$Q_{2*} = Q_2 - Q_2 \, \Theta^T R_{2*}^{-1} \Theta Q_2 \quad .$$

The uncorrelated property of $x_*$ and the estimated error $\tilde{x} = x - x_*$ is stated as the following lemma.

<u>Lemma A.4</u>

The estimated state $x_*$ and the state estimation error $\tilde{x}$ is uncorrelated, i.e.,

$$X_{12} = E[x_* \, \tilde{x}^T] = 0 \quad .$$

Proof:

Using equations for $x$ and $x_*$ stated above and $u = Kx_*$, the equations of $x_*$ and $\tilde{x}$ are given by

$$\dot{x}_* = (A + BK)x_* - GC_2 x - G(\nu + \Theta w)$$

$$\dot{\tilde{x}} = (A + GC_2)\tilde{x} + \Gamma w + G(\nu + \Theta w) \quad .$$

From the above two equations, $X_{12}$ satisfies the following equation

$$\dot{X}_{12} = (A + BK)X_{12} - GC_2P - GE[(\nu + \Theta w)\tilde{x}^T]$$
$$+ X_{12}(A + GC_2)^T + E[x_*(\nu + \Theta w)^T]G^T$$
$$+ E[x_*w^T]\Gamma^T .$$

It can be shown(6) that

$$E[\nu\tilde{x}^T] = \frac{1}{2} R_2G^T$$

$$E[w\ \tilde{x}^T] = \frac{1}{2} Q_2(\Gamma + G\Theta)^T$$

$$E[x_*\ w^T] = E[x\ w^T] - E[\tilde{x}\ w^T]$$
$$= -\frac{1}{2} G\ \Theta\ Q_2$$

$$E[x_*\ \nu^T] = E[x\ \nu^T] - E[\tilde{x}\ \nu^T]$$
$$= -\frac{1}{2} GR_2 \quad .$$

Substituting above equation into the equation of $X_{12}$ yields

$$\dot{X}_{12} = (A + BK)X_{12} + X_{12}(A + GC_2)^T$$
$$- G[C_2\ P + R_2\ G^T + \Theta\ Q_2\ \Gamma^T + \Theta Q_2\ \Theta^T\ G^T].$$

Using the expression for G given in equation (2.3.5), after some matrix manipulation, the above equation yields

$$\dot{X}_{12} = (A + BK)X_{12} + X_{12}(A + GC_2)^T .$$

For the stationary case, $\dot{X}_{12} = 0$, it yields

$$0 = (A + BK)X_{12} + X_{12}(A + GC_2)^T .$$

It is shown by Kalman[7] in 1960 that, if the system is controllable and observable, the eigenvalues of A+BK and $A+GC_2$ are all in the open left-half plan. The unique solution of the above Lyapunov equation is equal to zero, i.e.,

$$X_{12} = 0 .$$

——— Q.E.D. ———

# APPENDIX B

## SOME PROPERTIES OF THE HAMILTONIAN MATRIX

Lemma B.1

The eigenvalues of the Hamiltonian matrix

$$H = \begin{bmatrix} A & -R \\ -Q & -A^T \end{bmatrix} \quad \text{(R, Q symmetric)}, \qquad \text{(B.1.1)}$$

are symmetric with respect to the imaginary axis in the eigenvalue plane, i.e., if $\lambda$ is a eigenvalue of H, then $-\lambda$ is also a eigenvalue of H.

Proof:

If $\lambda$ is an eigenvalue of H, then there exist a eigenvector, such that

$$\begin{bmatrix} A & -R \\ -Q & -A^T \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} .$$

The above equation can be rewritten as

$$\begin{bmatrix} A^T & -Q \\ -R & -A \end{bmatrix} \begin{bmatrix} x_2 \\ -x_1 \end{bmatrix} = -\lambda \begin{bmatrix} x_2 \\ -x_1 \end{bmatrix} ,$$

i.e., $\lambda$ is the eigenvalue of $H^T$. The eigenvalues are invariant under the transposition of the matrix. Thus $-\lambda$ is an eigenvalue of $H$.

——— Q.E.D. ———

Proofs of the following classical theorems (listed as Lemmas B.2-B.5) will not be included here. The interested reader may refer to the indicated references.

Lemma B.2: (Kalman[8], 1960)

The algebraic Riccati equation

$$SA + A^TS + C^TC - SBB^TS = 0 \qquad (B.1.2)$$

has a unique positive definite solution, S, if the matrices (A,B) are controllable and (A,C) are observable. Also the matrix $A-BB^TS$ has eigenvalues in the open left-half plane. It is pointed out by Wonham[27] in 1968 that the properties of controllability and observability can be changed to stabilizability and detectability respectively.

Lemma B.3: (MacFarlane[11], 1963)

Under the conditions of Lemma B.2 the eigenvalues of the matrix $A-BB^TS$ are the same as the left half plane eigenvalues of the Euler-Lagrange system matrix

$$H = \begin{bmatrix} A & -BB^T \\ & \\ -C^TC & -A^T \end{bmatrix} . \tag{B.1.3}$$

Lemma B.4: (Rutherford[9], 1932)

The solution, X, for the linear matrix equation

$$AX - XB = 0$$

is unique and equal to zero if A and B have no common eigenvalues.

Lemma B.5: (Schur[28], 1909)

There exists an orthogonal transformation

$$\begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$$

which transforms the matrix H into quasi-upper triangular form

$$\begin{bmatrix} U_{11} & U_{12} \\ & \\ 0 & U_{22} \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} \\ & \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} A & -BB^T \\ & \\ -C^TC & -A^T \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} \\ & \\ P_{21} & P_{22} \end{bmatrix}^T$$

From Lemma B.1, B.2, and B.3 the orthogonal transformation matrix can be chosen so that $U_{11}$ has all eigenvalues in the left (right) half plane and $U_{22}$ has all eigenvalues in the right (left) half plane.

Theorem B.1

Under the conditions of Lemma B.1, namely that the given system is controllable and observable, the symmetric positive definite solution of the algebraic Riccati equation (3.2.1) and (3.2.4) satisfies the following relations

a. For the linear regulator problem

$$P_{11} S = P_{12} \tag{B.1.4}$$

where $P_{11}$, $P_{12}$ satisfy the following equation.

$$\begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} A_* & -BR_*^{-1}B^T \\ -Q_* & -A_*^T \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}^T \tag{B.1.5}$$

with $U_{22}$ having all the eigenvalues in the right half plane.

b. For the stationary Kalman filter

$$P_{12}\, P = P_{11} \tag{B.1.6}$$

where $P_{11}$, $P_{12}$ satisfy the following equation

$$\begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} A_{**} & \Gamma Q_{2*} \Gamma^T \\ C_2^T R_{2*}^{-1} C_2 & -A_{**}^T \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}^T \tag{B.1.7}$$

with $U_{22}$ having all eigenvalues in the left half plane.

Proof:

a. Pre-multiplying equation (B.1.5) by the orthogonal matrix yields

$$\begin{bmatrix} P_{11}^T & P_{21}^T \\ P_{12}^T & P_{22}^T \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} = \begin{bmatrix} A_* & -BR_*^{-1}B^T \\ -Q_* & -A_*^T \end{bmatrix} \begin{bmatrix} P_{11}^T & P_{21}^T \\ P_{12}^T & P_{22}^T \end{bmatrix}$$

Expanding yields four equations, two of which are

$$A_* P_{11}^T - BR_*^{-1} B^T P_{12}^T = P_{11}^T U_{11} \tag{B.1.8}$$

$$-Q_* P_{11}^T - A_*^T P_{12}^T = P_{12}^T U_{11} . \qquad (B.1.9)$$

Solving the Riccati equation (3.2.1) for $Q_*$ yields

$$Q_* = S BR_*^{-1} B^T S - SA_* - A_*^T S . \qquad (B.1.10)$$

Using (B.1.10) in (B.1.9) gives

$$-(SBR_*^{-1} B^T S - SA_* - A_*^T S) P_{11}^T - A^T{}_* P_{12}^T = P_{12}^T U_{11} . \qquad (B.1.11)$$

Using (B.1.8) in (B.1.11) yields

$$-SBR_*^{-1} B^T S P_{11}^T + S(P_{11}^T U_{11} + BR_*^{-1} B^T P_{12}^T)$$
$$+ A_*^T SP_{11}^T - A_*^T P_{12}^T = P_{12}^T U_{11}$$

which can be rearranged to give

$$(A_* - BR_*^{-1} B^T S)^T (SP_{11}^T - P_{12}^T) + (SP_{11}^T - P_{12}^T) U_{11} = 0 . \qquad (B.1.12)$$

From Lemma B.2, B.3, B.4, B.5

$$SP_{11}^T - P_{12}^T = 0$$

or

$$P_{11} S = P_{12} \quad . \quad \text{Q.E.D.}$$

b. The same procedure can be followed for the Kalman filter case:

Expanding equation (B.1.7) yields

$$A_{**}P_{11}^T + \Gamma Q_{2*}\Gamma^T P_{12}^T = P_{11}^T U_{11} \tag{B.1.13}$$

$$C_2^T R_{2*}^{-1} C_2 P_{11}^T - A_{**}^T P_{12}^T = P_{12}^T U_{11}. \tag{B.1.14}$$

Solving the Riccati equation (3.2.4) for $\Gamma Q_{2*}\Gamma^T$ yields

$$\Gamma Q_{2*}\Gamma^T = PC_2^T R_{2*}^{-1} C_2 P - A_{**}P - PA_{**}^T. \tag{B.1.15}$$

Using (B.1.15) in (B.1.13) gives

$$A_{**}P_{11}^T + PC_2^T R_{2*}^{-1} C_2 PP_{12}^T - A_{**}PP_{12}^T$$
$$- PA_{**}^T P_{12}^T = P_{11}^T U_{11}. \tag{B.1.16}$$

Using (B.1.14) in (B.1.16) yields

$$A_{**}P_{11}^T + PC_2^T R_{2*}^{-1} C_2 PP_{12}^T - A_{**}PP_{12}^T$$
$$- PC_2^T R_{2*}^{-1} C_2 P_{11}^T + PP_{12}^T U_{11} = P_{11}^T U_{11}$$

which can be rearranged to give

$$(A_{**} - PC_2{}^T R_{2*}{}^{-1} C_2)(P_{11}{}^T - PP_{12}{}^T) - (P_{11}{}^T - PP_{12}{}^T)U_{11} = 0 .$$

From Lemma B.2, B.3, B.4, B.5

$$P_{11}{}^T - PP_{12}{}^T = 0$$

or

$$P_{12}\, P = P_{11} .$$

——— Q.E.D. ———

<u>Theorem B.2</u>

Under the conditions of Theorem B.1, the matrices $P_{11}$ and $P_{12}$ in equations (B.1.4) and (B.1.6) are non-singular.

Proof:

Since the same procedure can be followed for the Kalman filter case, only the regulator case will be proved here. Expanding equation (B.1.5) yields four equations, one of which is

$$U_{11} = P_{11} A_* P_{11}{}^T - P_{11} B R_*{}^{-1} B^T P_{12}{}^T$$
$$- P_{12} Q_* P_{11}{}^T - P_{12} A_*{}^T P_{12}{}^T . \qquad \text{(B.1.17)}$$

Using equation (B.1.10), (B.1.4) in (B.1.17) gives

$$U_{11} = P_{11}(I + S^2)(A_* - BR_*^{-1}B^TS)P_{11}^T . \qquad (B.1.18)$$

Since $U_{11}$, $S$, and $A_*-BR_*^{-1}B^TS$ are non-singular, $P_{11}$ must also be non-singular. In addition, $P_{12} = P_{11}\ S$ is non-singular.

——— Q.E.D. ———

Theorem B.3

If C is symmetric negative definite and if A has all eigenvalues in the open left half plane, then the solution of the Lyapunov equation

$$AX + XA^T = C \quad , \qquad (B.1.19)$$

X, is symmetric positive definite.

Proof:

Using classical system theory, the solution of equation (B.1.19), as described by Kwakernaak and Sivan(29), is given by

$$X = -\int_0^\infty e^{At}\ C\ e^{A^Tt}\ dt$$

when the system is asymptotically stable.

The matrix is symmetric positive definite since C is symmetric negative definite.

——— Q.E.D. ———

# APPENDIX C

## THE ERROR ANALYSIS OF THE SIMILARITY REDUCTION TO QUASI-TRIANGULAR FORM

# APPENDIX C-1

## Error Analysis of the Algorithm Proposed in Chapter III

In this appendix, the following topics will be discussed:

1. The error analysis of the quasi-traingularization using the QR algorithm
2. The error analysis of the Householder type similarity transformation
3. The operations count of the program which solves the algebraic Riccati equation (the program is presented in Appendix G).

The following useful error bounds for some basic floating-point computations, which are presented by Wilkinson[21] [1965], are stated without proof. The interested reader may refer to the reference. The notation $fl(\cdot)$ is the result of floating point computation, $|\cdot|$ is absolut value, $||\cdot||_2$ is the 2-norm of the matrix, $t_1$ is defined by the relation $2^{-t_1} = (1.06)2^{-t}$ with the t-digit mantissa machine.

$$\text{(i)} \quad fl\left(\prod_{i=1}^{n} x_i\right) = (1 + E) \prod_{i=1}^{n} x_i \tag{C-1}$$

where $|E| < (n - 1)\ 2^{-t_1}$ .

(ii) $fl\left(\sum_{i=1}^{n} x_i\right. = \sum_{i-1}^{n} \{x_i\ (1 + E_i)\}$ (C-2)

where $|E_1| < (n-1)2^{-t_1}$, $|E_i| < (n+1-i)2^{-t_1}$ $i=2,\ldots n$.

(iii) $fl(\sum_{i-1}^{n} x_i y_i) = \sum_{i=1}^{n} \{x_i y_i\ (1 + E_i)\}$ (C-3)

where $|E_1| < n2^{-t_1}$, $|E_i| < (n+2-i)2^{-t_1}$ $i=2, \ldots, n$.

(iv) $fl\ (A + B) = A + B + F$ (C-4)

where $||F||_2 \leq 2^{-t}\ ||A + B||_2$.

(v) $fl\ (AB) = AB + F$ (C-5)

where $||F||_2 \leq n^2\ 2^{-t_1}\ ||A||_2\ ||B||_2$.

There are two important properties of the 2-norm of a matrix which is formed by the outer product of two vectors.

<u>Lemma C.1</u>

Let u and $\delta u$ be two column vectors. The following two identities hold:

(a) $||uu^T||_2 = ||u||_2^{\,2}$ (C-6)

(b) $||u\ \delta u^T||_2 = ||\delta u||_2\ ||u||_2$. (C-7)

Proof:

From the definitions of the 2-norm of a matrix and a vector, the following argument results:

$$||uu^T||_2 = [\max \lambda\ (uu^T uu^T)]^{1/2}$$

$$= [(u^T u) \max \lambda\ (uu^T)]^{1/2}$$

$$= ||u||_2 \ [\max \lambda\ (uu^T)]^{1/2} , \qquad \text{(C-8)}$$

where $\lambda(A)$ is an eigenvalue of A.

Since $uu^T$ is symmetric, the eigenvalues of $uu^T$ are equal to the square roots of the eigenvalues of $uu^T uu^T$, i.e.,

$$\lambda(uu^T) = [\lambda\ (uu^T\ uu^T)]^{1/2} . \qquad \text{(C-9)}$$

Substitute (C-9) into (C-8),

$$||uu^T||_2 = ||u||_2 \ [\max \lambda\ (uu^T\ uu^T)^{1/2}]^{1/2}$$

$$= ||u||_2 \ \{\ ||uu^T||_2^{1/2}\ \} ,$$

i.e., $||uu^T||_2^{1/2} = ||u||_2$, which is equation (C-6).

The same procedure can be followed for equation (C-7)

$$||u\,\delta u^T||_2 = [\max\lambda\ (u\,\delta u^T\ \delta u u^T)]^{1/2}$$

$$= [(\delta u^T \delta u)\ \max\lambda\ (uu^T)]^{1/2}$$

$$= ||\delta u||_2\ \{\ [\max\lambda\ (uu^T\ uu^T)]^{1/2}\ \}^{1/2}$$

$$= ||\delta u||_2\ \{\ ||uu^T||_2\}^{1/2}\ .$$

Substitute (C-6) into the above equality

$$||u\delta u^T||_2 = ||\delta u||_2\ \{\ ||u||_2^2\ \}^{1/2}$$

$$= ||\delta u||_2\ ||u||_2\ ,$$

the equation (C-7) results.

——————— Q.E.D. ———————

Lemma C.2

The floating point computation, fl [ ], of matrix multiplication is given by

$$fl\ (A_1A_2\ \ldots\ A_s) = A_1A_2\ \ldots\ A_s + F$$

where $||F||_2 \leq 1.06\ (s-1)\ n^2\ 2^{-t}\ [\prod_{j=1}^{s}\ ||A_j||_2]$ if $s \cdot 2^{-t} < 0.01$ .

Proof:

Analogous to Wilkinson[21] (1965), an error bound for the multiplication of two n by n matrices is given by

$$fl(AB) = AB + F$$

where $\quad ||F||_2 \le (1.01)n^2\ 2^{-t}\ ||A||_2\ ||B||_2$, if $n2^{-t} \le .01$.

The error bound for multiplication of several matrices is then given by

$$\begin{aligned} fl(A_1A_2 \ldots A_s) &= fl[fl(A_1 \ldots A_{s-1})A_s] \\ &= fl(A_1 \ldots A_{s-1})A_s + F_s \\ &= fl(A_1 \ldots A_{s-2})A_{2-1}A_s + F_{2-1}A_s + F_s \\ &\ \ \vdots \\ &= A_1 \ldots A_{s-1}A_s + F_2A_3A_4 \ldots A_s \\ &\quad + F_3A_4A_5 \ldots A_s + \ldots + F_{s-1}A_s + F_s \, , \end{aligned}$$

where $\quad ||F_i||_2 \le (1.06)n^2\ 2^{-t}\ [\prod_{j=1}^{n} ||A_j||_2]$.

The above expression can be rearranged as

$$fl(A_1 \ldots A_s) = A_1 \ldots A_s + F$$

where $\quad ||F||_2 = ||F_2A_3A_4 \ldots A_s + \ldots + F_{s-1}A_s + F_s||_2$

$$\leq (s-1)(1.06)n^2\ 2^{-t}\ [\prod_{j=1}^{s} ||A_j||_2] \quad .$$

——— Q.E.D. ———

The following error analysis of the QR quasi-triangularization follows the work done by Wilkinson[21] in 1965. The error bound for a sequence of orthogonal similarity transformations can be stated as a therorem.

Theorem C.1

Let $\bar{A}_s$ be the computed result of the similarity transformation; then

$$\bar{A}_s = G_1^T\ (E + A_o)\ G_1\ , \qquad \text{(C-10)}$$

where

$$G_1 = Q_1\ Q_2\ \ldots\ Q_s \qquad \text{(C-11)}$$

and $Q_i$ is the orthogonal matrix corresponding to the exact application of the pth step of the algorithm to $\bar{A}_{p-1}$ . Also, let the computed $\bar{Q}_i$ correspnding to $Q_i$ be given by

$$\bar{Q}_i = Q_i + e_i \tag{C-12}$$

where

$$||e_i||_2 \leq a(i,\ n)2^{-t} . \tag{C-13}$$

If the error bound of the matrix $F_p$ which satisfies the equation

$$\bar{A}_p = \bar{Q}_p^{\ T} \bar{A}_{p-1} \bar{Q}_p + F_p \tag{C-14}$$

is of the form

$$||F_p||_2 \leq f(p,n)2^{-t} ||\bar{A}_{p-1}||_2, \tag{C-15}$$

then, the error bound of E is given by

$$||E||_2 \leq 2^{-t} ||A_o||_2 \sum_{p=1}^{s} x_p \tag{C-16}$$

where

$$x_p = [2a(p,n) + a^2(p,n)2^{-t} + f(p,n)]y_{p-1} \tag{C-17}$$

and

$$y_p = \prod_{i=1}^{p} \{ (1 + a(p,n)2^{-t})^2 + f(i,n)2^{-t} \}$$

$$y_o = 1 . \qquad \text{(C-18)}$$

The values a (p,n) and f(p,n) depend on the algorithm used.

Proof:

Let $Q_p$ be the exact orthogonal matrix corresponding to the exact application of the $p^{th}$ step of the algorithm to the computed transformed matrix at the (p-1)st step $\bar{A}_{p-1}$. The computed matrix $\bar{Q}_p$ correponding to the matrix $Q_p$ satisfies the relation

$$\bar{Q}_p = Q_p + e_p . \qquad \text{(C-19)}$$

At the pth step, the orthogonal similarity transformation can be written as

$$\bar{A}_p = \bar{Q}_p^{\,T} \bar{A}_{p-1} \bar{Q}_p + F_p \qquad \text{(C-20)}$$

where $F_p$ is the difference between the accepted $\bar{A}_p$ and the exact product $\bar{Q}_p^{\,T} \bar{A}_{p-1} \bar{Q}_p$ . Substituting (C-19) into (C-20), the resulting equation is given by

$$\bar{A}_p = (Q_p + e_p)^T \bar{A}_{p-1} (Q_p + e_p) + F_p$$

$$= Q_p{}^T \bar{A}_{p-1} Q_p + Y_p \tag{C-21}$$

where

$$Y_p = Q_p{}^T \bar{A}_{p-1} e_p + e_p{}^T \bar{A}_{p-1} Q_p$$
$$+ e_p{}^T A_{p-1} e_p + F_p$$

and $Q_p{}^T \bar{A}_{p-1} Q_p$ is an exact orthogonal similarity transformation of $\bar{A}_{p-1}$. Combining equation (C-21) for p = 1,2, ... s, we have

$$\bar{A} = Y_s + G_s{}^T Y_{s-1} G_s + G_{s-1}{}^T Y_{s-2} G_{s-1} + \ldots$$
$$+ G_2{}^T Y_1 G_2 + G_1{}^T A_o G_1 \tag{C-23}$$

where $\quad G_p = Q_p Q_{p+1} \ldots Q_s \; .$ (C-24)

Equation (C-23) can be rewritten as

$$\bar{A}_s = Y + G_1{}^T A_o G_1 \tag{C-25}$$

with

$$Y = Y_s + G_s Y_{s-1} G_s^T + G_{s-1} Y_{s-2} G_{s-1}^T + \ldots G_2 Y_1 G_2^T \tag{C-26}$$

or alternatively

$$\overline{A}_s = G_1^T (E + A_o) G_1 \tag{C-27}$$

with

$$E = L_s Y_s L_s^T + L_{s-1} Y_{s-1} L_{s-1}^T + \dots + L_1 Y_1 L_1^T \tag{C-28}$$

and

$$L_p = Q_1 Q_2 \dots Q_p \quad . \tag{C-29}$$

Since $L_p$ is exactly orthogonal for p=1, 2, ... s, the 2-norm of $L_p$, $||L_p||_2$, is equal to one. Taking the 2-norm of equation (C-28), we obtain

$$\begin{aligned} ||E||_2 &\leq \sum_{p=1}^{s} \{ ||L_p||_2 \; ||Y_p||_2 \; ||L_p^T||_2 \} \\ &= \sum_{p=1}^{s} ||Y_p||_2 \quad . \end{aligned} \tag{C-30}$$

Assuming that the error bound of $e_p$ can be formulated as

$$||e_p||_2 \leq a(p,n) 2^{-t} \tag{C-31}$$

equation (C-22) gives

$$||Y_p|| \leq ||\overline{A}_{p-1}||_2 \{2a(p,n)2^{-t} + a^2(p,n)2^{-2t}\} + ||F_p||_2 \tag{C-32}$$

where $||Q_p||_2 = 1$, since $Q_p$ is orthogonal.

Using (C-32), equation (C-21) gives

$$||\bar{A}_p||_2 \leq ||\bar{A}_{p-1}||_2 + ||Y_p||_2$$

$$\leq (1+a(p,n)2^{-t})^2 \; ||\bar{A}_{p-1}||_2 + ||F_p||_2 \; . \qquad \text{(C-33)}$$

Also, if the error bound of $F_p$ can be formulated as

$$||F_p||_2 \leq f(p,n)2^{-t} \; ||\bar{A}_{p-1}||_2 \; , \qquad \text{(C-34)}$$

where f(p,n) is some function of p and n, then combing (C-32), (C-33), (C-34), gives

$$||\bar{A}_p||_2 \leq \{ \; (1+a(p,n)2^{-t})^2 + f(p,n)2^{-t} \; \} \; ||A_{p-1}||_2$$

$$\leq \prod_{i=1}^{p} \{ \; (1+a(i,n)2^{-t})^2 + f(i,n)2^{-t} \; \} \; ||A_o||_2 \qquad \text{(C-35)}$$

and

$$||Y_p||_2 \leq \{ \; 2a(p,n) + a^2(p,n)2^{-t} + f(p,n)\} \; 2^{-t} \; ||A_{p-1}||_2 \; . \qquad \text{(C-36)}$$

Substituting (C-35), (C-36) into (C-30), an a priori bound for the norm of the equivalent perturbation E in $A_o$ (see equation (C-27)) is given by

$$||E||_2 \le 2^{-t} \; ||A_o||_2 \sum_{p=1}^{s} x_p, \tag{C-37}$$

where

$$x_p = [2a(p,n) + a^2(p,n)2^{-t} + f(p,n)]y_{p-1} \tag{C-38}$$

and

$$y_p = \prod_{i=1}^{p} \{ (1 + a(i,n)2^{-t})^2 + f(i,n)2^{-t}\} . \tag{C-39}$$

The problem is reduced to finding expressions for a(p,n) and f(p,n) for the particular algorithm used.

Q.E.D.

<u>Theorem C.2</u>

A bound for the difference between the computed $\overline{Q}$ and the exact Q is given by

$$||e||_2 = ||\overline{Q} - Q||_2$$

$$\le (4.8\ell - 11.2)2^{-t} , \tag{C-40}$$

where

$$Q = I - \frac{uu^T}{2K^2} , \tag{C-41}$$

$$u^T = (x_1 \mp S,\ x_2,\ x_3,\ \ldots,\ x_\ell)$$

$$2K^2 = S^2 \mp x_1 S \qquad \text{(C-42)}$$

$$S^2 = x_1^2 + x_2^2 + \ldots + x_\ell^2 \ .$$

Proof:

The algorithm used to quasi-triangularize the Hamiltonian matrix is the Householder reduction. The orthogonal similarity transformation matrix is of the form

$$Q = I - \frac{uu^T}{2K^2} \ . \qquad \text{(C-43)}$$

where,

$$u^T = (x_1 \mp S,\ x_2,\ x_3,\ \ldots,\ x_\ell)$$

$$2K^2 = S^2 \mp x_1 S \qquad \text{(C-44)}$$

$$S^2 = x_1^2 + \ldots. + x_\ell^2 \ .$$

The error of the computed $\overline{Q}$ is given by

$$e = \overline{Q} - Q$$

$$= \frac{uu^T}{2K^2} - \frac{\overline{u}\overline{u}^T}{2\overline{K}^2} \qquad (C-45)$$

where $\overline{u}$, $\overline{K}$ are computed values of u and K respectively. By using the equation (C-3)

$$b = fl(x_1^2 + \ldots + x_\ell^2)$$

$$= \sum_{i=1}^{\ell} \{ x_i^2 \ (1 + E_i) \} \qquad (C-46)$$

$$|E_1| < \ell \ 2^{-t_1} \ , \ |E_r| < (\ell + 2-r) 2^{-t_1} \text{ for } r = 2, \ldots \ell \ ,$$

the following equation results

$$b = fl(x_1^2 + \ldots + x_\ell^2)$$

$$= (x_1^2 + \ldots + x_\ell^2) \ (1 + E) \qquad (C-47)$$

$$= S^2 (1 + E)$$

with

$$|E| < \ell \ 2^{-t_1} . \qquad (C-48)$$

The computed $\overline{S}$ is given by

$$\bar{S} = fl\ (\sqrt{x_1^{\ 2} + \ldots + x_\ell^{\ 2}}) = fl\,(b^{1/2})$$

$$= b^{1/2}\ (1 + \eta)$$

where we shall assume that

$$|\eta| < (1.00001)\,2^{-t}\ \text{(Wilkinson}^{(21)}\ 1965)\ . \qquad \text{(C-49)}$$

Substitute (C-47) into the above equation, the resulting equation is given by

$$\bar{S} = S(1 + E)^{1/2}\ (1 + \eta) \qquad \text{(C-50)}$$

where

$$= S(1 + \zeta)$$

$$(1 + \zeta) = (1 + E)^{1/2}\ (1 + \eta)$$

$$= [1 + .5E + \sigma(E^2)]\ (1 + \eta)$$

$$= [1 + .5E + \eta + \sigma(E^2) + \sigma(E\eta) + \ldots\ ]$$

$$\leq [1 + .500001\ E + \eta]\ . \qquad \text{(C-51)}$$

In the above equation, the reasonable assumption

$$\sigma(E^2) + \sigma(E\eta) + \ldots \leq .000001\ E$$

is made. From equation (C-48),(C-49),(C-51), the following inequality can be formed

$$|\zeta| \leq .500001\ |E| + |\eta|$$

$$\leq (.500001)\ \ell\ 2^{-t_1} + (1.00001)2^{-t}$$

$$\leq [(.500001)(1.06)\ell + 1.00001]2^{-t}$$

$$\leq [.530001\ \ell + 1.00001]2^{-t} \quad .$$

Equation (C-50) can be rewritten as

$$\bar{S} = S(1 + \zeta) \tag{C-52}$$

with

$$|\zeta| \leq [.53001\ \ell + 1.00001]2^{-t} \quad . \tag{C-53}$$

The computed K is given by

$$2\bar{K}^2 = fl((x_1^{\ 2} + \ldots + x_\ell^{\ 2}) + x_1\ \bar{S})$$

$$= [x_1^{\ 2}\ (1+\Theta_1) + \ldots + x_\ell^{\ 2}(1+\Theta_\ell) + x_1\ \bar{S}(1+\Theta_{\ell+1})]$$

where

$$|\Theta_1| < (\ell+1)2^{-t_1}, \quad |\Theta_r| < (\ell+3-r)2^{-t_1} \text{ for } r=2, \ldots, \ell+1$$

so that

$$2\overline{K}^2 = (x_1{}^2 + \ldots + x_\ell{}^2 + x_1 \overline{S})(1 + \Theta)$$

$$= [S^2 + x_1 S(1 + \zeta)] (1 + \Theta)$$

$$= [S^2 + x_1 S] (1 + \delta)(1 + \Theta) \qquad \text{(C-54)}$$

where

$$|\Theta| < (\ell+1)2^{-t_1}$$

and

$$\delta = \frac{x_1 S}{S^2 + x_1 S} \zeta$$

$$\leq \frac{1}{2} \zeta \qquad (\text{since } x_1 \leq S).$$

Let us define $\gamma$ by

$$(1 + \gamma) = (1 + \delta)(1 + \Theta)$$

$$= 1 + \delta + \Theta + \delta\Theta$$

then

$$|\gamma| \le |\delta| + |\Theta| + |\delta\Theta|$$

$$\le \frac{1}{2}\,(.530001\ell + 1.00001)2^{-t} + 1.06(\ell+1)2^{-t}$$

$$+ \{\, \frac{1}{2}(.530001\ell + 1.00001) \cdot 1.06(\ell+1) \,\}\, 2^{-2t}$$

$$\le (1.325001\ell + 1.560005)2^{-t} \,.$$

Again, the reasonable assumption

$$\{\, \frac{1}{2}\,(.530001\ell + 1.00001)(1.06)(\ell+1)\}\, 2^{-t} \le .0000005\ell$$

is made. The equation (C-54) can then be rewritten as

$$2\bar{K}^2 = 2K^2\,(1 + \gamma) \qquad \text{(C-55)}$$

with

$$|\gamma| \le (1.325001\ell + 1.560005)2^{-t} \,. \qquad \text{(C-56)}$$

From equation (C-44)

$$\bar{u}_1 = fl(x_1 + \bar{S})$$

$$= (x_1 + \bar{S})(1 + \phi)$$

$$= [x_1 + S(1 + \zeta)] \ (1 + \phi)$$

$$= (x_1 + S)(1 + \psi)$$

$$= u_1(1 + \psi)$$

where

$$|\phi| \leq 2^{-t}$$

and

$$|\psi| \leq |\zeta| + |\phi| + |\zeta\phi|$$

$$\leq (.530001 \ \ell + 2.00002)2^{-t}$$

with the reasonable assumption that

$$(.53001 \ \ell + 1.000001)2^{-t} \leq .00001 \quad .$$

The computed $\bar{u}$ can then be written as

$$\bar{u}^T = u^T + \delta u^T \tag{C-57}$$

with

$$\delta u^T = (\delta u_1, 0, \ldots , 0) \tag{C-58}$$

$$\delta u_1 = u_1 \psi \tag{C-59}$$

$$||\delta u||_2 = |u_1| \; |\psi|$$

$$\leq (.530001\, \ell + 2.00002) 2^{-t} \; ||u||_2 , \tag{C-60}$$

and

$$||\bar{u}||_2 = ||u + \delta u||_2$$

$$\leq \{ 1 + (.530001\, \ell + 2.00002) 2^{-t} \} \; ||u||_2 . \tag{C-61}$$

Equation (C-45) can be rewritten as

$$e = \bar{Q} - Q$$

$$= \frac{uu^T}{2K^2} - \frac{\overline{uu}^T}{2\bar{K}^2}$$

$$= \frac{uu^T - \overline{uu}^T}{2K^2} + \overline{uu}^T \left( \frac{1}{2K^2} - \frac{1}{2\bar{K}^2} \right)$$

$$= \frac{-(u\delta u^T + \delta u u^T + \delta u \delta u^T)}{2K^2} + \frac{\overline{uu}^T}{2K^2} \left(1 - \frac{2K^2}{2K^2(1+\gamma)} \right)$$

so that,

$$||e||_2 = ||\bar{Q} - Q||_2$$

$$\leq \frac{2||\delta u u^T||_2 + ||\delta u \delta u^T||_2}{2K^2} + \frac{||\overline{uu}^T||_2}{2K^2} \left|(1 - \frac{1}{1+\gamma})\right| .$$

Using the results stated in Lemma C.1, the above inequality becomes

$$||e||_2 \leq \frac{2||u||_2\ ||\delta u||_2 + ||\delta u||_2^2}{2K^2} + \frac{||\bar{u}||_2^2}{2K^2} \left|(1 - \frac{1}{1+\gamma})\right|$$

$$\leq \frac{2(.2650005\ell + 1.50001)2^{-t}\ ||u||_2^2 + ||\delta u||_2^2}{2K^2}$$

$$+ \frac{\{1 + (.2650005\ell + 1.50001)2^{-t}\}^2\ ||u||_2^2}{2K^2} \left|(1 - \frac{1}{1+\gamma})\right| .$$

But

$$\left|1 - \frac{1}{1+\gamma}\right| = |1 - (1 - \gamma + \sigma(\gamma^2)|$$

$$\leq |\gamma + \sigma(\gamma^2)|$$

$$\leq |\gamma| + |\sigma(\gamma^2)| ,$$

from equation (C-56) with the assumptions that

$$\sigma(\gamma^2) \leq .000001\ell\ 2^{-t}$$

and

$$||\delta u||_2^2 \leq .000001\ell\ 2^{-t}\ ||u||_2^2\ ,$$

the error bound of $||e||_2$ is given by

$$||e||_2 \leq \frac{(.530002\ell + 3.00002)2^{-t}\ ||u||_2^2}{2K^2}$$

$$+ \frac{\{\ 1 + (.2650005\ell + 1.50001)2^{-t}\ \}^2\ ||u||_2^2}{2K^2}$$

$$\cdot\ (1.325002\ell + 1.560005)2^{-t}\ .$$

Since

$$\frac{||u||_2^2}{2K^2} = 2,$$

the above inequality can be rewritten as

$$||e||_2 \leq (2.120006\ell + 8.00008)2^{-t}$$

$$+2(1.325003\ell + 1.560006)2^{-t}\quad ,$$

where the assumption

$$\{\ 2(.530001\ell + 2.00002)2^{-t} + (.530001\ell + 2.00002)^2$$

$$\cdot\ 2^{-2t}\ \}\ (1.325002\ell + 1.560006)2^{-t}$$

$$\leq .000001\ell\ 2^{-t}$$

is made. The error bound of e is thus given by

$$||e||_2 = ||\bar{Q} - Q||_2$$

$$\leq (4.77002\ell + 11.12009)2^{-t} . \qquad \text{(C-62)}$$

The quantity $a(p,\ell)$ in equation (C-37), (C-38), (C-39) is then given by

$$a(p,\ell) = 4.77002\ell + 11.12009 \qquad \text{(C-63)}$$

———— Q.E.D. ————

The error analysis of the QR algorithm will be discussed next.

<u>Theorem C.3</u>

The quasi-upper-triangularization of the Hamiltonian matrix stated in Chapter III is formulated in general as

$$\bar{A}_s = G_1{}^T (E + A_o) G_1 , \qquad \text{(C-64)}$$

where the a priori error is bounded by

$$||E||_2 \leq 2^{-t} ||A_o|| \{ 8kn^3 + (38.2 + 4k)n^2$$

$$+ 44.5n + 50.9 + (4kn^2 + 50.9)s \} \qquad \text{(C-65)}$$

with

$$k = 2.12$$

and s, the number of QR iterations, is given by

(i) if all the eigenvalues are real

$$s = 3n^2 - n$$

(ii) if all the eigenvalues are complex

$$s = \frac{3}{2} n^2 .$$

Proof: From equation (C-5), we have

$$\begin{aligned} fl(ABC) &= fl(fl(AB)C) \\ &= fl(AB)C + F_2 \\ &= ABC + F_1C + F_2 \end{aligned}$$

where m is the dimension of matrices A, B, C, and

$$||F_1||_2 \leq m^2 \, 2^{-t_1} \, ||A||_2 \, ||B||_2$$

$$\begin{aligned} ||F_2||_2 &\leq m^2 \, 2^{-t_1} \, ||AB||_2 \, ||C||_2 \\ &\leq m^2 \, 2^{-t_1} \, ||A||_2 \, ||B||_2 \, ||C||_2 , \end{aligned}$$

so that,

$$fl(ABC) = ABC + F$$

with

$$||F||_2 \leq 2m^2 \; 2^{-t_1} \; ||A||_2 \; ||B||_2 \; ||C||_2 \quad .$$

If $A_{p-1}$ is upper Hessenberg and $Q_p$ is orthogonal, then

$$fl(Q_p{}^T A_{p-1} Q_p) = Q_p{}^T A_{p-1} Q_p + F_p \tag{C-66}$$

with

$$||F_p||_2 \leq k \; m^2 \; 2^{-t} \; ||A_{p-1}||_2 \tag{C-67}$$

and

$$k = 2.12 \; .$$

Then quantity $f(p,m)$ in equation (C-37), (C-38), (C-39), is, then, given by

$$f(p,m) = k \; m^2 \tag{C-68}$$

where

$$k = 2.12 \; .$$

The algorithm proposed in Chapter III for quasi-upper triangularizing the Hamiltonian matrix has two major steps. First, m-1 Householder type similarity transformations are performed to transform the original Hamiltonian matrix into an upper Hessenberg matrix. Second, the QR algorithm is performed on the upper Hessenberg matrix. The error bound of E in equation (C-37) can be described as

$$||E||_2 \leq 2^{-t} \; ||A_o||_2 \; \{ \; \delta_o + \sum_{p=1}^{s} \delta_p \; \} \qquad \text{(C-69)}$$

where s is the number of similarity transformations required for quasi-triangularization

$\delta_o$ is due to the Hessenberg reduction and

$\delta_p$ is due to the pth iteration of QR algorithm.

The quantity $\delta_o$ is given by

$$\delta_o = \sum_{i=1}^{m-1} \{ \; [2a(i,m) + a^2(i,m)2^{-t} + km^2] \cdot y_{i-1} \} \qquad \text{(C-70)}$$

$$y_{i-1} = \prod_{j=1}^{i-1} \{ \; [1 + a(j,m)2^{-t}]^2 + km^2 \; 2^{-t} \; \} \qquad \text{(C-71)}$$

where m is the dimension of the Hamiltonian matrix.

The quantity $\delta_p$ can be found as follows. At the pth step, $m-n_e-1$ Householder type similarity transformations are required. Here $n_e$ is the number of eigenvalues which have already been isolated. For simplicity, $n_2$ is assumed to have the following form

$n_e = P - 1$ if all the eigenvalues are real

$n_e = 2(p - 1)$ if all the eigenvalues are complex.

Further more, if the eigenvalues are known a priori, it is estimated that the number of iterations, ITE, required to isolate a real (or complex pair of) eigenvalue(s) is

ITE = 2 if a real eigenvalue is to be isolated

ITE = 3 if a pair of eigenvalues is to be isolated.

Under these assumptions, $\delta_p$ can be expressed as

$$\delta_p = \{ 2a(p,3) + a^2(p,3)2^{-t} + km^2 \} y_{p-1} \tag{C-72}$$

$$y_{p-1} = \prod_{i=1}^{p-1} \{ [1 + a(i,3)2^{-t}]^2 + km^2 2^{-t} \} . \tag{C-73}$$

The number of similarity transformations required for triangularization is given by

$$s = \sum_{p=1}^{k_1} \text{ITE} * (m - n_e - 1) \tag{C-74}$$

where, for isolating n eigenvalues.

$k_1 = n$ if all the eigenvalues are real

$k_1 = \frac{n}{2}$ if all the eigenvalues are complex.

The reason for a(i,3) is that the QR algorithm is applied to a the Hessenberg matrix. Combing equation (C-63), (C-70), (C-71), gives

$$\delta_o = \sum_{i=1}^{m-1} \{ [9.54004m + 22.24081 + km^2 + (22.75309m^2 + 106.0861m + 123.6564)2^{-t}] \cdot y_{i-1} \} .$$

It is reasonable to assume that

$$(22.75309m^2 + 106.0861m + 123.6564)2^{-t} \leq .00001m$$

and

$$[1 + a(i,m)2^{-t}]^2 + km^2\ 2^{-t} \leq 1.000001 .$$

The expression of $\delta_o$ can then be simplified to be

$$\delta_o \leq \sum_{i=1}^{m-1} \{ [9.54005m + 22.24018 + km^2] \prod_{j=1}^{i-1} (1.000001] \}$$

$$= [9.54005m + 22.24018 + km^2] \sum_{i=1}^{n-1} (1.000001)^{i-1} .$$

With the aid of the identify

$$\sum_{i=1}^{m-1} (1.000001)^{i-1} = \sum_{j=0}^{m-2} (1.000001)^j$$

$$= \frac{(1.000001)^{m-1} - 1}{1.000001 - 1}$$

$$= [(1.000001)^{m-1} - 1]10^6$$

$$\leq \{ 1 + (m-1+1)(.000001) - 1]10^6$$

$$= m, \text{ if } m \text{ is less than } 10000. ,$$

we get,

$$\delta_o \leq km^3 + 9.54005m^2 + 22.24018m . \qquad (C-75)$$

The same procedure can be followed that under the assumptions

$$[1 + a(i,3)2^{-t}]^2 + km^2 2^{-t} \leq 1.000001,$$

and

$$a^2(p,3)2^{-t} \leq .00001,$$

the expression of $\delta_p$ can be simplified as

$$\delta_p \leq \{ 28.62015 + 22.24018 + km^2 \} (1.000001)^{p-1}$$

$$= \{ (50.86033 + km^2 \} (1.000001)^{p-1} . \qquad (C-76)$$

Substituting (C-75), (C-76) into (C-69), we obtain

$$||E||_2 \leq 2^{-t} ||A_o||_2 \{ km^3 + 9.54005m^2$$

$$+ 22.24018m + (50.86033 + km^2) \sum_{p=1}^{s} (1.000001)^{p-1} \} .$$

The dimension of the Hamiltonian matrix is equal to 2n, and the above expression can be simplified to

$$||E||_2 \leq 2^{-t} \; ||A_o||_2 \; \{ \; km^3 + 9.54005m^2 + 22.24018m$$

$$+ \; (km^2 + 50.86033)(s + 1) \; \} \quad .$$

The quantity s is given by

(i) if all the eigenvalues are real

$$s = \sum_{p=1}^{n} 2(2n - P)$$

$$= n(4n) - 2\frac{n}{2}(n + 1)$$

$$= 3n^2 - n \qquad \text{(C-77)}$$

(ii) if all the eigenvalues are complex

$$s = \sum_{p=1}^{n/2} 2(2n - 2P + 1)$$

$$= \frac{n}{2} \cdot 2 \cdot (2n + 1) - 4\,\frac{\frac{n}{2}}{2}\,(\frac{n}{2} + 1)$$

$$= 2n^2 + n - \frac{n^2}{2} - n$$

$$= \frac{3}{2}\,n^2 \; . \qquad \text{(C-78)}$$

The bound for $||E||_2$ is then given by

$$||E||_2 \leq 2^{-t} \; ||A_o||_2 \; \{ \; 8kn^3 + (38.1602 + 4k)n^2$$

$$+ 44.8036n + 50.86033 + (4kn^2 + 50.86033)s \; \} \quad \text{(C-79)}$$

where

$$k = 2.12$$

and s is given by equation (C-77) or (C-78).

Q.E.D.

C-2

The Operations Count of the Algorithm

The operations count for solving the algebraic Riccati equation has three major parts: (1) OP count for finding eigenvalues of the Hamiltonian matrix, (2) OP count for isolating n desired eigenvalues in the lower-right hand corner of the Hamiltonian matrix, (3) OP count for solving the linear system of equations. In this discussion, the OP count is found for each required subroutine.

1. Operations count for subroutine HESS

For the similarity transformations, the OP count is given by

$$OP_{HT} = \sum_{k=n1+1}^{n2-1} \{ (nn - k + 1) + 1 + (nn - k + 1) + 2(nn - k + 1)(m - k + 1) + 2(nn - k + 1)n2 \}$$

$$= \sum_{k=n1+1}^{n2-1} \{ 2(nn - k + 1)\ (m - k + n2 + 2) + 1 \}$$

where

$nn = \min \{(nz + k-1, n2 \}$

$nz$ = number of non-zero elements below the diagonal n1, n2 = row indices of current isolated diagonal block

m = matrix size = 2n in our case.

For accumulating the orthogonal similarity transformation matrix, the OP count is given by

$$OP_{HP} = \sum_{k=n1+2}^{n2-1} \{ 2(nn - k + 1)m \} \tag{C-80}$$

There are two situations in subroutine HESS.

When the first Hessenberg reduction is performed, i.e., nz = m - 1, n1 = 1, n2 = m, the OP count is given by

$$OP_{HTf} = \sum_{k=2}^{m-1} \{2(nn - k + 1)(2m - k + 2) + 1 \}$$

and

$$m = \min \{ (m + k - 2), m \}$$

$$= m, \text{ for all } k$$

so that,

$$OP_{HTf} = \sum_{k=2}^{m-1} \{ 2(m - k + 1)(2m - k + 2) + 1 \}$$

$$= \frac{5}{3} m^3 - m^2 - \frac{5}{3} m - 6. \tag{C-81}$$

The well known identities

$$\sum_{k=1}^{m} k = \frac{m}{2}(m + 1)$$

and

$$\sum_{k=1}^{m} k^2 = \frac{m}{6}(m + 1)(2m + 1)$$

are used in equation (C-81).

$$OP_{HPf} = \sum_{k=2}^{m-1} \{ 2m(m + 1 - k) \}$$

$$= m^3 - m^2 - 2m. \qquad \text{(C-82)}$$

When the QR algorithm is performed on the upper Hessenberg matrix (i.e, nz = 3, n1 = 1, n2 = m -s, and s is the number of eigenvalues which have been isolated), the OP count is given by

$$OP_{HTS} = \sum_{k=2}^{m-s-1} \{ 2(nn - k + 1)(2m - k - S + 2) + 1 \}$$

with

$$nn = \min \{ (2 + k), m - s \}$$

$$= \begin{cases} 2 + k, \text{ if } k \varepsilon [2, m - s - 2] \\ m - s, \text{ if } k = m - s - 1 \end{cases} .$$

Using the expression for nn, the $OP_{HTS}$ is given by

$$OP_{HTS} = (9m^2 - 10m - 26) - 4(3m + 1)\,s + 3s^2 \qquad \text{(C-83)}$$

$$OP_{HPS} = \sum_{k=2}^{m-s-1} \{\, 2m(nn - k + 1) \,\}$$

$$= (6m^2 - 14m) - 6ms \quad . \qquad \text{(C-84)}$$

In summary, we have the following OP counts for the subroutine HESS:

(a) For the first Hessenberg reduction

$$OP_{HTf} = \frac{5}{3} m^3 - m^2 - \frac{5}{3} m - 6 \qquad \text{(C-85)}$$

$$OP_{HPf} = m^3 - m^2 - 2m \qquad \text{(C-86)}$$

(b) For the QR algorithm

$$OP_{HTS} = (9m^2 - 10m - 26) - 4(3m + 1)\,s + 3s^2 \qquad \text{(C-87)}$$

$$OP_{HPS} = (6m^2 - 14m) - 6ms \;. \qquad \text{(C-88)}$$

2. Operations count for subroutine SHIFT2

For the similarity transformations, the OP count is given by

$$OP_{ST} = 11 + 6(m + 1 - n1) + 6n2 \ .$$

For accummulating the transformation matrix

$$OP_{SP} = 6m \ .$$

For the case, n1 = 1, n2 = m - s, the OP count is given by

$$OP_{ST} = 12m + 11 - 6s \tag{C-89}$$

$$OP_{SP} = 6m \ . \tag{C-90}$$

Again, s is the number of eigenvalues which have been isolated.

3. Operations count for subroutine TRIA

(a) For real eigenvalues

$$OP_{TT} = 4m + 15 \tag{C-91}$$

$$OP_{TP} = 4m \ . \tag{C-92}$$

(b) For complex eigenvalues

$$OP_{TT} = 4 \tag{C-93}$$

$$OP_{TP} = 0 \; . \tag{C-94}$$

4. Operations count of Subroutine QR

The OP count in subroutine QR is the sum of the OP count for first Hessenberg reduction and the OP count for QR iteration applied to an upper Hessenberg matrix. For similarity transformation, the OP count is given by

$$OP_{QT} = 2 + OP_{HTf} + \sum_{s=0}^{n_e} \{ IT * [2 + OP_{HTS} + OP_{ST}] \}$$
$$+ \sum_{s=1}^{n_e} OP_{TT} \tag{C-95}$$

where $n_e$ is the number of eigenvalues to be isolated and IT is the number of iterations required per eigenvalue.

Substituting equations (C-85) to (C-94) into equation (C-95), we have

$$\sum_{s=0}^{n_e} \{ IT * [2 + OP_{HTS} + OP_{ST}] \}$$

$$= IT * \sum_{s=0}^{n_e} [(2 + 9m^2 - 10m - 26) - 4(3m + 1)s + 3s^2$$
$$+12m + 11 - 6s]$$

$$= IT * \{ (n_e + 1)(9m^2 + 2m - 13) - (12m + 10) \frac{n_e(n_e+1)}{2}$$

$$+ \quad 3 \frac{n_e}{6} (n_e + 1)(2n_e + 1)\}$$

$$= IT * (n_e + 1)[9m^2 + 2m - 13 + n_e{}^2 - (6m + 4.5)n_e]$$

$$\widetilde{OP}_T = \sum_{s=1}^{n_e} OP_{TT} = \begin{cases} n_e(4m + 15) & \text{if eigenvalues are real} \\ \frac{n_e}{2} \cdot 4 & \text{if eigenvalues are complex} \end{cases}$$

so that

$$OP_{QT} = 2 + \frac{5}{3}m^3 - m^2 - \frac{5}{3}m - 6 + IT*(n_e + 1)[9m^2$$

$$+ 2m - 13 + n_e{}^2 - (6m + 4.5)n_e] + \widetilde{OP}_T \,. \quad \text{(C-96)}$$

The number of iterations required per eigenvalue, IT, is estimated to be equal to 2.

The OP count for the orthogonal transformation matrix accumulation is given by

$$OP_{QP} = m^3 - m^2 - 2m + \sum_{s=0}^{n_e} \{ IT * [6m^2 - 14m - 6ms$$

$$+ 6m] \} + \sum_{s=1}^{n_e} OP_{TP} \,,$$

and,

$$\widetilde{OP}_P = \sum_{s=1}^{n_e} OP_{TP} = \begin{cases} 4\ n_e & \text{if eigenvalues are all real} \\ 0 & \text{if eigenvalues are all complex} \end{cases}$$

The $OP_{QP}$ is then given by

$$OP_{QP} = m^3 - m^2 - 2m + IT * (n_e + 1)\ [6m^2 - 8m - 3mn_e] + \widetilde{OP}_P \ .$$

The total OP count for upper-quasi triangularizing the Hamiltonian matrix is given by

$$OP_{uQ} = OP_{QT}^{n_e=2n-2} + OP_{QT}^{n_e=n} + OP_{QP}^{n_e=n} \ . \qquad \text{(C-97)}$$

Since m = 2n and IT = 2, the $OP_{uQ}$ is given by

$$OP_{QT}^{n_e=2n-2} = 2 + \frac{40}{3} n^3 - 4n^2 - \frac{10}{3} n - 6 + 2 * (2n-1)$$
$$[36n^2 + 4n - 13 + (2n-2)^2$$
$$- (12n + 4.5)(2n - 2)] + \widetilde{OP}_T$$
$$= \frac{232}{3} n^3 + 8n^2 - \frac{76}{3} n - 4 + \widetilde{OP}_T^{n_e=2n-2} \qquad \text{(C-98)}$$

$$OP_{QT}^{n_e=n} = 2 + \frac{40}{3} n^3 - 4n^2 - \frac{10}{3} n - 6 + 2 * (n + 1)$$
$$[36n^2 + 4n - 13 + n^2 - (12n + 4.5)n] + \widetilde{OP}_T$$
$$= \frac{190}{3} n^3 + 45n^2 - \frac{91}{3} n - 30 + \widetilde{OP}_T^{n_e=n} \qquad \text{(C-99)}$$

$$OP_{QP}^{n_e=n} = 8n^3 - 4n^2 - 4n + 2\ (n+1)\ [24n^2 - 16n$$

$$- 6n^2] + \widetilde{OP}_P$$

$$= 44n^3 - 36n + \widetilde{OP}_P^{n_e=n} . \qquad \text{(C-100)}$$

Substituting (C-98), (C-99), (C-100) into (C-97), the $OP_{uQ}$ is given by

(a) for real eigenvalues

$$OP_{uQ} = \frac{232}{3}\ n^3 + 8n^2 - \frac{76}{3}\ n - 4 + (2n - 2)(8n + 15)$$

$$+ \frac{190}{3}\ n^3 + 45n^2 - \frac{91}{3}\ n - 30 + n\ (8n + 15)$$

$$+ 44n^3 - 36n + 4n$$

$$= \frac{554}{3}\ n^3 + 77\ n^2 - \frac{176}{3}\ n - 64 \qquad \text{(C-101)}$$

(b) for complex eigenvalues

$$OP_{uQ} = \frac{232}{3}\ n^3 + 8n^2 - \frac{76}{3}\ n - 4 + 4\ n - 4$$

$$+ \frac{190}{3}\ n^3 + 45\ n^2 - \frac{91}{3}\ n - 30 + 2\ n + 44\ n^3 - 36\ n$$

$$= \frac{554}{3}\ n^3 + 53\ n^2 - \frac{257}{3}\ n - 38 . \qquad \text{(C-102)}$$

5. OP count of subroutine HSOLVE

The OP count of subroutine HSOLVE is given by

$$OP_L = 1 + \sum_{k=1}^{n-1} \{ (n - k + 1) + 2 + 4(n - k) + 1 \}$$

$$+ n \{ \sum_{k=1}^{n-1} [ (2n + 3) - k] + 1 \}$$

$$= \frac{3}{2} n^3 + 4 n^2 + \frac{5}{2} n - 3 .$$

Total OP count required for solving the algebraic Riccati equation is given by

$$OP_{Total} = OP_{uQ} + OP_L$$

(a) for real eigenvalues

$$OP_{Total} = \frac{1117}{6} n^3 + 81n^2 - \frac{337}{6} n - 67$$

(b) for complex eigenvalues

$$OP_{Total} = \frac{1117}{6} n^3 + 57n^2 - \frac{499}{6} n - 41 .$$

In summary, we have the following theorem.

Theorem C.4

If it is estimated that two QR iterations are needed for isolating one eigenvalue, then the operations count of the algorithm proposed in Chapter III for solving the algebraic Riccati equation is given by

$$OP_{max} = 186.2\ n^3 + 81\ n^2 - 56.2\ n - 67$$

and

$$OP_{min} = 186.2\ n^3 + 57\ n^2 - 83.2\ n - 41$$

where

$OP_{max}$ means the larger OP count assuming all real eigenvalues

$OP_{min}$ means the smaller OP count assuming all complex eigenvalues

n is the order of the system.

# APPENDIX D

# MEASUREMENT ELIMINATION THEOREMS AND THE SYSTEM MODEL

# APPENDIX D

# MEASUREMENT ELIMINATION THEOREMS AND THE SYSTEM MODEL

In Appendix D, proofs of the theorems in Chapter III are given. A supporting Lemma is also described.

## Proof of Equation (4.2.16)

$$I = E[y^T Q_1 y + u^T R_1 u \quad = tr[S\Gamma Q_2 \Gamma^T + K^T(R_1 + D_1{}^T Q_1 D_1)KP]$$

As discussed in Chapter II, the solution of the regulator Riccati equation S, the covariance of the estimated scale $X_*$ and the solution of the filter Riccati P satisfy following equations

$$0 = -SA_* - A_*{}^T S + SBR_*{}^{-1} B^T S - Q_*$$

$$0 = (A + BK)X_* + X_* (A + BK)^T + GR_2 G^T$$

$$0 = A_{**}P + PA_{**}{}^T + \Gamma Q_{2*} \Gamma^T - PC_2{}^T R_{2*}{}^{-1} C_2 P,$$

where the definition of matrices is given in Chapter II. With these three equations, the following equation is established

$$\Omega = 0 = (-\ SA_* - A_*^TS + SBR_*^{-1}B^TS - Q_*)X_*$$
$$+ (-\ SA_* - A_*^TS + SBR_*^{-1}B^TS - Q_*)P$$
$$+ S[(A + BK)X_* + X_*\ (A + BK)^T + GR_{2*}G^T]$$
$$+ S[A_{**}P + PA_{**}^T + \Gamma Q_{2*}\Gamma^T - PC_2^TR_{2*}^{-1}C_2P] \quad .$$

After some matrix manipulation, the above equation becomes

$$\Omega = 0 = -\ SBKP - A^TSX_* - C_1^TQ_1C_1X_*$$
$$- C_1^TQ_1D_1KX_* - A^TSP - C_1^TQ_1C_1P$$
$$- C_1^TQ_1D_1KP + SX_*A^T + SX_*K^TB^T$$
$$+ SPA^T + S\Gamma Q_2\Gamma \ .$$

The expectation of the integrand is then given by

$$I = E[y^TQ_1y + u^TR_1u]$$
$$= tr[Q_1Y + R_1U]$$
$$= tr[Q_1Y + R_1U + \Omega],$$

where $Y = E[yy^T]$, $U = E[uu^T]$.

Substituting equation of Y, U from Chapter II and $\Omega$ from above, the quantity I is given by

$$I = \mathrm{tr}[Q_1C_1X_*C_1{}^T + Q_1C_1PC_1{}^T + Q_1C_1X_*K^TD_1{}^T$$
$$+ Q_1D_1KX_*C_1{}^T + Q_1D_1KX_*K^TD_1{}^T + R_1KX_*K^T$$
$$- SBKP - A^TSX_* - C_1{}^TQ_1C_1X_* - C_1{}^TQ_1D_1KX_*$$
$$- A^TSP - C_1{}^TQ_1C_1P - C_1{}^TQ_1D_1KP + SX_*A^T$$
$$+ SX_*K^TB^T + SPA^T + S\Gamma Q_2\Gamma^T]$$

$$= \mathrm{tr}[- SBKP - A^TSP - C_1{}^TQ_1D_1KP + SPA^T + S\Gamma Q_2\Gamma^T]$$

$$= \mathrm{tr}[S\Gamma Q_2\Gamma^T - C_1{}^TQ_1D_1KP - SBKP] ,$$

where the equality tr[AB] = tr[BA] was used to simplify the above result. Also, since $-C_1{}^TQ_1D_1 - SB = K^TR_*$, the above equation reduced to the form

$$I = E[y^TQ_1y + u^TR_1u]$$

$$= \mathrm{tr}[S\Gamma Q_2\Gamma^T + K^T(R_1 + D_1{}^TQ_1D_1)KP] .$$

——— Q.E.D. ———

Lemma D.1

If the matrix U is symmetric positive semi-definite, then, the matrix product of K and U

$$K\ U\ K^T$$

is also symmetric positive semi-definite, i.e.,

$$K\ U\ K^T \geq 0 \text{ if } U \geq 0 \quad .$$

Proof:

For the symmetric property, take the transpose of $KUK^T$ and use the fact that U is symmetric. The result is given by

$$(K\ U\ K^T)^T = K\ U^T\ K^T = K\ U\ K^T$$

which is symmetric.

For the positive semi-definite property, consider the following. Since $U \geq 0$, there exists a matrix V such that $U = VV^T$. Then

$$K\ U\ K^T = KV\ V^T\ K^T = (KV)(KV)^T$$

which is positive semi-definite.

——— Q.E.D. ———

With the above lemma, an important theorem of the measurement elimination procedure can then be established.

Theorem D.1

Let $P_i$, $P_{ik}$ be the covariance of the state estimation error without $i^{th}$ and without $i^{th}$ and $k^{th}$ measurements respectively. These covariance matrices satisfy the following algebraic Riccati equations

$$A_{**}P_i + P_iA_{**}^T + \Gamma Q_{2*}\Gamma^T - P_iR_iP_i = 0$$

$$A_{**}P_{ik} + P_{ik}A_{**}^T + \Gamma Q_2\Gamma^T - P_{ik}R_{ik}P_{ik} = 0 \ .$$

The matrices in above equations are defined in equations (2.3.7) of Chapter II except that the $R_i$ and $R_{ik}$ are the term $C_2^TR_{2*}^{-1}C_2$ with $i^{th}$ and with $i^{th}$ and $k^{th}$ measurements eliminated. Also, assuming that the estimation error dynamic equation is asymptotically stable, i.e., the eigenvalues of $A_{**} - P_{ik}R_{ik}$ and $A_{**} - P_iR_i$ are all in the open left half plane.

Then

$$\Delta P = P_{ik} - P_i \geq 0$$

if

$$\Delta R = R_i - R_{ik} \geq 0 \ .$$

Also,

$$(\text{rms } \tilde{x}_j)_{ik} - (\text{rms } \tilde{x}_j)_i \geq 0$$

if

$$\Delta R = R_i - R_{ik} \geq 0 \ ,$$

where the subscript j means the $j^{th}$ element of the error state vector $\tilde{x}$ and the subscripts ik and i mean that the $i^{th}$ and $k^{th}$ and that the $i^{th}$ measurements are eliminated.

Proof:

Subtracting the equations for $P_i$ and $P_{ik}$, the resulting equation is given by

$$A_{**}\Delta P + \Delta P A_{**}{}^T - P_{ik}R_{ik}P_{ik} + P_i R_i P_i = 0 \ .$$

Using the equations of $\Delta R$ and $\Delta P$, the above equation becomes

$$\begin{aligned} 0 &= A_{**}\Delta P + \Delta P A_{**}{}^T - P_{ik}R_{ik}(\Delta P + R_i) \\ &\quad + P_i R_i P_i - \Delta P R_{ik} P_{ik} + \Delta P R_{ik} P_{ik} \\ &= (A_{**} - P_{ik}R_{ik})\Delta P + \Delta P (A_{**} - P_{ik}R_{ik})^T \\ &\quad - P_{ik}R_{ik}P_i + P_i R_i P_i + \Delta P R_{ik} P_{ik} \\ &= (A_{**} - P_{ik}R_{ik})\Delta P + \Delta P (A_{**} - P_{ik}R_{ik})^T \\ &\quad - P_{ik}R_{ik}P_i + P_i \Delta R P_i + P_i R_{ik} P_i + \Delta P R_{ik} P_{ik} \ . \end{aligned}$$

So that

$$0 = (A_{**} - P_{ik}R_{ik})\Delta P + \Delta P\,(A_{**} - P_{ik}R_{ik})^T$$
$$+ \Delta P R_{ik} \Delta P + P_i \Delta R P_i \quad .$$

Since $R_{ik} \geq 0$, from lemma D.1, $\Delta P R_{ik} \Delta P + P_i \Delta R_{\ i} \geq 0$ if $\Delta R \geq 0$. Also, from theorem B.3,

$$\Delta P \geq 0$$

if

$$(A_{**} - P_{ik}R_{ik})$$

is asymptotically stable. The result

$$(\text{rms } \tilde{x}_j)_{ik} - (\text{rms } \tilde{x}_j)_i \geq 0$$

follows immediately.

Q.E.D.

Theorem D.2

Under the same condition as in theorem D.1, the following results are concluded.

If

$$\Delta R = R_i - R_{ik} \geq 0$$

then

a. $\tilde{U}_{ik} - \tilde{U}_i \geq 0$

and

$(\text{rms } \tilde{u}_j)_{ik} - (\text{rms } \tilde{u}_j)_i \geq 0$

b. $\tilde{Y}_{ik} - \tilde{Y}_i \geq 0$

and

$(\text{rms } \tilde{y}_j)_{ik} - (\text{rms } \tilde{y}_j)_i \geq 0$

c. $I_{ik} - I_i \geq 0$,

where subscripts j, ik, and i have the meaning as in the above theorem and $\tilde{U}_i$, $\tilde{Y}_i$, $\tilde{u}_j$, $\tilde{y}_j$ etc. are defined in equations (4.2.10) through (4.2.16) of Chapter IV.

Proof:

From the theorem D.1, if $\Delta R \geq 0$, then $\Delta P \geq 0$. The following results follows.

$$\Delta U_{ik} - \Delta U_i = K(P_{ik} - P_i)K^T$$

$$= K\ \Delta P\ K^T \geq 0$$

$$\Delta Y_{ik} - \Delta Y_i = (C_1 + D_1K)\ \Delta P(C_1 + D_1K)^T \geq 0 .$$

The result

$$(\text{rms } \Delta u_j)_{ik} - (\text{rms } \Delta u_j)_i \geq 0$$

and

$$(\text{rms } \Delta y_j)_{ik} - (\text{rms } \Delta y_j)_{ik} \geq 0$$

follow immediately.

$$I_{ik} - I_i = \text{tr}[R^*\ \Delta P] ,$$

where

$$R^* = K^T(R_1 + D_1{}^TQ_1D_1)K \geq 0.$$

Since $R^* \geq 0$, there exists a matrix L such that $R^* = L^TL$, the quantity $I_{ik} - I_i$ is given by

$$\begin{aligned} I_{ik} - I_i &= \text{tr}[R^*\ \Delta P] \\ &= \text{tr}[L^T\ L\ \Delta P] \\ &= \text{tr}[L^T\ \Delta P\ L] . \end{aligned}$$

Since

$$\Delta P \geq 0,\ L^T \Delta P L \geq 0,$$

so that,

$$I_{ik} - I_i = tr[L^T \ \Delta P \ L] \geq 0 \quad .$$

——— Q.E.D. ———

## The System Model

The model of the aircraft used is given by (see equation (4.3.3))

$$\dot{x} = Ax + Bu + \Gamma W$$

$$y = C_1 x + D_1 u$$

$$z = C_2 x + D_2 u + \nu$$

where

$$x^T = [p,\ r,\ v,\ \phi,\ v_o,\ v_x,\ u_y,\ w_y]$$

$$u^T = [\delta a,\ \delta r]$$

$$y^T = [a_y,\ r,\ f_t]$$

$$z^T = [p,\ r,\ \phi,\ a_{ty},\ a_y,\ f_t,\ v,\ v-v_o].$$

The notation has the following meaning:

$p$ = roll rate, rad/s

$r$ = yaw rate, rad/s

$v$ = lateral aircraft velocity, ft/s

$\phi$ = roll angle, rad

$v_o$ = lateral gust velocity, ft/s

$v_x$ = longitudinal gradient of the lateral velocity, $sec^{-1}$

$u_y$ = lateral gradient of the longitudinal velocity, $sec^{-1}$

$w_y$ = lateral gradient of normal gust velocity, $sec^{-1}$

$\delta a$ = aileron deflection, rad

$\delta r$ = rudder deflection, rad

$a_y$ = lateral acceleration per unit gravitational acceleration

$\dot{r}$ = yaw acceleration ($rad/sec^2$)

$f_t$ = side force on vertical tail per unit weight of the aircraft

$a_{ty}$ = tail side acceleration per unit gravitation acceleration

$v-v_o$ = the relative side velocity of the aircraft with respect to air

The matrices are given as follows.

A MATRIX

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| -5.9362 | 2.5275 | -.28504 | 0. | .28504 | -1.2637 | 1.2637 | 5.9362 |
| .30729 | -.60799 | .30205E-01 | 0. | -.30205E-01 | .30400 | -.30400 | -.30729 |
| -.54272 | -777.99 | -.28452 | 32.173 | .28452 | -1.3568 | 1.3568 | .54272 |
| 1.0000 | 0. | 0. | 0. | 0. | 0. | 0. | 0. |
| 0. | 0. | 0. | 0. | 0. | 780.70 | 0. | 0. |
| 0. | 0. | 0. | 0. | -.11771 | -90.885 | 0. | 0. |
| 0. | 0. | 0. | 0. | .19619E-01 | 0. | -16.788 | 0. |
| 0. | 0. | 0. | 0. | 0. | 0. | 0. | -25.181 |

GAMA MATRIX

| | | |
|---|---|---|
| 0. | 0. | 0. |
| 0. | 0. | 0. |
| 0. | 0. | 0. |
| 0. | 0. | 0. |
| 0. | 0. | 0. |
| .11771 | -.26847E-01 | 0. |
| 0. | .36289E-01 | 0. |
| 0. | 0. | .45566E-01 |

C1 MATRIX

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| -.16869E-01 | .84345E-01 | -.88436E-02 | 1.0000 | .88436E-02 | -.42173E-01 | .42173E-01 | .16869E-01 |
| .30729 | -.60799 | .30205E-01 | 0. | -.30205E-01 | .30400 | -.30400 | -.30729 |
| -.16869E-01 | .84345E-01 | -.47166E-02 | 0. | .47166E-02 | -.84345E-01 | 0. | .16869E-01 |

Q2 MATRIX

| | | |
|---|---|---|
| 1.9214 | 0. | 0. |
| 0. | 1.9214 | 0. |
| 0. | 0. | 1.9214 |

B MATRIX

| | |
|---|---|
| 58.751 | 9.5538 |
| -4.4198 | -7.5595 |
| -18.510 | 44.425 |
| 0. | 0. |
| 0. | 0. |
| 0. | 0. |
| 0. | 0. |
| 0. | 0. |

D1 MATRIX

| | |
|---|---|
| -.57535 | 1.3808 |
| -4.4198 | -7.5595 |
| -.34521 | 1.3808 |

R1 MATRIX

| | |
|---|---|
| 0. | 0. |
| 0. | 0. |

Q1 MATRIX

| | | |
|---|---|---|
| 1.0000 | 0. | 0. |
| 0. | .57132 | 0. |
| 0. | 0. | 11.111 |

THETA MATRIX

| | | |
|---|---|---|
| 0. | 0. | 0. |
| 0. | 0. | 0. |
| 0. | 0. | 0. |
| 0. | 0. | 0. |
| 0. | 0. | 0. |
| 0. | 0. | 0. |
| 0. | 0. | 0. |
| 0. | 0. | 0. |

C2 MATRIX

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1.0000 | 0. | 0. | 0. | 0. | 0. | 0. | 0. |
| 0. | 1.0000 | 0. | 0. | 0. | 0. | 0. | 0. |
| 0. | 0. | 0. | 1.0000 | 0. | 0. | 0. | 0. |
| -.24914 | .54390 | -.31674E-01 | 1.0000 | .31674E-01 | -.27195 | .27195 | .24914 |
| -.16869E-01 | .84345E-01 | -.88436E-02 | 1.0000 | .88436E-02 | -.42173E-01 | .42173E-01 | .16869E-01 |
| -.16869E-01 | .84345E-01 | -.47166E-02 | 0. | .47166E-02 | -.84345E-01 | 0. | .16869E-01 |
| 0. | 0. | 1.0000 | 0. | 0. | 0. | 0. | 0. |
| 0. | 0. | 1.0000 | 0. | -1.0000 | 0. | 0. | 0. |

D2 MATRIX

| | |
|---|---|
| 0. | 0. |
| 0. | 0. |
| 0. | 0. |
| 2.7654 | 7.0947 |
| -.57535 | 1.3808 |
| -.34521 | 1.3808 |
| 0. | 0. |
| 0. | 0. |

R2 MATRIX

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| .93871E-04 | 0. | 0. | 0. | 0. | 0. | 0. | 0. |
| 0. | .93871E-04 | 0. | 0. | 0. | 0. | 0. | 0. |
| 0. | 0. | .14625E-05 | 0. | 0. | 0. | 0. | 0. |
| 0. | 0. | 0. | .19014E-04 | 0. | 0. | 0. | 0. |
| 0. | 0. | 0. | 0. | .19014E-04 | 0. | 0. | 0. |
| 0. | 0. | 0. | 0. | 0. | .11230E-05 | 0. | 0. |
| 0. | 0. | 0. | 0. | 0. | 0. | .12703E-02 | 0. |
| 0. | 0. | 0. | 0. | 0. | 0. | 0. | .33141 |

# APPENDIX E

# USER'S MANUAL

# I. THEORY OF THE LINEAR QUADRATIC REGULATOR AND STATIONARY KALMAN FILTER

## I.1 Introduction

In part I, the basic theory of the linear quadratic regulator and the stationary Kalman filter will be presented. These problems are classified into five catagories:

1. The control problem without process noise,
2. The control problem with process noise,
3. The state estimation problem,
4. The stochastic control problem, and
5. The stationary behavior of the system with zero control.

The solution of the five problems stated above will be discussed sequentially in the following.

In this report, a linear time-invariant system will be assumed. Mathematically, the system equations can be formulated as follows:

$$\begin{aligned} \dot{x} &= Ax + Bu + \Gamma w \\ y &= C_1 x + D_1 u \\ z &= C_2 x + D_2 u + \nu + \Theta w \end{aligned} \tag{1.1}$$

where
- $x$ system state, $n \times 1$ vector
- $u$ control variable, $\ell \times 1$ vector
- $w$ process noise, $p \times 1$ vector
- $y$ output, $m \times 1$ vector
- $z$ measurement, $q \times 1$ vector
- $\nu$ measurement noise, $q \times 1$ vector
- $A$ system dynamics matrix, $n \times n$ matrix
- $B$ control distribution matrix, $n \times \ell$ matrix
- $\Gamma$ process noise distribution matrix, $n \times p$ matrix
- $C_1$ output scaling matrix for state, $m \times n$ matrix
- $D_1$ output scaling matrix for control, $m \times \ell$ matrix
- $C_2$ measurement scaling matrix for state, $q \times n$ matrix
- $D_2$ measurement scaling matrix for control, $q \times \ell$
- $\Theta$ measurement coupling matrix for process noise, $q \times p$ matrix.

The power spectral densities of the zero mean, white, process and measurement noise are $Q_2$ and $R_2$ respectively.

I.2 The Control Problem without Process Noise

The problem is formulated as follows:

$$\min_{u} J = \frac{1}{2} \int_{o}^{\infty} (y^T Q_1 y + u^T R_1 u) \, dt \tag{1.2}$$

subject to constraints of the system and the output equations

$$\dot{x} = Ax + Bu$$
$$y = C_1 x + D_1 u \tag{1.3}$$

the matrices $Q_1$ and $R_1$ are assumed to be symmetric and positive definite.

It is shown in the thesis (1) that the solution of the problem defined by equations (1.2) and (1.3) is given by

$$u = K_1 x$$

where $$K_1 = C_* - R_*^{-1} B^T S \tag{1.4}$$

and the matrix S satisfies the algebraic Riccati equation

$$0 = -SA_* - A_*^T S + SBR_*^{-1} B^T S - Q_*$$

The remaining matrices are defined by

$$R_* = D_1^T Q_1 D_1 + R_1$$

$$C_* = -R_*^{-1} D_1^T Q_1 C_1$$

$$A_* = A + B\, C_* \tag{1.5}$$

$$Q_* = C_1^T Q_1 C_1 - C_*^T R_* C_* \quad .$$

The closed loop dynamic equation is

$$\dot{x} = (A + BK_1)\, x \quad . \tag{1.6}$$

I.3 The Control Problem with Process Noise

The problem is formulated as follows:

$$\min_u J = E\{\frac{1}{2} \int_0^\infty (y^T Q_1 y + u^T R_1 u) dt\} \tag{1.7}$$

subject to constraints of the system and the output equations

$$\begin{aligned} \dot{x} &= Ax + Bu + \Gamma w \\ y &= C_1 x + D_1 u \end{aligned} \tag{1.8}$$

Again, the matrices $Q_1$ and $R_1$ are assumed to be symmetric and positive definite.

Davis and others (2, 3, 4) have shown that the solution of the problem defined by equations (1.7) and (1.8) is the same as the solution given in section I.2 which is rewritten here:

$$u = K_1 x$$

where

$$K_1 = C_1 - R_*^{-1} B^T S \tag{1.9}$$

and the matrix S satisfies the algebraic Riccati equation

$$0 = -SA_* - A_*^T S + SBR_*^{-1} B^T S - Q_*$$

The remaining matrices are defined in equation (1.5). The closed loop dynamics equation is given by

$$\dot{x} = (A + BK_1)x + \Gamma w \quad . \tag{1.10}$$

The statistical properties are described by the following:

a. The state covariance matrix is defined as

$$X = E\{x(t)x^T(t)\} \quad . \tag{1.11}$$

It is shown in the thesis (1), that X satisfies the following Lyapunov equation for the stationary case:

$$0 = (A + BK)X + X(A + BK)^T + \Gamma Q_2 \Gamma^T \quad . \tag{1.12}$$

The rms response of the state is given by the square root of the diagonal elements of X.

b. The control covariance matrix is defined as

$$U = E\{u(t)\, u^T(t)\} \quad , \tag{1.13}$$

which is given by

$$U = K_1 X K_1^T \quad . \tag{1.14}$$

c. The output covariance matrix is defined as

$$Y = E\{y(t) y^T(t)\} \tag{1.15}$$

and is given by

$$Y = (C_1 + D_1 K_1) X (C_1 + D_1 K_1)^T \quad . \tag{1.16}$$

I.4 The State Estimation Problem.

The problem is formulated as follows:

$$\max_x J_1 = p[x(t)|z(\tau),\ \tau \leq t] \quad , \tag{1.17}$$

where $p[x|z]$ is the conditional probability of the system state, subject to constraints of the system and the measurement equations

$$\begin{aligned} \dot{x} &= Ax + Bu + \Gamma w \\ z &= C_2 x + D_2 u + \nu + \Theta w \end{aligned} \tag{1.18}$$

It is shown by Sage and Melsa (5) that, if the conditioned probability function and the joint density function of x and z are Gaussian and if w and $\nu$ are causally related to z, maximizing the conditional probability function, $p[x|z]$, is equivalent to minimizing another performance criteria (for the stationary case)

$$\min_{w,\nu} J = E\{\frac{1}{2} \int_{-\infty}^{t} (w^T Q_2^{-1} w + \nu^T R_2^{-1} \nu) dt\} \tag{1.19}$$

The solution of the problem defined by equations (1.18) and (1.19) is given by the stationary Kalman filter (see thesis (1)).

The dynamic equation of the stationary Kalman filter

$$\dot{x}_* = Ax_* + Bu - K_2(z - C_2x_* - D_2u)$$

The filter gain

$$K_2 = -PC_2^TR_{2*}^{-1} - \Gamma Q_2\Theta^TR_{2*}^{-1} \tag{1.20}$$

and the matrix P satisfies the algebraic Riccati equation

$$0 = A_{**}P + PA_{**}^T + \Gamma Q_{2*}\Gamma^T - PC_2^TR_{2*}^{-1}C_2P \quad .$$

$x_*$ is the filtered state estimate and the remaining matrices are defined by

$$\begin{aligned} R_{2*} &= R_2 + \Theta Q_2\Theta^T \\ A_{**} &= A - \Gamma Q_2\Theta^TR_{2*}^{-1}C_2 \\ Q_{2*} &= Q_2 - Q_2\Theta^TR_{2*}^{-1}\Theta Q_2 \quad . \end{aligned} \tag{1.21}$$

The state estimation error covariance matrix is defined by

$$\tilde{X} = E\{\tilde{x}(t)\tilde{x}^T(t)\} \quad , \tag{1.23}$$

where the estimation error $\tilde{x} = x - x_*$.
It is shown in the thesis (1), that the state estimation error covariance matrix for the stationary case is given by the solution of the algebraic Riccati equation in equation (1.20), i.e.,

$$\tilde{X} = P \quad . \tag{1.24}$$

## I.5 The Stochastic Control Problem

The stochastic control problem is a combination of problems stated in section I.3 and I.4. The problem is formulated as follows:

Find u as a function of $z(\tau)$, $\tau \leq t$, to minimize

$$J = E\{\frac{1}{2}\int_0^\infty (y^T Q_1 y + u^T R_1 u)dt\} \quad , \tag{1.25}$$

subject to constraints of the system, output and measurement equations

$$\begin{aligned} \dot{x} &= Ax + Bu + \Gamma w \\ y &= C_1 x + D_1 u \\ z &= C_2 x + D_2 u + \nu + \Theta w \quad . \end{aligned} \tag{1.26}$$

According to the separation theorem (6, 7, 8), the problem defined by equations (1.25) and (1.26) can be treated as two separate problems: 1) the optimal control problem (section I.3), and 2) the state estimation problem (section I.4). The solution is given by a combination of the solutions of the optimal control problem and the state estimation problem:

The stationary Kalman filter implementation dynamic equation

$$\dot{x}_* = A_1 x_* - K_2 z$$

with

$$A_1 = A + BK_1 + K_2C_2 + K_2D_2K_1 \tag{1.27}$$

The controller

$$u = K_1 x_*$$

with the control gain

$$K_1 = C_* - R_*^{-1}B^T S$$

where S satisfies the algebraic Riccati equation

$$0 = -SA_* - A_*^T S + SBR_*^{-1}B^T S - Q_*$$

and the filter gain

$$K_2 = -PC_2^T R_{2*}^{-1} - \Gamma Q_2^T \Theta^T R_{2*}^{-1}$$

where P satisfies the algebraic Riccati equation

$$0 = A_{**}P + PA_{**}^T + \Gamma Q_{2*}\Gamma^T - PC_2^T R_{2*}^{-1} C_2 P \quad .$$

The definition of the matrices in equation (1.27) are given in equations (1.5) and (1.21).

The statistical properties are described by the following:

a. It is shown in the thesis (1), that the covariance matrix of the estimated state

$$X_* = E\{x_*(t)x_*^T(t)\} \tag{1.28}$$

is given by the solution of the following Lyapunov equation

$$0 = (A + BK_1)X_* + X_*(A + BK_1)^T + K_2(R_2 + \Theta Q_2 \Theta^T)K_2^T \tag{1.29}$$

b. It is also shown in the thesis (1) that the covariance matrix of the actual state is given by

$$\begin{aligned} X &= E\{x(t)x^T(t)\} \\ &= X_* + P \quad . \end{aligned} \tag{1.30}$$

c. The covariance matrix of the control is given by

$$\begin{aligned} U &= E\{u(t)u^T(t)\} \\ &= K_1 X_* K_1^{\,T} \quad . \end{aligned} \tag{1.31}$$

d. The covariance matrix of the output is given by

$$\begin{aligned} Y &= E\{y(t)y^T(t)\} \\ &= (C_1 + D_1K_1)X_*(C_1 + D_1K_1)^T + C_1PC_1^{\,T} \ . \end{aligned} \tag{1.32}$$

I.6 Stationary Response of the System with Zero Control

The system equation and the output with zero control is given by

$$\begin{aligned} \dot{x} &= Ax + \Gamma w \\ y &= C_1 x \quad . \end{aligned} \tag{1.33}$$

The statistical properties are described by the following:

a. The covariance matrix of the system state

$$X = E\{x(t)x^T(t)\} \tag{1.34}$$

is given by the solution of the following Lyapunov equation for the stationary case

$$0 = AX + XA^T + \Gamma Q_2 \Gamma^T \tag{1.35}$$

b. The covariance matrix of the output is given by

$$\begin{aligned} Y &= E\{y(t)y^T(t)\} \\ &= C_1 X C_1^{\,T} \quad . \end{aligned} \tag{1.36}$$

# II. LOGICAL CONSTRUCTION OF THE ALGORITHM AND DESCRIPTION OF THE SUBROUTINES

## II.1 Introduction

From the discussion in part I, we know that in order to solve the problems of the linear quadratic regulator and the stationary Kalman filter, the algebraic Riccati equations (1.4), (1.9), (1.20) and (1.27) have to be solved. Many authors have suggested methods to solve the algebraic Riccati equation. One of the methods that has been most successful is the eigenvector decomposition method first proposed by MacFarlane (9) and by Potter (10). In this method, the eigenvalues and the corresponding eigenvectors of the Euler-Lagrange system are determined. The eigenvectors associated with eigenvalues whose real parts are all of the same sign are partitioned into two matrices. These matrices form a set of linear equations which yield the solution of the algebraic Riccati equation.

The success of this method hinges on the requirement that the partitioned eigenvector matrices be nonsingular. Unfortunately, in the case when one or more of the eigenvalues are repeated, the resulting matrices may be singular. The singularity can be removed by using the generalized eigenvectors. However, this method is not entirely satisfactory, because when the eigenvalues are nearly equal, the partitioned eigenvector matrices are not singular, but they remain ill-conditioned. This ill-conditioning can lead to errors in the computed solution. Also, small perturbations in the system matrix elements can lead to drastic changes in the partitioned eigenvector matrices, which, in turn, causes poor numerical stability.

In order to alleviate these difficulties, the method presented in this report is proposed. In this method, the Hamiltonian matrix is transformed into a quasi-upper triangular matrix, such that the lower n x n corner of the matrix contains all the positive (negative) eigenvalues for the regulator (filter) problem. The highly stable QR algorithm is used to accomplish this orthogonal similarity transformation. The orthogonal matrix is then partitioned into four n x n matrices. These matrices form a set of linear equations which yield the solution of the algebraic Riccati equation.

In section II.2, the logical construction of the proposed algorithm will be presented. The supporting theorems of the proposed algorithm will be given without proof. Such proof is found in references 11, 12, 13. In section II.3, a description of the main program and subroutines will be presented.

## II.2 Proposed Algorithm for Solving the Algebraic Riccati Equation.

The proposed algorithm for solving the algebraic Riccati equation is given by the following steps:

1. Determine the Hamiltonian matrix H of the Euler-Lagrange system.

   a. For the linear quadratic regulator, the resulting algebraic Riccati equation (1.4) is in the form

$$SA_* + A_*^T S + Q_* - SBR_*^{-1}B^T S = 0 \quad . \tag{2.1}$$

   The corresponding Hamiltonian matrix is given by

$$H = \begin{bmatrix} A_* & -BR_*^{-1}B^T \\ -Q_* & -A_*^T \end{bmatrix} \tag{2.2}$$

   b. For the stationary Kalman filter, the resulting algebraic Riccati equation (1.20) is in the form

$$A_{**}P + PA_{**}^T + \Gamma Q_{2*}\Gamma^T - PC_2^T R_{2*}^{-1} C_2 P = 0 \quad . \tag{2.3}$$

   The corresponding Hamiltonian matrix is given by

$$H = \begin{bmatrix} A_{**} & \Gamma Q_{2*}\Gamma^T \\ C_2^T R_{2*}^{-1} C_2 & -A_{**}^T \end{bmatrix} \quad . \tag{2.4}$$

2. Determine the eigenvalues of Hamiltonian matrix.

   The 2n eigenvalues of H can be found by the following procedure:

   a. Use the Householder reduction (17) to transform H into upper Hessenberg form. This method transforms H into the upper Hessenberg matrix H' by the reduction

$$H' = Q\, H\, Q^T \tag{2.5}$$

   where $Q = Q_{n-2}, Q_{n-3} \ldots Q_1$, with $Q_i$ of the form

$$Q_i = I - \frac{u_i u_i^T}{\beta_i}$$

b. Use the highly stable QR algorithm (18) with implicit double shifts of origin to transform the upper Hessenberg matrix H' into the quasi-upper triangular matrix H". The QR algorithm with double shift of origin is described as follows:

At the j iteration of the algorithm,

$$H_{j+2} = Q_j H_j Q_j^T \tag{2.7}$$

$$\text{given } R_j = Q_j(H_j - k_j I)(H_j - k_{j+1} I) \tag{2.8}$$

where $R_j$ is a triangular matrix and $Q_j$ is an orthogonal matrix. The matrices $R_j$ and $Q_j$ are again determined by Householder's algorithm. In the program, the algorithm is implemented in implicit form where the first Householder transformation is followed by a Hessenburg reduction (see 12.).

3. Isolate the eigenvalues with proper sign of real part in lower right-hand corner.

Use the QR algorithm with implicit double shifts of origin at the eigenvalues previously computed from step 2 to transform the Hamiltonian matrix, H, into the form

$$QHQ^T = \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} \tag{2.9}$$

with the orthogonal similarity transformation matrix

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} , \tag{2.10}$$

and $U_{22}$ containing all the eigenvalues with positive (negative) real part for regulator (filter) problem. In cases when any undesired zero sub-diagonal elements appear and/or any undesired eigenvalues are isolated, an arbitrary Householder type similarity transformation is performed to remove the undesired zero sub-diagonal elements and/or the undesired eigenvalues.

4. Solve the linear system of equations.

a. The solution of the regulator Riccati equation (2.1) is given by

$$S = Q_{11}^{-1} Q_{12} \quad . \tag{2.11}$$

b. The solution of filter Riccati equation (2.3) is given by

$$P = Q_{12}^{-1} Q_{11} \quad . \tag{2.12}$$

The matrices $Q_{11}$ and $Q_{12}$ are nonsingular if the systems associated with equations (2.1) and (2.3) are controllable and observable.

Once the solution of the algebraic Riccati equation (2.11) and (2.12) are found, the calculation of the control gain, filter gain, closed loop dynamics matrix, etc. are simple matrix computations.

## II.3 Description of main program OPTIMAL and Subroutines

In this section, the main program OPTIMAL and the following fifteen subroutines are described:

1. READM
2. PRINT
3. PREVAL
4. CONTRL
5. FILTER
6. RICCAT
7. QR
8. HESS
9. SHIFT2
10. PERMUT
11. TRIA
12. HSOLVE
13. EIGVC
14. SOLYAP
15. LYAPUN

0. Main program OPTIMAL

The main program carries out input, output, and also drives the subroutines to compute solutions.

1. Subroutine READM

This subroutine reads in necessary information to set up an $n_1$ by $n_2$ matrix, then prints it out.

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      SUBROUTINE READM(A,N1,N2,N3,NT,NA,LP)                          C
C      DIMENSION A(N3,N2)                                             C
C     READ AND PRINT N1*N2 MATRIX A                                   C
C     INPUT...                                                        C
C        N1=ROW DIMENSION OF A (NUMBER OF ROWS)                       C
C        N2=COLUMN DIMENSION OF A                                     C
C        N3=DECLARED ROW DIMENSION OF A                               C
C        NT=Z IF A IS ZERO MATRIX                                     C
C        NT=I IF A IS IDENTITY MATRIX                                 C
C        NT=S IF A IS SYMMETRIC MATRIX                                C
C        NT=D IF A IS DIAGONAL MATRIX                                 C
C        NA=NAME OF MATRIX A                                          C
C        LP=0 FROM REGULAR LINE PRINTER                               C
C        LP=1 FROM TERMINAL PRINTER                                   C
C     SUBROUTINE REQD.- PRINT                                         C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

2. Subroutine PRINT

This subroutine prints out a given $n_1$ by $n_2$ matrix.

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      SUBROUTINE PRINT(A,N1,N2,ND,ID)                                C
C      DIMENSION A(ND,N2)                                             C
C     PRINT N1*N2 MATRIX A                                            C
C     INPUT...                                                        C
C        A=MATRIX TO BE PRINTED                                       C
C        N1=ROW DIMENSION OF A (NUMBER OF ROWS)                       C
C        N2=COLUMN DIMENSION OF A                                     C
C        ND=DECLARED ROW DIMENSION OF A                               C
C        ID=0 REGULAR LINE PRINTER (120 COLUMN)                       C
C        ID=1 TERMINAL PRINTER (80 COLUMN)                            C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

3. Subroutine PREVAL

This subroutine prints out the given eigenvalues and the corresponding normalized eigenvectors, if desired.

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      SUBROUTINE PREVAL(VAL,EL,ND,N,ID,S40)                          C
C      DIMENSION VAL(ND),EL(ND,ND)                                    C
C      COMPLEX VAL                                                    C
C     PRINT EIGENVALUES AND EIGENVECTORS. THE EIGENVECTORS WILL       C
C     BE NORMALLIZED                                                  C
C     INPUT....                                                       C
C        VAL=EIGENVALUES (FROM QR)                                    C
C        EL=EIGENVECTOR MATRIX (FROM EIGVC)                           C
C        ND=DECLARED DIMENSION OF VAL,EL                              C
C        N=DIMENSION OF VAL,EL                                        C
C        S40=PRECISION OF EIGENVALUES COMPUTATION                     C
C        ID=0 ,PRINT EIGENVALUES ONLY                                 C
C        ID=1 ,BOTH EIGENVALUES AND EIGENVECTORS PRINTED              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

4. Subroutine CONTRL

This subroutine takes the given matrices to set up the regulator Hamiltonian matrix of equation (2.2). The proper subroutines are called in this program to solve the Riccati equation (2.1). Also, the control gain is computed.

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C     SUBROUTINE CONTRL(A,B,C,D,Q,R,S,GAIN,H,P,VAL,N,M,L,ND,ND2,NDM,        C
C    1NDL,W,ID)                                                             C
C     DIMENSION A(ND,ND),B(ND,NDL),C(NDM,ND),D(NDM,NDL),Q(ND,ND),           C
C    1R(ND,ND),S(ND,ND),GAIN(NDL,ND),H(ND2,ND2),P(ND2,ND2),VAL(ND2),        C
C    1W(ND2)                                                                C
C     COMPLEX VAL                                                           C
C   SOLVE THE REGULATOR PROBLEM                                             C
C      DX/DT=A*X+B*U                                                        C
C      Y=C*X+D*U                                                            C
C   WITH QUADRATIC PERFORMANCE CRITERION                                    C
C      J=(1/2)*(INTG(YT*Q*Y+UT*R*U))                                        C
C   INPUT...                                                                C
C   1...                                                                    C
C     A,B,C,D,Q,R, ARE MATRICES DEFINED ABOVE WITH DIMENSIONS               C
C     N*N,N*L,M*N,M*L,M*M,L*L RESPECTIVELY                                  C
C   2..                                                                     C
C     N,M,L,DIMENSION OF MATRIX                                             C
C     ND,NDM,NDL ARE DECLARED DIMENSIONS ASSOCIATED WITH N,M,L              C
C   3... ND2=ND*2 DECLARED DIMENSION OF MATRICES H,P,VAL,W                  C
C     ID=0 ,D MATRIX IS EQ. ZERO , OTHERWISE =1                             C
C   OUTPUT...                                                               C
C     S=N*N MATRIX , SOLUTION OF RICCATI EQUATION                           C
C     GAIN=L*N MATRIX, CONTROL GAIN ,U=GAIN*X                               C
C     VAL= FIRST N ELEMENTS ARE EIGENVALUES( CLOSED LOOP)                   C
C          LAST N ELEMENTS ARE WORKING SPACE                                C
C   SOME NOTES....                                                          C
C     A,B,C,D, WILL BE SAVED                                                C
C     Q,R, WILL BE DESTROYED                                                C
C     H,P,W ARE WORKING SPACE                                               C
C   SUBROUTINES REQD.-HSOLVE,RICCAT,QR,HESS,SHIFT2,TRIA,PERMUT              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

5. Subroutine FILTER

This subroutine takes the given matrices to set up the filter Hamiltonian matrix of equation (2.4). The proper subroutines are called in this program to solve the Riccati equation (2.3). Also, the filter gain is computed.

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C     SUBROUTINE FILTER(A,C,GAMA,THETA,Q,R,S,GAIN,H,P,VAL,N,M,L,ND,         C
C    1ND2,NDM,NDL,W,ID)                                                     C
C     DIMENSION A(ND,ND),C(NDM,ND),GAMA(ND,NDL),THETA(NDM,NDL),             C
C    1Q(ND,ND),R(ND,ND),H(ND2,ND2),P(ND2,ND2),W(ND2),S(ND,ND),              C
C    1VAL(ND2),GAIN(ND,NDM)                                                 C
C     COMPLEX VAL                                                           C
C   SOLVE FOR STATIONARY KALMAN FILTER OF THE SYSTEM                        C
C     DX/DT=A*X+B*U+GAMA*W                                                  C
C     Z=C*X+D*U+V+THETA*W                                                   C
C     Q AND R ARE POWER SPECTRAL DENSITIES OF W AND V                       C
C   INPUT...                                                                C
C     A,C,GAMA,THETA,Q,R AS DEFINED ABOVE ARE N*N,M*N,N*L,M*L,L*L,M*M       C
C     RESPECTIVELY.                                                         C
C     N,L,M,ND,NDL,NDM ARE DIMENSION AND DECLARED DIMENSION RESPECTIVELY.   C
C     ND2=ND*2 DECLARED DIMENSION OF H,P,VAL,W                              C
C     ID=0 IF MATRIX THETA=0. ,=1 OTHERWISE                                 C
C   OUTPUT...                                                               C
C     S,VAL,GAIN ARE SOLUTION OF RICCATI EQUATION, CLOSED LOOP              C
C     EIGENVALUES, AND FILTER GAIN RESPECTIVELY.                            C
C   NOTICE.. Q,R WILL BE DESTROYED.                                         C
C     H,P,W ARE WORKING SPACE                                               C
C   SUBROUTINES REQD.- HSOLVE,RICCAT,QR,HESS,SHIFT2,TRIA,PERMUT             C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

6. Subroutine RICCAT

This subroutine takes the given Hamiltonian matrix to call the proper subroutines for performing the Riccati equation solution. The Hamiltonian matrix is transformed into the form given by equation (2.9). The solution of the corresponding algebraic Riccati equation is computed corresponding to equation (2.11) or equation (2.12).

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C     SUBROUTINE RICCAT(H,P,S,VAL,K1,N,ND,ND2,W,QQ)                    C
C     DIMENSION H(ND2,ND2),P(ND2,ND2),S(ND,ND),VAL(ND2),W(ND2),QQ(ND,ND) C
C     COMPLEX VAL,VAT                                                  C
C   REGULATOR PROBLEM--- S*A+AT*S-Q+S*R*S=0.                           C
C   FILTER PROBLEM--- A*S+S*AT+R-S*Q*S=0.                              C
C   INPUT...                                                           C
C      H=2N*2N MATRIX, DEFINED AS FOLLOWS...                           C
C      H11=A                                                           C
C      H12=R                                                           C
C      H21=Q                                                           C
C      H22=-AT                                                         C
C      A,Q,R AS IN THE ABOVE RICCATI EQUATION FOR REGULATOR AND FILTER C
C      K1=CONTROL PARAMETER,                                           C
C         K1=1 FOR REGULATOR(ISOLATE RIGHT HALF PLANE EIGENVALUES)     C
C         K1=-1 FOR FILTER(ISOLATE LEFT HALF PLANE EIGENVALUES)        C
C      N=DIMENSION OF S                                                C
C      ND=DECLARED DIMENSION OF S AND QQ                               C
C      ND2=DECLARED DIMENSION OF H AND P(USUALLY 2*ND)                 C
C      W IS WORKING SPACE, 2N*1                                        C
C      QQ IS N*N WORKING SPACE                                         C
C   OUTPUT....                                                         C
C      S=SOLUTION OF RICCATI EQUATION,N*N MATRIX                       C
C      H=QUASI-UPPER TRIANGULAR MATRIX,2N*2N                           C
C      P=ORTHOGONAL SIMILARITY TRANSFORMATION MATRIX, 2N*2N            C
C      VAL=N LEFT HALF PLANE EIGENVALUES AND N LOCATIONS OF WORKING SPACE C
C   SUBROUTINES REQD.- QR,HSOLVE,HESS,SHIFT2,TRIA,PERMUT               C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

7. Subroutine QR

This subroutine performs the quasi-upper triangularization of the given matrix by using the QR algorithm with implicit double shifts of origin. Knowledge of the eigenvalues can be given or not. Also, the number of eigenvalues to be isolated is specified by the calling program. The order of the computed eigenvalues corresponds to the ordering of the isolated diagonal blocks in the resulting quasi-upper triangular matrix. When a priori eigenvalue information is given, this ordering is forced to be the same as the input eigenvalues.

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C     SUBROUTINE QR(H,P,N,ND,VAL,MV,SNO,ITR,I1,K1)                     C
C     COMPLEX VAL,V                                                    C
C     DIMENSION H(ND,ND),P(ND,ND),VAL(ND),V(2)                         C
C   QR REDUCTION TO QUASI-UPPER TRIANGULAR FORM                        C
C   INPUT.....                                                         C
C      H= MATRIX TO BE REDUCED                                         C
C      ITR=DIMENSION OF PREVIOUSLY TRIANGULARIZED CORNER OF H          C
C      I1= 0, NO A PRIORI EIGENVALUE KNOWLEDGE                         C
C      I1= 1, SHIFT ON KNOWN EIGENVALUES                               C
C      K1= 0, NO CUMMULATED P CALCULATED                               C
C      K1= 1, CUMMULATED P CALCULATED                                  C
C      MV=NUMBER OF EIGENVALUES TO BE ISOLATED                         C
C      VAL=A PRIORI EIGENVALUES ( IF KNOWN )                           C
C      SNO=REQUIRED PRECISION AS A FRACTION OF THE MATRIX NORM         C
C      N=DIMENSION OF MATRIX H                                         C
C      ND=NUMBER OF ROWS IN MAIN PROGRAM DIMENSION OF H                C
C   OUTPUT..............                                               C
C      H=QUASI-UPPER TRIANGULAR MATRIX                                 C
C      P=ORTHOGONAL MATRIX ( IF K1= 1 )                                C
C      VAL=CALCULATED EIGENVALUES                                      C
C      SNO=SNO*NORM-USED FOR DISTINGUISHING ZERO SUBDIAGONAL ELEMENTS  C
C          IN RESULTING QUASI-TRIANGULAR MATRIX                        C
C   SUBROUTINES REQD.- HESS,SHIFT2,TRIA,PERMUT                         C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

8. Subroutine HESS

This subroutine performs the orthogonal similarity transformation to transform the given matrix into the upper Hessenberg form. Advantage is taken of zero elements if the lower right triangle is known to be null.

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C     SUBROUTINE HESS(A,P,N,ND,NZ,N1,N2,SMALL,K1)                       C
C     DIMENSION A(ND,ND),P(ND,ND)                                       C
C  PERFORMES ORTHOGONAL SIMILARITY REDUCTION TO UPPER HESSENBERG FORM.  C
C  MATRIX IS FULL N*N OR HAS ZERO LOWER-LEFT TRIANGLE.                  C
C  PARAMETERS- A- INPUT- N*N MATRIX TO BE REDUCED.                      C
C                 OUTPUT- REDUCED MATRIX                                C
C              P- ACCUMULATED ORTHOGONAL TRANSFORMATION MATRIX.         C
C              N- MATRIX SIZE                                           C
C              ND- NUMBER OF ROWS IN MAIN PROGRAM DIMENSION FOR A AND P C
C              NZ- NUMBER OF NON-ZERO ELEMENTS BELOW THE DIAGONAL IN A. C
C              N1,N2- ROW INDICIES OF CURRENT ISOLATED DIAGONAL BLOCK   C
C              SMALL- A SMALL NUMBER = MACH. PREC. * NORM(A)            C
C              K1=0 NO ACCUMULATED P MATRIX                             C
C              K1=1 ACCUMULATE ORTHOGONAL TRANSFORMATIONS               C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

9. Subroutine SHIFT2

This subroutine performs the first step of the QR algorithm with implicit double shifts of origin. To complete one iteration of the QR algorithm, the resulting matrix is transformed back to upper Hessenberg form using subroutine HESS above.

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C     SUBROUTINE SHIFT2(A,P,N,ND,N1,N2,C1,C2,SMALL,K1)                  C
C     DIMENSION A(ND,ND),P(ND,ND)                                       C
C  PERFORMS DOUBLE SHIFT WITH ORIGINS S1,S2 WHICH SATISFY C1=S1+S2      C
C  AND C2=S1*S2, FOLLOWED BY THE FIRST SIMILARITY TRANSFORMATION        C
C  IN QR FACTORIZATION. MATRIX A IS UPPER HESSENBERG ON INPUT.          C
C  PARAMETERS- A- INPUT-N*N UPPER HESSENBERG MATRIX                     C
C                 OUTPUT- TRANSFORMED MATRIX WITH 3 ELEMENTS BELOW DIAG C
C              P- N*N ACCUMULATED ORTHOGONAL TRANSFORMATION MATRIX      C
C              N- MATRIX SIZE                                           C
C              ND- NUMBER OF ROWS IN MAIN PROGRAM DIMENSION FOR A AND P C
C              N1,N2- ROW INDICIES OF ISOLATED DIAGONAL BLOCK           C
C              C1, C2- CONSTANTS FOR DOUBLE ORIGIN SHIFT                C
C              SMALL- A SMALL NUMBER = MACH. PREC. * NORM(A)            C
C              K1=0 NO ACCUMULATED P MATRIX                             C
C              K1=1 ACCUMULATE ORTHOGONAL TRANSFORMATIONS               C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

10. Subroutine PERMUT

This subroutine performs an orthogonal similarity transformation on the given matrix by using a Householder type orthogonal matrix. The purpose of this subroutine is to remove any undesirable zero sub-diagonal elements. Notice that the resulting matrix is no longer in upper Hessenberg form.

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C     SUBROUTINE PERMUT(H,P,N,ND,NX,N2,S40,K1,IP)                     C
C     DIMENSION H(ND,ND),P(ND,ND)                                     C
C   ARBITRARY ORTHOGONAL( HOUSEHOLDER TYPE ) SIMILARITY               C
C   TRANSFORMATION TO REMOVE AN UNDESIRABLE ZERO ON SUBDIAGONAL       C
C   OR EXCHANGE POSITION OF TWO REAL EIGENVALUES ON 2*2 DIAGONAL BLOCK C
C   INPUT....                                                         C
C       H=MATRIX TO BE TRANSFORMED                                    C
C       N=DIMENSION OF H,P                                            C
C       ND=DECLARED DIMENSION OF H,P                                  C
C       NX,N2=ROW INDICIES OF THE BLOCK TO BE TRANSFORMED             C
C       K1=0 NO CUMMULATED P CALCULATED                               C
C       K1=1 CUMMULATED P CALCULATED                                  C
C       IP=1 IF POSITION EXCHANGE PERFORMED OTHERWISE=0               C
C       S40=PRECISION CRITERION FROM QR SUBROUTINE                    C
C   OUTPUT.........                                                   C
C       H=TRANSFORMED MATRIX                                          C
C       P IS ORTHOGONAL MATRIX                                        C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

11. Subroutine TRIA

This subroutine performs an orthogonal similarity transformation to triangularize an isolated 2 x 2 diagonal block if the eigenvalues of that block are real.

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C     SUBROUTINE TRIA(A,P,N,NDIM,N2,VAL,SMALL,K1)                     C
C     DIMENSION A(NDIM,NDIM),P(NDIM,NDIM),VAL(2)                      C
C     COMPLEX VAL                                                     C
C  COMPUTES EIGENVALUES OF 2*2 DIAGONAL BLOCK . IF EIGENVALUES ARE    C
C  REAL , AN ORTHOGONAL SIMILARITY TRANSFORMATION IS PERFORMED TO     C
C  TRIANGULARIZE DIAGONAL BLOCK.                                      C
C       B=P*A*PT AND SET A=B                                          C
C  INPUT....                                                          C
C       A=QUASI-TRIANGULAR MATRIX TO BE TRANSFORMED                   C
C       P=THE ORTHOGONAL MATRIX FROM PREVIOUS SIMILARITY TRANSFORMATION C
C       N=DIMENSION OF MATRIX A                                       C
C       NDIM=DECLEARED DIMENSION OF ARRAY A                           C
C       N2=INDEX OF 2*2 BLOCK = ROW AND COLUMN NUMBER OF LOWER        C
C          RIGHT CORNER                                               C
C       SMALL=NORM*MACHINE PRECISION                                  C
C       K1=0, NO CUMMULATED P CALCULATED                              C
C       K1=1, CUMMULATED P CALCULATED                                 C
C  OUTPUT....                                                         C
C       A= QUASI-UPPER TRIANGULAR MATRIX B                            C
C       P=CUMMULATED ORTHOGONAL TRANSFORMATION MATRIX                 C
C       VAL=COMPUTED EIGENVALUES IN VAL(1) AND VAL(2)                 C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

12. Subroutine HSOLVE

This subroutine solves the linear system equation A * X = B by using the Householder reduction.

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C     SUBROUTINE HSOLVE(A,W,B,N,M,NDIM,ID,IE)                         C
C     DIMENSION A(NDIM,NDIM),W(NDIM),B(NDIM,M)                        C
C   SOLVE THE LINEAR EQUATION A*X=B BY HOUSEHOLDER METHOD             C
C   INPUT...                                                          C
C      A=THE MATRIX OF N*N ORDER OF A*X=B                             C
C      B=MATRIX OF A*X=B ,N*M                                         C
C      W=WORKING SPACE                                                C
C      N=DIM.OF A                                                     C
C      NDIM= DECLARED ROW DIMENSION OF A,B, AND W                     C
C      ID= INDICATOR. ID=0 - TRIANGULARIZE A AND SOLVE,               C
C                     ID=1 - BYPASS TRIANGULARIZATION (PREVIOUS CALL REQD.) C
C   OUTPUT....                                                        C
C      A=Q*A=TRIANGULARIZED MATRIX                                    C
C      B=THE SOLUTION X                                               C
C      IE= ERROR INDICATOR, IE=0 NO ERROR                             C
C                           IE=1 NONUNIQUE SOLUTION (NO ERROR)        C
C                           IE=2 UNDEFINED SOLUTION                   C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

13. Subroutine EIGVC

This subroutine computes the matrix of eigenvectors for a given matrix which was previously put in quasi-upper triangular form using subroutine QR. The knowledge of the orthogonal similarity transformation matrix, which reduced the original matrix to quasi-upper triangular form, is then used to compute the eigenvectors of the original matrix.

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      SUBROUTINE EIGVC(A,VAL,N,NDIM,EL,Q,C,SMO,IE)                        C
C      DIMENSION A(NDIM,NDIM),VAL(NDIM),EL(NDIM,NDIM),A1(2,2),A2(2,2),     C
C     1A3(2,2),C(NDIM),Q(NDIM,NDIM)                                        C
C      COMPLEX VAL                                                         C
C  COMPUTE THE EIGENVECTORS OF A QUASI-UPPER TRIANGULAR MATRIX             C
C  INPUT...                                                                C
C      A=QUASI-UPPER TRIANGULAR MATRIX FROM ROUTINE QR                     C
C      VAL=THE EIGENVALUES OF A COMPUTED IN ROUTINE QR                     C
C      Q=SIMILARITY TRANSFORMATION MATRIX FROM QR                          C
C      N=DIMENSION OF A , EL , Q                                           C
C      NDIM=DECLARED DIMENSION OF A , EL , Q                               C
C      SMO=CONVERGENCE CRITERION FROM QR=MACH.PREC.*NORM A                 C
C  OUTPUT...                                                               C
C      EL=THE EIGENVECTOR MATRIX OF A                                      C
C      C=INTERMEDIATE STORAGE VECTOR                                       C
C      IE=0 INDEPENDENT EIGENVECTOR                                        C
C      IE=1 DEFECTIVE EIGENSYSTEM                                          C
C  SUBROUTINES REQD.- LYAPUN,HSOLVE                                        C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

14. Subroutine SOLYAP

This subroutine solves the Lyapunov equation $A * X + X * A^T = C$. Before entering this subroutine, the matrix A must be transformed into quasi-upper triangular form. With the knowledge of the orthogonal similarity transformation matrix, the resulting solution X will be the solution of the original Lyapunov equation.

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      SUBROUTINE SOLYAP(A,C,N,ND,SMO,IE,P,W)                              C
C      DIMENSION A(ND,ND),C(ND,ND),AA(2,2),CC(2,2),AAT(2,2),P(ND,ND),      C
C     1W(ND)                                                               C
C    SOLVE A*X+X*AT=C                                                      C
C    INPUT...                                                              C
C       A=N*N MATRIX IN QUASI-UPPER TRIANGULAR FORM FROM QR SUBROUTINE     C
C       C=N*N SYMMETRIC MATRIX                                             C
C       P= SIMILARITY TRANSFORMATION MATRIX FROM QR SUBROUTINE             C
C       W=WORKING SPACE                                                    C
C       ND=DECLARED DIMENSION OF A , C , P                                 C
C       SMO=CONVERGENCE CRITERION FROM QR=MACH.PREC.*NORM A                C
C    OUTPUT...                                                             C
C       C=X, THE SOLUTION OF LYAPUNOV EQUATION A*X+X*AT=C                  C
C       IE=0 NO ERROR                                                      C
C       IE=1 ERROR-SOLUTION NOT COMPUTED                                   C
C    SUBROUTINES REQD.- LYAPUN,HSOLVE                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

15. Subroutine LYAPUN

This subroutine solves the Lyapunov equation $A * X + X * B = C$ when the dimensions of A, B are 1 x 1 or 2 x 2.

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      SUBROUTINE LYAPUN(A,NA,B,NB,C,IE)                                  C
C      DIMENSION AA(4,4),A(2,2),B(2,2),C(2,2),W(3)                        C
C  SOLVE A*X+X*B=C AND SET C=X                                            C
C  INPUT....                                                              C
C    A AND B ARE 1*1 OR 2*2                                               C
C    NA=DIMENSION OF A (1 OR 2)                                           C
C    NB=DIMENSION OF B (1 OR 2)                                           C
C    C=NA*NB MATRIX                                                       C
C  OUTPUT...                                                              C
C    C=SOLUTION X                                                         C
C    IE=0  NON-SINGULAR CASE                                              C
C    IE=1  NON-UNIQUE SOLUTION                                            C
C    IE=2  UNDEFINED SOLUTION                                             C
C  SUBROUTINE REQD.- HSOLVE                                               C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

## III. THE USER'S MANUAL FOR OPTIMAL AND ITS SUBROUTINES WITH SEVERAL EXAMPLES

### III.1 Introduction

In this part, a user's manual for the main program OPTIMAL and the subroutines is presented. The error and warning messages which are printed directly from the subroutines are also discussed. Several examples are given at the end of this part.

### III.2 User's manual for the control and estimation problem

In this section, the user's manual for the main program OPTIMAL will be presented. Also, the user's manual for the subroutines CONTRL, FILTER, RICCAT will be given in case the user prefers using his own main program.

A. The user's manual for the main program OPTIMAL

<u>INPUT requirement</u>:

The program OPTIMAL is designed so that the required matrices can be input in arbitrary order preceeded by a matrix ID card. Two control cards must be input before the required matrices for any given problem. The required data cards are described sequentially as follows:

<u>Card A</u>: print control card (NN, IO, LP), FORMAT (A6, 2I3)

NN = PRINT

IO = 0, if no optional print wanted.

= 1, if optional print wanted.

LP = 0, if regular line printer will be used.

= 1, if terminal line printer will be used.

<u>Card B</u>: Eigensystem control card (NN, LO, LV), FORMAT (A6, 2I3)

NN = OPTION

LO = 0, no optional open loop eigensystem.

= 1, open loop eigensystem will be computed.

LV = 0, eigenvectors will not be computed.

= 1, eigenvectors will be computed.

The following set of cards is for reading in the required matrices.

Card C: matrix ID card (NN, N1, N2, NT), FORMAT (A6, 2I3, 2X, A1)

NN = name of the matrix to be input. This name is the same as the name used in part I, such as A, B, C1, D1, C2, D2, R1, R2, Q1, Q2, except GAMA for $\Gamma$ and THETA for $\Theta$.

N1 = number of rows of the matrix.

N2 = number of columns of the matrix.

NT = blank for a regular matrix.

= Z for a zero matrix.

= D for a diagonal matrix.

= I for an identity matrix.

= S for a symmetric matrix.

Matrix input cards (free format):

1. If NT = blank, the matrix is entered by rows. For example,

$$A = \begin{bmatrix} 1. & 2. & 3. \\ 4. & 5. & 6. \\ 7. & 8. & 9. \end{bmatrix}$$

requires three cards to be typed.

first card 1., 2., 3.

second card 4., 5., 6.

third card 7., 8., 9.

2. If NT = Z or I, no matrix input cards required.

3. If NT = D, the diagonal elements only are entered. For example,

$$R1 = \begin{bmatrix} 1. & 0. & 0. \\ 0. & 2. & 0. \\ 0. & 0. & 3. \end{bmatrix},$$

requires one card to be typed.

card 1., 2., 3.

4. If NT = S, the matrix is entered by row with the lower triangular part only. For example,

$$Q2 = \begin{bmatrix} 1. & 2. & 4. \\ 2. & 3. & 5. \\ 4. & 5. & 6. \end{bmatrix}$$

requires three cards to be typed.

<u>first card</u> 1.

<u>second card</u> 2., 3.

<u>third card</u> 4., 5., 6.

Following all the data cards for the required matrices for a given problem, a blank card is used to indicate another set of data cards for a second problem will follow.

<u>Output Information</u>

1. The stochastic control problem.

The output information contains items:

a. solution of regulator Riccati equation
b. regulator control gain
c. closed loop dynamics matrix
d. closed loop eigenvalues for controller
e. closed loop eigenvectors for controller
f. open loop eigenvalues
g. open loop eigenvectors
h. rms control
i. control covariance matrix
j. rms output
k. output covariance matrix
l. solution of the filter Riccati equation
m. filter gain
n. rms estimation error
o. estimation error dynamics matrix
p. estimation eigenvalues
q. estimation eigenvectors
r. rms state
s. state covariance matrix

t. filter implementation dynamics matrix

u. eigenvalues of filter implemantation

v. eigenvectors of filter implementation

If IO = 0, items c, i, k, o, s, t, u, v will not be printed.

If LO = 0, items f, g will not be computed or printed.

If LV = 0, items e, g, q, v will not be computed or printed.

2. The control problem without process noise.

The output information contains items: a, b, c, d, e, f, g.

If IO = 0, item c will not be printed.

If LO = 0, items f, g will not be printed.

IF LV = 0, items e, g will not be printed.

3. The control problem with process noise.

The output information contains items: a, b, c, d, e, f, g, h, i, j, k plus

w. rms regulator state

x. regulator state covariance matrix.

If IO = 0, items c, i, k, x will not be printed.

If LO = 0, items f, g will not be computed or printed.

If LV = 0, items e, g will not be computed or printed.

4. The state estimation problem.

The output information contains items: f, g, l, m, n, o, p, q.

If IO = 0, item o will not be printed.

If LO = 0, items f, g will not be computed or printed.

If LV = 0, items g, q will not be computed or printed.

5. Steady state response of the system with zero control.

The output information contains items: f, g, j, k, w, x.

If IO = 0, items k, x will not be printed.

If LV = 0, item g will not be computed or printed.

Parameter LO does not play a role in this problem.

B. Subroutines CONTRL, FILTER, and RICCAT.

Under some circumstances, the user may prefer to use his own main program. In this case, subroutines CONTRL and FILTER can be used for control and estimation purposes. In cases when the given problem can not be formulated as one of the cases in part I, the subroutine RICCAT can be used to solve control and estimation problems if the Hamiltonian matrix of the Euler-Lagrange system can be formulated. With the given Hamiltonian matrix, RICCAT will give the solution of the corresponding Riccati equation as well as the closed loop eigenvalues. The calling procedure of subroutine CONTRL, FILTER and RICCAT is given in section II.3.

III.3 User's manual for other utility subroutines.

There are several subroutines that can be used for various purposes. The required calling procedures are as follows:

1. Solving the linear system equation A * X = B
   a. for single B (i.e., solve A * X = B only)
      call HSOLVE (A, W, B, N, M, NDIM, 0, IE)
   b. solve A * $X_1$ = $B_1$, then solve A * $X_2$ = $B_2$ etc.
      call HSOLVE (A, W, B1, N, M1, NDIM, 0, IE),
      call HSOLVE (A, W, B2, N, M2, NDIM, 1, IE),
      etc.
2. Computing the eigenvalues and eigenvectors of matrix A
   a. compute the eigenvalues of A only
      call QR (A, P, N, ND, VAL, MV, SNO, 0 , 0, 0)
   b. compute the eigenvalues and eigenvectors of A
      call QR (A, P, N, ND, VAL, N, SNO, 0, 0, 1)
      call EIGVC (A, VAL, N, ND, EL, P, C, SNO, IE).
   c. determination of invariant subspace
      Let A be an n x n matrix. In order to determine the invariant subspace associated with m desired eigenvalues, the following calling sequence is needed.
      Set P = A
      call QR (P, A, N, ND, VAL1, N, SNO, 0, 0, 0)
      take n-m undesired eigenvalues and set VAL2 to these eigenvalues, then
      call QR (A, P, N, ND, VAL2, N-M, SNO, 0, 1, 1).
      The first m rows of the matrix P form an orthogonal basis of the desired invariant subspace.
3. Solving the Lyapunov equation A * X + X * $A^T$ = C
   call QR (A, P, N, ND, VAL, N, SNO, 0, 0, 1)
   call SOLYAP (A, C, N, ND, SNO, IE, P, W).

III.4 Error and Warning Messages

In the subroutine QR, if a particular eigenvalue or eigenvalue pair takes more than 20 iterations of the QR algorithm to converge, then the iteration process will be terminated and an error message - QR NOT CONVERGING - will be printed out. Usually, the algorithm will converge with an error equal to the machine precision times norm of the matrix within 20 iterations. However, in the case of repeated eigenvalues, the algorithm is very likely to converge only with the square root of the machine precision times norm of the matrix. To avoid the error termination in this case, the convergence criterion is changed to the square root of the machine precision times norm of the matrix on the twentieth iteration step.

In subroutine RICCAT, if the Hamiltonian matrix has repeated eigenvalues on (or near) the imaginary axis and/or the origin, the message - WARNING HAMILTONIAN HAS EIGENVALUES ON IMAGINARY AXIS -, or - WARNING HAMILTONIAN MATRIX NEARLY SINGULAR - will be printed. In this situation, the problem is ill-conditioned, but the execution is not terminated. When this happens, it is suggested to check the solution of the Riccati equation as well as the eigenvalues of the closed loop system. Since it is very unlikely that the Hamiltonian matrix would have more than two pairs of eigenvalues at the origin or have more than one pair of eigenvalues at the same point on the imaginary axis, no error message is printed out to indicate these possible program failure modes.

## III.5. Examples

In this section, five example problems and the corresponding input and output for the program are given.

Input Cards:

```
1234567890123456789012345678901234567890
PRINT   1  0
OPTION  1  0
A       2  2  S
1.
2.,3.
B       2  1
1.
2.
C1      2  2  I
D1      2  1
2.
3.
Q1      2  2  D
2.,1.
R1      1  1
2.

PRINT   1  0
OPTION  0  1
A       4  4
0.,1.,0.,0.
0.,-.415,-.0111,0.
9.8,-1.43,-.0198,0.
0.,0.,1.,0.
B       4  1
0.
6.27
9.8
0.
C1      1  4
0.,0.,0.,1.
D1      1  1  Z
GAMA    4  1
0.
-.0111
-.0198
0.
```

(continued on next page)

```
Q1       1  1
.25
R1       1  1
131.3316
Q2       1  1
490.

PRINT    0  0
OPTION   1  1
A        2  2  S
1.
2.,3.
B        2  1
1.
2.
C2       2  2  I
D2       2  1  Z
GAMA     2  2  D
1.,1.
THETA    2  2  Z
Q2       2  2  D
2.,1.
R2       1  1
2.

PRINT    1  0
OPTION   1  1
A        4  4
0.,.1,0.,0.
0.,-.415,-.0111,0.
9.8,-1.43,-.0198,0.
0.,0.,1.,0.
B        4  1
0.
6.27
9.8
0.
GAMA     4  1
0.
-.0111
-.0198
0.
C1       1  4
0.,0.,0.,1.
D1       1  1  Z
C2       2  4
0.,0.,0.,1.
1.,0.,0.,0.
D2       2  1  Z
THETA    2  1  Z
Q1       1  1
.25
R1       1  1
131.3316
Q2       1  1
490.
R2       2  2  D
.272,.0000153

PRINT    0  0
OPTION   0  1
A        2  2  S
-7.
2.,-3.
GAMA     2  1
1.
2.
C1       1  2
3.,2.
Q2       1  1
3.
```

Results generated by the program:

```
A          MATRIX

  1.0000      2.0000
  2.0000      3.0000

B          MATRIX

  1.0000
  2.0000

C1         MATRIX

  1.0000     0.
 0.           1.0000

D1         MATRIX

  2.0000
  3.0000

Q1         MATRIX

  2.0000     0.
 0.           1.0000

R1         MATRIX

  2.0000

PRINT   ID=1    LP=0
OPTION  LD=1    LV=0

 ***CONTROL PROBLEM ***

SOLUTION OF REGULATOR RICCATI EQUATION

  9.5518      10.582
  10.582      22.133

REGULATOR CONTROL GAIN

 -1.8272     -3.0446

CLOSED LOOP EIGENVALUES FOR CONTROLLER


   EIGENVALUE( 1 )

   -.22402     +J 0.


   EIGENVALUE( 2 )

   -3.6924     +J 0.

OPEN LOOP EIGENVALUES


   EIGENVALUE( 1 )

    4.2361     +J 0.


   EIGENVALUE( 2 )

   -.23607     +J 0.
```

```
A          MATRIX

 0.          1.0000      0.           0.
 0.          -.41500     -.11100E-01 0.
 9.8000     -1.4300      -.19800E-01 0.
 0.          0.           1.0000      0.

B          MATRIX

 0.
 6.2700
 9.8000
 0.

C1         MATRIX

 0.          0.          0.           1.0000

D1         MATRIX

 0.

GAMA       MATRIX

 0.
 -.11100E-01
 -.19800E-01
 0.

Q1         MATRIX

  .25000

R1         MATRIX

  131.33

Q2         MATRIX

  490.00

PRINT   IO=1    LP=0
OPTION  LO=0    LV=1

 *****CONTROL PROBLEM WITH PROCESS NOISE***

SOLUTION OF REGULATOR RICCATI EQUATION

  41.393      10.252      4.8968      2.6306
  10.252      5.7935      .64536      .13263
  4.8968      .64536      .73072      .49984
  2.6306      .13263      .49984      .50035

REGULATOR CONTROL GAIN

 -.85487     -.32475     -.85337E-01 -.43630E-01

CLOSED LOOP DYNAMICS MATRIX OF CONTROLLER

 0.           1.0000     0.           0.
 -5.3600     -2.4512     -.54616     -.27356
  1.4223     -4.6125     -.85610     -.42757
 0.           0.          1.0000      0.
```

```
CLOSED LOOP EIGENVALUES AND VECTORS FOR CONTROLLER

   EIGENVALUE( 1 )                 EIGENVECTOR( 1 )

   -.41983     +J  1.1353
                                -.62002E-01 +J  .85322E-01
                                -.70835E-01 +J -.10621
                                 .76029     +J 0.
                                -.21786     +J -.58913

   EIGENVALUE( 2 )                 EIGENVECTOR( 2 )

   -.41983     +J -1.1353
                                -.62002E-01 +J -.85322E-01
                                -.70835E-01 +J  .10621
                                 .76029     +J 0.
                                -.21786     +J  .58913

   EIGENVALUE( 3 )                 EIGENVECTOR( 3 )

   -1.2338     +J  .55452
                                -.81159E-01 +J  .18382E-01
                                 .89942E-01 +J -.67685E-01
                                 .79621     +J 0.
                                -.53688     +J -.24129

   EIGENVALUE( 4 )                 EIGENVECTOR( 4 )

   -1.2338     +J -.55452
                                -.81159E-01 +J -.18382E-01
                                 .89942E-01 +J  .67685E-01
                                 .79621     +J 0.
                                -.53688     +J  .24129

RMS  REGULATOR STATE

  .69020E-01  .12859      .46711      .62056

REGULATOR STATE COVARIANCE MATRIX

  .47637E-02 -.27756E-16 -.38103E-02 -.25286E-01
  .34694E-16  .16535E-01 -.20849E-01  .38103E-02
 -.38103E-02 -.20849E-01  .21819      .54262E-14
 -.25286E-01  .38103E-02  .58495E-14  .38509

RMS  CONTROL

  .63698E-01

CONTROL COVARIANCE MATRIX

  .40574E-02

RMS  OUTPUT

  .62056

OUTPUT COVARIANCE MATRIX

  .38509
```

```
A          MATRIX

  1.0000      2.0000
  2.0000      3.0000

B          MATRIX

  1.0000
  2.0000

C2         MATRIX

  1.0000     0.
 0.           1.0000

D2         MATRIX

 0.
 0.

GAMA       MATRIX

  1.0000     0.
 0.           1.0000

THETA      MATRIX

 0.          0.
 0.          0.

Q2         MATRIX

  2.0000     0.
 0.           1.0000

R2         MATRIX

  2.0000

PRINT   IO=0    LP=0
OPTION  LO=1    LV=1

 *****ESTIMATION PROBLEM*****

SOLUTION OF FILTER RICCATI EQUATION

  18.184      31.740
  31.740      62.626

FILTER GAIN

 -9.0920
 -15.870

RMS  ESTIMATION ERROR

  4.2643      7.9136

ESTIMATION EIGENVALUES AND VECTORS


   EIGENVALUE( 1 )               EIGENVECTOR( 1 )

   -.80876     +J 0.
                            -.26480
                            -.96430


   EIGENVALUE( 2 )               EIGENVECTOR( 2 )

   -4.2832     +J 0.
                             .46491
                             .88536
```

OPEN LOOP EIGENVALUES AND VECTORS

```
   EIGENVALUE( 1 )                EIGENVECTOR( 1 )

    4.2361     +J 0.
                                 .52573
                                 .85065

   EIGENVALUE( 2 )                EIGENVECTOR( 2 )

   -.23607     +J 0.
                                 .85065
                                -.52573
```

```
A          MATRIX

 0.           .10000     0.           0.
 0.          -.41500    -.11100E-01  0.
  9.8000     -1.4300    -.19800E-01  0.
 0.          0.           1.0000     0.

B          MATRIX

 0.
  6.2700
  9.8000
 0.

GAMA       MATRIX

 0.
 -.11100E-01
 -.19800E-01
 0.

C1         MATRIX

 0.          0.          0.           1.0000

D1         MATRIX

 0.

C2         MATRIX

 0.          0.          0.           1.0000
  1.0000     0.          0.          0.

D2         MATRIX

 0.
 0.

THETA      MATRIX

 0.
 0.

Q1         MATRIX

  .25000

R1         MATRIX

  131.33
```

```
Q2         MATRIX

 490.00

R2         MATRIX

  .27200     0.
 0.           .15300E-04

PRINT   IO=1    LP=0
OPTION  LO=1    LV=1

 *****CONTROL PLUS ESTIMATION PROBLEM*****

SOLUTION OF REGULATOR RICCATI EQUATION

  1305.2     -47.809     75.603     17.194
 -47.809      8.4142    -5.8381    -2.1559
  75.603     -5.8381     6.0205     1.9640
  17.194     -2.1559     1.9640      .99210

REGULATOR CONTROL GAIN

 -3.3590      .33927E-01 -.17053    -.43630E-01

CLOSED LOOP DYNAMICS MATRIX OF CONTROLLER

 0.           .10000     0.          0.
 -21.061     -.20228    -1.0803     -.27356
 -23.118     -1.0975    -1.6910     -.42757
 0.          0.          1.0000      0.

CLOSED LOOP EIGENVALUES AND VECTORS FOR CONTROLLER


   EIGENVALUE( 1 )               EIGENVECTOR( 1 )

   -.74398     +J  .39390
                                 .18996E-01 +J -.95168E-02
                                -.10384     +J  .14563
                                -.55985     +J  .29641
                                 .75251     +J 0.


   EIGENVALUE( 2 )               EIGENVECTOR( 2 )

   -.74398     +J -.39390
                                 .18996E-01 +J  .95168E-02
                                -.10384     +J -.14563
                                -.55985     +J -.29641
                                 .75251     +J 0.


   EIGENVALUE( 3 )               EIGENVECTOR( 3 )

   -.20266     +J  .58072
                                -.60327E-03 +J -.29726E-01
                                 .17385     +J  .56739E-01
                                -.16963     +J  .48608
                                 .83703     +J 0.


   EIGENVALUE( 4 )               EIGENVECTOR( 4 )

   -.20266     +J -.58072
                                -.60327E-03 +J  .29726E-01
                                 .17385     +J -.56739E-01
                                -.16963     +J -.48608
                                 .83703     +J 0.
```

SOLUTION OF FILTER RICCATI EQUATION

| | | | |
|---|---|---|---|
| .47608E-04 | .74876E-03 | .11386E-02 | .66270E-03 |
| .74876E-03 | .26823E-01 | .42586E-01 | .11909E-01 |
| .11386E-02 | .42586E-01 | .75314E-01 | .36511E-01 |
| .66270E-03 | .11909E-01 | .36511E-01 | .10979 |

FILTER GAIN

| | |
|---|---|
| -.24364E-02 | -3.1116 |
| -.43784E-01 | -48.939 |
| -.13423 | -74.420 |
| -.40365 | -43.314 |

RMS ESTIMATION ERROR

| | | | |
|---|---|---|---|
| .68998E-02 | .16378 | .27443 | .33135 |

ESTIMATION ERROR DYNAMICS MATRIX

| | | | |
|---|---|---|---|
| -3.1116 | .10000 | 0. | -.24364E-02 |
| -48.939 | -.41500 | -.11100E-01 | -.43784E-01 |
| -64.620 | -1.4300 | -.19800E-01 | -.13423 |
| -43.314 | 0. | 1.0000 | -.40365 |

ESTIMATION EIGENVALUES AND VECTORS

EIGENVALUE( 1 ) EIGENVECTOR( 1 )

-1.7955 +J 1.7644

| | |
|---|---|
| .10481E-01 | +J -.18999E-01 |
| .47797 | +J -.73433E-01 |
| .78127 | +J 0. |
| .19729 | +J -.34113 |

EIGENVALUE( 2 ) EIGENVECTOR( 2 )

-1.7955 +J -1.7644

| | |
|---|---|
| .10481E-01 | +J .18999E-01 |
| .47797 | +J .73433E-01 |
| .78127 | +J 0. |
| .19729 | +J .34113 |

EIGENVALUE( 3 ) EIGENVECTOR( 3 )

-.17958 +J .20152

| | |
|---|---|
| -.89984E-03 | +J -.20609E-04 |
| -.28326E-02 | +J -.24176E-02 |
| .17724 | +J .19356 |
| .96494 | +J 0. |

EIGENVALUE( 4 ) EIGENVECTOR( 4 )

-.17958 +J -.20152

| | |
|---|---|
| -.89984E-03 | +J .20609E-04 |
| -.28326E-02 | +J .24176E-02 |
| .17724 | +J -.19356 |
| .96494 | +J 0. |

RMS STATE

| | | | |
|---|---|---|---|
| .36583E-01 | .32692 | .52505 | 1.1939 |

STATE COVARIANCE MATRIX

| | | | |
|---|---|---|---|
| .13383E-02 | .72741E-12 | -.10573E-01 | -.14016E-01 |
| .60673E-12 | .10688 | .35638E-01 | .10573 |
| -.10573E-01 | .35638E-01 | .27567 | .21030E-11 |
| -.14016E-01 | .10573 | .34808E-11 | 1.4255 |

RMS CONTROL

.68533E-01

CONTROL COVARIANCE MATRIX

.46968E-02

RMS OUTPUT

1.1939

OUTPUT COVARIANCE MATRIX

1.4255

FILTER IMPLEMENTATION DYNAMICS MATRIX

| | | | |
|---|---|---|---|
| -3.1116 | .10000 | 0. | -.24364E-02 |
| -70.000 | -.20228 | -1.0803 | -.31734 |
| -97.538 | -1.0975 | -1.6910 | -.56181 |
| -43.314 | 0. | 1.0000 | -.40365 |

FILTER IMPLEMENTATION EIGENVALUES AND VECTORS

EIGENVALUE( 1 ) EIGENVECTOR( 1 )

-2.5838 +J 2.5680

.16424E-02 +J -.18667E-01
.48907 +J -.64200E-01
.80686 +J 0.
.41632E-01 +J -.32184

EIGENVALUE( 2 ) EIGENVECTOR( 2 )

-2.5838 +J -2.5680

.16424E-02 +J .18667E-01
.48907 +J .64200E-01
.80686 +J 0.
.41632E-01 +J .32184

EIGENVALUE( 3 ) EIGENVECTOR( 3 )

-.39114E-01 +J 0.

.57668E-02
.15325
-.10828
-.98222

EIGENVALUE( 4 ) EIGENVECTOR( 4 )

-.20189 +J 0.

-.45560E-02
-.10835
.32326E-02
.99410

```
OPEN LOOP EIGENVALUES AND VECTORS

   EIGENVALUE( 1 )               EIGENVECTOR( 1 )

    .29960E-01 +J  .14523
                             -.22300E-02 +J  .72784E-03
                             -.17251E-02 +J -.30205E-02
                              .29636E-01 +J  .14365
                              .98918     +J 0.

   EIGENVALUE( 2 )               EIGENVECTOR( 2 )

    .29960E-01 +J -.14523
                             -.22300E-02 +J -.72784E-03
                             -.17251E-02 +J  .30205E-02
                              .29636E-01 +J -.14365
                              .98918     +J 0.

   EIGENVALUE( 3 )               EIGENVECTOR( 3 )

   -.49472     +J 0.
                             -.12455E-01
                              .61618E-01
                              .44255
                             -.89454

   EIGENVALUE( 4 )               EIGENVECTOR( 4 )

   -.34168E-13 +J 0.
                             -.14319E-14
                              .73534E-14
                             -.42177E-13
                              1.0000


A          MATRIX

 -7.0000     2.0000
  2.0000    -3.0000

GAMA       MATRIX

  1.0000
  2.0000

C1         MATRIX

  3.0000     2.0000

Q2         MATRIX

  3.0000

PRINT   IO=0    LP=0
OPTION  LO=0    LV=1

 *****STEADY STATE COVARIANCE PROBLEM*****
```

OPEN LOOP EIGENVALUES AND VECTORS

| | |
|---|---|
| EIGENVALUE( 1 ) | EIGENVECTOR( 1 ) |
| -2.1716 +J 0. | |
| | .38268 |
| | .92388 |
| EIGENVALUE( 2 ) | EIGENVECTOR( 2 ) |
| -7.8284 +J 0. | |
| | .92388 |
| | -.38268 |

RMS STEADY STATE

.76312 1.6908

RMS OUTPUT FOR ZERO CONTROL

5.6688

## REFERENCES FOR MANUAL


1. Wei, S. Y., "Application of Linear Quadratic Control Design in Reduction of Aerodynamic Forces in Aircraft," Ph.D. dissertation, Mechanical Engineering Department, Oregon State University, 1980, Chapter 2.
2. Davis, M. H. A., Linear Estimation and Stochastic Control, a Halstead press book, John Wiley and Sons, NY, 1977, pp. 169.
3. Åström, K. J., Introduction to Stochastic Control Theory, Academic Press, NY and London, 1970, pp. 264-268, p. 286-289.
4. Fleming, W. H. and Rishel, R. W., "Deterministic and Stochastic Optimal Control," Application of Mathematics, Vol. 1, Springer-Verlag, NY, Heidelberg, Berlin, 1975, Chapter 6.
5. Sage, A. P. and Melsa, J. L., System Identification, Academic Press, NY, 1967, Chapter 3.
6. Bryson, A. E., Jr., and Ho, Y. C., Applied Optimal Control, John Wiley and Sons, NY, 1975, pp. 414.
7. Wonham, W. M., "Random Differential Equations in Control Theory," in Probabilistic Methods in Applied Mathematics, edited by Bharucha-Reid, A.T., Volume 2, Academic Press, NY and London, 1970.
8. Sage, A. P. and C. C. White, III, Optimum Systems Control, 2nd Ed., Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1968, chapter 9.
9. MacFarlane, A. G. J., "An Eigenvector Solution to the Optimal Regulator Problem," J. Electronics and Control, Vol. 14, June 1963, pp. 643-654.
10. Potter, J. E., "Matrix Quadratic Solution," SIAM J. Appl. Math., Vol. 14, May 1966, pp. 496-501.
11. Holley, W. E., and Wei, S. Y., "Improved Method for Solving the Algebraic Riccati Equation," AIAA Journal of Guidance and Control, Vol. 3, No. 2, 1980, pp. 190-192.
12. Wilkinson, J. H., and Reinsch, C., "Linear Algebra," in Handbook of Automatic Computation, Springer-Verlag, 1971.
13. Wilkinson, J. H., The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965.

# APPENDIX F

# THE OPTIMAL PROGRAM LISTING

```
      PROGRAM OPTIMAL (INPUT,OUTPUT,TAPE5=INPUT)
      DIMENSION A(30,30), B(30,10), C1(10,30), D1(10,10), C2(10,30), D2
     1(10,10), THETA(10,10), GAIN(10,30), GK(30,10), GAMA(30,10), Q1(30,
     230), R1(30,30), Q2(30,30), R2(30,30), S(30,30), H(60,60), P(60,60)
     3, VAL(60), W(60), EL(30,30), PV(30,30)
      COMPLEX VAL
      DATA ND/30/,NDD/60/,NDL/10/,NDM/10/,NDP/10/,NDQ/10/
      DATA NA/1HA/,NB/1HB/,NC1/2HC1/,ND1/2HD1/,NC2/2HC2/,ND2/2HD2/,NGA/
     14HGAMA/,NQ1/2HQ1/,NR1/2HR1/,NQ2/2HQ2/,NR2/2HR2/,NTH/5HTHETA/,NZ/
     21HZ/,NPR/5HPRINT/,NOP/6HOPTION/
      DATA SMALL/1.E-14/
      DATA IUNT/5LINPUT/
C
C     SOLVE OPTIMAL CONTROL AND ESTIMATION PROBLEMS.
C     INPUT*****
C     IO=0 NO OPTIONAL PRINT WANTED
C     IO=1 OPTIONAL PRINT WANTED
C     LP=0 REGULAR LINE PRINTER WILL BE USED
C     LP=1 TERMINAL LINE PRINTER WILL BE USED
C     LO=0  NO OPTIONAL OPEN LOOP EIGENSYSTEM
C     LO=1  COMPUTE OPEN LOOP EIGENSYSTEM
C     LV=0  EIGENVECTORS NOT COMPUTED
C     LV=1  COMPUTE EIGENVECTORS
C     INPUT NECCERSARY MATRICES
      IEOF = 0
      ID1 = 0
      ID2 = 0
      IS1 = 0
      IO = 0
      LP = 0
      LO = 0
      LV = 0
   20 IQ1 = 0
      IR1 = 0
      IR2 = 0
      IQ2 = 0
   30 READ (5,910) NN,N1,N2,NT
      IF (EOF(IUNT)) 40,50,40
   40 IEOF = 1
      GO TO 200
   50 IF (NN.EQ.NA) GO TO 80
      IF (NN.EQ.NPR) GO TO 60
      IF (NN.EQ.NOP) GO TO 70
      IF (NN.EQ.NB) GO TO 90
      IF (NN.EQ.NC1) GO TO 100
      IF (NN.EQ.ND1) GO TO 110
      IF (NN.EQ.NQ1) GO TO 120
      IF (NN.EQ.NR1) GO TO 130
      IF (NN.EQ.NGA) GO TO 140
      IF (NN.EQ.NQ2) GO TO 150
      IF (NN.EQ.NC2) GO TO 160
      IF (NN.EQ.ND2) GO TO 170
```

```
      IF (NN.EQ.NR2) GO TO 180
      IF (NN.EQ.NTH) GO TO 190
      GO TO 200
60    IO = N1
      LP = N2
      GO TO 30
70    LO = N1
      LV = N2
      GO TO 30
80    CALL READM (A,N1,N2,ND,NT,NA,LP)
      N = N1
      GO TO 30
90    CALL READM (B,N1,N2,ND,NT,NB,LP)
      N = N1
      L = N2
      GO TO 30
100   CALL READM (C1,N1,N2,NDM,NT,NC1,LP)
      M = N1
      N = N2
      GO TO 30
110   CALL READM (D1,N1,N2,NDM,NT,ND1,LP)
      M = N1
      L = N2
      ID1 = 1
      IF (NT.EQ.NZ) ID1 = 0
      GO TO 30
120   CALL READM (Q1,N1,N2,ND,NT,NQ1,LP)
      IQ1 = 1
      M = N1
      GO TO 30
130   CALL READM (R1,N1,N2,ND,NT,NR1,LP)
      IR1 = 1
      L = N1
      GO TO 30
140   CALL READM (GAMA,N1,N2,ND,NT,NGA,LP)
      N = N1
      NP = N2
      GO TO 30
150   CALL READM (Q2,N1,N2,ND,NT,NQ2,LP)
      NP = N1
      IQ2 = 1
      GO TO 30
160   CALL READM (C2,N1,N2,NDQ,NT,NC2,LP)
      NQ = N1
      N = N2
      GO TO 30
170   CALL READM (D2,N1,N2,NDQ,NT,ND2,LP)
      ID2 = 1
      IF (NT.EQ.NZ) ID2 = 0
      NQ = N1
      L = N2
      GO TO 30
```

```
 180  CALL READM (R2,N1,N2,ND,NT,NR2,LP)
      IR2 = 1
      NQ = N1
      GO TO 30
 190  CALL READM (THETA,N1,N2,NDQ,NT,NTH,LP)
      NQ = N1
      NP = N2
      IS1 = 1
      IF (NT.EQ.NZ) IS1 = 0
      GO TO 30
 200  IPR = IR1+IQ2+IR2+IR2
      IF (IPR.EQ.1.AND.IQ1.EQ.0) IPR = 0
      PRINT 920, IO,LP
      PRINT 930, LO,LV
C
C     IPR IS THE PROBLEM INDICATOR
C     =0 STEADY STATE COVARIANCE
C     =1 CONTROL WITHOUT PROCESS NOISE
C     =2 CONTROL WITH PROCESS NOISE
C     =3 ESTIMATION
C     =4 CONTROL + ESTIMATION
      IF(IPR.EQ.0) PRINT*,42H *****STEADY STATE COVARIANCE PROBLEM*****
      IF(IPR.EQ.1)PRINT*,23H ***CONTROL PROBLEM ***
      IF(IPR.EQ.2)PRINT*,43H *****CONTROL PROBLEM WITH PROCESS NOISE***
      IF(IPR.EQ.3)PRINT*,29H *****ESTIMATION PROBLEM*****
      IF(IPR.EQ.4)PRINT*,42H *****CONTROL PLUS ESTIMATION PROBLEM*****
      IF (IPR.EQ.0) GO TO 250
      IF (IPR.EQ.3) GO TO 470
      CALL CONTRL (A,B,C1,D1,Q1,R1,S,GAIN,H,P,VAL,N,M,L,ND,NDD,NDM,NDL,W
     1,ID1)
      PRINT 940
      CALL PRINT (S,N,N,ND,LP)
      PRINT 950
      CALL PRINT (GAIN,L,N,NDL,LP)
      IF (LV.EQ.1) GO TO 210
      PRINT 960
      CALL PREVAL (VAL,EL,ND,N,0,SMO)
      IF (IPR.EQ.1) GO TO 800
 210  DO 230 I=1,N
         DO 230 J=1,N
            SUM = 0.
            DO 220 K=1,L
 220        SUM = SUM+B(I,K)*GAIN(K,J)
 230  R1(I,J) = A(I,J)+SUM
      IF (IO.EQ.0) GO TO 240
      PRINT 970
      CALL PRINT (R1,N,N,ND,LP)
 240  IF (IPR.EQ.4.OR.IPR.EQ.1) GO TO 290
 250  DO 260 I=1,NP
         DO 260 J=1,N
            H(I,J) = 0.
            DO 260 K=1,NP
```

```
260  H(I,J) = H(I,J)+Q2(I,K)*GAMA(J,K)
     DO 230 I=1,N
        DO 230 J=1,N
           SUM = 0.
           DO 270 K=1,NP
270        SUM = SUM+GAMA(I,K)*H(K,J)
280  Q2(I,J) = -SUM
     IF (IPR.EQ.0) GO TO 310
290  SMO = SMALL
     CALL QR (R1,PW,N,ND,VAL,N,SMO,0,0,1)
     IF (LV.EQ.0) GO TO 300
     CALL EIGVC (R1,VAL,N,ND,EL,PW,W,SMO,IE)
     PRINT 980
     CALL PREVAL (VAL,EL,ND,N,1,SMO)
300  IF (IPR.EQ.1) GO TO 300
     IF (IPR.EQ.4) GO TO 420
     CALL SOLYAP (R1,Q2,N,ND,SMO,IE,PW,W)
     DO 310 I=1,N
310  W(I) = SQRT(Q2(I,I))
     PRINT 990
     CALL PRINT (W,1,N,1,LP)
     IF (IO.EQ.0) GO TO 320
     PRINT 1000
     CALL PRINT (Q2,N,N,ND,LP)
320  DO 330 I=1,N
        DO 330 J=1,L
           H(I,J) = 0.
           DO 330 K=1,N
330  H(I,J) = H(I,J)+Q2(I,K)*GAIN(J,K)
     DO 340 I=1,L
        DO 340 J=1,L
           P(I,J) = 0.
           DO 340 K=1,N
340  P(I,J) = P(I,J)+GAIN(I,K)*H(K,J)
     DO 350 I=1,L
350  W(I) = SQRT(P(I,I))
     PRINT 1010
     CALL PRINT (W,1,L,1,LP)
     IF (IO.EQ.0) GO TO 360
     PRINT 1020
     CALL PRINT (P,L,L,NDD,LP)
360  DO 380 I=1,M
        DO 380 J=1,N
           SUM = 0.
           IF (ID1.EQ.0) GO TO 380
           DO 370 K=1,L
370        SUM = SUM+D1(I,K)*GAIN(K,J)
380  S(I,J) = C1(I,J)+SUM
     DO 390 I=1,M
        DO 390 J=1,N
           R1(I,J) = 0.
           DO 390 K=1,N
```

```
 390  R1(I,J) = R1(I,J)+S(I,K)*Q2(K,J)
      DO 400 I=1,M
         DO 400 J=1,M
            H(I,J) = 0.
            DO 400 K=1,N
 400  H(I,J) = H(I,J)+R1(I,K)*S(J,K)
      DO 410 I=1,M
 410  W(I) = SQRT(H(I,I))
      PRINT 1030
      CALL PRINT (W,1,M,1,LP)
      IF (IO.EQ.0) GO TO 300
      PRINT 1040
      CALL PRINT (H,M,M,NDD,LP)
      GO TO 300
 420  IF (IS1.EQ.0) GO TO 440
      DO 430 I=1,NP
         DO 430 J=1,NQ
            H(I,J) = 0.
            DO 430 K=1,NP
 430  H(I,J) = H(I,J)+Q2(I,K)*THETA(J,K)
 440  DO 460 I=1,NQ
         DO 460 J=1,NQ
            SUM = 0.
            IF (IS1.EQ.0) GO TO 460
            DO 450 K=1,NP
 450        SUM = SUM+THETA(I,K)*H(K,J)
 460  Q1(I,J) = R2(I,J)+SUM
 470  CALL FILTER (A,C2,GAMA,THETA,Q2,R2,S,GK,H,P,VAL,N,NQ,NP,ND,NDD,NDQ
     1,NDP,W,IS1)
      PRINT 1050
      CALL PRINT (S,N,N,ND,LP)
      PRINT 1060
      CALL PRINT (GK,N,NQ,ND,LP)
      DO 480 I=1,N
 480  W(I) = SQRT(S(I,I))
      PRINT 1070
      CALL PRINT (W,1,N,1,LP)
      IF (LV.EQ.1) GO TO 490
      PRINT 1080
      CALL PREVAL (VAL,EL,ND,N,0,SMO)
      GO TO 530
 490  DO 510 I=1,N
         DO 510 J=1,N
            SUM = 0.
            DO 500 K=1,NQ
 500        SUM = SUM+GK(I,K)*C2(K,J)
 510  R2(I,J) = A(I,J)+SUM
      IF (IO.EQ.0) GO TO 520
      PRINT 1090
      CALL PRINT (R2,N,N,ND,LP)
 520  SMO = SMALL
      CALL QR (R2,Q2,N,ND,VAL,N,SMO,0,0,1)
```

```
      CALL EIGVC (R2,VAL,N,ND,EL,Q2,W,SMO,IE)
      PRINT 1100
      CALL PREVAL (VAL,EL,ND,N,1,SMO)
  530 IF (IPR.EQ.3) GO TO 300
      DO 540 I=1,NQ
         DO 540 J=1,N
            H(I,J) = 0.
            DO 540 K=1,NQ
  540 H(I,J) = H(I,J)+Q1(I,K)*GK(J,K)
      DO 560 I=1,N
         DO 560 J=1,N
            SUM = 0.
            DO 550 K=1,NQ
  550       SUM = SUM+GK(I,K)*H(K,J)
  560 Q1(I,J) = -SUM
      CALL SOLYAP (R1,Q1,N,ND,SMO,IE,PW,W)
      DO 570 I=1,N
         DO 570 J=1,N
  570 PW(I,J) = Q1(I,J)+S(I,J)
      DO 580 I=1,N
  580 W(I) = SQRT(PW(I,I))
      PRINT 1110
      CALL PRINT (W,1,N,1,LP)
      IF (IO.EQ.0) GO TO 590
      PRINT 1120
      CALL PRINT (PW,N,N,ND,LP)
  590 DO 600 I=1,N
         DO 600 J=1,L
            H(I,J) = 0.
            DO 600 K=1,N
  600 H(I,J) = H(I,J)+Q1(I,K)*GAIN(J,K)
      DO 610 I=1,L
         DO 610 J=1,L
            P(I,J) = 0.
            DO 610 K=1,N
  610 P(I,J) = P(I,J)+GAIN(I,K)*H(K,J)
      DO 620 I=1,L
  620 W(I) = SQRT(P(I,I))
      PRINT 1130
      CALL PRINT (W,1,L,1,LP)
      IF (IO.EQ.0) GO TO 630
      PRINT 1140
      CALL PRINT (P,L,L,NDD,LP)
  630 DO 640 I=1,N
         DO 640 J=1,M
            Q1(I,J) = 0.
            DO 640 K=1,N
  640 Q1(I,J) = Q1(I,J)+PW(I,K)*C1(J,K)
      IF (ID1.EQ.0) GO TO 680
      DO 650 I=1,N
         DO 650 J=1,M
            S(I,J) = 0.
```

```
          DO 650 K=1,L
650   S(I,J) = S(I,J)+H(I,K)*D1(J,K)
      DO 660 I=1,M
         DO 660 J=1,M
            H(I,J) = 0.
            DO 660 K=1,N
660   H(I,J) = H(I,J)+C1(I,K)*S(K,J)
      DO 670 I=1,L
         DO 670 J=1,M
            S(I,J) = 0.
            DO 670 K=1,L
670   S(I,J) = S(I,J)+P(I,K)*D1(J,K)
680   DO 710 I=1,M
         DO 710 J=1,M
            SUM = 0.
            DO 690 K=1,N
690         SUM = SUM+C1(I,K)*Q1(K,J)
            XW = 0.
            IF (ID1.EQ.0) GO TO 710
            DO 700 K=1,L
700         XW = XW+D1(I,K)*S(K,J)
            XW = XW+H(I,J)+H(J,I)
710   P(I,J) = SUM+XW
      DO 720 I=1,M
720   W(I) = SQRT(P(I,I))
      PRINT 1150
      CALL PRINT (W,1,M,1,LP)
      IF (IO.EQ.0) GO TO 800
      PRINT 1160
      CALL PRINT (P,M,M,NDD,LP)
      DO 740 I=1,NQ
         DO 740 J=1,N
            SUM = 0.
            IF (ID2.EQ.0) GO TO 740
            DO 730 K=1,L
730         SUM = SUM+D2(I,K)*GAIN(K,J)
740   PW(I,J) = C2(I,J)+SUM
      DO 770 I=1,N
         DO 770 J=1,N
            SUM = 0.
            DO 750 K=1,L
750         SUM = SUM+B(I,K)*GAIN(K,J)
            XW = 0.
            DO 760 K=1,NQ
760         XW = XW+GK(I,K)*PW(K,J)
770   R1(I,J) = A(I,J)+SUM+XW
      PRINT 1170
      CALL PRINT (R1,N,N,ND,LP)
      SMO = SMALL
      CALL QR (R1,PW,N,ND,VAL,N,SMO,0,0,1)
      IF (LV.EQ.1) GO TO 780
      PRINT 1180
```

```
      GO TO 790
  780 CALL EIGVC (R1,VAL,N,ND,EL,PW,W,SMO,IE)
      PRINT 1190
  790 CALL PREVAL (VAL,EL,ND,N,LV,SMO)
  800 IF (LO.EQ.0) GO TO 900
  810 SMO = SMALL
      DO 820 I=1,N
         DO 820 J=1,N
  820 R1(I,J) = A(I,J)
      CALL QR (R1,PW,N,ND,VAL,N,SMO,0,0,1)
      IF (LV.EQ.1) GO TO 830
      PRINT 1200
      GO TO 840
  830 CALL EIGVC (R1,VAL,N,ND,EL,PW,W,SMO,IE)
      PRINT 1210
  840 CALL PREVAL (VAL,EL,ND,N,LV,SMO)
      IF (IPR.NE.0) GO TO 900
      CALL SOLYAP (R1,Q2,N,ND,SMO,IE,PW,W)
      DO 850 I=1,N
  850 W(I) = SQRT(Q2(I,I))
      PRINT 1220
      CALL PRINT (W,1,N,1,LP)
      IF (IO.EQ.0) GO TO 860
      PRINT 1230
      CALL PRINT (Q2,N,N,ND,LP)
  860 DO 870 I=1,M
         DO 870 J=1,N
            H(I,J) = 0.
            DO 870 K=1,N
  870 H(I,J) = H(I,J)+C1(I,K)*Q2(K,J)
      DO 890 I=1,M
         DO 880 J=1,M
            Q2(I,J) = 0.
            DO 880 K=1,N
  880    Q2(I,J) = Q2(I,J)+H(I,K)*C1(J,K)
  890 W(I) = SQRT(Q2(I,I))
      PRINT 1240
      CALL PRINT (W,1,M,1,LP)
      IF (IO.EQ.0) GO TO 900
      PRINT 1250
      CALL PRINT (Q2,M,M,ND,LP)
  900 PRINT 1260
      IF (IEOF.NE.0) STOP
      GO TO 20
C
  910 FORMAT (A6,2I3,2X,A1)
  920 FORMAT (/,12H PRINT   IO=,I1,7H    LP=,I1)
  930 FORMAT (12H OPTION  LO=,I1,7H    LV=,I1,/)
  940 FORMAT (/,39H SOLUTION OF REGULATOR RICCATI EQUATION)
  950 FORMAT (/,23H REGULATOR CONTROL GAIN)
  960 FORMAT (/,39H CLOSED LOOP EIGENVALUES FOR CONTROLLER)
  970 FORMAT (/,42H CLOSED LOOP DYNAMICS MATRIX OF CONTROLLER)
```

```
980  FORMAT (/,51H CLOSED LOOP EIGENVALUES AND VECTORS FOR CONTROLLER)
990  FORMAT (/,21H RMS  REGULATOR STATE)
1000 FORMAT (/,34H REGULATOR STATE COVARIANCE MATRIX)
1010 FORMAT (/,13H RMS  CONTROL)
1020 FORMAT (/,26H CONTROL COVARIANCE MATRIX)
1030 FORMAT (/,12H RMS  OUTPUT)
1040 FORMAT (/,25H OUTPUT COVARIANCE MATRIX)
1050 FORMAT (/,36H SOLUTION OF FILTER RICCATI EQUATION)
1060 FORMAT (/,12H FILTER GAIN)
1070 FORMAT (/,22H RMS  ESTIMATION ERROR)
1080 FORMAT (/,23H ESTIMATION EIGENVALUES)
1090 FORMAT (/,33H ESTIMATION ERROR DYNAMICS MATRIX)
1100 FORMAT (/,35H ESTIMATION EIGENVALUES AND VECTORS)
1110 FORMAT (/,10H RMS STATE)
1120 FORMAT (/,24H STATE COVARIANCE MATRIX)
1130 FORMAT (/,12H RMS CONTROL)
1140 FORMAT (/,26H CONTROL COVARIANCE MATRIX)
1150 FORMAT (/,11H RMS OUTPUT)
1160 FORMAT (/,25H OUTPUT COVARIANCE MATRIX)
1170 FORMAT (/,33H FILTER IMPLEMENTATION DYNAMICS MATRIX)
1180 FORMAT (/,34H FILTER IMPLEMENTATION EIGENVALUES)
1190 FORMAT (/,46H FILTER IMPLEMENTATION EIGENVALUES AND VECTORS)
1200 FORMAT (/,22H OPEN LOOP EIGENVALUES)
1210 FORMAT (/,34H OPEN LOOP EIGENVALUES AND VECTORS)
1220 FORMAT (/,18H RMS  STEADY STATE)
1230 FORMAT (/,31H STEADY STATE COVARIANCE MATRIX)
1240 FORMAT (/,29H RMS  OUTPUT FOR ZERO CONTROL)
1250 FORMAT (/,35H OUTPUT COVARIANCE FOR ZERO CONTROL)
1260 FORMAT (///)
     END
```

```
      SUBROUTINE READM (A,N1,N2,N3,NT,NA,LP)
      DIMENSION A(N3,N2)
      DATA NZ/1HZ/,ND/1HD/,NS/1HS/,NI/1HI/
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C     READ AND PRINT N1*N2 MATRIX A                                    C
C     INPUT...                                                         C
C        N1=ROW DIMENSION OF A (NUMBER OF ROWS)                        C
C        N2=COLUMN DIMENSION OF A                                      C
C        N3=DECLARED ROW DIMENSION OF A                                C
C        NT=Z IF A IS ZERO MATRIX                                      C
C        NT=I IF A IS IDENTITY MATRIX                                  C
C        NT=S IF A IS SYMMETRIC MATRIX                                 C
C        NT=D IF A IS DIAGONAL MATRIX                                  C
C        NA=NAME OF MATRIX A                                           C
C        LP=0 FROM REGULAR LINE PRINTER                                C
C        LP=1 FROM TERMINAL PRINTER                                    C
C     SUBROUTINE REQD.- PRINT                                          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IF (NT.EQ.NZ) GO TO 20
      IF (NT.EQ.ND) GO TO 40
      IF (NT.EQ.NS) GO TO 60
      IF (NT.EQ.NI) GO TO 80
      READ (5,*) ((A(I,J),J=1,N2),I=1,N1)
 10   PRINT 110, NA
      CALL PRINT (A,N1,N2,N3,LP)
      RETURN
 20   DO 30 I=1,N1
         DO 30 J=1,N2
 30   A(I,J) = 0.
      GO TO 10
 40   DO 50 I=1,N1
         DO 50 J=1,N2
 50   A(I,J) = 0.
      READ (5,*) (A(I,I),I=1,N1)
      GO TO 10
 60   READ (5,*) ((A(I,J),J=1,I),I=1,N1)
      DO 70 I=1,N1
         DO 70 J=I,N1
 70   A(I,J) = A(J,I)
      GO TO 10
 80   DO 100 I=1,N1
         DO 90 J=1,N1
 90      A(I,J) = 0.
 100  A(I,I) = 1.
      GO TO 10
C
 110  FORMAT (/,1X,A10,7H MATRIX)
      END
```

```
      SUBROUTINE PRINT (A,N1,N2,ND,ID)
      DIMENSION A(ND,N2)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C     PRINT N1*N2 MATRIX A                                             C
C     INPUT...                                                         C
C        A=MATRIX TO BE PRINTED                                        C
C        N1=ROW DIMENSION OF A (NUMBER OF ROWS)                        C
C        N2=COLUMN DIMENSION OF A                                      C
C        ND=DECLARED ROW DIMENSION OF A                                C
C        ID=0 REGULAR LINE PRINTER (120 COLUMN)                        C
C        ID=1 TERMINAL PRINTER (60 COLUMN)                             C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      PRINT*," "
      MD = 10
      IF (ID.EQ.1) MD = 5
      J1 = 1
      J2 = MD
      IF (J2.GT.N2) J2 = N2
      GO TO 20
 10   IF (J2.GE.N2) RETURN
      PRINT*," "
      J1 = J1+MD
      J2 = J2+MD
      IF (J2.GT.N2) J2 = N2
 20   DO 30 I=1,N1
 30   PRINT 40, (A(I,J),J=J1,J2)
      GO TO 10
C
 40   FORMAT (1X,10G12.5)
      END
```

```
      SUBROUTINE PREVAL (VAL,EL,ND,N,ID,SMO)
      DIMENSION VAL(ND) , EL(ND,ND)
      COMPLEX VAL
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C     PRINT EIGENVALUES AND EIGENVECTORS, THE EIGENVECTORS WILL        C
C     BE NORMALLIZED                                                   C
C     INPUT....                                                        C
C        VAL=EIGENVALUES (FROM QR)                                     C
C        EL=EIGENVECTOR MATRIX (FROM EIGVC)                            C
C        ND=DECLARED DIMENSION OF VAL,EL                               C
C        N=DIMENSION OF VAL,EL                                         C
C        SMO=PRECISION OF EIGENVALUES COMPUTATION                      C
C        ID=0 ,PRINT EIGENVALUES ONLY                                  C
C        ID=1 ,BOTH EIGENVALUES AND EIGENVECTORS PRINTED               C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      DO 110 I=1,N
         IF (ID.EQ.1) GO TO 10
         PRINT 120, I
         GO TO 20
 10      PRINT 130, I,I
 20      RE = REAL(VAL(I))
         AIM = AIMAG(VAL(I))
         PRINT 140, RE,AIM
         IF (ID.EQ.0) GO TO 110
         IF (ABS(AIM) .LE.SMO) GO TO 70
         K = I-1
         KP1 = I
         SIGN = 1.
         IF (AIM.LT.0.) GO TO 50
         K = I
         KP1 = I+1
         SIGN = -1.
         SUM = 0.
         XMAX = 0.
         DO 30 J=1,N
            X = EL(J,K)*EL(J,K)+EL(J,KP1)*EL(J,KP1)
            IF (X.LE.XMAX) GO TO 30
            XMAX = X
            JMAX = J
 30      SUM = SUM+X
         SUM = SQRT(SUM*XMAX)
         X1 = EL(JMAX,K)
         X2 = EL(JMAX,KP1)
         DO 40 J=1,N
            XX = (EL(J,K)*X1+EL(J,KP1)*X2)/SUM
            EL(J,KP1) = (-EL(J,K)*X2+EL(J,KP1)*X1)/SUM
 40      EL(J,K) = XX
 50      DO 60 J=1,N
            XW = EL(J,KP1)*SIGN
 60      PRINT 150, EL(J,K),XW
         GO TO 110
 70      SUM = 0.
```

```
      DO 30 J=1,N
30    SUM = SUM+EL(J,I)*EL(J,I)
      SUM = SQRT(SUM)
      DO 90 J=1,N
90    EL(J,I) = EL(J,I)/SUM
      DO 100 J=1,N
100   PRINT 160, EL(J,I)
110   CONTINUE
      RETURN
C
120   FORMAT (//,3X,12H EIGENVALUE(,I2,2H ))
130   FORMAT (//,3X,12H EIGENVALUE(,I2,2H ),15X,13H EIGENVECTOR(,I2,2H )
     1)
140   FORMAT (/,3X,G12.5,3H +J,G12.5)
150   FORMAT (30X,G12.5,3H +J,G12.5)
160   FORMAT (30X,G12.5)
      END
```

```
      SUBROUTINE CONTRL (A,B,C,D,Q,R,S,GAIN,H,P,VAL,N,M,L,ND,ND2,NDM,NDL
     1,W,ID)
      DIMENSION A(ND,ND) , B(ND,NDL) , C(NDM,ND) , D(NDM,NDL) , Q(ND,ND) , R
     1(ND,ND) , S(ND,ND) , GAIN(NDL,ND) , H(ND2,ND2) , P(ND2,ND2) , VAL(ND2) ,
     2 W(ND2)
      COMPLEX VAL
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C    SOLVE THE REGULATOR PROBLEM                                        C
C       DX/DT=A*X+B*U                                                   C
C       Y=C*X+D*U                                                       C
C    WITH QUADRATIC PERFORMANCE CRITERION                               C
C       J=(1/2)*(INTG(YT*Q*Y+UT*R*U))                                   C
C    INPUT...                                                           C
C    1...                                                               C
C      A,B,C,D,Q,R, ARE MATRICES DEFINED ABOVE WITH DIMENSIONS          C
C      N*N,N*L,M*N,M*L,M*M,L*L RESPECTIVELY                             C
C    2..                                                                C
C      N,M,L,DIMENSION OF MATRIX                                        C
C      ND,NDM,NDL ARE DECLARED DIMENSIONS ASSOCIATED WITH N,M,L         C
C    3... ND2=ND*2 DECLARED DIMENSION OF MATRICES H,P,VAL,W             C
C      ID=0 ,D MATRIX IS EQ. ZERO , OTHERWISE =1                        C
C    OUTPUT...                                                          C
C      S=N*N MATRIX , SOLUTION OF RICCATI EQUATION                      C
C      GAIN=L*N MATRIX, CONTROL GAIN ,U=GAIN*X                          C
C      VAL= FIRST N ELEMENTS ARE EIGENVALUES( CLOSED LOOP)              C
C           LAST N ELEMENTS ARE WORKING SPACE                           C
C    SOME NOTES....                                                     C
C      A,B,C,D, WILL BE SAVED                                           C
C      Q,R, WILL BE DESTROYED                                           C
C      H,P,W ARE WORKING SPACE                                          C
C    SUBROUTINES REQD.-HSOLVE,RICCAT,QR,HESS,SHIFT2,TRIA,PERMUT         C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      NP = N
      IF (ID.EQ.0) GO TO 60
C     P11=Q*D
      DO 10 I=1,M
         DO 10 J=1,L
            P(I,J) = 0.
            DO 10 K=1,M
 10   P(I,J) = P(I,J)+Q(I,K)*D(K,J)
C     R=R+DT*Q*D
      DO 30 I=1,L
         DO 30 J=1,L
            SUM = 0.
            DO 20 K=1,M
 20         SUM = SUM+D(K,I)*P(K,J)
 30   R(I,J) = SUM+R(I,J)
C     GAIN=P12=DT*Q*C
      DO 50 I=1,L
         DO 50 J=1,N
            SUM = 0.
            DO 40 K=1,M
```

```
 40         SUM = SUM+P(K,I)*C(K,J)
            GAIN(I,J) = SUM
 50   P(I,J+N) = SUM
      NP = N+N
C     P11=BT
 60   DO 80 I=1,L
         DO 70 J=1,N
 70      P(I,J) = B(J,I)
         DO 80 J=1,L
 80   H(I,J) = R(I,J)
C     P11=INV(R)*BT AND P12=INV(R)*DT*Q*C
      CALL HSOLVE (H,W,P,L,NP,ND2,0,IE)
C     H=Q*C
      DO 90 I=1,M
         DO 90 J=1,N
            H(I,J) = 0.
            DO 90 K=1,M
 90   H(I,J) = H(I,J)+Q(I,K)*C(K,J)
C     H21=CT*Q*D*INV(R)*DT*Q*C-CT*Q*C
      DO 130 I=1,N
         NPI = N+I
         DO 130 J=1,N
            NPJ = N+J
            SUM = 0.
            IF (ID.EQ.0) GO TO 110
            DO 100 K=1,L
 100        SUM = SUM+GAIN(K,I)*P(K,NPJ)
 110        DO 120 K=1,M
 120        SUM = SUM-C(K,I)*H(K,J)
 130  H(NPI,J) = SUM
C     R=-INV(R)*BT AND GAIN=-INV(R)*DT*Q*C
      DO 140 I=1,L
         DO 140 J=1,N
            IF (ID.EQ.0) GO TO 140
            GAIN(I,J) = P(I,J+N)
 140  R(I,J) = P(I,J)
C     H11=A-B*INV(R)*DT*Q*C AND H22=-H11T AND H12=-B*INV(R)*BT
      DO 180 I=1,N
         NPI = N+I
         DO 180 J=1,N
            NPJ = N+J
            SUM = 0.
            IF (ID.EQ.0) GO TO 160
            DO 150 K=1,L
 150        SUM = SUM+B(I,K)*GAIN(K,J)
 160        H(I,J) = A(I,J)-SUM
            H(NPJ,NPI) = -H(I,J)
            SUM = 0.
            DO 170 K=1,L
 170        SUM = SUM-B(I,K)*R(K,J)
 180  H(I,NPJ) = SUM
      CALL RICCAT (H,P,S,VAL,1,N,ND,ND2,W,Q)
```

```
C       GAIN=INV(R)*DT*Q*C*+INV(R)*BT*S
        DO 200 I=1,L
           DO 200 J=1,N
              SUM = 0.
              DO 190 K=1,N
 190          SUM = SUM-R(I,K)*S(K,J)
              IF (ID.NE.0) SUM = SUM-GAIN(I,J)
 200    GAIN(I,J) = SUM
        RETURN
        END
```

```
      SUBROUTINE FILTER (A,C,GAMA,THETA,Q,R,S,GAIN,H,P,VAL,N,M,L,ND,ND2,
     1NDM,NDL,W,ID)
      DIMENSION A(ND,ND) , C(NDM,ND) , GAMA(ND,NDL) , THETA(NDM,NDL) , Q(ND,
     1ND) , R(ND,ND) , H(ND2,ND2) , P(ND2,ND2) , W(ND2) , S(ND,ND) , VAL(ND2) ,
     2 GAIN(ND,NDM)
      COMPLEX VAL
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C    SOLVE FOR STATIONARY KALMAN FILTER OF THE SYSTEM                     C
C      DX/DT=A*X+B*U+GAMA*W                                               C
C      Z=C*X+D*U+V+THETA*W                                                C
C      Q AND R ARE POWER SPECTRAL DENSITIES OF W AND V                    C
C    INPUT...                                                             C
C      A,C,GAMA,THETA,Q,R AS DEFINED ABOVE ARE N*N,M*N,N*L,M*L,L*L,M*M    C
C      RESPECTIVELY.                                                      C
C      N,L,M,ND,NDL,NDM ARE DIMENSION AND DECLARED DIMENSION RESPECTIVELY. C
C      ND2=ND*2 DECLARED DIMENSION OF H,P,VAL,W                           C
C      ID=0 IF MATRIX THETA=0.  ,=1 OTHERWISE                             C
C    OUTPUT...                                                            C
C      S,VAL,GAIN ARE SOLUTION OF RICCATI EQUATION, CLOSED LOOP           C
C      EIGENVALUES, AND FILTER GAIN RESPECTIVELY.                         C
C    NOTICE.. Q,R WILL BE DESTROYED.                                      C
C      H,P,W ARE WORKING SPACE                                            C
C    SUBROUTINES REQD.- HSOLVE,RICCAT,QR,HESS,SHIFT2,TRIA,PERMUT          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IF (ID.EQ.0) GO TO 20
      DO 10 I=1,M
         DO 10 J=1,L
            S(I,J) = 0.
            DO 10 K=1,L
 10   S(I,J) = S(I,J)+THETA(I,K)*Q(K,J)
 20   DO 40 I=1,M
         DO 40 J=1,M
            SUM = 0.
            IF (ID.EQ.0) GO TO 40
            DO 30 K=1,L
 30         SUM = SUM+S(I,K)*THETA(J,K)
 40   H(I,J) = R(I,J)+SUM
      DO 70 I=1,M
         DO 50 J=1,N
 50      P(I,J) = C(I,J)
         IF (ID.EQ.0) GO TO 70
         DO 60 J=1,L
 60      P(I,J+N) = S(I,J)
 70   CONTINUE
      NL = N+L
      IF (ID.EQ.0) NL = N
      CALL HSOLVE (H,W,P,M,NL,ND2,0,IE)
      IF (ID.EQ.0) GO TO 100
      DO 90 I=1,L
         DO 90 J=1,L
            SUM = 0.
            DO 80 K=1,M
```

```
80        SUM = SUM+S(K,I)*P(K,J+N)
90    Q(I,J) = Q(I,J)-SUM
100   DO 110 I=1,N
         DO 110 J=1,N
            S(I,J) = 0.
            DO 110 K=1,M
110   S(I,J) = S(I,J)+C(K,I)*P(K,J)
      IF (ID.EQ.0) GO TO 130
      DO 120 I=1,N
         DO 120 J=1,M
            R(I,J) = 0.
            DO 120 K=1,L
120   R(I,J) = R(I,J)+GAMA(I,K)*P(J,K+N)
130   DO 140 I=1,N
         DO 140 J=1,M
140   GAIN(I,J) = -P(J,I)
      DO 170 I=1,N
         DO 170 J=1,N
            SUM = 0.
            IF (ID.EQ.0) GO TO 160
            DO 150 K=1,M
150         SUM = SUM+R(I,K)*C(K,J)
160         H(I,J) = A(I,J)-SUM
            H(J+N,I+N) = -H(I,J)
170   H(I+N,J) = S(I,J)
      DO 180 I=1,L
         DO 180 J=1,N
            P(I,J) = 0.
            DO 180 K=1,L
180   P(I,J) = P(I,J)+Q(I,K)*GAMA(J,K)
      DO 200 I=1,N
         DO 200 J=1,N
            SUM = 0.
            DO 190 K=1,L
190         SUM = SUM+GAMA(I,K)*P(K,J)
200   H(I,J+N) = SUM
      CALL RICCAT (H,P,S,VAL,-1,N,ND,ND2,W,Q)
      DO 230 J=1,M
         DO 210 I=1,N
            W(I) = 0.
            DO 210 K=1,N
210      W(I) = W(I)+S(I,K)*GAIN(K,J)
         DO 220 I=1,N
220      GAIN(I,J) = W(I)
230   CONTINUE
      IF (ID.EQ.0) GO TO 250
      DO 240 I=1,N
         DO 240 J=1,M
240   GAIN(I,J) = GAIN(I,J)-R(I,J)
250   RETURN
      END
```

```
      SUBROUTINE RICCAT (H,P,S,VAL,K1,N,ND,ND2,W,QQ)
      DIMENSION H(ND2,ND2), P(ND2,ND2), S(ND,ND), VAL(ND2), W(ND2), QQ
     1(ND,ND)
      COMPLEX VAL,VAT
      DATA SMALL/1.E-14/
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C   REGULATOR PROBLEM--- S*A+AT*S-Q+S*R*S=0.                            C
C   FILTER PROBLEM--- A*S+S*AT+R-S*Q*S=0.                               C
C   INPUT...                                                            C
C      H=2N*2N MATRIX, DEFINED AS FOLLOWS...                            C
C      H11=A                                                            C
C      H12=R                                                            C
C      H21=Q                                                            C
C      H22=-AT                                                          C
C      A,Q,R AS IN THE ABOVE RICCATI EQUATION FOR REGULATOR AND FILTER  C
C      K1=CONTROL PARAMETER,                                            C
C         K1=1 FOR REGULATOR(ISOLATE RIGHT HALF PLANE EIGENVALUES)      C
C         K1=-1 FOR FILTER(ISOLATE LEFT HALF PLANE EIGENVALUES)         C
C      N=DIMENSION OF S                                                 C
C      ND=DECLARED DIMENSION OF S AND QQ                                C
C      ND2=DECLARED DIMENSION OF H AND P(USUALLY 2*ND)                  C
C      W IS WORKING SPACE, 2N*1                                         C
C      QQ IS N*N WORKING SPACE                                          C
C   OUTPUT....                                                          C
C      S=SOLUTION OF RICCATI EQUATION,N*N MATRIX                        C
C      H=QUASI-UPPER TRIANGULAR MATRIX,2N*2N                            C
C      P=ORTHOGONAL SIMILARITY TRANSFORMATION MATRIX, 2N*2N             C
C      VAL=N LEFT HALF PLANE EIGENVALUES AND N LOCATIONS OF WORKING SPACE C
C   SUBROUTINES REQD.- QR,HSOLVE,HESS,SHIFT2,TRIA,PERMUT                C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SMO = SMALL
      N2 = N+N
      DO 10 I=1,N2
         DO 10 J=1,N2
 10   P(I,J) = H(I,J)
C     CALCULATE EIGENVALUES
      CALL QR (P,H,N2,ND2,VAL,N2,SMO,0,0,0)
C     REORDER EIGENVALUES
      DO 20 I=1,N2
         RE = REAL(VAL(I))*K1
         IF (RE.LT.0.) VAL(I) = CMPLX(0.,0.)
 20   CONTINUE
      DO 40 J=1,N
         IPP = N2+1-J
         VAT = VAL(IPP)
         AMX = ABS(REAL(VAT))
         IPP1 = IPP-1
         IPP2 = 0
         DO 30 I=1,IPP1
            AMI = ABS(REAL(VAL(I)))
            IF (AMI.LE.AMX) GO TO 30
            IPP2 = I
```

```
            AMX = AMI
 30      CONTINUE
         IF (IPP2.EQ.0) GO TO 40
         VAL(IPP) = VAL(IPP2)
         VAL(IPP2) = VAT
 40   CONTINUE
      DO 50 I=1,N
 50   VAL(I) = VAL(I+N)
      RE = REAL(VAL(1))
      AIM = AIMAG(VAL(1))
      S1 = SQRT(SMO)
      S2 = SQRT(S1)
      DISO = SQRT(RE*RE+AIM*AIM)
      IF (DISO.GT.S2) GO TO 70
      IF (ABS(AIM) .GT.SMO) GO TO 60
      RE = REAL(VAL(2))
      IF (ABS(RE) .LE.S2) GO TO 60
      IF (DISO.GT.S1) GO TO 80
 60   PRINT 150
      GO TO 80
 70   IF (ABS(RE) .GT.S1) GO TO 80
      PRINT 160
 80   SMO = SMALL
C     CALCULATE QUASI-TRIANGULAR DECOMPOSITION
      CALL QR (H,P,N2,ND2,VAL,N,SMO,0,1,1)
      IF (K1.EQ.-1) GO TO 100
      DO 90 J=1,N
         JJ = J+N
         DO 90 I=1,N
            QQ(I,J) = P(I,J)
 90   S(I,J) = P(I,JJ)
      GO TO 120
 100  DO 110 J=1,N
         JJ = J+N
         DO 110 I=1,N
            S(I,J) = P(I,J)
 110  QQ(I,J) = P(I,JJ)
C
C     SOLVE FOR S, THE RICCATI EQN. SOLUTION
C     K1=1 , P11*S=P12
C     K1=-1 , P12*S=P11
C
 120  CALL HSOLVE (QQ,W,S,N,N,ND,0,IE)
      IF (K1.EQ.-1) GO TO 140
      DO 130 I=1,N
 130  VAL(I) = -VAL(I)
 140  RETURN
C
 150  FORMAT (//,47H **WARNING.HAMILTONIAN MATRIX NEARLY SINGULAR**,//)
 160  FORMAT (//,61H **WARNING.HAMILTONIAN HAS EIGENVALUE AT IMAGINARY
     1   AXIS**,//)
      END
```

```
      SUBROUTINE QR (H,P,N,ND,VAL,MV,SNO,ITR,I1,K1)
      COMPLEX VAL,V
      DIMENSION H(ND,ND) , P(ND,ND) , VAL(ND) , V(2)
      DATA SMALL/1.E-14/
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C    QR REDUCTION TO QUASI-UPPER TRIANGULAR FORM                               C
C    INPUT.....                                                                C
C       H= MATRIX TO BE REDUCED                                                C
C       ITR=DIMENSION OF PREVIOUSLY TRIANGULARIZED CORNER OF H                 C
C       I1= 0, NO A PRIORI EIGENVALUE KNOWLEDGE                                C
C       I1= 1, SHIFT ON KNOWN EIGENVALUES                                      C
C       K1= 0, NO CUMMULATED P CALCULATED                                      C
C       K1= 1, CUMMULATED P CALCULATED                                         C
C       MV=NUMBER OF EIGENVALUES TO BE ISOLATED                                C
C       VAL=A PRIORI EIGENVALUES ( IF KNOWN )                                  C
C       SNO=REQUIRED PRECISION AS A FRACTION OF THE MATRIX NORM                C
C       N=DIMENSION OF MATRIX H                                                C
C       ND=NUMBER OF ROWS IN MAIN PROGRAM DIMENSION OF H                       C
C    OUTPUT..............                                                      C
C       H=QUASI-UPPER TRIANGULAR MATRIX                                        C
C       P=ORTHOGONAL MATRIX ( IF K1= 1 )                                       C
C       VAL=CALCULATED EIGENVALUES                                             C
C       SNO=SNO*NORM-USED FOR DISTINGUISHING ZERO SUBDIAGONAL ELEMENTS         C
C           IN RESULTING QUASI-TRIANGULAR MATRIX                               C
C    SUBROUTINES REQD.- HESS,SHIFT2,TRIA,PERMUT                                C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IPERVP = 0
      IPERV = 0
      IF (ITR.NE.0) GO TO 60
      IF (K1.EQ.0) GO TO 30
C     INITIALLIZE P AS IDENTITY
      DO 20 I=1,N
         DO 10 J=1,N
 10      P(I,J) = 0.
 20   P(I,I) = 1.
 30   OMN = 0.
C     CALCULATE THE MAXIMUM NORM OMN
      DO 50 I=1,N
         AA = 0.
         DO 40 J=1,N
 40      AA = AA+ABS(H(I,J))
         IF (AA.GT.OMN) OMN = AA
 50   CONTINUE
      SMO = SMALL*OMN
      SNO = SNO*OMN
 60   N1 = 1
      N2 = N-ITR
      M = MV
      CALL HESS (H,P,N,ND,N-1,N1,N2,SMO,K1)
C     START MAJOR LOOP
 70   IJ = 0
```

```
      IF (M.LE.0) RETURN
      N2M1 = N2-1
      N2M2 = N2-2
 80   N2MN1 = N2-N1
      IF (N2MN1.GE.0) GO TO 90
      N1 = 1
      N2MN1 = N2-N1
 90   IF (N2MN1.EQ.1) GO TO 150
      IF (N2MN1.EQ.0) GO TO 160
C     START QR ITTERATION
      DO 140 IT=1,20
         IF (I1.EQ.0) GO TO 100
         IF (IT.GT.10) GO TO 100
         IF (IJ.NE.0) GO TO 110
         RE = REAL(VAL(M))
         AIM = AIMAG(VAL(M))
         C1 = (RE+RE)
         C2 = RE*RE+AIM*AIM
         IJ = 1
         GO TO 110
 100     C1 = H(N2M1,N2M1)+H(N2,N2)
         C2 = H(N2,N2)*H(N2M1,N2M1)-H(N2M1,N2)*H(N2,N2M1)
         IF (IT.NE.10) GO TO 110
C     PERFORM EXCEPTIONAL SHIFT
         C1 = ABS(H(N2,N2M1))+ABS(H(N2M1,N2M2))
         C2 = C1*C1
         C1 = 1.5*C1
 110     CALL SHIFT2 (H,P,N,ND,N1,N2,C1,C2,SMO,K1)
         CALL HESS (H,P,N,ND,3,N1,N2,SMO,K1)
C     TEST FOR CONVERGENCE
         S1 = SNO
         IF (IT.EQ.20) S1 = SQRT(SNO)
         DO 130 I=1,N2MN1
            LH = N2-I+1
            IF (ABS(H(LH,LH-1)).GE.S1) GO TO 130
            IF (IPERVP.GT.2) GO TO 120
            IF (I.GT.2.AND.I1.EQ.1) GO TO 300
C     ZERO SUBDIAGONAL ELEMENT
 120        N1 = LH
            GO TO 80
 130     CONTINUE
 140  CONTINUE
      PRINT 330
      STOP
C     QR HAS CONVERGED - CALCULATE EIGENVALUES
 150  CALL TRIA (H,P,N,ND,N2,V,SNO,K1)
      NV = 2
      GO TO 170
 160  V(1) = CMPLX(H(N2,N2),0.)
      NV = 1
 170  IF (I1.NE.0) GO TO 190
      DO 180 K=1,NV
```

```
          VAL(M) = V(NV-K+1)
          M = M-1
 180  N2 = N2-1
      GO TO 70
C     FIND NEAREST GIVEN E-VALUE
 190  DO 260 K=1,NV
         KK = NV-K+1
         II = 1
         DMIN = CABS(VAL(1)-V(KK))
         IF (M.LT.2) GO TO 210
         DO 200 I=2,M
            CC = CABS(VAL(I)-V(KK))
            IF (CC.GT.DMIN) GO TO 200
            DMIN = CC
            II = I
 200     CONTINUE
 210     IF (IPERV.GE.1.AND.N2.EQ.2) GO TO 220
         IF (IPERVP.LE.2) GO TO 230
 220     PRINT 340, M
         GO TO 240
 230     IF (DMIN.GT.(GNO**.25)) GO TO 270
         IF (REAL(VAL(II))*REAL(V(KK)).LT.0.) GO TO 270
 240     IF (M.EQ.II) GO TO 250
         VAL(II) = VAL(M)
 250     VAL(M) = V(KK)
         IF (M.LE.1) RETURN
         M = M-1
 260  N2 = N2-1
      IPERV = 0
      IPERVP = 0
      GO TO 70
C     REJECT EIGENVALUE
 270  IF (N2.LE.1) GO TO 320
      IF (ABS(H(N2,N2-1)).GT.S1) GO TO 290
      IF(N2.EQ.2) GO TO 280
      IF(ABS(H(N2-1,N2-2)).LE.S1) GO TO 280
      GO TO 290
 280  IF (IPERV.GE.1) GO TO 290
      CALL PERMUT (H,P,N,ND,N1,N2,S40,K1,1)
      IPERV = IPERV+1
      GO TO 160
 290  IF (N2.EQ.2) GO TO 310
 300  CALL PERMUT (H,P,N,ND,1,N2,S40,K1,0)
      CALL HESS (H,P,N,ND,N2-1,1,N2,S40,K1)
 310  IPERVP = IPERVP+1
      N1 = 1
      GO TO 70
 320  VAL(M) = V(1)
      PRINT 340, M
      RETURN
C
 330  FORMAT (25H ERROR- QR NOT CONVERGING)
```

```
340  FORMAT (/,20H COMPUTED EIGENVALUE,I3,27H DOES NOT MATCH INPUT VALU
    1E)
     END
```

```
      SUBROUTINE HESS (A,P,N,ND,NZ,N1,N2,SMALL,K1)
      DIMENSION A(ND,ND) , P(ND,ND)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C  PERFORMES ORTHOGONAL SIMILARITY REDUCTION TO UPPER HESSENBERG FORM.     C
C  MATRIX IS FULL N*N OR HAS ZERO LOWER-LEFT TRIANGLE.                     C
C  PARAMETERS- A- INPUT- N*N MATRIX TO BE REDUCED.                         C
C                 OUTPUT- REDUCED MATRIX                                   C
C              P- ACCUMULATED ORTHOGONAL TRANSFORMATION MATRIX.            C
C              N- MATRIX SIZE                                              C
C              ND- NUMBER OF ROWS IN MAIN PROGRAM DIMENSION FOR A AND Q    C
C              NZ- NUMBER OF NON-ZERO ELEMENTS BELOW THE DIAGONAL IN A.    C
C              N1,N2- ROW INDICIES OF CURRENT ISOLATED DIAGONAL BLOCK      C
C              SMALL- A SMALL NUMBER = MACH. PREC. * NORM(A)               C
C              K1=0 NO ACCUMULATED P MATRIX                                C
C              K1=1 ACCUMULATE ORTHOGONAL TRANSFORMATIONS                  C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      N2M1 = N2-1
      N1P1 = N1+1
      DO 120 K=N1P1,N2M1
         KM1 = K-1
         NN = KM1+NZ
         IF (NN.GT.N2) NN = N2
         S = 0.
         DO 10 I=K,NN
 10      S = S+A(I,KM1)*A(I,KM1)
         S = SQRT(S)
         IF (S.LT.SMALL) GO TO 100
         IF (A(K,KM1).GT.0.) S = -S
         A(K,KM1) = A(K,KM1)-S
         C = SQRT(-S*A(K,KM1))
         DO 20 I=K,NN
 20      A(I,KM1) = A(I,KM1)/C
         DO 40 J=K,N
            C = 0.
            DO 30 I=K,NN
 30         C = C+A(I,J)*A(I,KM1)
            DO 40 I=K,NN
 40      A(I,J) = A(I,J)-C*A(I,KM1)
         DO 60 I=1,N2
            C = 0.
            DO 50 J=K,NN
 50         C = C+A(I,J)*A(J,KM1)
            DO 60 J=K,NN
 60      A(I,J) = A(I,J)-C*A(J,KM1)
         IF (K1.EQ.0) GO TO 90
         DO 80 J=1,N
            C = 0.
            DO 70 I=K,NN
 70         C = C+P(I,J)*A(I,KM1)
            DO 80 I=K,NN
 80      P(I,J) = P(I,J)-C*A(I,KM1)
 90      A(K,KM1) = S
```

```
100      KP1 = K+1
         DO 110 I=KP1,NN
110      A(I,KM1) = 0.
120   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SHIFT2 (A,P,N,ND,N1,N2,C1,C2,SMALL,K1)
      DIMENSION A(ND,ND), P(ND,ND)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C  PERFORMS DOUBLE SHIFT WITH ORIGINS S1,S2 WHICH SATISFY C1=S1+S2     C
C  AND C2=S1*S2, FOLLOWED BY THE FIRST SIMILARITY TRANSFORMATION       C
C  IN QR FACTORIZATION. MATRIX A IS UPPER HESSENBERG ON INPUT.         C
C  PARAMETERS- A- INPUT-N*N UPPER HESSENBERG MATRIX                    C
C                 OUTPUT- TRANSFORMED MATRIX WITH 3 ELEMENTS BELOW DIAG C
C              P- N*N ACCUMULATED ORTHOGONAL TRANSFORMATION MATRIX     C
C              N- MATRIX SIZE                                          C
C              ND- NUMBER OF ROWS IN MAIN PROGRAM DIMENSION FOR A AND Q C
C              N1,N2- ROW INDICIES OF ISOLATED DIAGONAL BLOCK          C
C              C1, C2- CONSTANTS FOR DOUBLE ORIGIN SHIFT               C
C              SMALL- A SMALL NUMBER = MACH. PREC. * NORM(A)           C
C              K1=0 NO ACCUMULATED P MATRIX                            C
C              K1=1 ACCUMULATE ORTHOGONAL TRANSFORMATIONS              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      N1P1 = N1+1
      N1P2 = N1+2
      X1 = A(N1,N1)-C1
      W1 = A(N1,N1)*X1+C2+A(N1,N1P1)*A(N1P1,N1)
      W2 = (X1+A(N1P1,N1P1))*A(N1P1,N1)
      W3 = A(N1P2,N1P1)*A(N1P1,N1)
      S = SQRT(W1*W1+W2*W2+W3*W3)
      IF (S.LT.SMALL) RETURN
      IF (W1.GT.0.) S = -S
      W1 = W1-S
      C = SQRT(-S*W1)
      W1 = W1/C
      W2 = W2/C
      W3 = W3/C
      DO 10 J=N1,N
         S = A(N1,J)*W1+A(N1P1,J)*W2+A(N1P2,J)*W3
         A(N1,J) = A(N1,J)-S*W1
         A(N1P1,J) = A(N1P1,J)-S*W2
 10   A(N1P2,J) = A(N1P2,J)-S*W3
      DO 20 I=1,N2
         S = A(I,N1)*W1+A(I,N1P1)*W2+A(I,N1P2)*W3
         A(I,N1) = A(I,N1)-S*W1
         A(I,N1P1) = A(I,N1P1)-S*W2
 20   A(I,N1P2) = A(I,N1P2)-S*W3
      IF (K1.EQ.0) RETURN
      DO 30 J=1,N
         S = P(N1,J)*W1+P(N1P1,J)*W2+P(N1P2,J)*W3
         P(N1,J) = P(N1,J)-S*W1
         P(N1P1,J) = P(N1P1,J)-S*W2
 30   P(N1P2,J) = P(N1P2,J)-S*W3
      RETURN
      END
```

```
      SUBROUTINE PERMUT (H,P,N,ND,NX,N2,SMO,K1,IP)
      DIMENSION H(ND,ND) , P(ND,ND)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C   ARBITRARY ORTHOGONAL( HOUSEHOLDER TYPE ) SIMILARITY                 C
C   TRANSFORMATION TO REMOVE AN UNDESIRABLE ZERO ON SUBDIAGONAL         C
C   OR EXCHANGE POSITION OF TWO REAL EIGENVALUES ON 2*2 DIAGONAL BLOCK  C
C   INPUT....                                                           C
C       H=MATRIX TO BE TRANSFORMED                                      C
C       N=DIMENSION OF H,P                                              C
C       ND=DECLARED DIMENSION OF H,P                                    C
C       NX,N2=ROW INDICIES OF THE BLOCK TO BE TRANSFORMED               C
C       K1=0 NO CUMMULATED P CALCULATED                                 C
C       K1=1 CUMMULATED P CALCULATED                                    C
C       IP=1 IF POSITION EXCHANGE PERFORMED OTHERWISE=0                 C
C       SMO=PRECISION CRITERION FROM QR SUBROUTINE                      C
C   OUTPUT.........                                                     C
C       H=TRANSFORMED MATRIX                                            C
C       P IS ORTHOGONAL MATRIX                                          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IF (IP.NE.0) GO TO 100
      X = 2./(N2-NX+1)
      DO 30 I=NX,N
         NM = N2
         IF (I+1.LT.N2) NM = I+1
         SUM = 0.
         DO 10 K=NX,NM
 10      SUM = SUM-H(K,I)
         DO 20 J=NX,N2
 20      H(J,I) = H(J,I)+X*SUM
 30   CONTINUE
      DO 60 I=1,N2
         SUM = 0.
         DO 40 K=NX,N2
 40      SUM = SUM-H(I,K)
         DO 50 J=NX,N2
 50      H(I,J) = H(I,J)+X*SUM
 60   CONTINUE
      IF (K1.EQ.0) RETURN
      DO 90 I=1,N
         SUM = 0.
         DO 70 K=NX,N2
 70      SUM = SUM-P(K,I)
         DO 80 J=NX,N2
 80      P(J,I) = P(J,I)+X*SUM
 90   CONTINUE
      RETURN
 100  N2M1 = N2-1
      N2M2 = N2-2
      N2P1 = N2+1
      A = H(N2M1,N2M1)
      B = H(N2M1,N2)
      C = H(N2,N2)
```

```
      IF (ABS(B).GE.SMO) GO TO 110
      X11 = 0.
      X12 = 1.
      X22 = 0.
      GO TO 120
110   X = A-C
      XX = SQRT(X*X+B*B)
      X22 = B/XX
      X11 = -X22
      X12 = X/XX
120   H(N2M1,N2M1) = C
      H(N2,N2) = A
      H(N2M1,N2) = X11*(X12*A+X22*B)+X12*X22*C
      IF (N2P1.GT.N) GO TO 140
      DO 130 I=N2P1,N
         XA = X11*H(N2M1,I)+X12*H(N2,I)
         H(N2,I) = X12*H(N2M1,I)+X22*H(N2,I)
130   H(N2M1,I) = XA
140   IF (N2M2.LE.0) GO TO 160
      DO 150 I=1,N2M2
         XA = X11*H(I,N2M1)+X12*H(I,N2)
         H(I,N2) = X12*H(I,N2M1)+X22*H(I,N2)
150   H(I,N2M1) = XA
160   IF (K1.EQ.0) RETURN
      DO 170 I=1,N
         XA = X11*P(N2M1,I)+X12*P(N2,I)
         P(N2,I) = X12*P(N2M1,I)+X22*P(N2,I)
170   P(N2M1,I) = XA
      RETURN
      END
```

```
      SUBROUTINE TRIA (A,P,N,NDIM,N2,VAL,SMALL,KI)
      DIMENSION A(NDIM,NDIM) , P(NDIM,NDIM) , VAL(2)
      COMPLEX VAL
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C  COMPUTES EIGENVALUES OF 2*2 DIAGONAL BLOCK . IF EIGENVALUES ARE         C
C  REAL , AN ORTHOGONAL SIMILARITY TRANSFORMATION IS PERFORMED TO          C
C  TRIANGULARIZE DIAGONAL BLOCK.                                           C
C        B=P*A*PT AND SET A=B                                              C
C  INPUT....                                                               C
C        A=QUASI-TRIANGULAR MATRIX TO BE TRANSFORMED                       C
C        P=THE ORTHOGONAL MATRIX FROM PREVIOUS SIMILARITY TRANSFORMATION   C
C        N=DIMENSION OF MATRIX A                                           C
C        NDIM=DECLEARED DIMENSION OF ARRAY A                               C
C        N2=INDEX OF 2*2 BLOCK = ROW AND COLUMN NUMBER OF LOWER            C
C           RIGHT CORNER                                                   C
C        SMALL=NORM*MACHINE PRECISION                                      C
C        KI=0, NO CUMMULATED P CALCULATED                                  C
C        KI=1, CUMMULATED P CALCULATED                                     C
C  OUTPUT....                                                              C
C        A= QUASI-UPPER TRIANGULAR MATRIX B                                C
C        P=CUMMULATED ORTHOGONAL TRANSFORMATION MATRIX                     C
C        VAL=COMPUTED EIGENVALUES IN VAL(1) AND VAL(2)                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      N2M1 = N2-1
      IF (ABS(A(N2,N2M1)).LE.SMALL) GO TO 50
      IF (ABS(A(N2M1,N2)).GT.SMALL) GO TO 10
      P11 = 0.
      P22 = 0.
      P12 = 1.
      P21 = 1.
      GO TO 20
 10   C1 = A(N2M1,N2M1)-A(N2,N2)
      K = C1*C1+4.*A(N2M1,N2)*A(N2,N2M1)
      IF (K.LT.0.) GO TO 70
      CX = (C1+SQRT(K))*.5
      CC = SQRT(CX*CX+A(N2,N2M1)*A(N2,N2M1))
      P11 = CX/CC
      P21 = A(N2,N2M1)/CC
      P12 = ABS(P21)
      P22 = P11
      IF (P21.GT.0.) P22 = -P22
 20   DO 30 I=1,N2
         AN2M1 = A(I,N2M1)*P11+A(I,N2)*P21
         AN2 = A(I,N2M1)*P12+A(I,N2)*P22
         A(I,N2M1) = AN2M1
 30   A(I,N2) = AN2
      DO 40 I=N2M1,N
         AN2M1 = A(N2M1,I)*P11+A(N2,I)*P21
         AN2 = A(N2M1,I)*P12+A(N2,I)*P22
         A(N2M1,I) = AN2M1
 40   A(N2,I) = AN2
      IF (KI.EQ.0) GO TO 50
```

```
      DO 50 I=1,N
         PN2M1 = P(N2M1,I)*P11+P(N2,I)*P21
         PN2 = P(N2M1,I)*P12+P(N2,I)*P22
         P(N2M1,I) = PN2M1
   50 P(N2,I) = PN2
   60 VAL(2) = CMPLX(A(N2,N2),0.)
      VAL(1) = CMPLX(A(N2M1,N2M1),0.)
      GO TO 80
C
C     CALCULATE THE COMPLEX EIGENVALUES
C
   70 RE = (A(N2M1,N2M1)+A(N2,N2))/2.
      AIM = SQRT(-X)/2.
      VAL(1) = CMPLX(RE,AIM)
      VAL(2) = CMPLX(RE,-AIM)
   80 RETURN
      END
```

```
      SUBROUTINE HSOLVE (A,W,B,N,M,NDIM,ID,IE)
      DIMENSION A(NDIM,NDIM), W(NDIM), B(NDIM,M)
      DATA SMALL/1.E-14/
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C    SOLVE THE LINEAR EQUATION A*X=B BY HOUSEHOLDER METHOD              C
C    INPUT...                                                           C
C       A=THE MATRIX OF N*N ORDER OF A*X=B                              C
C       B=MATRIX OF A*X=B  ,N*M                                         C
C       W=WORKING SPACE                                                 C
C       N=DIM.OF A                                                      C
C       NDIM= DECLARED ROW DIMENSION OF A,B, AND W                      C
C       ID= INDICATOR, ID=0 - TRIANGULARIZE A AND SOLVE,                C
C                      ID=1 - BYPASS TRIANGULARIZATION (PREVIOUS CALL REQD.) C
C    OUTPUT....                                                         C
C       A=Q*A=TRIANGULARIZED MATRIX                                     C
C       B=THE SOLUTION X                                                C
C       IE= ERROR INDICATOR, IE=0 NO ERROR                              C
C                            IE=1 NONUNIQUE SOLUTION (NO ERROR)         C
C                            IE=2 UNDEFINED SOLUTION                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IE = 0
      IF (ID.EQ.1) GO TO 30
      OMN = 0.
      DO 20 I=1,N
         AA = 0.
         DO 10 J=1,N
 10      AA = AA+ABS(A(I,J))
         IF (AA.LT.OMN) GO TO 20
         OMN = AA
 20   CONTINUE
      SMO = SMALL*OMN
      IF (N.EQ.1) GO TO 30
C     TRIANGULARIZE A
      NM1 = N-1
      DO 70 K=1,NM1
         R = 0.
         DO 30 I=K,N
 30      R = R+A(I,K)*A(I,K)
         R = SQRT(R)
         IF (A(K,K).LT.0.) R = -R
         W(K) = A(K,K)+R
         C = SQRT(R*W(K))
         A(K,K) = -R
         IF (ABS(C).LT.SMO) GO TO 70
         KP1 = K+1
         W(K) = W(K)/C
         DO 40 I=KP1,N
 40      A(I,K) = A(I,K)/C
         DO 60 J=KP1,N
            RR = W(K)*A(K,J)
            DO 50 L=KP1,N
 50         RR = RR+A(L,K)*A(L,J)
```

```
            A(K,J) = A(K,J)-RR*W(K)
            DO 60 I=KP1,N
 60      A(I,J) = A(I,J)-RR*A(I,K)
 70   CONTINUE
C     TRANSFORM B AND BACKSOLVE
 80   DO 180 IJ=1,M
         IF (N.EQ.1) GO TO 120
         DO 110 K=1,NM1
            RR = W(K)*B(K,IJ)
            KP1 = K+1
            DO 90 J=KP1,N
 90         RR = RR+A(J,K)*B(J,IJ)
            B(K,IJ) = B(K,IJ)-RR*W(K)
            DO 100 I=KP1,N
 100        B(I,IJ) = B(I,IJ)-RR*A(I,K)
 110     CONTINUE
 120     IF (ABS(A(N,N)).GT.SMO) GO TO 130
         IE = 2
         IF (ABS(B(N,IJ)).GT.SMO) RETURN
         IE = 1
         B(N,IJ) = 1.
         GO TO 140
 130     B(N,IJ) = B(N,IJ)/A(N,N)
 140     IF (N.EQ.1) GO TO 180
         DO 170 I=1,NM1
            J = N-I
            L = N-I+1
            XX = B(J,IJ)
            DO 150 K=L,N
 150        XX = XX-A(J,K)*B(K,IJ)
            IF (ABS(A(J,J)).GT.SMO) GO TO 160
            IE = 2
            IF (ABS(XX).GT.SMO) RETURN
            IE = 1
            B(J,IJ) = 1.
            GO TO 170
 160        B(J,IJ) = XX/A(J,J)
 170     CONTINUE
 180  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE EIGVC (A,VAL,N,NDIM,EL,Q,C,S4O,IE)
      DIMENSION A(NDIM,NDIM), VAL(NDIM), EL(NDIM,NDIM), A1(2,2), A2(2,2)
     1, A3(2,2), C(NDIM), Q(NDIM,NDIM)
      COMPLEX VAL
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C  COMPUTE THE EIGENVECTORS OF A QUASI-UPPER TRIANGULAR MATRIX          C
C  INPUT...                                                             C
C      A=QUASI-UPPER TRIANGULAR MATRIX FROM ROUTINE QR                  C
C      VAL=THE EIGENVALUES OF A COMPUTED IN ROUTINE QR                  C
C      Q=SIMILARITY TRANSFORMATION MATRIX FROM QR                       C
C      N=DIMENSION OF A , EL , Q                                        C
C      NDIM=DECLARED DIMENSION OF A , EL , Q                            C
C      S4O=CONVERGENCE CRITERION FROM QR=MACH.PREC.*NORM A              C
C  OUTPUT...                                                            C
C      EL=THE EIGENVECTOR MATRIX OF A                                   C
C      C=INTERMEDIATE STORAGE VECTOR                                    C
C      IE=0 INDEPENDENT EIGENVECTOR                                     C
C      IE=1 DEFECTIVE EIGENSYSTEM                                       C
C  SUBROUTINES REQD.- LYAPUN,HSOLVE                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      INITIALIZE EL
C
      IE = 0
      DO 10 I=1,N
         DO 10 J=1,N
 10   EL(I,J) = 0.
      K = N
 20   DELTA = REAL(VAL(K))
      W = AIMAG(VAL(K))
      IF (ABS(W).GT.S4O) GO TO 30
      N2 = 1
      A2(1,1) = DELTA
      EL(K,K) = 1.
      IF (K.EQ.1) GO TO 150
      GO TO 40
 30   N2 = 2
      A2(1,1) = DELTA
      A2(2,2) = DELTA
      A2(1,2) = W
      A2(2,1) = -W
      KM1 = K-1
      EL(K,K) = 0.
      EL(K,KM1) = 1.
      AC = A(KM1,KM1)-DELTA
      AA = AC**2+W**2
      EL(KM1,K) = -(A(KM1,K)*W)/AA
      EL(KM1,KM1) = -(A(KM1,K)*AC)/AA
      IF (K.EQ.2) GO TO 150
      ELK = EL(KM1,K)
      ELKM1 = EL(KM1,KM1)
 40   II = K-N2
```

```
      KK = II
      LL = N2
 50   N1 = 1
      IF (KK.EQ.1) GO TO 60
      IF (ABS(A(KK,KK-1)).GT.S40) N1 = 2
 60   NN = KK-N1
      DO 70 I=1,N1
         DO 70 J=1,N1
 70   A1(I,J) = -A(NN+I,NN+J)
      DO 90 I=1,N1
         DO 90 J=1,N2
            SUM = 0.
            DO 80 L=1,LL
 80         SUM = SUM+A(NN+I,KK+L)*EL(KK+L,II+J)
 90   A3(I,J) = SUM
      CALL LYAPUN (A1,N1,A2,N2,A3,IT)
      IF (IT.EQ.2) GO TO 110
      DO 100 I=1,N1
         DO 100 J=1,N2
 100  EL(NN+I,II+J) = A3(I,J)
      GO TO 140
 110  IE = 1
      NNP1 = NN+1
      NNP2 = NN+2
      IIP1 = II+1
      IIP2 = II+2
      EL(NNP1,IIP1) = 1.
      DO 120 I=NNP2,K
 120  EL(I,IIP1) = 0.
      IF (N2.EQ.1) GO TO 140
      EL(NNP1,IIP1) = ELKM1
      EL(NNP1,IIP2) = ELK
      EL(NNP2,IIP1) = 1.
      DO 130 I=NNP2,K
 130  EL(I,IIP2) = 0.
 140  KK = NN
      LL = LL+N1
      IF (KK.GT.0) GO TO 50
      K = II
      IF (K.GT.0) GO TO 20
C
C     EL=QT*EL
C
 150  DO 190 J=1,N
         DO 170 I=1,N
            C(I) = 0.
            DO 160 KI=1,N
 160        C(I) = C(I)+Q(KI,I)*EL(KI,J)
 170     CONTINUE
         DO 180 L=1,N
 180     EL(L,J) = C(L)
 190  CONTINUE
```

```
RETURN
END
```

```
      SUBROUTINE SOLYAP (A,C,N,ND,SMO,IE,P,W)
      DIMENSION A(ND,ND) , C(ND,ND) , AA(2,2) , CC(2,2) , AAT(2,2) , P(ND,ND)
     1, W(ND)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C    SOLVE A*X+X*AT=C                                                    C
C    INPUT...                                                            C
C       A=N*N MATRIX IN QUASI-UPPER TRIANGULAR FORM FROM QR SUBROUTINE   C
C       C=N*N SYMMETRIC MATRIX                                           C
C       P= SIMILARITY TRANSFORMATION MATRIX FROM QR SUBROUTINE           C
C       W=WORKING SPACE                                                  C
C       ND=DECLARED DIMENSION OF A , C , P                               C
C       SMO=CONVERGENCE CRITERION FROM QR=MACH.PREC.*NORM A              C
C    OUTPUT...                                                           C
C       C=X, THE SOLUTION OF LYAPUNOV EQUATION A*X+X*AT=C                C
C       IE=0 NO ERROR                                                    C
C       IE=1 ERROR-SOLUTION NOT COMPUTED                                 C
C    SUBROUTINES REQD.- LYAPUN,HSOLVE                                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      DO 20 I=1,N
         DO 10 J=1,N
            W(J) = 0.
            DO 10 K=1,N
 10      W(J) = W(J)+C(I,K)*P(J,K)
         DO 20 J=1,N
 20   C(I,J) = W(J)
      DO 40 J=1,N
         DO 30 I=1,N
            W(I) = 0.
            DO 30 K=1,N
 30      W(I) = W(I)+P(I,K)*C(K,J)
         DO 40 I=1,N
 40   C(I,J) = W(I)
      IE = 0
      M = N
 50   NA = 2
      IF(M.GT.1) GO TO 51
      NA=1
      GO TO 52
 51   IF(ABS(A(M,M-1)).LE.SMO) NA=1
 52   MM=M-NA
      DO 60 I=1,NA
         DO 60 J=1,NA
            NNI = MM+I
            NNJ = MM+J
            AA(I,J) = A(NNI,NNJ)
            CC(I,J) = C(NNI,NNJ)
 60   AAT(I,J) = A(NNJ,NNI)
      CALL LYAPUN (AA,NA,AAT,NA,CC,IT)
      IF (IT.NE.0) IE = 1
      NN = MM
      NB = NA
 70   DO 80 I=1,NA
```

```
         DO 80 J=1,NB
80    C(NN+I,MM+J) = CC(I,J)
      IF (NN.LE.0) GO TO 130
      DO 100 I=1,NN
         DO 100 J=1,NB
            NNJ = MM+J
            SUM = 0.
            DO 90 K=1,NA
               NNK = NN+K
90          SUM = SUM+A(I,NNK)*C(NNK,NNJ)
100   C(I,NNJ) = C(I,NNJ)-SUM
      NA = 2
      IF(NN.GT.1) GO TO 101
      NA=1
      GO TO 102
101   IF(ABS(A(NN,NN-1)).LE.SMO) NA=1
102   NNA = NN-NA
      DO 110 I=1,NA
         DO 110 J=1,NA
110   AA(I,J) = A(NNA+I,NNA+J)
      DO 120 I=1,NA
         DO 120 J=1,NB
120   CC(I,J) = C(NNA+I,MM+J)
      CALL LYAPUN (AA,NA,AAT,NB,CC,IT)
      IF (IT.NE.0) IE = 1
      NN = NNA
      GO TO 70
130   IF (MM.LE.0) GO TO 160
      M = MM
      DO 150 I=1,M
         DO 150 J=1,M
            SUM = 0.
            DO 140 K=1,NB
               MPK = M+K
140         SUM = SUM+A(I,MPK)*C(J,MPK)+C(I,MPK)*A(J,MPK)
150   C(I,J) = C(I,J)-SUM
      GO TO 50
160   DO 170 I=2,N
         II = I-1
         DO 170 J=1,II
170   C(I,J) = C(J,I)
      DO 190 I=1,N
         DO 180 J=1,N
            W(J) = 0.
            DO 180 K=1,N
180      W(J) = W(J)+C(I,K)*P(K,J)
         DO 190 J=1,N
190   C(I,J) = W(J)
      DO 210 J=1,N
         DO 200 I=1,N
            W(I) = 0.
            DO 200 K=1,N
```

```
200       W(I) = W(I)+P(K,I)*C(K,J)
          DO 210 I=1,N
210   C(I,J) = W(I)
      RETURN
      END
```

```
      SUBROUTINE LYAPUN (A,NA,B,NB,C,IE)
      DIMENSION AA(4,4) , A(2,2) , B(2,2) , C(2,2) , W(3)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C  SOLVE A*X+X*B=C AND SET C=X                                          C
C  INPUT....                                                            C
C    A AND B ARE 1*1 OR 2*2                                             C
C    NA=DIMENSION OF A (1 OR 2)                                         C
C    NB=DIMENSION OF B (1 OR 2)                                         C
C    C=NA*NB MATRIX                                                     C
C  OUTPUT...                                                            C
C    C=SOLUTION X                                                       C
C    IE=0  NON-SINGULAR CASE                                            C
C    IE=1  NON-UNIQUE SOLUTION                                          C
C    IE=2  UNDEFINED SOLUTION                                           C
C  SUBROUTINE REQD.- HSOLVE                                             C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      DO 10 I=1,NA
         DO 10 J=1,NA
            AA(I,J) = A(I,J)
            AA(I+NA,J) = 0.
            AA(I,J+NA) = 0.
 10   AA(I+NA,J+NA) = A(I,J)
      II = 1
      DO 30 I=1,NB
         JJ = 1
         DO 20 J=1,NB
            AA(II,JJ) = AA(II,JJ)+B(J,I)
            IF (NA.GT.1) AA(II+1,JJ+1) = AA(II+1,JJ+1)+B(J,I)
 20      JJ = JJ+NA
 30   II = II+NA
      N = NA*NB
      IF (NB.EQ.2.AND.NA.EQ.1) C(2,1) = C(1,2)
      CALL HSOLVE (AA,W,C,N,1,4,0,IE)
      IF (NB.EQ.2.AND.NA.EQ.1) C(1,2) = C(2,1)
      RETURN
      END
```

# APPENDIX G

## THE SENSOR ELIMINATION PROGRAM LISTING

```
      PROGRAM SENSOR(TAPE5,INPUT,OUTPUT)
      DIMENSION A(30,30),B(30,10),C1(10,30),D1(10,10),C2(10,30),
     1D2(10,10),THETA(10,10),GAIN(10,30),GK(30,10),GAMA(30,10),
     1Q1(30,30),R1(30,30),Q2(30,30),R2(30,30),S(30,30),H(60,60),
     1P(60,60),VAL(60),W(60),EL(30,30),PW(30,30)
      DIMENSION R11(30,30),Q22(30,30),R22(30,30),IELEM(10),
     +C22(10,30),D22(10,10),SITA2(10,10),INDEX(10),XMAX(10)
      DIMENSION SO(30,30),UO(10,10),YO(10,10),WS(30),WU(10),WY(10)
      COMPLEX VAL
      DATA ND/30/,NDD/60/,NDL/10/,NDM/10/,NDP/10/,NDQ/10/
      DATA NA/"A"/,NB/"B"/,NC1/"C1"/,ND1/"D1"/,NC2/"C2"/,ND2/"D2"/,
     1NGA/"GAMA"/,NQ1/"Q1"/,NR1/"R1"/,NQ2/"Q2"/,NR2/"R2"/,
     1NZ/"Z"/,NTH/"THETA"/
      DATA SMALL/1.E-14/
      DATA IUNT/5LINPUT/
C
C    SOLVE SENSOR COMBINATION TESTING
C    INPUT*****
C    IO=0 FIND INFORMATION WITH ALL SENSORS
C    IO=1  SENSORS ELEMINATION
C    IP=0 NO OPTIONAL PRINT
C    IP=1 OPTINAL PRINT
C    IP=2 PRINT EVERYTHING
C    LP=0 REGULAR PRINTER
C    LP=1 TERMINAL PRINTER
C    ICR=1,2,3,OR 4 CRITERION CHOOSING  (RELATIVE ERROR OF RMS X,U,Y,TR)
C    CRITER= THE ACCEPTABLE CRITERIA OF THE RELATIVE ERRORS
C    INPUT NECCERSARY MATRICES
  300 READ(5,*) IO,IP,LP,ICR
      IF(EOF(5).NE.0) STOP
      ID1=0
      ID2=0
      IS1=0
    1 READ(5,100)NN,N1,N2,NT
  100 FORMAT(A6,2I3,2X,A1)
   44 IF(NN.EQ.NA) GO TO 11
      IF(NN.EQ.NB) GO TO 12
      IF(NN.EQ.NC1) GO TO 13
      IF(NN.EQ.ND1) GO TO 14
      IF(NN.EQ.NQ1) GO TO 15
      IF(NN.EQ.NR1) GO TO 16
      IF(NN.EQ.NGA) GO TO 17
      IF(NN.EQ.NQ2) GO TO 18
      IF(NN.EQ.NC2) GO TO 19
      IF(NN.EQ.ND2) GO TO 20
      IF(NN.EQ.NR2) GO TO 21
      IF(NN.EQ.NTH) GO TO 22
      GO TO 23
   11 CALL READM(A,N1,N2,ND,NT,NA,LP)
      N=N1
      GO TO 1
   12 CALL READM(B,N1,N2,ND,NT,NB,LP)
```

```
      N=N1
      L=N2
      GO TO 1
   13 CALL READM(C1,N1,N2,NDM,NT,NC1,LP)
      M=N1
      N=N2
      GO TO 1
   14 CALL READM(D1,N1,N2,NDM,NT,ND1,LP)
      M=N1
      L=N2
      ID1=1
      IF(NT.EQ.NZ) ID1=0
      GO TO 1
   15 CALL READM(Q1,N1,N2,ND,NT,NQ1,LP)
      M=N1
      GO TO 1
   16 CALL READM(R1,N1,N2,ND,NT,NR1,LP)
      L=N1
      GO TO 1
   17 CALL READM(GAMA,N1,N2,ND,NT,NGA,LP)
      N=N1
      NP=N2
      GO TO 1
   18 CALL READM(Q2,N1,N2,ND,NT,NQ2,LP)
      NP=N1
      IF(IO.EQ.0) GO TO 1
      DO 73 I=1,NP
      DO 73 J=1,NP
   73 Q22(I,J)=Q2(I,J)
      GO TO 1
   19 CALL READM(C2,N1,N2,NDQ,NT,NC2,LP)
      NQ=N1
      N=N2
      IF(IO.EQ.0) GO TO 1
      DO 74 I=1,NQ
      DO 74 J=1,N
   74 C22(I,J)=C2(I,J)
      GO TO 1
   20 CALL READM(D2,N1,N2,NDQ,NT,ND2,LP)
      NQ=N1
      L=N2
      ID1=1
      IF(NT.EQ.NZ) ID1=0
      IF(IO.EQ.0) GO TO 1
      DO 75 I=1,NQ
      DO 75 J=1,L
   75 D22(I,J)=D2(I,J)
      GO TO 1
   21 CALL READM(R2,N1,N2,ND,NT,NR2,LP)
      NQ=N1
      IF(IO.EQ.0) GO TO 1
      DO 76 I=1,NQ
```

```
      DO 76 J=1,NQ
   76 R22(I,J)=R2(I,J)
      GO TO 1
   22 CALL READ4(THETA,N1,N2,NDQ,NT,NTH,LP)
      NQ=N1
      NP=N2
      IS1=1
      IF(NT.EQ.NZ) IS1=0
      IF(IO.EQ.0) GO TO 1
      DO 77 I=1,NQ
      DO 77 J=1,NP
   77 SITA2(I,J)=THETA(I,J)
      GO TO 1
   23 PRINT 471
  471 FORMAT(//," *** SENSOR COMBINATION TESTING***",//)
      IF(ID1.EQ.0) GO TO 500
      DO 31 I=1,L
      DO 31 J=1,M
      H(I,J)=0.
      DO 31 K=1,M
   31 H(I,J)=H(I,J)+D1(K,I)*Q1(K,J)
  500 DO 32 I=1,L
      DO 32 J=1,L
      SUM=0.
      IF(ID1.EQ.0) GO TO 32
      DO 33 K=1,M
   33 SUM=SUM+H(I,K)*D1(K,J)
   32 EL(I,J)=R1(I,J)+SUM
      CALL CONTRL(A,B,C1,D1,Q1,R1,S,GAIN,H,P,VAL,N,M,L,ND,NDD,NDM,
     1NDL,W,ID1)
      DO 37 I=1,N
      DO 37 J=1,NP
      H(I,J)=0.
      DO 37 K=1,N
   37 H(I,J)=H(I,J)+S(I,K)*GAMA(K,J)
      DO 38 I=1,N
      DO 38 J=1,NP
      P(I,J)=0.
      DO 38 K=1,NP
   38 P(I,J)=P(I,J)+H(I,K)*Q2(K,J)
      TRACE=0.
      DO 39 I=1,N
      SUM=0.
      DO 40 K=1,NP
   40 SUM=SUM+P(I,K)*GAMA(I,K)
   39 TRACE=TRACE+SUM
      DO 93 I=1,M
      DO 93 J=1,N
      SUM=0.
      IF(ID1.EQ.0) GO TO 93
      DO 94 K=1,L
   94 SUM=SUM+D1(I,K)*GAIN(K,J)
```

```
 93 R11(I,J)=C1(I,J)+SUM
    DO 60 I=1,N
    DO 60 J=1,L
    PW(I,J)=0.
    DO 60 K=1,L
 60 PW(I,J)=PW(I,J)+GAIN(K,I)*EL(K,J)
    DO 61 I=1,N
    DO 61 J=1,N
    EL(I,J)=0.
    DO 61 K=1,L
 61 EL(I,J)=EL(I,J)+PW(I,K)*GAIN(K,J)
    ISEN=0
    NTP2=NQ+2
    NQP2=NQ+2
    NQQ=NQ
    IF(IO.EQ.0) GO TO 433
    DO 434 I=1,NQ
    IELEM(I)=I
434 INDEX(I)=I
433 IN=0
    DO 103 IX=1,NQP2
    IF(ISEN.EQ.0) GO TO 25
    IF(ISEN.EQ.1) GO TO 428
426 IF(IX.LT.NQP2) GO TO 428
    IF(IN.GT.0) GO TO 429
    NQ=NQQ
    IXN=0
    GO TO 435
429 IF(IN.EQ.1) GO TO 430
    AMIN=XMAX(1)
    INEL=1
    IXN=INDEX(1)
    IXE=IELEM(1)
    DO 105 I=2,IN
    IF(XMAX(I).GE.AMIN) GO TO 105
    IXN=INDEX(I)
    INEL=I
    AMIN=XMAX(I)
    IXE=IELEM(I)
105 CONTINUE
    GO TO 435
430 IXN=INDEX(1)
    IXE=IELEM(1)
    GO TO 435
428 IM=IX
    IF(NQP2.EQ.NTP2) IM=IX-1
    IXN=INDEX(IM)
    IXE=IELEM(IM)
435 DO 30 I=1,NP
    DO 30 J=1,NP
 30 Q2(I,J)=Q22(I,J)
    IT=1
```

```
      DO 31 IY=1,NQQ
      IF(IY.EQ.IXN) GO TO 31
      DO 32 J=1,N
   32 C2(IT,J)=C22(IY,J)
      R2(IT,IT)=R22(IY,IY)
      DO 35 J=1,NP
   35 THETA(IT,J)=SITA2(IY,J)
      IT=IT+1
   31 CONTINUE
      IF(IN.EQ.0.AND.IX.EQ.NQP2) GO TO 705
      NQ=NQQ-1
      IF(IX.LT.NQP2) GO TO 440
      NQQ=NQ
      IF(IN.EQ.1) GO TO 436
      DO 441 I=1,NQ
      DO 442 J=1,N
  442 C22(I,J)=C2(I,J)
      DO 444 J=1,NP
  444 SITA2(I,J)=THETA(I,J)
  441 R22(I,I)=R2(I,I)
  436 PRINT 445,IXE
  445 FORMAT(///," ***SENSOR NO.",I3," WAS ELEMINATED***",///)
      IF(IN.EQ.1) GO TO 706
      GO TO 431
  440 PRINT 400,IXE
  400 FORMAT(//," ***TAKE OUT SENSOR NO.",I3," ***")
      GO TO 431
  705 PRINT 707
  707 FORMAT(////," NO MORE SENSOR CAN BE ELEMINATED")
  706 PRINT 708
  708 FORMAT(" FINAL FORM OF MATRICES CONCERNING SENSORS ARE",//)
  431 PRINT *," C2 MATRIX"
      CALL PRINT(C2,NQ,N,NDQ,LP)
      PRINT *," R2 MATRIX"
      CALL PRINT(R2,NQ,NQ,ND,LP)
      PRINT *," THETA MATRIX"
      CALL PRINT(THETA,NQ,NP,NDQ,LP)
      IF(IX.EQ.NQP2) GO TO 701
   25 CALL FILTER(A,C2,GAMA,THETA,Q2,R2,S,GK,H,P,VAL,N,NQ,NP,ND,NDD,
     1NDQ,NDP,W,I31)
      IF(IP.EQ.0.AND.ISEN.NE.0) GO TO 510
      PRINT 90
   90 FORMAT(/," STEADY STATE SOLUTION OF FILTER RICCATI EQUATION")
      CALL PRINT(S,N,N,ND,LP)
  510 IF(IO.EQ.0) GO TO 403
      IF(ISEN.NE.0) GO TO 401
      DO 402 I=1,N
      DO 402 J=1,N
  402 SO(I,J)=S(I,J)
      GO TO 403
  401 IF(IP.LE.1) GO TO 403
      DO 404 I=1,N
```

```
      DO 404 J=1,N
  404 PW(I,J)=(S(I,J)-SO(I,J))/SO(I,J)
      PRINT *," RELATIVE ERROR OF FILTER RICCATI SOLUTION"
      CALL PRINT(PW,N,N,ND,LP)
  403 DO 45 I=1,N
   45 W(I)=SQRT(S(I,I))
      IF(IP.EQ.0.AND.ISEN.NE.0) GO TO 511
      PRINT 46
   46 FORMAT(/," RMS RESPONSE OF ESTIMATE ERRORS")
      CALL PRINT(W,1,N,1,LP)
  511 IF(IO.EQ.0) GO TO 407
      IF(ISEN.NE.0) GO TO 405
      DO 406 I=1,N
  406 WS(I)=W(I)
      GO TO 407
  405 DO 408 I=1,N
  408 W(I)=(W(I)-WS(I))/WS(I)
      PRINT *," RELATIVE ERROR OF RMS RESPONSE OF ESTIMATED ERROR"
      CALL PRINT(W,1,N,1,LP)
      AMX=0.
      IF(ICR.NE.1) GO TO 407
      DO 102 I=1,N
      X=ABS(W(I))
      IF(X.GT.AMX) AMX=X
  102 CONTINUE
  407 DO 3 I=1,L
      DO 3 J=1,N
      PW(I,J)=0.
      DO 3 K=1,N
    3 PW(I,J)=PW(I,J)+GAIN(I,K)*S(K,J)
      DO 9 I=1,L
      DO 9 J=1,L
      Q1(I,J)=0.
      DO 9 K=1,N
    9 Q1(I,J)=Q1(I,J)+PW(I,K)*GAIN(J,K)
      IF(IP.EQ.0.AND.ISEN.NE.0) GO TO 512
      PRINT *," ERROR CONTROL COVARIANCE MATRIX"
      CALL PRINT(Q1,L,L,ND,LP)
  512 IF(IO.EQ.0) GO TO 411
      IF(ISEN.NE.0) GO TO 409
      DO 410 I=1,L
      DO 410 J=1,L
  410 UO(I,J)=Q1(I,J)
      GO TO 411
  409 IF(IP.LE.1) GO TO 411
      DO 412 I=1,L
      DO 412 J=1,L
  412 PW(I,J)=(Q1(I,J)-UO(I,J))/UO(I,J)
      PRINT *," RELATIVE ERROR OF ERROR CONTROL"
      CALL PRINT(PW,L,L,ND,LP)
  411 DO 35 I=1,L
   35 W(I)=SQRT(Q1(I,I))
```

```
      IF(IP.EQ.0.AND.ISEN.NE.0) GO TO 513
      PRINT *," RMS RESPONSE OF ERROR CONTROL"
      CALL PRINT(W,1,L,1,LP)
  513 IF(IO.EQ.0) GO TO 415
      IF(ISEN.NE.0) GO TO 413
      DO 414 I=1,L
  414 WU(I)=W(I)
      GO TO 415
  413 DO 416 I=1,L
  416 W(I)=(W(I)-WU(I))/WU(I)
      PRINT *," RELATIVE ERROR OF RMS RESPONSE OF ERROR CONTROL"
      CALL PRINT(W,1,L,1,LP)
      IF(ICR.NE.2) GO TO 415
      DO 110 I=1,L
      X=ABS(W(I))
      IF(X.GT.AMX) AMX=X
  110 CONTINUE
  415 DO 95 I=1,M
      DO 95 J=1,N
      Q1(I,J)=0.
      DO 95 K=1,N
   95 Q1(I,J)=Q1(I,J)+R11(I,K)*G(K,J)
      DO 96 I=1,M
      DO 96 J=1,M
      Q2(I,J)=0.
      DO 96 K=1,N
   96 Q2(I,J)=Q2(I,J)+Q1(I,K)*R11(J,K)
      IF(IP.EQ.0.AND.ISEN.NE.0) GO TO 514
      PRINT *," ERROR OUTPUT COVARIANCE MATRIX"
      CALL PRINT(Q2,M,M,ND,LP)
  514 IF(IO.EQ.0) GO TO 419
      IF(ISEN.NE.0) GO TO 417
      DO 418 I=1,M
      DO 418 J=1,M
  418 YO(I,J)=Q2(I,J)
      GO TO 419
  417 IF(IP.LE.1) GO TO 419
      DO 420 I=1,M
      DO 420 J=1,M
  420 PW(I,J)=(Q2(I,J)-YO(I,J))/YO(I,J)
      PRINT *," RELATIVE ERROR OF ERROR OUTPUT"
      CALL PRINT(PW,M,M,ND,LP)
  419 DO 97 I=1,M
   97 W(I)=SQRT(Q2(I,I))
      IF(IP.EQ.0.AND.ISEN.NE.0) GO TO 515
      PRINT *," RMS RESPONSE OF ERROR OUTPUT"
      CALL PRINT(W,1,M,1,LP)
  515 IF(IO.EQ.0) GO TO 423
      IF(ISEN.NE.0) GO TO 421
      DO 422 I=1,M
  422 WY(I)=W(I)
      GO TO 423
```

```
421 DO 424 I=1,M
424 W(I)=(W(I)-WY(I))/WY(I)
    PRINT *," RELATIVE ERROR OF RMS RESPONSE OF ERROR OUTPUT"
    CALL PRINT(W,1,M,1,LP)
    IF(ICR.NE.3) GO TO 423
    DO 104 I=1,M
    X=ABS(W(I))
    IF(X.GT.AMX) AMX=X
104 CONTINUE
423 TRACEX=0.
    DO 62 I=1,N
    SUM=0.
    DO 63 K=1,N
 63 SUM=SUM+EL(I,K)*S(K,I)
 62 TRACEX=TRACEX+SUM
    TRACEX=TRACEX+TRACE
    IF(IP.EQ.0.AND.ISEN.NE.0) GO TO 516
    PRINT 64,TRACEX
 64 FORMAT(//," ***TRACE=",G13.5," ***",//)
516 IF(IO.EQ.0) GO TO 301
    IF(ISEN.NE.0) GO TO 425
    TRACEO=TRACEX
    GO TO 700
425 TRACEX=(TRACEX-TRACEO)/TRACEO
    PRINT 427,TRACEX
427 FORMAT(//," $$$$$ RELATIVE ERROR OF TRACE=",G13.5," $$$$",//)
    IF(ICR.EQ.4) AMX=TRACEX
    ISEN=2
    IF(AMX.GT.CRITER) GO TO 103
    IN=IN+1
    INDEX(IN)=IXN
    IELEM(IN)=IXE
    XMAX(IN)=AMX
    GO TO 103
700 PRINT*," TYPE IN THE CRITERIA"
    READ*,CRITER
    PRINT 600,CRITER
600 FORMAT(/," ELEMINATION CRITERIA=",G12.4)
    ISEN=1
103 CONTINUE
701 IF(IN.LE.1) GO TO 301
    NQP2=IN
    IT=INEL
    ILP=IT+1
    DO 432 I=ILP,IN
    INDEX(IT)=INDEX(I)-1
    IELEM(IT)=IELEM(I)
    IT=IT+1
432 CONTINUE
    GO TO 433
301 GO TO 300
    END
```