Numerical Solution of Algebraic Matrix Riccati Equations

by

William Fredrick Arnold III
Electronic Warfare Department

FEBRUARY 1984

NAVAL WEAPONS CENTER
CHINA LAKE, CALIFORNIA 93555

Approved for public release; distribution unlimited.
FOREWORD

This report documents and distributes a dissertation submitted by the author in partial satisfaction of the requirements for the degree of Doctor of Philosophy in Electrical Engineering at the University of Southern California, Los Angeles.

The research was conducted during the period September 1981 to August 1983.

Approved by
P. B. HOMER, Head
Electronic Warfare Department

Released for publication by
B. W. HAYS
Technical Director
February 1984

Under authority of
K. A. DICKERSON
Capt., U.S. Navy
Commander

NWC Technical Publication 6521

Published by ........................................ Electronic Warfare Department
Collation ........................................................... Cover, 83 leaves
First printing ........................................................ 95 unnumbered copies
Numerical Solution of Algebraic Matrix Riccati Equations

William Fredrick Arnold III

Naval Weapons Center
China Lake, California 93555

Naval Weapons Center
China Lake, California 93555

February 1984

163

Approved for public release; distribution is unlimited.

See back of form.
(U) **Numerical Solution of Algebraic Matrix Riccati Equations** (U), by William Fredrick Arnold III. China Lake, Calif., Naval Weapons Center, February 1984. 163 pp. (NWC TP 6521, publication UNCLASSIFIED.)

(U) Numerical issues related to the computational solution of the algebraic matrix Riccati equation are studied. The approach uses the generalized eigenproblem formulation for the solution of general forms of algebraic Riccati equations arising in both continuous- and discrete-time applications. These general forms result from control and filtering problems for systems in generalized (or implicit or descriptor) state space form. A Newton-type iterative refinement procedure for the generalized Riccati solution is derived. The issue of numerical condition of the Riccati problem is addressed. Balancing to improve numerical condition is discussed. An overview of a software package coded in FORTRAN is given. Results of numerical experiments are reported.

(U) The special structure of models of physical systems given in linear second-order form is examined. Exploiting the structure in solving associated Riccati equations is studied. Tests for controllability and observability are derived in terms of the original second-order-model matrices.
CONTENTS

Introduction .................................................. 3
Main Contributions of the Thesis .............................. 5
Outline of Chapters ........................................... 6

Background .................................................... 9
Numerical Stability and Conditioning ......................... 9
Perturbation Analysis ......................................... 17
Solution of the Generalized Algebraic Riccati Equation
(CARE) as a Generalized Eigenproblem ....................... 18

Iterative Refinement by Newton's Method ...................... 31
Iterative Refinement of Continuous-Time Solution ............ 31
Iterative Refinement of Discrete-Time Solution ............... 40

Conditioning of the Algebraic Riccati Problem .............. 49
Previous Work ............................................... 49
First-Order-Perturbation Analysis of the GARE .............. 54
Balancing to Improve Condition ................................ 60

Numerical Experiments ....................................... 67
Software Package RICPACK .................................... 67
Algorithms and Software ...................................... 70
Numerical Examples .......................................... 70

Second-Order Models .......................................... 79
Second-Order-Model Structure in the LSS Framework .......... 80
The GARE for Second-Order Models ............................ 82
Controllability and Observability Criteria for
Second-Order Models ......................................... 84

Conclusion ..................................................... 93

References ..................................................... 97

Appendix A: Software Description ............................ 103
Figures:
2.1 Geometric View of the Effect of Data Uncertainty and Computational Errors in Solving a Problem ........ 10
2.2 Possible Combinations of Problem Conditioning and Computation Stability in the Numerical Solution of a Problem ......................... 11
2.3 Geometric Interpretation of the Condition Number, $\kappa$ .................................................. 13
2.4 Illustration of Backward Error ................................................. 15

Tables:
5.1 Default Options .............................................................. 68
5.2 Numerical Results for Example 1, $\varepsilon=10^{-N}$ .............. 72
5.3 Numerical Results for Example 2, $\varepsilon=10^{-N}$ .............. 74
5.4 Numerical Results for Example 3, Ward Balancing, $\varepsilon=10^{-N}$ ............................................. 76
5.5 Numerical Results for Example 3, System Balancing, $\varepsilon=10^{-N}$ ............................................. 77
CHAPTER 1
INTRODUCTION

In this study some numerical issues related to the computational solution of a generalized form of the algebraic matrix Riccati equation are examined. The approach herein utilizes the generalized eigenproblem formulation, which provides a powerful framework for the solution of quite general forms of algebraic Riccati equations arising in both continuous- and discrete-time applications. This general form is derived from control and filtering problems for systems in generalized (or implicit or descriptor) state space form. These equations play fundamental roles in the analysis, synthesis, and design of linear-quadratic-Gaussian control and estimation systems as well as in other areas of applied mathematics. A representative sample of applications may be found in [1]-[4]. It is not our purpose here to survey the extensive literature available on Riccati equations, but, rather we refer the reader to, for example [1]-[4] for references.

The method exploited here is a variant of the classical eigenvector approach to Riccati equations. Martensson [5] is one of the best summaries of the eigenvector approach to solving algebraic Riccati equations. However, the use of eigenvectors often encounters numerical difficulties in practical computation, especially when the corresponding eigenvalues are closely spaced. Thus, the method to be preferred and the basis of this work employ Schur vectors instead of eigenvectors because Schur vectors are more reliably and accurately computed. The Schur method was first examined in detail by Laub [6], [7] and
then extended by Pappas, et.al. [8] and Emami-Naeini [9] for the
discrete-time problem, and by Van Dooren [10] for the continuous- and
discrete-time problem. The very general formulation herein was derived
by Laub [11] and expanded by Lee [12].

The numerical issue first examined in this thesis is the deriv-
tion of a Newton-type iterative-refinement procedure for the general-
of this type for the standard Riccati equation in the continuous- and
discrete-time characterizations, respectively.

Perhaps the most important issue studied here is that of condi-
tioning of the Riccati problem, that is, to define a measure of the
sensitivity of the numerical solution to changes in the data. Although
results are available on conditioning for problems such as linear
equations [15], [16] and eigenvalues [17]-[19], little is published on
the conditioning of the Riccati problem. A recent work in this area
for the continuous-time problem is by Byers [20]. The results of
numerical experiments examining various condition measures are reported
herein.

Since the behavior of many physical systems in engineering is
first modeled by second-order differential equations with very special
properties (i.e., symmetry and definiteness), we examine exploiting
this structure in certain computations of interest. Namely, we exploit
the structure in solving associated Riccati problems and in testing for
controllability and observability of the second-order models.
1.1 **Main Contributions of the Thesis**

We regard the derivation of condition measures for the continuous- and discrete-time generalized algebraic Riccati equation and the evaluation of their numerical behavior in special situations as the main contributions of this thesis. More specifically, we may list the following contributions.

1. The derivation of a Newton-type iterative-refinement procedure for the continuous and discrete algebraic Riccati equation that has monotonic convergence that is quadratic in the vicinity of the solution.

2. Establishing the equivalence between the given generalized optimal control problem solution and the solution of a more standard optimal control problem in certain cases.

3. Derivation of condition numbers for the solution of the generalized Riccati equation.

4. A new algorithm for applying a change of coordinate transformation to the system model when the Riccati problem is solved using the generalized eigenproblem approach.

5. Numerical experimentation to illustrate the behavior of known and speculated condition estimates for the Riccati problem. The numerical results demonstrate that a single satisfactory condition estimate is not available.

6. Exploitation of the structure of second-order models to solve efficiently the velocity feedback optimal control problem using a generalized Riccati equation.
7. Formulation of useful tests for controllability and observability of second-order models directly in terms of conditions on the model matrices.


1.2 Outline of Chapters

In Chapter 2 we review some important concepts of numerical analysis, which are used heavily in the succeeding analysis. The concepts of numerical stability and conditioning are presented, and there is a brief discussion of first-order perturbation analysis. The application of Schur techniques for the solution of generalized algebraic Riccati equations is reviewed. Very general optimal control and filtering problems are formulated, which result in generalized algebraic Riccati equations for the continuous- and discrete-time cases.

Chapter 3 derives a Newton-type iterative procedure, which we employ for improving the numerical accuracy of the Schur solution of generalized Riccati equations or for calculating new solutions when problem parameters are changed by a small amount. Newton's method for the standard algebraic Riccati equation in continuous- and discrete-time formulations is reviewed first. Equivalence between the generalized solution and a certain standard solution is established. The Newton procedure is then extended to the generalized case.

The subject of conditioning of the generalized algebraic Riccati equation is examined in Chapter 4. The desired form of condition
estimate is defined. Previous work in the area of conditioning of the Riccati problem is reviewed. New condition estimates are then derived using a first-order perturbation analysis. Balancing to improve the condition of the problem is next considered. Balancing of the generalized eigenproblem is discussed first. A new algorithm for applying a change of coordinates transformation to the system model when the Riccati problem is solved in the generalized eigenproblem formulation is presented.

Numerical results for the solution of the generalized Riccati equation for special cases are presented in Chapter 5. The software package developed as a research tool to aid in the studies of this thesis is discussed. Highlights of the package capabilities are given. The numerical algorithms employed and sources for the software are discussed and referenced. The results of three specially designed examples are examined to illustrate the behavior of the Riccati solution and the ability of the known and new condition estimates to predict the numerical accuracy of the solution.

In Chapter 6 second-order models are considered. The second-order model structure is defined in the large space structure framework. The generalized algebraic Riccati equation is considered for this problem, and ways are explored to take computational advantage of the second-order formulation to solve the Riccati equation. The velocity feedback problem is shown to reduce to a simple form in this framework. Tests for controllability (stabilizability) and observability (detectability)
are then derived in terms of the model matrices for second order models.

Finally in Chapter 7 we state the conclusions we have drawn from this work and make recommendations for future research in this area.

An appendix is included which contains the description (from the actual software) of all the FORTRAN subroutines accumulated in the software package.
CHAPTER 2
BACKGROUND

In this chapter we review some important concepts and results which form the basis for the work in the succeeding chapters. The first section discusses the numerical analysis concepts of stability and conditioning. The second section briefly presents the perturbation theory necessary in later analysis. The third section reviews results employing Schur techniques for the solution of algebraic matrix Riccati equations. Very general optimal control and filtering problems are formulated which result in generalized algebraic Riccati equations (GARE) for the continuous- and discrete-time cases. Solutions for these GARE are stated in terms of a generalized eigenproblem.

2.1 Numerical Stability and Conditioning

We shall now introduce the important concepts of numerical stability and conditioning. The following framework will be useful for the understanding of these concepts:

\[ f : D \rightarrow S \] (2.1)

Give \( d \in D \), we want to compute \( f(d) \in S \). However, frequently only \( d^* \) (near \( d \)) is known and only \( f^* \) (an algorithm to approximate \( f \)) is available. Figure 2.1 is a geometrical view of this problem-solving process. Therefore, in this framework we seek \( f(d) \), but compute \( f^*(d^*) \).
2.1.1 Numerical Stability

**Definition 2.1:** An algorithm $f^*$ is numerically stable if for all $d$ contained in $D$, there exists a $d^*$ contained in $D$ and near $d$ such that $f(d^*)$ (the exact solution of a slightly perturbed problem) is near $f^*(d)$ (the computed solution).

That is, we expect that a stable algorithm will not introduce any inaccuracies into the solution larger than those present in the data.

2.1.2 Problem Conditioning

**Definition 2.2:** If $f(d^*)$ (the exact solution of a slightly perturbed problem) is near $f(d)$ (the true solution), the problem is said to be well-conditioned. If $f(d^*)$ may potentially differ greatly from $f(d)$, the problem is ill-conditioned.
The four possible combinations of stability and conditioning are shown in Figure 2.2.

![Diagram of possible combinations of stability and conditioning]

a) Well-conditioned problem
   Stable computation

b) Well-conditioned problem
   Unstable computation

c) Ill-conditioned problem
   Stable computation

d) Ill-conditioned problem
   Unstable computation

FIGURE 2.2 Possible combinations of problem conditioning and computation stability in the numerical solution of a problem.
Although it may be impossible for the numerical solution process to guarantee an accurate answer to an ill-conditioned problem, it is desirable for the program to recognize ill-conditioning of the computations and report this fact to the user. Therefore, it is desirable to associate data with a computing problem which reflects the overall sensitivity of the solution to changes in the data (i.e., a condition number). We can do this as follows [21]:

**Definition 2.3:** Let \( D \) and \( S \) be finite dimensional metric spaces with metrics \( P_d \) and \( P_s \), respectively. Let \( f(d) \) be the computing problem under consideration. As before

\[
f : D \rightarrow S
\]

The absolute asymptotic condition number is

\[
\kappa(f(d)) := \lim_{\delta \to 0} \sup_{p_d(d,d^*) = \delta} \frac{\overline{p_s(f(d), f(d^*))}}{\delta}
\]  

(2.2)

Relative condition can be defined similarly [21]. Figure 2.3 illustrates the concept of a condition number. One can see that \( \kappa \) is a measure of the sensitivity of the solution to perturbations in the data. \( \kappa \) can be interpreted as the factor by which the uncertainty, \( \delta \), is multiplied by in the problem-solving process. Condition numbers are generally estimated using perturbation theory.
2.1.3 Condition Number of a Matrix

Definition 2.4: Let \( \| \cdot \| \) be a consistent matrix norm on \( \mathbb{R}^{n \times n} \) satisfying

\[
\| \| = 1 \quad (2.3)
\]

with a consistent vector norm on \( \mathbb{R}^n \). Let \( A \in \mathbb{R}^{n \times n} \), then

\[
\kappa(A) = \frac{\| A \|}{\| A^{-1} \|} \quad (2.4)
\]

Now consider the computation of the inverse of a slightly perturbed matrix, i.e., find \( (A + \delta)^{-1} \).

Theorem 2.1: If \( \| A^{-1} \| \cdot \| \delta \| < 1 \), then

\[
\frac{\| A^{-1} - (A + \delta)^{-1} \|}{\| A^{-1} \|} \leq \frac{\kappa(A) \| \delta \|}{\| A \|} \quad (2.5)
\]
Proof: [15]

The left hand side of the inequality is the relative error in \((A + E)^{-1}\). If \(E\) is sufficiently small, the right side is effectively \(\kappa(A) \frac{\|E\|}{\|A\|}\). Therefore, the theorem states that the relative error in \(A + E\) may be magnified by as much as \(\kappa(A)\) in calculating \((A + E)^{-1}\). For this reason, \(\kappa(A)\) is called "the condition number of \(A\) with respect to inversion." Note that

\[ 1 - \|I\| = \|A\|^{-1} \leq \|A\| \|A^{-1}\| = \kappa(A) \tag{2.6} \]

2.1.4 Error Analysis

In the numerical solution of real problems, we compute \(f^*(d^*)\) instead of \(f(d)\), so we would like to estimate \(\|f(d) - f^*(d^*)\|\). This is the basic goal of error analysis. There are two main types of error analysis of numerical computation and are referred to as forward and backward error analysis.

In the forward error analysis, one attempts to obtain a bound on the error in the final result by starting with the original problem and following, step by step, the effect of computational errors (round-off) and original data uncertainty. However, the resulting bounds are usually hopelessly pessimistic, and the analysis itself is very complicated for all but simple problems [16], [22].

In backward error analysis, one does not attempt to compute \(\|f(d) - f^*(d^*)\|\), but rather one attempts to determine how close the problem actually solved is to the original problem. This is illustrated graphically in Figure 2.4. In this technique, one uses error bounds to
show that the computed solution of a given problem is near the exact solution of a slightly perturbed problem. This is sufficient to ensure that the algorithm that did the computations is stable.

![Figure 2.4 Illustration of backward error.](image)

2.1.5 Role of Orthogonal Matrices

The class of orthogonal transformations has a special role in numerical computations.

Definition 2.5: An orthogonal matrix $U$ is a square matrix with orthonormal columns. That is

$$U \in \mathbb{R}^{n \times n}, \quad U = (u_1, u_2, \ldots, u_n)$$

and

$$u_i^T u_j = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$$

Properties of orthogonal matrices important to numerical computations are

Theorem 2.2: Let $U$ be orthogonal, then
1) \( U^T U = U U^T = I \); 
2) \( I A U I_2 = U A I_2 = I A I_2 \). 

**Proof:** 1) \( U^T U = I \) follows from the orthonormality of the columns of \( U \). It follows then that \( U^T \) is the inverse of \( U \). Since by definition a matrix commutes with its inverse, we have \( U U^T = I \). Note that this implies that the rows of \( U \) are also orthonormal.

2) To prove this part, we make use of the facts that for \( A \in \mathbb{R}^{m \times n} \), \( I A_2^2 = I A^T A I_2 \) and \( I A^T I_2 = I A I_2 \) ([15], Theorem 4.2.10).

Now,

\[
I A U I_2^2 = I A (U A)^T U A I_2 = I A^T (U^T U A) I_2 = I A^T A I_2 = I A I_2^2
\]

and, therefore,

\[
I A U I_2 = I A (A U)^T I_2 = I A^T (U^T A) I_2 = I A^T A I_2 = I A I_2.
\]

From this it is easy to see that orthogonal matrices have three advantages. First, they are easy to invert; \( U^{-1} = U^T \). Second, orthogonal matrices are perfectly conditioned with respect to \( I \cdot I_2 \):

\[
\kappa(U) = I A I_2 I U^T I_2 = 1 \cdot 1 = 1.
\]

(2.8)

A third advantage of orthogonal transformations is that perturbations in the result can be accounted for by a perturbation of the same size in the original problem. For example, having computed \( U T A U \), we introduce an error \( F \) into the result. If we set \( E = UFU^T \), then \( I E I_2 = I F I_2 \) and

\[
U^T (A + E) U = U^T A U + F.
\]

(2.9)
This makes orthogonal transformations ideal for backward error analysis. That is, multiplications by orthogonal transformations are backward stable.

2.2 Perturbation Analysis

For the purposes of this thesis, we employ a technique commonly known as first-order-perturbation theory. This technique reveals the effects of perturbations in the problem on the solution. The technique is generally applied in a three-step procedure. First, a form is chosen for the approximate problem solution. Then the approximation is substituted into an equation, and all terms second-order or higher in small quantities are deleted. Finally, the resulting linear equation is solved for the unknown in the approximation.

To illustrate the technique, consider the problem of estimating $(A + E)^{-1}$, where $A$ is nonsingular and $E$ is a "small-perturbation" matrix. First we chose a form for the approximation as

$$(A + E)^{-1} = A^{-1}(I - H), \quad (2.10)$$

where $H$ is a "small perturbation" matrix. Substituting the approximation into the equation

$$(A + E)(A + E)^{-1} = I, \quad (2.11)$$

we obtain

$$(A + E)A^{-1}(I - H) = I - H + EA^{-1} - EA^{-1}H = I. \quad (2.12)$$

Dropping the term $EA^{-1}H$, which is assumed small in comparison to $E$ and $H$, and solving the resulting equation gives us

$$H = EA^{-1}, \quad (2.13)$$

or
We can now loosely derive the condition number of a matrix with respect to inversion. From (2.14) we have

\[ \frac{1}{\|A^{-1}\|_2} \leq \|I-A^{-1}\|_2 \|I\|_2 \]

(2.15)

Hence, we have

\[ \frac{1}{\|A^{-1}\|_2} \leq \|I\|_2 \|E\|_2 = \kappa(A) \frac{\|E\|_2}{\|A\|_2} \]

(2.16)

This differs from the bound of Theorem 2.1 (equation 2.5) by the factor

\[ \frac{1}{1 - \kappa(A) \|E\|_2} \]

(2.17)

which is negligible when \( \|E\|_2 \) is small.

2.3 Solution of the Generalized Algebraic Riccati Equation (GARE) as a Generalized Eigenproblem

In this section we shall present the method of solving the GARE (both continuous- and discrete-time cases) via a generalized eigenproblem. First, the GARE resulting from the optimal regulator problem will be derived. Then the optimal filtering problem will also be stated, and the corresponding GARE given. Finally, the solutions to the GARE will be formulated utilizing the generalized real Schur form of the generalized eigenproblem.

2.3.1 Optimal Regulator - Continuous-Time Problem

Consider the following general time-invariant deterministic linear optimal regulator problem:

System:

\[ \dot{x}(t) = Ax(t) + Bu(t) ; \quad x(t_0) = x_0 \]

(2.18)

\[ y(t) = Cx(t) \]
Criterion: \[ J = \frac{1}{2} \int_0^\infty (y^TQy + u^TRu + 2x^TSu)dt \] (2.19)

where

\[ x \in \mathbb{R}^n; \quad u \in \mathbb{R}^m; \quad y \in \mathbb{R}^r; \] (2.20)

\[ E, A \in \mathbb{R}^{nxn}; \quad B \in \mathbb{R}^{nxm}; \quad C \in \mathbb{R}^{rxn}, \]

Also,

\[ 0 = QT \in \mathbb{R}^{nxr}; \quad R = RT \in \mathbb{R}^{nxn}; \quad S \in \mathbb{R}^{nxm}; \]

and

\[ \begin{bmatrix} C^TQ & S \\ S^T & R \end{bmatrix} \succ 0; \quad Q, R > 0; \quad E \text{ nonsingular.} \]

Application of Hamilton-Jacobi theory gives rise to the following equations:

\[ h = \frac{1}{2} (x^TCQx + 2x^TSu + u^TRu) + p^T(Ax + Bu - Ex) \] (2.22)

\[ \frac{\partial h}{\partial p} - \frac{d}{dt} \left( \frac{\partial h}{\partial p} \right) = 0 = Ax + Bu - Ex \] (2.23)

\[ \frac{\partial h}{\partial x} - \frac{d}{dt} \left( \frac{\partial h}{\partial x} \right) = 0 = C^TQx + Su + A^Tp + E^Tp \] (2.24)

\[ \frac{\partial h}{\partial u} - \frac{d}{dt} \left( \frac{\partial h}{\partial u} \right) = 0 = S^T \dot{x} + Ru + B^Tp \] (2.25)

where \( h \) is the scalar Hamiltonian, and \( p(t) \) is the costate vector, \( p \in \mathbb{R}^n \).

In the usual treatment \( E = I \) and \( S = 0 \), equation (2.25) is solved for \( u \), and \( u \) is then substituted in (2.23) and (2.24) to obtain the Hamiltonian system

\[ \begin{bmatrix} \dot{x} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B^T \\ -C^TQ & -A^T \end{bmatrix} \begin{bmatrix} x \\ p \end{bmatrix} \] (2.26)

If one makes the "Riccati substitution" \( p = Xx \), we are led to the algebraic Riccati equation for \( X \)
If the pair \((A, B)\) is stabilizable and the pair \((C, A)\) is detectable, then a unique non-negative definite solution \(X = X^T > 0\) exists such that the linear control law
\[ u^0 = -R^{-1}B^TX \] (2.28)
stabilizes the system and is optimal in the sense that it minimizes the criterion (2.19) over all other control laws [1].

In the more general setting we have the following Hamiltonian system
\[
\begin{bmatrix}
E & 0 \\
0 & E^T
\end{bmatrix}
\begin{bmatrix}
\dot{x} \\
\dot{p}
\end{bmatrix} =
\begin{bmatrix}
A-BR^{-1}S^T & -BR^{-1}B^T \\
SR^{-1}S^T-C^TQC & -(A-BR^{-1}S^T)^T
\end{bmatrix}
\begin{bmatrix}
x \\
p
\end{bmatrix}
\] (2.29)
If one makes the "Riccati substitution" \(p = XEx\), we are led to the GARE
\[
0 = (A-BR^{-1}S^T)Xe + E^TX(A-BR^{-1}S^T) - E^TXBR^{-1}B^TXe + C^TQC - SR^{-1}S^T
\]
\[ = A^TXe + E^Txa - (B^TXe + S^TR^{-1}(B^TXe + S^T)) + C^TQC \] (2.30)
and the optimal control law is given by
\[ u^0 = -R^{-1}(S^T + B^TXe)x \] (2.31)
To solve for \(X = X^T > 0\), one can use the technique suggested by Pappas, et.al. [8] for the discrete time ARE as expanded by Lee [12]. Before presenting this technique, we will formulate the other problems of interest.

2.3.2 **Optimal Filter - Continuous-Time Problem**

Consider the following time-invariant linear optimal observer problem
System: $\dot{x}(t) = Ax(t) + Bu(t) + Dv(t)$, \hspace{1cm} (2.32)
$y(t) = Cx(t) + w(t)$, $t \geq t_0$,
where $\begin{bmatrix} v(t) \\ w(t) \end{bmatrix} \in \mathbb{R}^{l+r}$ is a white noise process with intensity

\begin{equation}
L \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} > 0
\end{equation}
(2.33)

Also, (2.20) applies, $E$ is nonsingular and $D \in \mathbb{R}^{nxl}$. Furthermore, the initial state $x(t_0)$ is uncorrelated with $v$ and $w$, and $u(t)$ is a given input to the system.

The optimal observer is given by

\begin{equation}
\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + K^0[y(t) - C\hat{x}(t)],
\end{equation}
(2.34)
which minimizes the weighted mean square reconstruction error

\begin{equation}
E\{e^T(t)We(t)\},
\end{equation}
(2.35)
where

\begin{equation}
e(t) = x(t) - \hat{x}(t); \quad W = W^T > 0.
\end{equation}
(2.36)

Let

\begin{equation}
\bar{S} = ES,
\end{equation}
(2.37)

Then $K^0$ is given by

\begin{equation}
K^0 = [\bar{S} + EXC^T]^{-1},
\end{equation}
(2.38)
where $X$ solves the following GARE:

\begin{equation}
0 = (A-\bar{S}R^{-1}C)XE^T + EX(A-\bar{S}R^{-1}C)^T - EXC^TR^{-1}CXE^T + DQD^T - \bar{S}R^{-1}\bar{S}^T.
\end{equation}
(2.39)

One can see that this GARE is the dual to (2.30), and, therefore, we can solve for $X$ by the same technique.
2.3.3. Optimal Regulator – Discrete-Time Problem

Consider the following general discrete-time, time-invariant deterministic optimal regulator problem:

System: \[ E_{x_{k+1}} = A_{x_k} + B_{u_k} ; \ x(0) = x_0 \]  \hspace{1cm} (2.40)
\[ y_k = C_{x_k} \]

Criterion: \[ J = \frac{1}{2} \sum_{k=0}^{\infty} (y_k^T Q y_k + u_k^T R u_k + 2x_k^T S u_k) \]  \hspace{1cm} (2.41)

where (2.20) and (2.21) apply also except we only require \( R \geq 0 \).

Application of the discrete maximum principle [23] gives rise to the set of equations

\[ h_k = \frac{1}{2} x_k^T C^T Q C x_k + u_k^T R u_k + 2x_k^T S u_k + p_{k+1}^T A_{x_k} + B_{u_k} \]  \hspace{1cm} (2.42)

\[ E_{x}^T p_k = \frac{\partial h_k}{\partial x_k} = C^T Q C x_k + S u_k + A^T p_{k+1} \]  \hspace{1cm} (2.43)

\[ \frac{\partial h_k}{\partial u_k} = 0 = R u_k + S^T x_k + B^T p_{k+1} \]  \hspace{1cm} (2.44)

\[ E_{x_k} = \frac{\partial h_k}{\partial p_{k+1}} = A_{x_k} + B_{u_k} \]  \hspace{1cm} (2.45)

where \( h_k \) is the scalar Hamiltonian, and \( p_k \in \mathbb{R}^n \) is the costate vector.

Assuming for the moment that \( R \) is invertible, we can solve (2.44) for \( u_k \) and substitute into (2.43) and (2.45). Also, we make the "Riccati substitution" \( p_k = X E_{x_k} \) to obtain the following discrete version of the GARE (for nonsingular \( X \):
This equation can be algebraically manipulated into the following equivalent forms:

\[ T^T X E = (A - BR^{-1}S^T)^T (X^{-1} + BR^{-1}B^T)^{-1} (A - BR^{-1}S^T) \]
\[ + CTQC - SR^{-1}S^T. \]  

(2.46)

Note that (2.47) does not require \( X \) explicitly, and (2.48) does not require \( X^{-1} \) or \( R^{-1} \) explicitly.

Once \( X \) is determined from the above GARE, the linear optimal feedback is given by

\[ u_k^0 = -(R + B^T XB)^{-1} (A^T XB + S)^T x_k. \]  

(2.49)

As in the continuous-time case, stabilizability and detectability assumptions assure the existence and uniqueness of \( X = X^T > 0 \) such that (2.49) is a stabilizing feedback.

2.3.4 Optimal Filter - Discrete-Time Problem

Consider the following discrete-time, time-invariant linear optimal observer problem

\[ \text{System: } EX_{k+1} = AX_k + Bu_k + Dv_k \]
\[ y_k = Cx_k + w_k \quad k > 0 \]  

(2.50)
where \( v_k \in \mathbb{R}^l \) and \( w_k \in \mathbb{R}^r \) are zero mean sequences with

\[
E \begin{bmatrix} v_k^T \\ w_k^T \end{bmatrix} = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \delta_{k l} \tag{2.51}
\]

\( Q = Q^T \in \mathbb{R}^{lxl}, \quad S \in \mathbb{R}^{lxr}, \quad R = R^T \in \mathbb{R}^{rxr} \)

Equation (2.20) applies, \( E \) is nonsingular and \( D \in \mathbb{R}^{nxl} \). The input sequence \( u_k \) is known, and the initial state \( x_0 \) is uncorrelated with \( v_k \) and \( w_k \).

It can be shown that the optimal observer is given by

\[
E\hat{x}_{k+1} = A\hat{x}_k + B u_k + K^0 (y_k - C \hat{x}_k) . \tag{2.52}
\]

The weighted mean square reconstruction error

\[
E \{ (x_k - \hat{x}_k)^T W (x_k - \hat{x}_k) \}, \tag{2.53}
\]

where \( W > 0 \), is minimized when \( K \) is chosen to be

\[
K^0 = (AXC^T + DS)(CXCT + R)^{-1} \tag{2.54}
\]

and \( X \) is found by solving the GARE:

\[
EXE^T = AXA^T + DQDT - (AXC^T + DS)(CXCT + R)^{-1}(AXC^T + DS)^T \tag{2.55}
\]

or equivalently

\[
EXE^T = (A - DSR^{-1}C)X(A - DSR^{-1}C)^T + DOD^T - DSR^{-1}S^TDT - (A - DSR^{-1}C)XC^T(CXCT + R)^{-1}CX (A - DSR^{-1}C)^T \tag{2.56}
\]

Equations (2.55) and (2.56) are dual to (2.48) and (2.47), and we can solve either problem using the same technique. We shall now present a technique for solving the GARE.
2.3.5 GARE Solution - Continuous-Time Case

We will work in the context of the regulator problem for convenience. If we define

\[
y = \begin{bmatrix} x \\ p \end{bmatrix}; \quad L = \begin{bmatrix} E & 0 \\ 0 & E^T \end{bmatrix}; \quad M = \begin{bmatrix} A^{\text{T}} - BR^{-1}S T & -BR^{-1}B^T \\ SR^{-1}S^T & C^T Q C \end{bmatrix};
\]  

Then we can associate with (2.29) a generalized eigenproblem

\[
\lambda L y = M y. \tag{2.58}
\]

**Theorem 2.3:** (Generalized Real Schur Form) let \( L, M \in \mathbb{R}^{2n \times 2n} \) and define the matrix pencil \( \lambda L - M \). The pencil is said to be regular if \( \det(\lambda L - M) \neq 0 \). There exist orthogonal transformations \( P \) and \( Z \) such that

\[
P^T(\lambda L - M)Z = \lambda P^TLZ - P^TMZ = \hat{\lambda} L - \hat{M} \tag{2.59}
\]

where \( \hat{L} \) is upper triangular and \( \hat{M} \) is quasi-upper triangular. For the \( 1 \times 1 \) diagonal blocks, the generalized eigenvalues are real \( \lambda_i = \hat{\lambda}_{ii}/\hat{\lambda}_{ii} \) (possibly "infinite" if \( \hat{\lambda}_{ii} = 0 \)). The \( 2 \times 2 \) blocks correspond to a finite pair of complex conjugate eigenvalues. Moreover, these eigenvalues can be arranged in any desired order.

**Proof:** [10]

It can be shown for \( L \) and \( M \), as defined in (2.57), that if \( \lambda \) is a generalized eigenvalue of \( \lambda L - M \), then \( -\lambda \) is also (Hamiltonian property of the eigenvalues). If the transformed pencil \( P^T(\lambda L - M)Z \) is partitioned as

\[
P^T(\lambda L - M)Z = \begin{bmatrix} \lambda P^TLZ & P^TMZ \end{bmatrix} = \begin{bmatrix} \hat{\lambda} L & -\hat{M} \end{bmatrix}
\]
where $L_{11}, M_{11} \in \mathbb{R}^{n \times n}$, then the vectors corresponding to the first $n$ columns of $Z$ span the eigenspace of $\lambda L_{11} - M_{11}$ [10]. Moreover, we can require that the real parts of the generalized eigenvalues of $\lambda L_{11} - M_{11}$ be negative. If we partition the corresponding $Z$ into $n \times n$ blocks as

$$Z = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix},$$

we can state the following:

**Theorem 2.4:** With respect to the assumptions made for this generalized eigenvalue formulation with $L$, $M$ and $Z$ as defined above,

1) Let $V = \begin{bmatrix} E & 0 \\ 0 & I \end{bmatrix} Z$ then $X = V_{21} V_{11}^{-1}$

solves (2.30) with $X = X^T \geq 0$;

2) The **generalized eigenvalues** of $\lambda L_{11} - M_{11}$ are the closed-loop eigenvalues of the system under the optimal feedback

$$u^0(t) = -R^{-1}(S^T + E^T X E)x(t)$$

**Proof:** [12]

The above method depends explicitly on $R^{-1}$ and can encounter computational difficulties when $R$ is ill-conditioned with respect to inversion. If this is the case, the following compression technique due to Van Dooren [10] can be employed. Arrange equations (2.23) through (2.25) as
Determine an orthogonal matrix $U \in \mathbb{R}^{(2n+m) \times (2n+m)}$ such that

$$
\begin{bmatrix}
E & 0 & 0 \\
0 & E^T & 0 \\
0 & 0 & 0 \\
\end{bmatrix} 
\begin{bmatrix}
\dot{x} \\
\dot{p} \\
\dot{\delta} \\
\end{bmatrix} = 
\begin{bmatrix}
A & 0 & B \\
-C^T QC & -A^T & -S \\
S^T & B^T & R \\
\end{bmatrix} 
\begin{bmatrix}
x \\
p \\
\delta \\
\end{bmatrix}.
$$

(2.63)

where $R \in \mathbb{R}^{m \times m}$ and is nonsingular. Then apply $U$ to the pencil corresponding to (2.63)

$$
U 
\begin{bmatrix}
E & 0 & 0 \\
0 & E^T & 0 \\
0 & 0 & 0 \\
\end{bmatrix} 
\begin{bmatrix}
A & 0 & B \\
-C^T QC & -A^T & -S \\
S^T & B^T & R \\
\end{bmatrix}
$$

(2.65)

It can be shown [10] that the pencils (2.66) and (2.58) are equivalent. Therefore, the pencil (2.66) can be used to solve for $X$ without having to invert $R$.

2.3.6 GAEB Solution - Discrete-Time Case

As in the previous section, we work in the context of the regulator problem for convenience. The matrix pencil appropriate for this problem can be formed from equations (2.43) through (2.45) as
Assuming for the moment that $R$ is invertible, we can derive a pencil equivalent to (2.67) in the form $\lambda L - M$ as

$$
\lambda \begin{bmatrix} E & 0 & 0 \\ 0 & A & 0 \\ 0 & B & 0 \\ 0 & -B^T & 0 \end{bmatrix} - \begin{bmatrix} A & 0 & B \\ -C^T Q C & E & -S \\ S^T & 0 & R \end{bmatrix} = (2.67)
$$

where $L$ and $M$ are defined appropriately. It can be shown that the generalized eigenvalues of (2.68) have the symplectic property, i.e., if $\lambda$ is a generalized eigenvalue of (2.68), then $\frac{1}{\lambda}$ is also [12].

We can now solve the discrete-time problem in a manner analogous to the continuous-time case. We first transform the pencil (2.68) to the form (2.60) using orthogonal $P$ and $Z$. Note if $R$ is singular or ill-conditioned with respect to inversion, we can compress the pencil (2.67) to a pencil equivalent to (2.68) using the technique of Van Dooren [10]. Now, we will require that $|\lambda|$ corresponding to the resulting pencil $\lambda L_{11} - M_{11}$ be less than unity. If we partition $Z$ as in (2.61), we have

**Theorem 2.5:** With respect to the assumptions made for this generalized eigenvalue formulation with $L$, $M$, and $Z$ as defined above,

1) Let $V = \begin{bmatrix} E & 0 \\ 0 & I \end{bmatrix}$ then

$$
X = V_{21} V_{11}^{-1}
$$

solves (2.48) with $X = X^T \geq 0$;
2) The generalized eigenvalues of $\lambda L_{11} - M_{11}$ are the closed-loop eigenvalues under the optimal feedback

$$u^0_k = -(R + B^T X B)^{-1} (S^T + B^T X A) x_k$$

Proof: [12]

Note that in this discrete-time problem, neither the solution of the GARE nor the optimal feedback depends explicitly on $R^{-1}$ as in the continuous-time problem. The special case $R = 0$ is called "deadbeat control" and is discussed for the case $S = 0, E = I$, utilizing the above methods for solution in [24].
Numerical implementation of the GARE solution methods of Chapter 2 is relatively straightforward. Proven stable algorithms exist that are coded into reliable FORTRAN software (see Chapter 5). However, the GARE may be ill-conditioned, and the resulting numerical solution may not be as accurate as desired. This chapter presents an iterative refinement procedure utilizing Newton's method. This procedure is presented for the continuous- and discrete-time problems. The continuous-time method is based on Kleinman [13], and the discrete-time method is based on Hewer [14].

3.1 Iterative Refinement of Continuous-Time Solution

The following result is due to Kleinman [13] for the controllable case, which was extended to the stabilizable case by Sandell [25]. This result is for the ARE and is presented here in detail, although from a slightly different approach, to provide the groundwork for extension to the GARE.

Given the system (2.18) and criterion (2.19) with $E=I$, $Q=I$ and $S=0$, the resulting ARE from (2.27) is

$$A^TX+XA-XBR^{-1}B^TX + C^TC = 0.$$ (3.1)

If the pair $[A,B]$ is stabilizable and the pair $[C,A]$ is detectable, then there exists a unique solution $X=X^T > 0$ to (3.1), such that the linear feedback
\[ u^0 = -R^{-1}B^T x \quad := -K x \quad \] (3.2)

stabilizes the closed-loop system and minimizes the criterion (2.19).

Furthermore, it can be shown that
\[ J(x_0; u^0) = \min_{u} J(x_0; u) = x_0^T x_0. \] (3.3)

To derive the Newton iterative procedure, consider at the \( k \)-th iteration that the solution \( X \) of (3.1) is of the form
\[ X = X_k + \delta X, \] (3.4)
where \( \delta X \) is "small", i.e., can neglect second-order terms in \( \delta X \).

Substituting (3.4) into (3.1), we have
\[ 0 = A^T X_k + X_k A - X_k B R^{-1} B^T X_k + C^T C + \delta X (A - B R^{-1} B^T X_k) + (A - B R^{-1} B^T X_k)^T \delta X. \] (3.5)

Denote the solution of (3.5) at the \( k \)-th iteration by \( \delta X_k \), and let
\[ \delta X_k = X_{k+1} - X_k. \] (3.6)

Substituting (3.6) into (3.5) and simplifying, we have
\[ 0 = (A - B K_k + 1)^T X_{k+1} + X_{k+1} (A - B K_k + 1) + C^T C + K_{k+1}^T R K_{k+1}, \] (3.7)
where
\[ K_{k+1} = R^{-1} B^T X_k, \] (3.8)
and \( A - B K_{k+1} \) is the closed-loop-system matrix.

Equation (3.7) is a Lyapunov equation in the unknown \( X_{k+1} \) and its unique non-negative definite solution is given by
\[ X_{k+1} = \int_0^\infty e^{(A-BK_{k+1})t} \left( C^T + K_{k+1}^T R K_{k+1} \right) e^{(A-BK_{k+1})t} dt \] (3.9)

which is finite, if and only if, the closed-loop-system matrix is stable (i.e., has eigenvalues with negative real parts). It can be shown that

\[ X_1 - X_2 = \int_0^\infty e^{(A-BK_2)t} \left[ (K_1 - K_2) T (R(K_1 - K_2) - (K_1 - K_2)) (B^T X_1 - R K_2) \right. \\
- \left. (B^T X_1 - R K_2) (K_1 - K_2) \right] e^{(A-BK_2)t} dt \] (3.10)

or, alternatively,

\[ X_1 - X_2 = \int_0^\infty e^{(A-BK_1)t} \left[ (K_1 - K_2) T (R(K_1 - K_2) - (K_1 - K_2)) (B^T X_2 - R K_2) \right. \\
- \left. (B^T X_2 - R K_2) (K_1 - K_2) \right] e^{(A-BK_1)t} dt \] (3.11)

We can now state and prove the main result of Kleinman [13] as extended by Sandell [25].

**Theorem 3.1:** Let \( X_k, k=0,1,\ldots \), be the unique non-negative definite solution of the linear algebraic equation

\[ 0 = (A-BK_k)^T X_k + X_k (A-BK_k) + C^T C + K_k^T R K_k \] (3.12)

where, recursively,

\[ K_k = R^{-1} B^T X_{k-1}, \quad k = 1,2,\ldots, \] (3.13)

and \( K_0 \) is chosen such that the matrix \( (A-BK_0) \) is stable. Then,

1) \( 0 \leq X \leq X_{k+1} \leq X_k \leq \cdots \leq X_0 \)

2) \( \lim_{k \to \infty} X_k = X \)

3) in the vicinity of \( X \), \( \|X_{k+1} - X\| \leq C_2 \|X_k - X\|^2 \),

where \( C_2 \) is a finite constant.
Proof: 1) Let \( X_0 \) satisfy (3.12) for the chosen \( K_0 \). Now set
\[
K_1 = R^{-1}B^T X_0
\]
and let \( X_1 \) be the associated solution to (3.9).

Using (3.10), we obtain
\[
X_0 - X_1 = \int_0^\infty e^{(A-BK_1)^T t}(K_0 - K_1)^T R(K_0 - K_1)e^{(A-BK_1)t} dt \geq 0,
\]
so that \( X_1 \leq X_0 \). In addition, we have by (3.11)
\[
X_1 - X = \int_0^\infty e^{(A-BK_1)^T t}(K_1 - K)^T R(K_1 - K)e^{(A-BK_1)t} dt \geq 0.
\]
Hence, \( X_1 \) is bounded above and below and, therefore, has finite norm.

Thus, \( (A-BK_1) \) is stable so \( X_1 \) satisfies (3.12) with \( k=1 \). Repeating the above argument for \( k=2,3,... \) yields the desired result.

2) Taking the limit of (3.12) as \( k \to \infty \), we obtain
\[
0 = A^T X_\infty + X_\infty A - X_\infty B R^{-1} B^T X_\infty + C^T C. \tag{3.14}
\]
Since \( X \) is the unique non-negative definite solution of (3.14), \( X_\infty = X \).

3) Set \( K_1 = R^{-1}B^T X_k \) and \( K_2 = R^{-1}B^T X \) in (3.11) and take the norm to find
\[
\|X_{k+1} - X\| \leq \int_0^\infty \|e^{(A-BK_{k+1})^T t}\|_2^2 dt \|BR^{-1}B^T\| \|X_k - X\|_2. \tag{3.15}
\]
Since \( (A-BK_{k+1}) \) is stable, it can be shown that, uniformly in \( k \),
\[
\int_0^\infty \|e^{(A-BK_{k+1})^T t}\|_2^2 dt \leq \text{constant} = C_1
\]

Let \( C_2 = C_1 \|BR^{-1}B^T\|_2 \), and the proof is complete.
It should be noted that another iterative scheme is possible and has been used in practice [26]. One simply solves (3.5) at iteration $k$ for $\delta X_k$ and then takes

$$X_{k+1} = X_k + \delta X_k \quad (3.16)$$

It has been shown [27] that

$$\lim_{k \to \infty} \delta X_k = 0 \quad (3.17)$$

and

$$\lim_{k \to \infty} X_k = X \quad (3.18)$$

This procedure, while theoretically equivalent to Theorem 3.1, has some computational drawbacks. Namely, it requires more machine operations and if the initial guess for $X_0$ is nonsymmetric, then $X_{k+1}$ will be nonsymmetric at each iteration.

Before we extend Theorem 3.1 to the general case, we first want to show that solving the GARE (2.27), which results from the generalized state space system (2.18) with criterion (2.19) and then applying the optimal feedback (2.31), is equivalent to solving a "standard" regulator problem when $E$ is invertible.

We will always assume that $E$ is invertible since when $E$ is singular the GARE (2.27) does not, in general, have a solution. However, even though $E$ is assumed nonsingular, it is computationally undesirable in many cases to invert $E$ and solve the equivalent standard problem. A method for solving the generalized regulator problem when $E$ is singular has been proposed by Cobb [28] and involves decomposing the problem into a standard part and a nonsingular $E$ part.
Theorem 3.2: Solving the generalized regulator problem of section 2.3.1 is equivalent to solving the following "standard" problem

\[ \dot{x} = \bar{A}x + \bar{B}u \]
\[ y = Cx, \quad (3.19) \]

with criterion

\[ J = \frac{1}{2} \int_0^\infty (y^TQy + 2x^TQ \dot{x} + u^TRu) \, dt \quad (3.20) \]

where

\[ \bar{A} = E^{-1}A \]
\[ \bar{B} = E^{-1}B. \quad (3.21) \]

Proof: The ARE corresponding to this problem is

\[ 0 = (\bar{A} - \bar{B}R^{-1}S^T) \bar{X} + \bar{X}(\bar{A} - \bar{B}R^{-1}S^T)^T - \bar{X}BR^{-1}B^T \bar{X} + C^TQC - SR^{-1}S^T \]
\[ = (A - BR^{-1}S^T)E^{-T}X + XE^{-1}A - BR^{-1}B^T \bar{X} \]
\[ + C^TQC - SR^{-1}S^T. \quad (3.22) \]

The optimal control law is

\[ u^* = -R^{-1}(\bar{B}^T \bar{X} + S^T)x \]
\[ = -R^{-1}(B^T E^{-T} \bar{X} + S^T)x, \quad (3.23) \]

which results in the closed-loop system

\[ \dot{x} = E^{-1}(A - BR^{-1}S^T - BR^{-1}B^T E^{-T} \bar{X})x. \quad (3.24) \]

Now, if we note that

\[ \bar{X} = E^T XE \quad (3.25) \]

then (3.22) is equivalent to (2.27), and (3.24) is equivalent to (2.18) with the feedback (2.31), that is,
\[ E^*_k = (A-BR^{-1}S^T - BR^{-1}B^T X_k) X_k. \] (3.26)

Now we will extend Theorem 3.1 to the general case.

**Theorem 3.3:** Let \( X_k, k=0,1, \ldots \) be the unique non-negative definite solution of the linear algebraic equation

\[ 0 = (A-BK_k)^T X_k E + E^T X_k (A-BK_k) + C^T QC + K_k^T R_k - S K_k - (SK_k)^T \] (3.27)

where, recursively,

\[ K_k = R^{-1}(S^T X_{k-1} E + S^T), \quad k=1,2, \ldots \] (3.28)

and where \( K_0 \) is chosen such that the matrix \( E^{-1}(A-BK_0) \) is stable. Then

1) \( 0 \leq X \leq X_{k+1} \leq X_k \leq \ldots \leq X_0 \)

2) \( \lim_{k \to \infty} X_k = \lambda \)

3) in the vicinity of \( X \), \( ||X_{k+1} - X|| \leq C_2 ||X_k - X||^2 \)

where \( X \) solves the GARE (2.27), and \( C_2 \) is a finite constant.

**Proof:** 1) We employ the results of Theorem 3.2 to convert the problem: from (3.25) let

\[ X_k = E^{-1} X_{k+1} \] (3.29)

and substitute in (3.27) and (3.28) to obtain

\[ 0 = [E^{-1}(A-B\overline{K}_k)]^T \overline{X}_k + \overline{X}_k E^{-1}(A-B\overline{K}_k) + C^T QC + \overline{K}_k^T R_k - S \overline{K}_k - (SK_k)^T, \] (3.30)

where

\[ \overline{K}_k = R^{-1}[(E^{-1}B)^T X_{k-1} + S^T]. \] (3.31)
Then

\[
\bar{x}_k = \int_0^\infty e^{-\frac{1}{2}(A-B\bar{k})t} \left[ C^TQC + \bar{k}_k^T R \bar{k}_k - S \bar{k}_k - (S \bar{k}_k)^T \right] e^{-\frac{1}{2}(A-B\bar{k})t} dt
\]

when \( E^{-1}(A-B\bar{k}) \) is stable. It can be shown that

\[
\bar{x}_1 - \bar{x}_2 = \int_0^\infty e^{-\frac{1}{2}(A-B\bar{k})t} \left[ (\bar{k}_1 - \bar{k}_2)^T R (\bar{k}_1 - \bar{k}_2) \ight.

- (\bar{k}_1 - \bar{k}_2)^T (E^{-1}B)^T \bar{x}_1 - R \bar{k}_2 + S^T

- \left. (E^{-1}B)^T \bar{x}_2 - R \bar{k}_2 + S^T \right] (\bar{k}_1 - \bar{k}_2) e^{-\frac{1}{2}(A-B\bar{k})t} dt
\]

or, alternatively,

\[
\bar{x}_1 - \bar{x}_2 = \int_0^\infty e^{-\frac{1}{2}(A-B\bar{k}_1)t} \left[ (\bar{k}_1 - \bar{k}_2)^T R (\bar{k}_1 - \bar{k}_2) \ight.

- (\bar{k}_1 - \bar{k}_2)^T (E^{-1}B)^T \bar{x}_1 - R \bar{k}_2 + S^T

- \left. (E^{-1}B)^T \bar{x}_2 - R \bar{k}_2 + S^T \right] (\bar{k}_1 - \bar{k}_2) e^{-\frac{1}{2}(A-B\bar{k}_1)t} dt
\]

Now let \( \bar{x}_0 \) satisfy (3.30) for a \( \bar{k}_0 \) chosen such that \( E^{-1}(A-B\bar{k}_0) \) is stable. We remark here that by Theorem 3.2, this is equivalent to stabilizing \( E^{-1}(A-B\bar{k}_0) \). Set \( \bar{k}_1 = R^{-1} ((E^{-1}B)^T \bar{x}_0 + S^T) \), and let \( \bar{x}_1 \) be the associated solution to (3.32). Using (3.33), we obtain

\[
\bar{x}_0 - \bar{x}_1 = \int_0^\infty e^{-\frac{1}{2}(A-B\bar{k}_1)t} \left[ (\bar{k}_0 - \bar{k}_1)^T R (\bar{k}_0 - \bar{k}_1) \right] e^{-\frac{1}{2}(A-B\bar{k}_1)t} dt > 0
\]

so that \( \bar{x}_1 \leq \bar{x}_0 \). In addition, we have by (3.34)
\[ \bar{x}_1 - \bar{x} = \int_0^\infty e^{(A-B\bar{K}_1)}T \{ (\bar{K}_1 - \bar{K})\bar{R}(\bar{K}_1 - \bar{K}) \} e^{-1}(A-B\bar{K}_1)t \, dt \geq 0. \] 

(3.36)

Hence, \( \bar{x}_1 \) is bounded above and below and, therefore, has finite norm. Thus, \( E^{-1}(A-B\bar{K}_1) \) is stable so \( \bar{x}_1 \) satisfies (3.30) with \( k=1 \). Repeating the above argument for \( k=2,3,\ldots \) yields

\[ 0 < \bar{x} < \bar{x}_2 < \bar{x}_3 < \ldots < \bar{x}_0 \]

Since by (3.29) \( E \) is a congruence transformation on \( X \) to yield \( \bar{x} \), the desired result is implied.

2) Taking the limit of (3.27) as \( k=\infty \), we obtain after some manipulation

\[ 0 = (A-BR^{-1}S^T)^T \bar{x}_\infty + E^T \bar{x}_\infty (A-BR^{-1}S^T) - E^T X BR^{-1}S^T \bar{x}_\infty E^T \]

\[ + C^T QC = SR^{-1}S^T. \] 

(3.37)

Since \( X \) is the unique non-negative definite solution to (3.37), \( \bar{x}_\infty = X \).

3) Set \( \bar{x}_1 = R^{-1} \{ (E^{-1}B)^T \bar{x}_k + S^T \} \) and \( \bar{x}_2 = R^{-1} \{ (E^{-1}B)^T \bar{x} + S^T \} \) in (3.34) to find

\[ \bar{x}_{k+1} - \bar{x} = \int_0^\infty e^{(A-B\bar{K}_k+1)}t \{ (\bar{K}_k - \bar{K})e^{-1}BR^{-1}S^T \bar{x}_\infty \} dt. \] 

(3.38)

Substitute (3.25) into (3.38) to obtain
\[ E^T(X_{k+1} - X)E = \int_0^\infty [E^{-1}(A-Bk_{k+1})]^T e^T(X_{k+1} - X)BR^{-1}B^T(X_{k+1} - X)E^{-1}(A-Bk_{k+1}) dt. \]

(3.39)

Pre-multiply (3.39) by \( E^{-T} \), post-multiply by \( E^{-1} \), and take the norm to find

\[ \| X_{k+1} - X \| \leq \int_0^\infty \| E^{-1}(A-Bk_{k+1})t \|_2^2 dt \| E^{-1} \|_2^2 \| E^{T} \|_2 \| X_{k} - X \|_2. \]

(3.40)

It can be shown that, uniformly in \( k \),

\[ \int_0^\infty \| E^{-1}(A-Bk_{k+1})t \|_2^2 dt \leq \text{constant} = C_1. \]

Let \( C_2 = C_1 x_2(E) \| B \|_2 \| B^T \|_1 \), and the proof is complete.

We remark that the proof to part 3 indicates that the square of the condition of \( E \) with respect to inversion, \( x^2(E) \), influences the convergence rate of the algorithm multiplicatively.

### 3.2 Iterative Refinement of Discrete-Time Solution

An iterative scheme is possible for the discrete-time formulation, and it was first reported by Hewer [14]. The result is for the discrete-time ARE, and the proof given here is similar to the continuous-time case.

Given the system (2.40) with criterion (2.41) with \( E=I, Q=I, \) and \( S=0 \), the resulting ARE from (2.48) is

\[ X = A^T X A - A^T X B (R+B^T X B)^{-1} B^T X A + C^T C. \]

(3.41)

If the pair \([A,B]\) is stabilizable and the pair \([C,A]\) is detectable, then there exists a unique solution \( X = X^T \geq 0 \) to (3.41) such that the linear feedback
\[ u_k^0 = -(R+B^T XB)^{-1}B^T X_{k} = -K_k \] (3.42)

stabilizes the closed-loop system and minimizes the criterion (2.41).

One can follow a first-order perturbation-type derivation to arrive at the Lyapunov equation that forms the basis of the following theorem in a manner similar to that of the previous section for the continuous-time problem. The following result is due to Hewer [14].

**Theorem 3.4:** Let \( X_k, k=0,1,..., \) be the unique non-negative definite solution of the linear algebraic equation

\[ X_k = (A-BK_k)^T X_k (A-BK_k) + K_k^T R K_k + C^T C, \] (3.43)

where, recursively,

\[ K_k = (R+B^T X_{k-1} B)^{-1}B^T X_{k-1} A, k=1,2,..., \] (3.44)

and \( K_0 \) is chosen such that the closed-loop-system matrix \( (A-BK_0) \) is stable (i.e., has eigenvalues whose magnitudes are less than unity).

Then

1) \( 0 \leq X \leq X_{k+1} \leq X_k \leq \ldots \leq X_0 \)

2) \( \lim_{k \to \infty} X_k = X \)

3) in the vicinity of \( X, \| X_{k+1} - X \| \leq C_2 \| X_k - X \|^2. \)

where \( C_2 \) is a finite constant.

**Proof:**

1) If \( (A-BK_k) \) is stable, the unique non-negative definite solution \( X_k \) of (3.43) may be written as
\[ X_k = \frac{1}{N} \sum_{n=0}^{N} [(A-Bk_k)^T]^{n} (K_k R_k + C^T C) (A-Bk_k)^N. \] (3.45)

It can be shown that for \((A-Bk_1)\) and \((A-Bk_2)\) stable then

\[ X_1 - X_2 = (A-Bk_1)^T (X_1 - X_2) (A-Bk_1) + (K_1 - K_2)^T (R+B^T X_2) (K_1 - K_2) \]
\[ + [(R+B^T X_2)(K_2 - B^T X_2 A)]^T (K_1 - K_2) + (K_1 - K_2)^T [ (R+B^T X_2) K_2 \]
\[ - B^T X_2 A] \]
\[ = \frac{1}{N} \sum_{n=0}^{N} [(A-Bk_1)^T]^{n} [(K_1 - K_2)^T (R+B^T X_2) (K_1 - K_2) + [ (R+B^T X_2) K_2 \]
\[ - B^T X_2 A] + (K_1 - K_2)^T [ (R+B^T X_2) K_2 - B^T X_2 A)] (A-Bk_1)^N. \] (3.46)

or, alternatively,

\[ X_1 - X_2 = (A-Bk_2)^T (X_1 - X_2) (A-Bk_2) + (K_1 - K_2)^T (R+B^T X_2) (K_1 - K_2) \]
\[ + [(R+B^T X_2)(K_2 - B^T X_2 A)]^T (K_1 - K_2) + (K_1 - K_2)^T [ (R+B^T X_2) K_2 \]
\[ - B^T X_2 A] \]
\[ = \frac{1}{N} \sum_{n=0}^{N} [(A-Bk_2)^T]^{n} [(K_1 - K_2)^T (R+B^T X_2) (K_1 - K_2) + [ (R+B^T X_2) K_2 \]
\[ - B^T X_2 A] + (K_1 - K_2)^T [ (R+B^T X_2) K_2 - B^T X_2 A)] \]
\[ = (A-Bk_2)^N. \] (3.47)

Let \( X_0 \) satisfy (3.43) for the chosen \( k_0 \). Now set \( k_1 = (R+B^T X_0) C \)

\( B^T X_0 A \) and let \( X_1 \) be the associated solution to (3.43). Using (3.49) we obtain

\[ X_0 - X_1 = \frac{1}{N} \sum_{n=0}^{N} [(A-Bk_1)^T]^{n} [(K_0 - K_1)^T (R+B^T X_0) (K_0 - K_1)] (A-Bk_1)^N > 0, \]

so that \( X_1 \leq X_0 \). In addition, we have by (3.47)

\[ X_1 - X = \frac{1}{N} \sum_{n=0}^{N} [(A-Bk_1)^T]^{n} [(K_1 - K)^T (R+B^T X B) (K_1 - K)] (A-Bk_1)^N > 0. \]
Hence, $X_1$ is bounded above and below and, therefore, has finite norm. Thus, $(A-BK_1)$ is stable so $X_1$ satisfies (3.43) with $k=1$. Repeating the above argument for $k=2,3,...$ yields the desired result.

2) Taking the limit of (3.43) as $k\to\infty$, we obtain

$$X_\infty = A^TX_\infty A - A^TX_\infty B(R+B^TX_\infty B)^{-1}B^TX_\infty A + C^TC. \tag{3.50}$$

Since $X$ is the unique non-negative definite solution of (3.50), $X_\infty = X$.

3) Set $K_1 = (R+B^TX_kB)^{-1}B^TX_kA$ and $K_2 = (R+B^TX_kB)^{-1}B^TXA$ in (3.47), and take the norm to find

$$\|X_{k+1} - X\| \leq \|X_k - X\|^2 + B(R+B^TX_kB)^{-1}B^T \sum_{n=0}^{N-1} \|(A-BK_{k+1})^{n+1}\|_I. \tag{3.51}$$

Since $(A-BK_{k+1})$ is stable, it can be shown that, uniformly in $k$,

$$\sum_{n=0}^{N-1} \|(A-BK_{k+1})^{n+1}\|_I \leq \text{constant} = C_1$$

Let $C_2 = C_1\|B(R+B^TX_kB)^{-1}B^T\|_I$, and the proof is complete.

It should be noted that another iterative scheme is possible here, as in the continuous case, where one solves for a correction term to the solution at each iteration. This procedure faces the same computational drawbacks as in the continuous case and, therefore, will not be discussed further.

Before we extend Theorem 3.4 to the general case, we first want to show that solving the GARE (2.48), which results from the generalized state space system (2.40) with criterion (2.41) and then applying the optimal feedback (2.49), is equivalent to solving a "standard" regulator problem when $E$ is invertible.
**Theorem 3.5:** Solving the generalized regulator problem of section 2.3.3 is equivalent to solving the following "standard" problem

\[ x_{k+1} = \tilde{A}x_k + \tilde{B}u_k \]
\[ y_k = Cx_k \]  \hspace{1cm} (3.51)

with criterion

\[ J = \frac{1}{2} \sum_{k=0}^{T} (y_k^TQy_k + 2x_k^Tu_k + u_k^Tu_k) \]  \hspace{1cm} (3.52)

where

\[ \tilde{A} = E^{-1}A \]
\[ \tilde{B} = E^{-1}B \]  \hspace{1cm} (3.53)

**Proof:** The ARE corresponding to this problem is

\[ \bar{X} = A^T\bar{X}A - (A^T\bar{X}B + \bar{S})(R + B^T\bar{X}B)^{-1}(A^T\bar{X}B + \bar{S})^T + C^TQC \]
\[ = A^TTE^{-T}X^{-1}A - (AE^{-T}X^{-1}B+S)(R+BT_{E^{-T}X^{-1}B+S})^{-1}(A^T_{E^{-T}X^{-1}B+S})^T \]
\[ + C^TQC \]  \hspace{1cm} (3.54)

The optimal control law is

\[ u_k^0 = -(R+BT_{E^{-T}X^{-1}B+S})^{-1}(A_{E^{-T}X^{-1}B+S})^Tx_k \]
\[ = -(R+BT_{E^{-T}X^{-1}B+S})^{-1}(A_{E^{-T}X^{-1}B+S})^Tx_k \]  \hspace{1cm} (3.55)

which results in the closed-loop system

\[ x_{k+1} = E^{-1}[A-B(R+BT_{E^{-T}X^{-1}B+S})^{-1}(A_{E^{-T}X^{-1}B+S})^T]x_k \]  \hspace{1cm} (3.56)

Now, if we note that

\[ \bar{X} = E^TXE \]  \hspace{1cm} (3.57)
then (3.54) is equivalent to (2.48), and (3.56) is equivalent to (2.40) with the feedback (2.49); that is,

\[ E_{k+1} = [A-B(R+B^T XB)^{-1}(A^T XB+S)^T]x_k. \]  \hfill (3.58)

Now we will extend Theorem 3.4 to the general case.

**Theorem 3.6:** Let \( X_k, k=0,1,\ldots \) be the unique non-negative definite solution of the linear algebraic equation

\[ E^T X_k E = (A-BK_k)^T X_k (A-BK_k) + K_k^T R K_k + C^T Q C \]

\[ - SK_k - (S K_k)^T \]  \hfill (3.59)

where, recursively,

\[ K_k = (R+B^TX_{k-1}B)^{-1}(B^TX_{k-1}A+S^T), \quad k=1,2,\ldots, \] \hfill (3.60)

and \( K_0 \) is chosen such that the closed loop system matrix \( E^{-1}(A-BK_0) \) is stable. Then

1) \( 0 \leq X \leq X_{k+1} \leq X_k \leq \cdots \leq X_0 \)

2) \( \lim_{k \to \infty} X_k = X \)

3) in the vicinity of \( X \),

\[ \| X_{k+1} - X \| \leq C_2 \| X_k - X \|^2 \]

where \( X \) solves the GARE (2.48), and \( C_2 \) is a finite constant.

**Proof:** 1) We employ the results of Theorem 3.5 to convert the problem; from (3.57) let

\[ X_k = E^{-T} X_k E^{-1} \]  \hfill (3.61)

and substitute in (3.59) and (3.60) to obtain
\[ \bar{x}_k = [E^{-1}(A-B\bar{K}_k)]^T \bar{x}_k E^{-1}(A-B\bar{K}_k) + \bar{K}_k^T R \bar{K}_k + C^T Q C \]

\[ - S \bar{K}_k - (S\bar{K}_k)^T \]  \hspace{1cm} (3.62)

where

\[ \bar{K}_k = (R+B E^{-T} \bar{x}_{k-1} E^{-1} B)^{-1} (B E^{-T} \bar{x}_{k-1} E^{-1} A + S^T) . \]  \hspace{1cm} (3.63)

Then

\[ \bar{x}_k = \sum_{N=0}^{\infty} \left[ (E^{-1}(A-B\bar{K}_k))^T \right]^N \left[ \bar{K}_k^T R \bar{K}_k + C^T Q C - S \bar{K}_k - (S\bar{K}_k)^T \right] [E^{-1}(A-B\bar{K}_k)]^N \]  \hspace{1cm} (3.64)

when \( E^{-1}(A-B\bar{K}_k) \) is stable. It can be shown that

\[ \bar{x}_1 - \bar{x}_2 = \sum_{N=0}^{\infty} \left[ (E^{-1}(A-B\bar{K}_1))^T \right]^N \left( (\bar{K}_1 - \bar{K}_2)^T (R+B E^{-T} \bar{x}_2 E^{-1} B) (\bar{K}_1 - \bar{K}_2) \right. \]
\[ + \left( (R+B E^{-T} \bar{x}_2 E^{-1} B) \bar{K}_2 - B E^{-T} \bar{x}_2 E^{-1} A - S^T \right)^T (\bar{K}_1 - \bar{K}_2) \]
\[ + (\bar{K}_1 - \bar{K}_2)^T \left[ (R+B E^{-T} \bar{x}_2 E^{-1} B) \bar{K}_2 - B E^{-T} \bar{x}_2 E^{-1} A - S^T \right] \]
\[ \left. [E^{-1}(A-B\bar{K}_1)]^N \right. \]  \hspace{1cm} (3.65)

or, alternatively

\[ \bar{x}_1 - \bar{x}_2 = \sum_{N=0}^{\infty} \left[ (E^{-1}(A-B\bar{K}_2))^T \right]^N \left( (\bar{K}_1 - \bar{K}_2)^T (R+B E^{-T} \bar{x}_1 E^{-1} B) (\bar{K}_1 - \bar{K}_2) \right. \]
\[ + \left( (R+B E^{-T} \bar{x}_1 E^{-1} B) \bar{K}_2 - B E^{-T} \bar{x}_1 E^{-1} A - S^T \right)^T (\bar{K}_1 - \bar{K}_2) \]
\[ + (\bar{K}_1 - \bar{K}_2)^T \left[ (R+B E^{-T} \bar{x}_1 E^{-1} B) \bar{K}_2 - B E^{-T} \bar{x}_1 E^{-1} A - S^T \right] \]
\[ \left. [E^{-1}(A-B\bar{K}_2)]^N \right. . \]  \hspace{1cm} (3.66)
Now let \( \bar{X}_0 \) satisfy (3.62) for a \( \bar{K}_0 \) chosen such that \( E^{-1}(A - BK_0) \) is stable. Let \( \bar{X}_1 \) be the associated solution to (3.64). Using (3.66) we obtain
\[
\bar{X}_0 - \bar{X}_1 = \sum_{n=0}^{\infty} \left[ (E^{-1}(A-B\bar{K}_1))^T \right]^N \left[ (\bar{K}_0 - \bar{K}_1)^T (R+BE^{-1}E+B)(\bar{K}_1 - \bar{K}_2) \right]
\]
so that \( \bar{X}_1 \leq \bar{X}_0 \). In addition, we have by (3.65)
\[
\bar{X}_1 - \bar{X} = \sum_{n=0}^{\infty} \left[ (E^{-1}(A-B\bar{K}_1))^T \right]^N \left[ (\bar{K}_1 - \bar{K})^T (R+BE^{-1}E+B)(\bar{K}_1 - \bar{K}) \right]
\]
\[
\left[ E^{-1}(A-B\bar{K}_1) \right]^N \geq 0.
\]
Hence, \( \bar{X}_1 \) is bounded above and below and, therefore, has finite norm.

Thus, \( E^{-1}(A-B\bar{K}_1) \) is stable so \( \bar{X}_1 \) satisfies (3.62) with \( k=1 \). Repeating the above argument for \( k=2, 3, \ldots \) yields
\[
0 < \bar{X} < \bar{X}_{k+1} < \bar{X}_k < \ldots < \bar{X}_0.
\]

Since by (3.61) \( E \) is a congruence transformation on \( X \) to yield \( \bar{X} \), the desired result is implied.

2) Taking the limit of (3.59) as \( k \rightarrow \infty \) we obtain
\[
E^TX_\infty E = A^TX_\infty A -(A^TX_\infty B+S)(R+BE^{-1}E+B)^{-1}(A^TX_\infty B+S)^T + C^TQC. \quad (3.67)
\]
Since \( X \) is the unique non-negative definite solution to (3.67), \( X_\infty = X \).

3) It is easily shown from (3.65) that
\[ E^T(x_1 - x_2)E = \sum_{n=0}^{N} [(A - BK_1)^T] \sum_{(K_1 - K_2)^T(R+BTx_2B)(K_1 - K_2)}^N \]

\[ + \sum_{(K_1 - K_2)^T((R+BTx_2B)K_2 - BTx_2A-S^T)}^N [(A-BK_1)^N]. \]

(3.68)

Set \[ K_1 = (R+BTx_2B)^{-1}(B^T(B+ST)^{-1}) \text{ and } K_2 = (R+BTXB)^{-1}(B^T(B+ST)^{-1}) \]

in (3.68), and take norms to find

\[ \|x_{k+1} - x_1\| \geq \|x_k - x_1\| \sum_{n=0}^{N} \|A-BK_{k+1}\|^{N+1}. \]

Since \((A-BK_{k+1})\) is stable, it can be shown that, uniformly in \(k\),

\[ \sum_{n=0}^{N} \|A-BK_{k+1}\|^{N+1} \leq \text{constant} = C_1. \]

Let \(C_2 = C_1 \sum_{n=0}^{N} \|B(R+BTXB)^{-1}B^T\| \), and the proof is complete.
CHAPTER 4
CONDITIONING OF THE ALGEBRAIC RICCATI PROBLEM

As stated in Chapter 2, it is desirable to associate a measure with a computing problem which reflects the overall sensitivity of the solution to changes in the data. It is the intent of this chapter to derive such a measure (i.e., condition number) for the problem of computing a solution to the GARE. To be most useful, the condition number must be easily computable. It should not require additional computations on the same order of computing the solution itself.

We will first state the desired form for the condition number and review previous work on condition estimates for the Riccati problem. Then first order perturbation analysis will be employed to derive new condition estimates for the continuous- and discrete-time Riccati equations. Ways of employing balancing to improve the numerical accuracy of the calculated solution will be discussed. An efficient method for incorporating a change of model coordinates is presented.

4.1 Previous Work

We seek a relative condition estimate for the Riccati problem in the following form:

\[
\frac{|X - \bar{X}|}{|X|} = C \cdot \kappa(X) \cdot \epsilon_m,
\]

where \( X \) is the exact solution to the GARE, \( \bar{X} \) is the computed solution, \( C \) is a constant possibly depending on the size of the problem, \( \kappa(X) \) is the desired condition number, and \( \epsilon_m \) is the machine epsilon (precision). Machine epsilon is defined as the smallest positive
number that can be added to 1.0 such that the machine recognizes that the sum is different from 1.0. In this form, if $\kappa(X)$ is $O(10^N)$, then the computed solution can be expected to differ from the true solution by $N$ significant digits.

We should note here that this type of indicator of solution accuracy is desired because the more traditional method of examining the size of the residual is not always reliable. The residual is the remainder quantity obtained when the computed solution is substituted in the original problem. That is, for the GARE (2.30) the residual is

$$\text{residual} := (A-BR^{-1}ST)T + E^T(A-BR^{-1}ST) - E^TBR^{-1}ET + CTQC - SR^{-1}ST.$$  

(4.2)

Stewart [15] explores in detail the behavior of the residual in the linear equations problem $Ax=b$ and cites examples where the residual $b - Ax$ is not a reliable indicator of solution ($\bar{x}$) accuracy. There is no reason to expect that the more complex problem of the GARE produces residuals that are better behaved, even though counter-examples are more difficult to construct.

We look for $\kappa(X)$ of the form:

$$\kappa(X) = f(A,B,C,E,Q,R,S).$$  

(4.3)

That is, $\kappa(X)$ should be a function of the matrices involved in the GARE. We want the norms involved in (4.1) or in the computation of $\kappa(X)$ to be the 1, $\infty$ or Frobenius norm versus the 2 norm to minimize the number of calculations involved.
There have been several attempts to derive bounds for the solution to the ARE \([29]-[34]\), none of which are of the form \((4.1)\). The following result by Byers \([20]\) is for the continuous-time ARE \((2.27)\):

\[
\kappa_B(X) = \frac{1C^TQC1 + 2IA1XI + IBR^{-1}B^TIXI^2}{IXI \text{SEP}[A^T_C, -A_C]}
\]  

\((4.4)\)

where

\[
A_C = A - BR^{-1}B^TX
\]  

\((4.5)\)

\[
\text{SEP}(F,G) := \inf_{IPI = 1} |IPF - GPI|
\]  

\((4.6)\)

Byers arrives at this bound via a first-order-perturbation approximation to \((2.27)\). A notable feature of this condition estimate is that it includes the effect of the separation of the closed-loop spectrum \((4.6)\) on the condition of the problem. For a detailed definition of \(\text{SEP}(F,G)\) and a discussion of its properties see Stewart \([35],[36]\). Some speculation \([6],[20]\) and empirical results suggest that a small separation of the closed-loop spectrum causes loss of numerical accuracy in the computed solution (see Chapter 5). A potential drawback to Byers' result \((4.4)\) is the appearance of \(IXI^2\) in the numerator and \(IXI\) in the denominator. This would suggest ill-conditioning of the problem when \(IXI\) is large or small, which is not necessarily true.

Martensson \([5]\) showed that \(V_{11}\), which must be inverted to form the Riccati \((2.27)\) solution, is singular if the model is unstabilizable for the case when eigenvectors are used as the basis for the stable eigenspace in the solution process. Laub \([7]\), Pappas, et.al. \([8]\), and Emami-Naeini \([9]\) prove analogous results when Schur vectors are used as
the basis for the stable eigenspace. This fact and the fact that $V_{11}$ must be inverted (that is, a linear system of the form $XV_{11} = V_{21}$ must be solved for $X$) has prompted speculation [6], [7] that the condition of $V_{11}$ with respect to inversion, $\kappa(V_{11})$, might be a good indicator of the condition of the Riccati problem (2.27). Indeed, empirical results show that when $V_{11}$ is ill-conditioned with respect to inversion, say $\kappa(V_{11})$ is $\mathcal{O}(10^N)$; then, in general, $N$ digits of accuracy are lost in the solution $X$. Since for the GARE solution, inversion of $V_{11}$ involves the inversion of $E$; then $\kappa(V_{11})$ can be expected to reflect any ill-effects that near singularity of $E$ might have on the solution. However, examples do exist (see Chapter 5) where the Riccati problem is ill-conditioned, but this fact is not indicated by the magnitude of $\kappa(V_{11})$.

Another indicator of numerical accuracy in the solution of the ARE has been suggested by Paige and Van Loan [37]. Their reasoning is based on the following result:

**Theorem 4.1**: Let the unitary matrix $E \in \mathbb{C}^{n \times n}$ be partitioned in the form

$$W = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix}$$

where $W_{11} \in \mathbb{C}^{m \times m}$ with $m \leq \frac{n}{2}$. Then there are unitary matrices $U = \text{diag}(U_1, U_2)$ and $V = \text{diag}(V_1, V_2)$; $U_1, V_1 \in \mathbb{C}^{m \times m}$; $U_2, V_2 \in \mathbb{C}^{(n-m) \times (n-m)}$ such that

$$U^H W V = \begin{bmatrix} \Gamma & -\Delta & 0 \\ \Delta & \Gamma & 0 \\ 0 & 0 & \mathbb{I} \end{bmatrix},$$  \hspace{1cm} (4.7)
where
\[ \Gamma = \text{diag} \left( \gamma_1, \gamma_2, \ldots, \gamma_m \right) \geq 0 \]
\[ \Delta = \text{diag} \left( \delta_1, \delta_2, \ldots, \delta_m \right) \geq 0 \]
\[ \gamma_1 \leq \gamma_2 \leq \cdots \leq \gamma_m \leq 1 \quad . \quad (4.8) \]

**Proof:** [38]

The use of this theorem in [37] suggests that if we perform this special singular value decomposition on the orthogonal \( Z \) found in our reduction of the generalized eigenproblem (2.59) we have

**Theorem 4.2:** Given \( Z_{11}, Z_{21} \) as found in (2.59), (2.61)

1) Then there exists orthogonal \( U \) and \( V \in \mathbb{R}^{n \times n} \) such that
\[ U^T Z_{11} V = \Gamma = \text{diag} \left( \gamma_1, \gamma_2, \ldots, \gamma_n \right) ; \]
\[ U^T Z_{21} V = \Delta = \text{diag} \left( \delta_1, \delta_2, \ldots, \delta_n \right) ; \]
\[ \gamma_1^2 + \delta_1^2 = 1; \quad 0 \leq \gamma_1 \leq \gamma_2 \leq \cdots \leq \gamma_n \leq 1 \quad . \quad (4.9) \]

2) If \( \Gamma \) is nonsingular, \( X = U \text{diag}(\frac{\delta_1}{\gamma_1}, \ldots, \frac{\delta_n}{\gamma_n}) U^T \) solves the ARE (2.27).

**Proof:** [37]

The size of \( \gamma_1 \) is suggested in [37] as being an indicator of potential numerical difficulties in the solution of the ARE because
\[ 1/X_{12} = \left| \frac{\delta_1}{\gamma_1} \right| = (1-\gamma_1^2)^{1/2} / \gamma_1 = \cotan \gamma_1 \quad (4.10) \]
and if \( X \) is computed on a computer having a machine precision \( \varepsilon \), then rounding errors of order \( \varepsilon / \gamma_1 \) can be expected to contaminate the result. Therefore, \( 1/\gamma_1 \) may indicate the condition of the ARE.
We can relate $1/\gamma_1$ to $\kappa(Z_{11})$ ($\kappa(Z_{11}) = \kappa(V_{11})$ for the ARE (2.27)), which was suggested previously as a condition measure. Recall,

$$
\kappa(Z_{11}) = \|Z_{11}\|_2 \|Z_{11}^{-1}\|_2 \\
= \max(\sigma(Z_{11})) \frac{1}{\min(\sigma(Z_{11}))}
$$

(4.11)

where $\sigma(Z_{11})$ denotes the set of singular values of $Z_{11}$ [15]. Therefore,

$$
\kappa(Z_{11}) = \frac{\gamma_n}{\gamma_1} \leq \frac{1}{\gamma_1} .
$$

(4.12)

So we see the two estimates are of the same order unless all of the singular values of $Z_{11}$ are small.

4.2 First-Order-Perturbation Analysis of the GARE

To facilitate our analysis of the continuous-time Riccati problem in this section, we start with the GARE (2.30) and make the following definitions:

$$
D := A - BR^{-1}S^T \\
G := BR^{-1}B^T \\
H := C^TQC - SR^{-1}S^T .
$$

(4.12)

The GARE can now be written as

$$
0 = D^T\dot{X} + E^T\dot{X} - E^T X G X E + H =: F_c(X) .
$$

(4.13)

Suppose that there is a small perturbation in $F_c(X)$, say $\delta F_c$; we would like to know the effect of this perturbation on the solution $X$.

**Theorem 4.3:** For consistent matrix norms, and given a small perturbation $\delta F_c$ to (4.13), we have for the resultant change in the solution $\delta X$
\[ 16X_1 \geq \frac{16F_c I}{I^T 11A_c I + 1111A_c I} \quad (4.14) \]

and

\[ 16X_1 \leq \frac{\kappa(E)\kappa(E^T)16F_c I}{I^T 1111E^T 1SEF[A_c E^{-1}, -E_c(A_c E^{-1})^T]} \quad (4.15) \]

where

\[ \kappa(E) := \|E^{-1}\| = \text{condition of } E \text{ with respect to inversion} \]

\[ A_c := D - GXE = \text{closed-loop-system matrix}. \]

**Proof:** Starting with (4.13), define

\[ F_c(X-\delta X) := F(X) + \delta F_c \quad (4.16) \]

\[ = D^T(X-\delta X)E + E^T(X-\delta X)D - E^T(X-\delta X)G(X-\delta X)E + H. \]

Neglecting second-order terms in \( \delta X \),

\[ F_c(X-\delta X) = D^TXE + E^TXD - E^T(X-\delta X)G(X-\delta X)E + H - (D-GXE)^T\deltaXE \]

\[ - E^T\deltaX(D-GXE) \]

\[ = F_c(X) - (D-GXE)^T\deltaXE - E^T\deltaX(D-GXE) \]

Therefore,

\[ -\delta F_c = A_c^T\deltaXE + E^T\deltaX A_c \quad (4.17) \]

Taking norms in (4.17), we have

\[ 16F_c I \leq 1A_c^T 115X_1 E_1 + 1E^T 116XI1A_c I \]

which yields (4.14). We pre-multiply (4.17) by \( E^{-T} \) and post-multiply by \( E^{-1} \) to obtain
\[-E^{-T} \delta F_c E^{-1} = (A_c E^{-1})^T \delta X + \delta X A_c E^{-1} \cdot\]

Taking norms we have

\[1E^{-T} \delta F_c E^{-1} I \leq \frac{IZA E^{-1} + (A_c E^{-1})^T T Z I}{IZI} 16X I\]

where \( Z = \frac{\delta X}{16X I} \). Therefore,

\[1E^{-T} 11E^{-1} 115F_c I \leq \inf_{I P I - 1} \frac{1P A_c E^{-1} + (A_c E^{-1})^T P I}{I P I} 16X I\]

\[= 16X I SEP[A_c E^{-1}, -(A_c E^{-1})^T]\]

which yields (4.15).

**Theorem 4.4:** Given the GARE (2.30), a relative condition number for the solution \( X \) is

\[ \kappa_A C(X) := \frac{\kappa(E)\kappa(E^T)I C^TQC - SR - I S^T I}{1X I 11E I E^T I SEP[A_c E^{-1}, -(A_c E^{-1})^T]} \]  

(4.18)

where \( A_c = (A-BR - S_B^T BR - XE) \).

**Proof:** To facilitate the analysis, we consider perturbations in \( F_c(X) \) on the order of a perturbation in the constant term, i.e.,

\[16F_c I \leq 1H I \cdot \varepsilon_m, \]  

(4.19)

so that from Theorem 4.3 we obtain

\[16X I \leq \frac{\kappa(E)\kappa(E^T)1 H I}{1X I 11E I E^T I SEP[A_c E^{-1}, -(A_c E^{-1})^T]} \cdot \varepsilon_m. \]  

(4.20)

From (4.20) we can make the following definition:
\[ \kappa_{c}(X) := \frac{\kappa(E)\kappa(E^{T})\text{HH}^{\text{T}}}{\text{IXIX} \text{EXE} \text{SEP}[A E^{-1},-(A E^{-1})^{T}]} \]  

:= relative condition number for the continuous-time GARE.

We remark here that the condition number of Theorem 4.4 has the advantage that it includes the effects of ill-conditioning of E with respect to inversion, the separation of the closed-loop spectrum and the ill-conditioning of R with respect to inversion when S is non-zero. It does have the disadvantage of having IXI in the denominator, as did Byers' result. Clearly, the form of the bound on the perturbation, \( \delta F_{c} \), dictates the form of the resulting condition number. One could consider perturbations to all matrices involved in the problem with a corresponding increase in the complexity of the resulting condition number. However, at this point it is not clear that this is necessary, and it is counterproductive to our goal of a simple, easily computable measure. The proper form of the bound on \( \delta F_{c} \) is a topic of continuing research.

We now turn our attention to the discrete-time Riccati problem. To facilitate our analysis, we start with the GARE (2.48) and make the following definition:

\[ F_{d}(X) := A^{T}XA - E^{T}XE - (A^{T}XB + S)(R + B^{T}XB)^{-1}(A^{T}XB + S)^{T} + C^{T}QC = 0 \]  

(4.22)
As in the continuous-time case, suppose there is a small perturbation in \( F_d(X) \), say \( \delta F_d \), we would like to know the effect of this perturbation on the solution \( X \).

**Theorem 4.5:** For consistent matrix norms, and given a small perturbation \( \delta F_d \) to (4.22), we have for the resultant change in the solution \( \delta X \)

\[
\| \delta X \| \geq \frac{\| \delta F_d \|}{\| E \| E^{-1} + \| A_d \| A_d^{-1} \|}
\]

and

\[
\| \delta X \| \leq \frac{\| \delta F_d \|}{\| E \| E^{-1} \| / \kappa(E) \kappa(E^T) - \| A_d \| A_d^{-1} \|}
\]

if

\[
\| E \| E^{-1} \| \geq \kappa(E) \kappa(E^T) \| A_d \| A_d^{-1} \|
\]

where

\[ \kappa(E) := \| E \| E^{-1} \| = \text{condition of } E \text{ with respect to inversion} \]

\[ A_d := (A - BK) = \text{closed-loop-system matrix} \]

\[ X := (R + BT)XB = (A_TXB + S)^T. \]

**Proof:** Starting with (4.22) define

\[
F_d(X + \delta X) := F_d(X) + \delta F_d
\]

\[
= A^T(X + \delta X)A - E^T(X + \delta X)E + C^TQC - [A^T(X + \delta X)B + S][R + B^T(X + \delta X)B]^{-1}[A^T(X + \delta X)B + S]^T.
\]

After expanding, neglecting second- or higher-order terms in \( \delta X \), and some algebraic manipulation we obtain
\[ F_d(x+\delta x) = F_d(x) - E^T \delta x E + (A-BK)^T \delta x (A-BK). \]

Therefore,
\[ \delta F_d = -E^T \delta x E + A_d^T \delta x A_d. \quad (4.26) \]

Taking norms in (4.26) we have
\[ \|\delta F_d\| \leq \|E^T\| \|\delta x\| E + \|A_d^T\| \|\delta x\| \|A_d\|. \]

which yields (4.23). We pre-multiply (4.26) by \( E^{-T} \), and post-multiply by \( E^{-1} \) to obtain
\[ E^{-T} \delta F_d E^{-1} = -\delta x + (A_d E^{-1})^T \delta x A_d E^{-1}. \]

or
\[ \delta x = (A_d E^{-1})^T \delta x A_d E^{-1} - E^{-T} \delta F_d E^{-1}. \]

Taking norms yields (4.24).

**Theorem 4.6:** Given the GARE (2.48), a relative condition number for the solution \( X \) is
\[ \kappa_{A_d}(X) := \frac{\|C^T QC\|^T}{\|X\| \|E\|^T \|\|E\|^{-T} \|A_d\| \|A_d\|^T \|E\|^{-T} \|X\|^T}. \quad (4.27) \]

where \( A_d = A - B(R+B^T XB)^{-1}(A^T XB+S)^T. \)

**Proof:** With reasoning similar to that for the continuous-time problem, we consider a perturbation in \( F_d(x) \) on the order of a perturbation in the constant term, i.e.,
\[ \|\delta F_d\| \leq \|C^T QC\| \cdot \epsilon_m \quad (4.28) \]
and the desired result follows from Theorem 4.5.

We remark here that when $E = I$, the requirement (4.25) becomes $1 > \|A_d^T \| A_d \|$. This is always true for "some" norm since $A_d$ is the closed-loop-system matrix and all of its eigenvalues have magnitude less than unity. Therefore, its spectral radius is less than one and there always exists a norm that is arbitrarily close to the spectral radius $\rho(A_d)$ ([15], Theorem 6.3.8). Thus, one could argue in this case that

$$\kappa_{A_d}(X) = \frac{I^T Q^T C^T}{I X (1 - p^2(A_d))}$$

and the term $1 - p^2(A_d)$ would have the effect of raising the condition number when the spectrum of the closed-loop system has a pole near the unit circle. For the case $E \neq I$, (4.25) can be a very restrictive requirement and is a subject of continuing research.

4.3 Balancing to Improve Condition

Considerable success has been attained in increasing the numerical accuracy of the solutions to linear equations and eigenvalue problems by appropriate scaling of problem parameters [39], [40]. Therefore, it is reasonable to expect that some sort of balancing or problem scaling could improve the accuracy of the numerical computations involved in computing the solution to the GARE and, hence, the overall "condition" of the problem. One could argue that since the numerical solution depends fundamentally on the QZ algorithm for the generalized eigenproblem [41], that balancing the eigensystem for the generalized eigenproblem $\lambda L y = My$ could improve the Riccati solution. Ward [42]
has proposed a balancing algorithm specifically designed to precede QZ-type algorithms. This balancing consists of permutations and two-sided diagonal transformations. The strategy is to scale \( L \) and \( M \) so that their elements have magnitudes as close to unity as possible. Thus, the large elements of \( L \) and \( M \) cannot mask the effect of the small elements, as can often be the case.

Those interested in the details of the algorithm are referred to [42] where the scaling strategy is discussed. Numerical results for example eigenproblems are also given in [42]. Numerical experience in solving the GARE has shown that this balancing strategy can increase the accuracy of the solution; an example is given in Chapter 5. This balancing can also increase the reliability of \( \kappa(V_{11}) \) as an indicator of condition of the Riccati problem. However, one does pay the price in increased computational time, about 10% of the QZ algorithm computation time.

An alternate approach to the direct balancing of the eigenproblem is to attempt some sort of coordinate change in the problem which generates the Riccati equation. That is, some nonsingular transformation \( T \) to change coordinates in the original model as follows:

\[
x(t) = Tw(t)
\]

or

\[
x_k = Tw_k.
\]

There is reason to believe that the coordinate balancing transformation proposed by Moore [43] may result in a transformed Riccati equation that is more amenable to numerical solution. Moore has shown [44] that
by a change of coordinates, one can scale the relative size of components of $e^{At}B$ and $e^{At}C^T$. Since, as we have already pointed out, stabilizability of the model influences the computational accuracy of the Riccati solution, one could argue that a change of coordinates increasing the relative size of components of $e^{At}B$ would be beneficial prior to computing the Riccati solution. Specifically, Moore's "internally balanced" coordinates in which the reachability and observability gramians are equal and diagonal or "input-normal" coordinates where the reachability gramian is identity are logical choices.

Laub [6] illustrated the effect of this coordinate type balancing on the following linear optimal control problem: Find a feedback controller $u(t) = Kx(t)$ which minimizes the performance index

$$J(u) = \int_0^\infty [x^T(t)Qx(t) + u^T(t)Ru(t)] \, dt$$

with plant dynamics given by

$$\dot{x}(t) = Ax(t) + Bu(t); \quad x(0) = x_0.$$  

Assume $Q = Q^T \geq 0$, $R = R^T > 0$, $(A, B)$ stabilizable and $(A, C)$ detectable, where $C^TC = 0$ and rank $(C) = \text{rank} \,(Q)$. Then the optimal control is well known to be

$$u(t) = -R^{-1}B^Txx$$

where $X$ solves the ARE

$$A^TX + XA - XBR^{-1}B^TX + Q = 0.$$
Suppose we change coordinates via (4.29). Then in terms of the new state $w(t)$, our problem is to minimize

$$\int_0^T [w^T(t)T^TFw(t) + u^T(t)R(t)u(t)] dt$$

subject to

$$\dot{w}(t) = (T^{-1}F)w(t) + (T^{-1}B)u(t).$$

The associated solution $X_w$ of the transformed Riccati equation is related to the original $X$ by

$$X = T^{-T}X_w T^{-1}$$

One can see from (4.32) and (4.33) that if $T$ is ill-conditioned with respect to inversion and the balancing is applied as these equations indicate, then this technique can potentially introduce more error into the solution than would originally have appeared. Hence, the opposite of the intended effect could occur. Of course, one could be careful to only choose a well-conditioned $T$, like a diagonal scaling matrix for instance. However, if $T$ is a modal or system balancing [43] transformation, it could be quite ill-conditioned.

We can reduce this problem significantly in our generalized problem formulation framework as the following result will show.

**Theorem 4.7:** If a change of coordinates of the form (4.29) is made in the continuous-time generalized optimal regulator problem of section 2.3.1 or a change of coordinates of the form (4.30) is made in the discrete-time generalized optimal regulator problem of Section 2.3.3, then the solution $X$ to the original problem may be found as follows:
1) replace $A$ by $AT$, $C$ by $CT$, $S$ by $T^TS$ and $E$ by $ET$ in the appropriate matrix pencils ((2.51) or (2.65) for the continuous problem, (2.67) or (2.68) for the discrete problem)

2) determine the appropriate $\tilde{p}$, $\tilde{Z}$ transformations for the modified pencils and partition $\tilde{Z}$ as in (2.61)

3) let $\tilde{V} = \begin{bmatrix} ET & 0 \\ 0 & I \end{bmatrix} \tilde{Z}$, then $X = \tilde{V}_{21} \tilde{V}_{11}^{-1}$ (4.34)

solves the original GARE ((2.30) for the continuous problem, (2.48) for the discrete problem) with $X = X^T > 0$.

**Proof:** For the continuous-time case, the problem becomes in the transformed coordinates

System: $ET \dot{w}(t) = ATw(t) + Bu(t)$

$y(t) = CTw(t)$ (4.35)

Criterion: $J = \frac{1}{2} \int_0^\infty (y^TQy + u^TRu + 2w^TS^Tu)dt.$ (4.36)

Application of Hamilton–Jacobi theory as in Section 2.3.1 gives rise to a set of equations to which we can associate the following matrix pencil:

$$\lambda \begin{bmatrix} ET & 0 \\ 0 & T^TE \end{bmatrix} - \begin{bmatrix} (A-\text{BR}^{-1}S^T)^T & -\text{BR}^{-1}B^T \\ -T(C^TQC-SR^{-1}S^T)^T & -T(A-\text{BR}^{-1}S^T)^T \end{bmatrix}$$ (4.37)

This pencil can also be obtained by performing step 1 of the theorem on the pencil associated with (2.57). Now we can factor (4.37) as

$$\begin{bmatrix} I & 0 \\ 0 & T^T \end{bmatrix} \lambda \begin{bmatrix} E & 0 \\ 0 & ET \end{bmatrix} - \begin{bmatrix} (A-\text{BR}^{-1}S^T) & -\text{BR}^{-1}B^T \\ -(C^TQC-SR^{-1}S^T)^T & -(A-\text{BR}^{-1}S^T)^T \end{bmatrix} \begin{bmatrix} T & 0 \\ 0 & I \end{bmatrix}$$ (4.38)
If we perform the ordered transformation on (4.38) by finding a $\bar{P}$ and $\bar{Z}$ as in (2.60) (as stated in step 2 of the theorem) and compare the result with (2.60), we find

$$Z = \begin{bmatrix} T & 0 \\ 0 & I \end{bmatrix} \bar{Z}.$$  \hspace{1cm} (4.39)

Recall from Theorem 2.4 that

$$X = V_{21}V_{11}^{-1} = Z_{21}(EZ_{11})^{-1}.$$ \hspace{1cm} (4.40)

Substitute (4.39) into (4.40) and the theorem is proved for the continuous case. An analogous procedure proves the discrete case. We omit those details.

Note from the theorem that the numerous occurrences of the inverse of $T$ have been eliminated. The inversion of $T$ is essentially required only once in the process of solving the linear system

$$X\bar{V}_{11} = \bar{V}_{21}$$

for the Riccati solution $X$. Computational advantage has also been realized because the number of matrix multiplications required in the solution process has been reduced.
CHAPTER 5
NUMERICAL EXPERIMENTS

The methods for solution of the GARE presented in Chapter 2, the iterative refinement procedure derived in Chapter 3, and the condition estimates for the computed solution of Chapter 4 are all geared to the efficient and reliable numerical computation of the Riccati solution. This chapter describes a FORTRAN software package (RICPACK) developed to aid in the study of the numerical conditioning of the algebraic Riccati equation and other closely related topics. First, a brief description of the package will be given. A short discussion of the algorithms and software will follow. Finally, numerical examples and results will be given to illustrate relevant aspects of the numerical solution.

5.1 Software Package RICPACK

RICPACK was developed to aid in the study of the numerical conditioning of the algebraic Riccati equation, methods for improving the condition, and iterative refinement of the solution. The FORTRAN subroutines in the package were written in modular form and are designed to facilitate their incorporation in some larger computer-aided control system design (CACSD) package or computer-aided systems and control analysis and design environment (CASCADE).

A FORTRAN driver program has also been written, primarily as a research tool, that is user-friendly for use in an interactive "terminal" type environment. The program prompts for all necessary input.
Convenient input default options exist not only for ease of data input, but also for exploitation by the subroutines to reduce the number and complexity of the computations. These options are also designed to speed the input of more "standard" type problems. Table 5.1 lists the default options available.

**TABLE 5.1**

<table>
<thead>
<tr>
<th>Input Matrix</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>Identity</td>
</tr>
<tr>
<td>O</td>
<td>Identity</td>
</tr>
<tr>
<td>S</td>
<td>Zero</td>
</tr>
<tr>
<td>R</td>
<td>Identity (or enter diagonal elements only, if diagonal)</td>
</tr>
</tbody>
</table>

Highlights of RICPACK capabilities include:

(a) Choice of calculation of the stabilizing (non-negative definite), anti-stabilizing (non-positive definite), or just any (possibly) indefinite or nonsymmetric solution to the GARE.

(b) Coordinate or system balancing of the system model [43], [45]; i.e., a special coordinate transformation on (2.18) or (2.40) of the form

\[
x(t) = T w(t) \quad (5.1)
\]

or

\[
x_k = T w_k \quad (5.2)
\]

respectively, such that the observability and reachability gramians of the transformed system are equal and diagonal.

68
(c) Ward's balancing [42] of the generalized eigenproblem (2.58) or (2.68) (prior to the OZ transformation) which consists of permutations and two-sided-diagonal transformations.

(d) Direct handling of singular control weighting or singular measurement noise covariance by compression of the extended pencils (2.65) or (2.67).

(e) Direct handling of cross-weighting or noise correlation; i.e., \( S \neq 0 \).

(f) Provision for robustness recovery procedure; i.e., to replace the \( C^TC \) term in the GARE with \( Q + \gamma C^TC \) and iterate on \( \gamma \), the driver program need only modify one block of the matrix pencil at each iteration.

(g) Iterative refinement (or new solutions for small parameter perturbations) by Newton's method and Sylvester equations; i.e., iteratively solving equations (3.27) or (3.59). As of this writing, this is implemented for \( E = \) identity only because although algorithms exist for solving these general Sylvester equations, reliable software implementing these algorithms is not readily available.

(h) Model unstabilizability detection as indicated by the condition of \( V_{11} \) with respect to inversion in (2.62) or (2.69); \( V_{11} \) is singular for an unstabilizable model.

(i) Calculation of unique stabilizing solution for stabilizable models with undetectable modes.

(j) Residual calculation of the form

\[
\mathbf{r} = \frac{||\text{Residual}_1||}{||X_1||}. \tag{5.3}
\]
(k) Condition estimates for the Riccati problem derived in Chapter 4.

5.2 **Algorithms and Software**

Most of the algorithmic computations (i.e., those beyond matrix algebra) performed in RICPACK employ proven stable algorithms coded into reliable, portable FORTRAN software. Two sources for the software are LINPACK [39] and EISPACK [40], [46]. Modified LINPACK software is used for linear equation solving (the BLAS are replaced with in-line code), estimating the condition of a non-diagonal cost matrix $R$, solving the system $XV_{11} = V_{21}$ and in estimating the condition of $V_{11}$. The LINPACK singular value decomposition (SVD) is used in compressing the $(2n+m) \times (2n+m)$ pencil when necessary. EISPACK-type $QZ$ software is used for performing the $QZ$ transformation and generalized eigenvalue calculations. Other EISPACK software is used in calculating the coordinate balancing [43], [45] transformation, in addition to software based on the Bartels-Stewart algorithm [47] for Lyapunov equations. The Bartels-Stewart based software is also used in condition estimation and Newton's iteration calculations. Ward's [42] software is used for the eigenproblem balancing, and Van Dooren's software [44] is used for ordering the transformed pencil. A description of each subroutine used is given in the Appendix.

5.3 **Numerical Examples**

The following simple continuous-time example was used to illustrate the numerical properties of RICPACK when stabilizability is the key factor:
Example 1
\[
\dot{x} = \begin{bmatrix} 1 & 0 \\ 0 & -2 \end{bmatrix} x + \begin{bmatrix} \varepsilon \\ 0 \end{bmatrix} u \\
y = [1 \ 1]x
\] (5.4)

\[
\text{minimize } \int_0^\infty (y^T y + u^T u) dt
\] (5.5)

This system is stabilizable for \( \varepsilon \neq 0 \) and completely reconstructible.

The applicable ARE is
\[
A^T X + XA - XBB^T X + C^T C = 0.
\] (5.6)

The "true" solution for \( X \) can be hand calculated for comparison purposes as
\[
X = \begin{bmatrix}
\frac{1 + \sqrt{1 + \varepsilon^2}}{\varepsilon^2} & \frac{1}{2 + \sqrt{1 + \varepsilon^2}} \\
\frac{1}{2 + \sqrt{1 + \varepsilon^2}} & 1 - \frac{\varepsilon^2}{4(2 + \sqrt{1 + \varepsilon^2})^2}
\end{bmatrix} > 0.
\] (5.7)

Note that as \( \varepsilon \to 0 \) the system approaches unstabilizability and the (1,1) element of \( X \) tends to infinity.

The solution to this problem was numerically computed on a DEC KL-10 (under TOPS-20) in double precision using RICPACK. The machine precision is near \( 10^{-18} \) in this case. The results of interest are summarized in Table 5.2. The only measure of condition included in the table is the condition of \( V_{11} \) with respect to inversion because the other measures did not give an indication that the solution accuracy was degenerating as \( \varepsilon \to 0 \). We note here that the data in Table 5.2 and the succeeding tables that are expressed as a power of 10 are rounded to the nearest power of 10.
### TABLE 5.2

Numerical Results for Example 1, $\epsilon = 10^{-N}$

<table>
<thead>
<tr>
<th>N</th>
<th>$\kappa(V_{11})$</th>
<th>$r$ (5.3)</th>
<th>Acc*</th>
<th>Newton iterations</th>
<th>$r$ (5.3)</th>
<th>Acc*</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$10^0$</td>
<td>$10^{-18}$</td>
<td>17</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>$10^4$</td>
<td>$10^{-14}$</td>
<td>14</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>$10^8$</td>
<td>$10^{-10}$</td>
<td>10</td>
<td>2</td>
<td>$10^{-18}$</td>
<td>17</td>
</tr>
<tr>
<td>6</td>
<td>$10^{12}$</td>
<td>$10^{-8}$</td>
<td>6</td>
<td>3</td>
<td>$10^{-20}$</td>
<td>17</td>
</tr>
<tr>
<td>8</td>
<td>$10^{16}$</td>
<td>$10^{-2}$</td>
<td>2</td>
<td>4</td>
<td>$10^{-34}$</td>
<td>17</td>
</tr>
<tr>
<td>9</td>
<td>$10^{18}$</td>
<td>$10^{-1}$</td>
<td>0</td>
<td>6</td>
<td>$10^{-18}$</td>
<td>17</td>
</tr>
<tr>
<td>10</td>
<td>$10^{20}$</td>
<td>$10^0$</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

The following data includes Ward balancing effects

<table>
<thead>
<tr>
<th>N</th>
<th>$\kappa(V_{11})$</th>
<th>$r$ (5.3)</th>
<th>Acc*</th>
<th>Newton iterations</th>
<th>$r$ (5.3)</th>
<th>Acc*</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$10^0$</td>
<td>$10^{-18}$</td>
<td>17</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>$10^7$</td>
<td>$10^{-15}$</td>
<td>15</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>$10^{12}$</td>
<td>$10^{-9}$</td>
<td>9</td>
<td>2</td>
<td>$10^{-18}$</td>
<td>17</td>
</tr>
<tr>
<td>11</td>
<td>$10^{16}$</td>
<td>$10^{-7}$</td>
<td>7</td>
<td>3</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>12</td>
<td>$10^{17}$</td>
<td>$10^{-7}$</td>
<td>7</td>
<td>3</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>13</td>
<td>$10^{17}$</td>
<td>$10^{-6}$</td>
<td>7</td>
<td>3</td>
<td>$10^{-18}$</td>
<td>17</td>
</tr>
<tr>
<td>14</td>
<td>sing.</td>
<td>$10^1$</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

*Accuracy in correct significant digits.

Some useful observations can be made on this data. One can see that for this example, $\kappa(V_{11})$ and the residual ($r$) are both good indicators of the numerical accuracy. Since machine precision is near $10^{-18}$, one would expect about 17 correct digits for a well-conditioned problem (which is the case for $\epsilon=1$). The data indicates that one digit
of accuracy is lost for each power of 10 change in $\kappa(V_{11})$ and $r$. This is desirable behavior of a condition estimate. Note that Ward balancing improves the condition of $V_{11}$ and reduces the value of $r$ for the same value of $\varepsilon$. Ward balancing enables solution calculation for smaller values of $\varepsilon$, but the accuracy is not as smooth a function of $\kappa(V_{11})$. However, the residual is still a good indicator of accuracy. Note that in all cases with a reasonable starting guess a few iterations of Newton’s method restores full accuracy. The generalized eigenvalue solution was used as a starting guess and was considered reasonable if $\kappa(V_{11}) < 1/\text{(machine precision)}$. When this condition was not satisfied, the Newton iteration failed to converge to the desired solution.

The above example illustrates that stabilizability of the model does indeed influence the numerical accuracy of the Riccati solution. Also, $\kappa(V_{11})$ and $r$ are good indicators of solution accuracy as the model approaches unstabilizability. However, $\kappa(V_{11})$ may not be a good indicator in other situations as the following example will show:

Example 2

$$
\begin{bmatrix}
-\varepsilon & 1 & 0 & 0 \\
-1 & -\varepsilon & 0 & 0 \\
0 & 0 & \varepsilon & 1 \\
0 & 0 & -1 & \varepsilon
\end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} u
$$

$$
y = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} x
$$

minimize $\int_0^\infty (y^T y + u^T u) \, dt$  \hspace{1cm} (5.9)

The model is completely controllable and observable. The open-loop poles are at $\pm \varepsilon \pm j$, and the applicable ARE is
The solution to this problem was numerically computed on a UNIVAC 1100/83 in double precision using RICPACK. The machine precision is near $10^{-18}$ in this case. The results of interest are summarized in Table 5.3. Although an exact hand solution was not possible in this case, the behavior of the residual can be used to judge the solution accuracy. This example was designed to assess the effect of the separation of the closed-loop spectrum on solution accuracy and the ability of condition estimates to detect degrading accuracy. The column CLP in Table 5.3 indicates the position (real part) of the closed-loop pole nearest the imaginary axis in the complex plane.

**TABLE 5.3**

<table>
<thead>
<tr>
<th>N</th>
<th>CLP</th>
<th>$\kappa(V_{11})$</th>
<th>$\kappa_A_c(X)$</th>
<th>$\kappa_B(X)$</th>
<th>$r$</th>
<th>Newton</th>
<th>$r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$10^0$</td>
<td>$10^1$</td>
<td>$10^0$</td>
<td>$10^2$</td>
<td>$10^{-16}$</td>
<td>1</td>
<td>$10^{-16}$</td>
</tr>
<tr>
<td>3</td>
<td>$10^{-6}$</td>
<td>$10^0$</td>
<td>$10^6$</td>
<td>$10^7$</td>
<td>$10^{-12}$</td>
<td>1</td>
<td>$10^{-14}$</td>
</tr>
<tr>
<td>5</td>
<td>$10^{-10}$</td>
<td>$10^0$</td>
<td>$10^{10}$</td>
<td>$10^{11}$</td>
<td>$10^{-8}$</td>
<td>1</td>
<td>$10^{-12}$</td>
</tr>
<tr>
<td>7</td>
<td>$10^{-14}$</td>
<td>$10^0$</td>
<td>$10^{18}$</td>
<td>$10^{15}$</td>
<td>$10^{-8}$</td>
<td>2</td>
<td>$10^{-16}$</td>
</tr>
<tr>
<td>8</td>
<td>$10^{-16}$</td>
<td>$10^0$</td>
<td>$10^{16}$</td>
<td>$10^{17}$</td>
<td>$10^{-1}$</td>
<td>2</td>
<td>$10^{-16}$</td>
</tr>
<tr>
<td>9</td>
<td>$10^{-18}$</td>
<td>$10^0$</td>
<td>$10^{18}$</td>
<td>$10^{18}$</td>
<td>$10^{-1}$</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

One can see from this example that $\kappa(V_{11})$ provides no indication of loss of accuracy in the solution. However, in this example $\kappa_A_c(X)$ and $\kappa_B(X)$ correlate directly with the behavior of the residual, and thus, the solution accuracy. Ward balancing of the eigenproblem had no
noticeable effect on solution accuracy for a given value of $\varepsilon$. Newton iterations did significantly improve solution accuracy, as measured by the residual, until the condition $\|A_\varepsilon(X) - B(X)\|_\infty = 1.0/(\text{machine precision})$. At this point, the iterations failed to converge to the desired solution, as was the case in example 1.

This example shows that separation of the closed-loop spectrum does indeed influence the numerical accuracy of the Riccati solution and that the condition estimates in which separation is a factor provide good indicators of solution degeneracy.

The following example illustrates the effect of ill-conditioning of the $R$ weighting matrix, with respect to inversion, on the numerical solution for the continuous-time case. Recall that ill-conditioning of $R$ is not necessarily a problem in the discrete-time case since its inverse is not explicitly required.

**Example 3**

$$
\begin{align*}
\dot{x} &= \begin{bmatrix} -0.1 & 0 \\ 0 & -0.02 \end{bmatrix} x + \begin{bmatrix} 0.1 \\ 0.001 \end{bmatrix} u \\
y &= \begin{bmatrix} 10. \\ 100. \end{bmatrix} x
\end{align*}
$$

(5.11)

$$
\begin{align*}
\text{minimize } & \int_0^\infty (y^T y + u^T \begin{bmatrix} 1+\varepsilon \\ 1 \\ 1 \end{bmatrix} u) \, dt
\end{align*}
$$

(5.12)

The system is completely controllable and observable. As $\varepsilon \to 0$, the $R$ matrix approaches singularity. The applicable ARE is

$$
A^T x + x A - x B R^{-1} B^T x + C^T C = 0.
$$

(5.13)
The solution to this problem was numerically computed on a UNIVAC 1100/83 in double precision using RICPACK. The machine precision is near $10^{-18}$ in this case. The results of interest are summarized in Tables 5.4 and 5.5. Ward balancing was employed in the calculations for Table 5.4, and coordinate balancing was employed for Table 5.5. Results of calculations where no balancing was applied were nearly identical to those of Table 5.5 for coordinate balancing.

**TABLE 5.4**

**Numerical Results for Example 3, Ward Balancing, $c=10^{-N}$**

<table>
<thead>
<tr>
<th>$N$</th>
<th>$\kappa(R)$</th>
<th>$\kappa(V_{11})$</th>
<th>$\kappa A(X)$</th>
<th>$\kappa B(X)$</th>
<th>$r$</th>
<th>Newton iterations</th>
<th>$r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$10^1$</td>
<td>$10^1$</td>
<td>$10^0$</td>
<td>$10^2$</td>
<td>$10^{-17}$</td>
<td>1</td>
<td>$10^{-17}$</td>
</tr>
<tr>
<td>2</td>
<td>$10^2$</td>
<td>$10^2$</td>
<td>$10^2$</td>
<td>$10^6$</td>
<td>$10^{-17}$</td>
<td>1</td>
<td>$10^{-17}$</td>
</tr>
<tr>
<td>4</td>
<td>$10^4$</td>
<td>$10^3$</td>
<td>$10^3$</td>
<td>$10^{10}$</td>
<td>$10^{-14}$</td>
<td>2</td>
<td>$10^{-14}$</td>
</tr>
<tr>
<td>6</td>
<td>$10^6$</td>
<td>$10^4$</td>
<td>$10^3$</td>
<td>$10^{11}$</td>
<td>$10^{-12}$</td>
<td>10</td>
<td>$10^{-12}$</td>
</tr>
<tr>
<td>8</td>
<td>$10^8$</td>
<td>$10^5$</td>
<td>$10^3$</td>
<td>$10^{13}$</td>
<td>$10^{-10}$</td>
<td>10</td>
<td>$10^{-10}$</td>
</tr>
<tr>
<td>10</td>
<td>$10^{10}$</td>
<td>$10^6$</td>
<td>$10^3$</td>
<td>$10^{15}$</td>
<td>$10^{-8}$</td>
<td>10</td>
<td>$10^{-8}$</td>
</tr>
<tr>
<td>12</td>
<td>$10^{12}$</td>
<td>$10^7$</td>
<td>$10^3$</td>
<td>$10^{17}$</td>
<td>$10^{-6}$</td>
<td>10</td>
<td>$10^{-6}$</td>
</tr>
<tr>
<td>14</td>
<td>$10^{14}$</td>
<td>$10^8$</td>
<td>$10^3$</td>
<td>$10^{19}$</td>
<td>$10^{-5}$</td>
<td>10</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>16</td>
<td>$10^{16}$</td>
<td>$10^9$</td>
<td>$10^3$</td>
<td>$10^{21}$</td>
<td>$10^{-3}$</td>
<td>10</td>
<td>$10^{-3}$</td>
</tr>
</tbody>
</table>
The data in Table 5.4 indicate that \( \kappa(R) \) with respect to inversion accurately reflects the behavior of the residual. Also, \( \kappa(V_{11}) \) with respect to inversion is too optimistic in its estimation of the problem condition and \( \kappa(B(X)) \) is too pessimistic. \( \kappa(A_c(X)) \) provides no information in this case. A maximum of 10 Newton iterations was allowed, and where 10 appears in the table, convergence did not occur. The value for the residual in that case is the residual associated with the solution at the tenth iteration. One can see that the ill-conditioning of \( R \) begins to dominate the numerical accuracy when \( \kappa(R) > 10^6 \). Since \( R^{-1} \) is involved in the Newton iteration calculations, iterative improvement does not improve accuracy when \( \kappa(R) \) dominates.

### TABLE 5.5

Numerical Results for Example 3, System Balancing, \( \varepsilon=10^{-N} \)

<table>
<thead>
<tr>
<th>N</th>
<th>( \kappa(R) )</th>
<th>( \kappa(V_{11}) )</th>
<th>( \kappa(A_c(X)) ) (4.18)</th>
<th>( \kappa(B(X)) ) (4.4)</th>
<th>( r ) (5.3)</th>
<th>Newton iterations</th>
<th>( r ) (5.3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10^1</td>
<td>10^3</td>
<td>10^0</td>
<td>10^3</td>
<td>10^{-15}</td>
<td>2</td>
<td>10^{-17}</td>
</tr>
<tr>
<td>2</td>
<td>10^2</td>
<td>10^3</td>
<td>10^2</td>
<td>10^6</td>
<td>10^{-15}</td>
<td>2</td>
<td>10^{-17}</td>
</tr>
<tr>
<td>4</td>
<td>10^4</td>
<td>10^3</td>
<td>10^3</td>
<td>10^9</td>
<td>10^{-12}</td>
<td>7</td>
<td>10^{-14}</td>
</tr>
<tr>
<td>6</td>
<td>10^6</td>
<td>10^3</td>
<td>10^3</td>
<td>10^{11}</td>
<td>10^{-12}</td>
<td>10</td>
<td>10^{-12}</td>
</tr>
<tr>
<td>8</td>
<td>10^8</td>
<td>10^3</td>
<td>10^3</td>
<td>10^{13}</td>
<td>10^{-9}</td>
<td>10</td>
<td>10^{-11}</td>
</tr>
<tr>
<td>10</td>
<td>10^{10}</td>
<td>10^3</td>
<td>10^3</td>
<td>10^{15}</td>
<td>10^{-6}</td>
<td>10</td>
<td>10^{-9}</td>
</tr>
<tr>
<td>12</td>
<td>10^{12}</td>
<td>10^3</td>
<td>10^3</td>
<td>10^{17}</td>
<td>10^{-6}</td>
<td>10</td>
<td>10^{-6}</td>
</tr>
<tr>
<td>14</td>
<td>10^{14}</td>
<td>10^3</td>
<td>10^3</td>
<td>10^{19}</td>
<td>10^{-2}</td>
<td>10</td>
<td>10^{-4}</td>
</tr>
<tr>
<td>16</td>
<td>10^{16}</td>
<td>10^3</td>
<td>10^3</td>
<td>10^{21}</td>
<td>10^{-1}</td>
<td>10</td>
<td>10^{-2}</td>
</tr>
</tbody>
</table>
convergence criteria used for the Newton iterations do not recognize this fact and stop the iterations.

The data in Table 5.5 are essentially the same as that in Table 5.4 except for one important aspect. \( \kappa(V_{11}) \) with respect to inversion provides no information on the problem conditioning once \( \kappa(R) \geq \kappa(V_{11}) \). This may indicate that Ward balancing will cause other sources of problem ill-conditioning, beside unstabilizability of the model and singularity of \( E \), to be reflected in \( \kappa(V_{11}) \).

The preceding examples illustrate that none of the potential measures of conditioning are reliable indicators by themselves. However, numerical experience to date has shown that in all cases at least one of the measures will detect the degeneracy of numerical accuracy as it occurs.
The behavior of many physical systems in engineering can be modeled by the following system of equations:

\[ M\ddot{x}(t) + (D + G)x(t) + Kx(t) = f(t) \]  

(6.1)

where \( x, f \in \mathbb{R}^n \) and \( M, D, G \) and \( K \in \mathbb{R}^{n \times n} \). Moreover, in describing physical systems one can make the following assumptions without loss of generality:

\[ M = M^T > 0, \quad \text{generalized capacitive storage} \quad (6.2) \]

\[ D = D^T > 0, \quad \text{generalized energy dissipators} \]

\[ G = -G^T, \quad \text{generalized conservative elements} \]

\[ K = K^T > 0, \quad \text{generalized inductive storage}. \]

The model (6.1) can describe electrical, mechanical, thermal, and other systems by appropriate choice of "through" variables, \( f \) (current in electrical systems, force in mechanical systems, etc.) and "across" variables, \( x \) (voltage, displacement, etc.) [48]. Analogies exist among the various types of systems.

The model (6.1) can result directly from lumped parameter models, or finite approximations to distributed parameter systems described by partial differential equations. One large class of systems of current importance are large space structures (LSS), which are large distributed parameter systems that are most often discretized by the finite element method into the form (6.1) [49]. The problem of controlling LSS motivated the studies of this chapter, although the results are applicable to any system described by (6.1). The first section defines
the LSS framework and explores the inherent structure of (6.1) and (6.2) that might be computationally exploitable. The second section considers specifically the solution of the GARE associated with second-order models. In the third section, criteria are discussed for the determination of controllability, stabilizability, observability or detectability of (6.1).

6.1 Second-Order-Model Structure in the LSS Framework

In the LSS framework, $x$ is a displacement vector, $f$ is a force vector, $M$ is the mass matrix, $K$ is the stiffness matrix, $D$ is the damping matrix, and the $G$ matrix gives rise to gyroscopic forces. In general, $n$ is initially of very high order and the aforementioned matrices may be sparse. The force vector $f$ is of the form

$$f(t) = Fu(t) \quad (6.3)$$

where $u \in \mathbb{R}^m$ is the control input, and $F \in \mathbb{R}^{nxm}$ is the input matrix. One usually considers an output equation of the form

$$y(t) = Px(t) + Vx(t) \quad (6.4)$$

where

$$y \in \mathbb{R}^n \quad \text{and} \quad P, V \in \mathbb{R}^{rxn}.\quad \cdot$$

The traditional method for dealing with models of the form (6.1) is to transform to the equivalent standard matrix first-order (state variable) form

$$\dot{z}(t) = Az(t) + Bu(t) \quad (6.5)$$
where \( z = \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \) is the state of dimension \( 2n \). That is, the model (6.1) would be transformed to the following

\[
\begin{bmatrix}
\dot{x}(t) \\
\dot{\dot{x}}(t)
\end{bmatrix} = \begin{bmatrix}
0 & I \\
-M^{-1}K & -M^{-1}(D+G)
\end{bmatrix} \begin{bmatrix}
x(t) \\
\dot{x}(t)
\end{bmatrix} + \begin{bmatrix}
0 \\
F
\end{bmatrix} u(t).
\] (6.6)

Unfortunately, this procedure, although it is conceptually simple, faces practical computational drawbacks. First, the symmetry, definiteness and sparsity structure of the \( M, D, G \) and \( K \) matrices are not exploited. In general, \( M^{-1}K \) and \( M^{-1}(D+G) \) are not symmetric and are dense even though \( M, D, G \) and \( K \) are symmetric and sparse. Moreover, if \( M \) is nearly singular, then \( M^{-1} \) may be computationally ill-determined.

Consider the following generalized first-order realization of (6.1), (6.3), and (6.4):

\[
\begin{bmatrix}
I & 0 \\
0 & M
\end{bmatrix} \begin{bmatrix}
\dot{x} \\
\ddot{x}
\end{bmatrix} + \begin{bmatrix}
0 & I \\
-K & -(D+G)
\end{bmatrix} \begin{bmatrix}
x \\
\dot{x}
\end{bmatrix} + \begin{bmatrix}
0 \\
F
\end{bmatrix} u
\]

\[
y = [P \ \ V] \begin{bmatrix}
x \\
\dot{x}
\end{bmatrix} \] (6.7)

Pre-multiplying (6.7) by \( \begin{bmatrix} I & 0 \\ 0 & M \end{bmatrix}^{-1} \) yields the "standard" first-order model (6.6). System (6.7) is of the generalized first-order form

\[
\dot{z} = Az + Bu
\]

\[
y = Cz
\] (6.8)

by appropriate definition of \( z, A, B, C \) and \( E \).
Other first-order realizations of this form which are of potential interest include:

\[
\begin{bmatrix}
D+G & M \\
M & 0
\end{bmatrix}
\begin{bmatrix}
z \\
0
\end{bmatrix}
= 
\begin{bmatrix}
-K & 0 \\
0 & M
\end{bmatrix}
\begin{bmatrix}
z \\
0
\end{bmatrix}
+ 
\begin{bmatrix}
F \\
0
\end{bmatrix}
\begin{bmatrix}
u \\
0
\end{bmatrix}
\tag{6.9}
\]

\[
\begin{bmatrix}
D+G & M \\
-M & 0
\end{bmatrix}
\begin{bmatrix}
z \\
0
\end{bmatrix}
= 
\begin{bmatrix}
-K & 0 \\
0 & -M
\end{bmatrix}
\begin{bmatrix}
z \\
0
\end{bmatrix}
+ 
\begin{bmatrix}
F \\
0
\end{bmatrix}
\begin{bmatrix}
u \\
0
\end{bmatrix}
\tag{6.10}
\]

\[
\begin{bmatrix}
-K & 0 \\
0 & M
\end{bmatrix}
\begin{bmatrix}
z \\
0
\end{bmatrix}
= 
\begin{bmatrix}
0 & -K \\
-K & -(D+G)
\end{bmatrix}
\begin{bmatrix}
z \\
0
\end{bmatrix}
+ 
\begin{bmatrix}
F \\
0
\end{bmatrix}
\begin{bmatrix}
u \\
0
\end{bmatrix}
\tag{6.11}
\]

Note that (6.9) and (6.11) have symmetric \( E \) and \( A \) matrices when \( G = 0 \), while (6.1) has skew-symmetric \( E \) and symmetric \( A \) when \( D = 0 \). These properties are computationally advantageous in the problems in the remainder of this chapter.

6.2 The GARE for Second-Order Models

Consider the model (6.7), (6.4) in the context of the continuous-time optimal regulator problem of section 2.3.1 and let \( \bar{D} = D+G \). The GARE for this problem from (2.30) is

\[
0 = 
\begin{bmatrix}
I & 0 \\
0 & M
\end{bmatrix}
X
\begin{bmatrix}
0 & I \\
-I & \bar{D}
\end{bmatrix}
+ 
\begin{bmatrix}
0 & -K \\
I & -\bar{D}^T
\end{bmatrix}
X
\begin{bmatrix}
I & 0 \\
0 & M
\end{bmatrix}
+ 
\begin{bmatrix}
P^T \\
V^T
\end{bmatrix}
Q[P V]
- 
\begin{bmatrix}
I & 0 \\
0 & M
\end{bmatrix}
X
\begin{bmatrix}
0 & P \bigl(0 P^T\bigr) \bigl(0 P^T\bigr) X
\end{bmatrix}
\begin{bmatrix}
I & 0 \\
0 & M
\end{bmatrix}
\tag{6.12}
\]

and the optimal control law is given by

\[
u^0 = -R^{-1}[0 P^T] \begin{bmatrix}
I & 0 \\
0 & M
\end{bmatrix}
\begin{bmatrix}
x \\
x
\end{bmatrix}
\tag{6.13}
\]

If one solves the 2n x 2n GARE (6.12) employing the method of section 2.3, then one is faced with a 4n x 4n generalized eigenproblem. Clearly, this is undesirable if \( n \) is itself a large number.
Instead we expand $X = X^T \geq 0$ into $n \times n$ blocks as 

$$
\begin{bmatrix}
W & Y \\
Y^T & Z
\end{bmatrix} \geq 0, \quad W = W^T \geq 0, \quad WW^T = Y, \quad Z - Y^T W^T \geq 0.
$$

(6.14)

Substituting (6.14) into (6.12) and multiplying out yields the following three equations:

1. $0 = -YK - KY^T - YFR^{-1}F_{Y}^T + P^T Q P$

(6.15)

2. $0 = -MZ \ddot{D} - \ddot{D}_{Y}^T ZM - MZF^{-1}F_{Z}^T ZM + V^T Q V + MY^T + YM$

(6.16)

3. $0 = W - Y \ddot{D} - KZM - YFR^{-1}F_{Z}^T ZM + P^T Q V$

(6.17)

We see that (6.15) is an $n \times n$ Riccati-type equation from which $Y$ could be obtained. Once $Y$ is determined, (6.16) becomes an $n \times n$ GARE from which we can solve for $Z$. Equation (6.17) then simply defines $W$.

If only the feedback (6.13) is desired, we see (by substituting (6.14) into (6.13) and expanding) that

$$u^0 = -R^{-1}F_Y (Y^T X + Z M X) .$$

(6.18)

Therefore, one need not solve (6.17) for $W$.

It is computationally advantageous to only require the solution of two $n \times n$ Riccati equations rather than one $2n \times 2n$ Riccati equation, since the solution of an $N \times N$ Riccati equation takes $O(N^3)$ operations. To realize even greater computational advantage, we consider two common special cases. First, we consider the case when there is no damping or gyroscopic forces, i.e., $\ddot{D} = 0$. One can see that (6.16) is then no longer a GARE; it reduces to a simpler quadratic equation.
Next, consider the velocity output case, i.e., \( P = 0 \). Then, a solution for (6.15) is \( Y = 0 \). One can solve the GARE (6.16) for \( Z \) and the optimal control becomes

\[
u^0 = -R^{-1}F^TZx(t)\]

(6.19)

Obviously, solving one \( nxn \) GARE is computationally simpler than solving a \( 2n \times 2n \) GARE regardless of the size of \( n \). Based on these results, we can state the following theorem.

**Theorem 6.1:** Given the system

\[
M \ddot{x}(t) + D \dot{x}(t) + K x(t) = F u(t)
\]

(6.20)

with output

\[y(t) = V x(t)\]

(6.21)

and criterion

\[J = \int_0^\infty (y^T Q y + u^T R u) dt.\]

(6.22)

Then the unique stabilizing control which minimizes (6.22) is given by

\[u^0(t) = -R^{-1}F^TZx(t)\]

(6.23)

where \( Z = Z^T > 0 \) satisfies the GARE

\[
0 = -MZD - D^T Z - MZFR^{-1}F^T Z M + V^T Q V.
\]

(6.24)

**Proof:** The theorem is a restatement of the results of this section.

6.3 Controllability and Observability Criteria for Second-Order Models

We now seek to establish controllability and observability criteria for the model (6.1), (6.3) and (6.4). Controllability and observability of this model have been shown to provide important insights into modal behavior of the system and to furnish information...
on the number and positioning of sensors and actuators [51], [52].
Also, controllability and observability information can be used in
determining which modes to retain when performing model reduction [53].

One could apply the traditional methods for determining controllability and observability to the transformed first-order model (6.6) with state dimensions 2n. Unfortunately, the resulting standard tests (cast in terms of a 2n-th order "A" matrix) do not take advantage of the symmetry, definiteness and sparsity structure of the matrices M, D, G, and K. Also, computational problems may exist if M is near singular or n is very large.

Other conditions have been derived by Hughes and Skelton [51], which exploit the specialized structure of (6.1) and (6.2) for the cases D = 0 and D+G = 0. However, these conditions could suffer from computational difficulties since they require knowledge of the full modal transformation matrix, whose columns are the eigenvectors corresponding to the eigenvalues λ of the generalized eigenproblem

\[ [\lambda M + K]x = 0 \] (6.25)

where \( \pm \lambda^{1/2} \) is the modal frequencies. An equivalent form of (6.25) is the simple eigenproblem

\[ -M^{-1}Kx = \lambda x \] (6.26)

since M is nonsingular. But this form is computationally undesirable for the reasons already mentioned. In addition, computation of the eigenvectors of (6.25) and (6.26) is ill-conditioned whenever the λ is repeated or nearly equal [15], which is often the case in LSS.
The remainder of this section focuses on conditions for controllability (or, more generally, stabilizability) and observability (or, more generally, detectability) which take advantage of the structure of (6.1) and (6.2), but extend the results of [51] and are computationally more tractable. Most of the computational attractiveness of the new criteria accrue from the fact that an initial modal transformation is not necessary. Thus, if just a few "important" modes are known—and there exist techniques to determine just selected modes, e.g., [54], [55]—these modes can be tested for, say controllability, by a test involving just the model matrices M, D, G, K, and F.

**Definition 6.1:**

\[ \Omega := \{ \lambda_1 : \lambda_1 \text{ is a generalized eigenvalue of the problem} \]
\[
\begin{bmatrix}
0 & I \\
-K & -(D+G)
\end{bmatrix}
- \lambda
\begin{bmatrix}
I & 0 \\
0 & M
\end{bmatrix}
\]

\[ := \{ \text{modes of the system (6.7)} \} \]

Also, \( \Omega^+ := \{ \lambda \in \Omega : \Re \lambda > 0 \} \)

\( \Omega^- := \{ \lambda \in \Omega : \Re \lambda < 0 \} \)

For \( M \) nonsingular there are \( 2n \) modes, and \( \Omega^+ \) and \( \Omega^- \) are the sets of the unstable and stable modes, respectively. Alternatively, \( \Omega \) can be determined in terms of the generalized eigenvalue problems

\[
\begin{bmatrix}
-K & 0 \\
O & M
\end{bmatrix}
- \lambda
\begin{bmatrix}
D+G & M \\
M & 0
\end{bmatrix}
\]

\[ = \begin{bmatrix}
-K & 0 \\
O & -M
\end{bmatrix}
- \lambda
\begin{bmatrix}
D+G & M \\
-M & G
\end{bmatrix}
\]

\[
\begin{bmatrix}
0 & -K \\
-K & -(D+G)
\end{bmatrix}
- \lambda
\begin{bmatrix}
-K & 0 \\
0 & M
\end{bmatrix}
\]

86
whichever yields the greatest computational advantage. Computation of eigenvalues for problems of the form (6.32) is discussed in [54] for the case $G = 0$.

Theorem 6.2: (Hautus [56]) The system

$$\dot{x} = Ax + Bu; \ x(t) \in \mathbb{R}^n$$

(6.33)

is

a) Controllable (stabilizable) if and only if

$$\text{rank } (A - \lambda I, B) = n; \text{ for all } \lambda \in \Lambda(A) \ (\Lambda^+(A))$$

(6.34)

b) Observable (detectable) if and only if

$$\text{rank } \left[ \begin{array}{c} C \\ A - \lambda I \end{array} \right] = n; \text{ for all } \lambda \in \Lambda(A) \ (\Lambda^+(A)),$$

where $\Lambda(A) := \{ \lambda : (A - \lambda I)x = 0, x \neq 0 \}$

$$:= \text{Spectrum of } A$$

(6.35)

Proof: [56]

Clearly then, the system (6.8) with $E$ nonsingular is controllable (stabilizable) if and only if

$$\text{rank } (A - \lambda E, B) = 2n \text{ for all } \lambda \in \Lambda(E^{-1}A) \ (\Lambda^+(E^{-1}A))$$

(6.37)

and observable (detectable) if and only if

$$\text{rank } \left[ \begin{array}{c} C \\ A - \lambda E \end{array} \right] = 2n \text{ for all } \lambda \in \Lambda(E^{-1}A) \ (\Lambda^+(E^{-1}A))$$

(6.38)

We now exploit the structure of $A$, $B$, $C$ to derive controllability, etc., criteria directly in terms of $M$, $D$, $G$, $K$, $F$, $P$, and $V$.

Theorem 6.3: The system (6.1), (6.3) is controllable (stabilizable) if and only if

$$\text{rank } [\lambda^2M + \lambda(D+G) + K, F] = n; \text{ for all } \lambda \in \Omega \ (\Omega^+)$$

(6.39)
Proof: By Theorem 6.2, the system (6.1), (6.3) is controllable (stabilizable) if and only if
\[ 2n = \text{rank} \begin{bmatrix} A - \lambda E, B \end{bmatrix}; \text{ for all } \lambda \in \Omega (\Omega^+) \]
\[ = \text{rank} \begin{bmatrix} -\lambda I & I & 0 \\ -K & -D-G-\lambda M & F \end{bmatrix} \text{ from (6.7) and (6.8)} \]
\[ = \text{rank} \begin{bmatrix} \lambda M+D+G & I \\ I & 0 \end{bmatrix} \begin{bmatrix} -\lambda I & -I & 0 \\ -K & -D-G-\lambda M & F \end{bmatrix} \begin{bmatrix} -I & 0 & 0 \\ 0 & 0 & I \end{bmatrix} \]
\[ = \text{rank} \begin{bmatrix} \lambda^2 M+\lambda(D+G)+K & 0 & F \\ 0 & I & 0 \end{bmatrix} \]
Clearly this obtains if and only if
\[ \text{rank} \begin{bmatrix} \lambda^2 M+\lambda(D+G)+K, F \end{bmatrix} = n \text{ for all } \lambda \in \Omega (\Omega^+). \]
Note that \( \lambda \) is a scalar, so that sparsity in the problem is preserved. Also, no inverses and no initial transformations are necessary. Finally, note that each mode of the system can be checked individually without transforming the system to modal coordinates.

Theorem 6.4: The system (6.1), (6.4) is observable (detectable) if and only if
\[ \text{rank} \begin{bmatrix} \lambda V+F \\ \lambda^2 M+\lambda(D+G)+K \end{bmatrix} = n \text{ for all } \lambda \in \Omega (\Omega^+). \]  
(6.40)

Proof: By Theorem 6.2 the system (6.1), (6.4) is observable (detectable) if and only if
2n = rank \[
\begin{bmatrix}
C \\
A - \lambda E \\
P -\lambda I & V \\
-K & -D-G-\lambda M
\end{bmatrix}
\]; for all \( \lambda \in \Omega (\Omega^+) \)

\[= \text{rank } \begin{bmatrix}
1 & -V & 0 \\
0 & -\lambda M-D-G & -I \\
0 & I & 0
\end{bmatrix}
\begin{bmatrix}
-P & V \\
-I & I \\
-K & -D-G-\lambda M
\end{bmatrix}
\begin{bmatrix}
I & 0 \\
\lambda I & 1
\end{bmatrix}
\]

\[= \text{rank } \begin{bmatrix}
\lambda V+P \\
\lambda^2 M+\lambda(D+G)+K \\
0 & 1
\end{bmatrix}
\]

Clearly, this obtains if and only if

\[\text{rank } \begin{bmatrix}
\lambda V+P \\
\lambda^2 M+\lambda(D+G)+K \\
0 & 1
\end{bmatrix} = n \text{ for all } \lambda \in \Omega (\Omega^+) \]

Note that an alternative proof of Theorems 6.3 and 6.4 is to observe that the results are essentially restatements of the spectral criteria for left or right coprimeness of the appropriate polynomial matrices (quadratic, in this case). However, we have exploited here the numerically useful characterization of \( \Omega \) in Definition 6.1, and the proofs are direct and require no polynomial matrix theory. Several special cases of Theorems 6.3 and 6.4 are of interest in many systems and are now stated as corollaries.

**Corollary 6.3.1**: When \( D+G = 0 \) (i.e., no damping or gyroscopic forces), the system (6.1), (6.3) is controllable if and only if

\[\text{rank } [-\omega_i^2 M+K, F] = n ; \ i = 1, \ldots, n \] (6.41)

where \( \omega_i = \sqrt{\lambda_i} \ ; \ \lambda_i \in \Omega(N^{-1}K) \) (Note: \( \lambda_i \geq 0 \)).

**Proof**: When \( D+G = 0 \)

\[\Omega = \{ \pm j \omega_i \ ; \ i = 1, \ldots, n \}\]
and (6.41) follows directly from Theorem 6.3.

**Corollary 6.3.2:** When $D+G = 0$ and the system (6.1), (6.3) is in modal form, then (6.1), (6.3) is controllable if and only if

$$\text{rank } F_r = n_r; \ (r = 1, \ldots, R)$$

(6.42)

where $F_r$ are partitioned rows of the modally transformed $F$ matrix corresponding to the multiplicities $n_i$ of the $\omega_i$; $n_1 + \ldots + n_R = n$, and $R$ is the number of distinct $\omega_i$. This is Theorem 1 of [51].

**Proof:** In modal form

$$M = I, \ K = \text{diag}[\omega_1^2, \ldots, \omega^2], \ F_n = \text{modally transformed } F \text{ matrix.}$$

Then (6.42) follows directly from Corollary 6.3.1.

**Corollary 6.3.3:** When $K = 0$ the system (6.1), (6.3) is controllable if and only if $\text{rank } F = n$.

**Proof:** When $K = 0$, $\lambda = 0 \in \Omega$ and the corollary follows directly from Theorem 6.3.

Similarly, we can state

**Corollary 6.4.1:** When $D+G = 0$ the system (6.1), (6.4) is observable if and only if

$$\text{rank } \begin{bmatrix} j\omega_iV+P \\ -\omega_i^2M+K \end{bmatrix} = n; \ i = 1, \ldots, n.$$  

(6.43)

**Proof:** This result follows directly from the proof of Corollary 6.3.1 and Theorem 6.4.

**Corollary 6.4.2:** When $D+G = 0$ and $V = 0$ (i.e., no rate feedback) the system (6.1), (6.4) is observable if and only if
Proof: Follows directly from Corollary 6.4.1.

**Corollary 6.4.3:** When \( D + G = 0 \) and the system (6.1), (6.4) is in modal form, then (6.1), (6.4) is observable if and only if

\[
\text{rank} \begin{bmatrix} P \\ -\omega_i^2 M + K \end{bmatrix} = n; \quad i = 1, \ldots, n. \quad (6.44)
\]

where \( [j\omega_i V_r + P_r] \) are the suitably partitioned columns of the modally transformed \( [j\omega_i V + P] \) matrix.

Proof: In modal form \( M = I, K = \text{diag}[\omega_1^2, \ldots, \omega_n^2] \), \( [j\omega_i V + P] \) = modally transformed \( [j\omega_i V + P] \) matrix and (6.45) follows directly from (6.43).

**Corollary 6.4.4:** When \( D + G = 0 \), the system (6.1), (6.4) is in modal form, and \( V = 0 \) then (6.1), (6.4) is observable if and only if

\[
\text{rank} P_r = n_r; \quad (r = 1, \ldots, R). \quad (6.46)
\]

Proof: Follows directly from Corollary 6.4.3

**Corollary 6.4.5:** When \( K = 0 \) the system (6.1), (6.4) is observable if and only if \( \text{rank} P = n \).

Proof: Set \( \omega_i = 0 \) and \( K = 0 \) in (6.43).

Note that the above Corollaries have obvious analogues in terms of stabilizability and detectability, as appropriate.
CHAPTER 7
CONCLUSION

We have examined some numerical issues regarding the solution of a very general form of the algebraic Riccati equation in both the continuous- and discrete-time formulations. These generalized equations resulted from control and filtering problems for systems in generalized state space form with performance criteria that included cross-coupling between the state and input. The basic solution method considered was the Schur technique. The Schur technique was preferred because of its good numerical properties, especially when some closed-loop-system eigenvalues were closely spaced. The generalized eigenproblem framework employed allows solutions when \( E \) and \( R \) are ill-conditioned with respect to inversion without undue influence of the ill-conditioning on the solution process. Singular \( R \) matrices were permissible in the discrete-time case.

A Newton-type iterative refinement procedure for the Riccati solution was derived. In the most general case, it required solution of a Sylvester equation at each iteration. Numerical results indicated that the iterative refinement improved numerical accuracy significantly in all cases where accuracy was lost due to ill-conditioning of the Riccati problem except when \( R \) was ill-conditioned with respect to inversion in the continuous-time problem. Lack of improvement in this case was attributed to the fact that the iterative procedure required \( R^{-1} \) explicitly at each iteration.
Condition estimates for the Riccati problem were examined. New condition estimates were also derived. The behavior of these estimates and their ability to detect degraded accuracy of the Riccati solution were evaluated in numerical examples. All numerical examples employed the software package RICPACK for the solution of the Riccati equation. RICPACK was developed as a research tool to aid in the study of numerical issues related to the solution of algebraic Riccati equations. It was found that no single condition measure considered herein would reliably reflect the accuracy of the Riccati solution in all cases. However, ill-conditioning was always detected by at least one of them.

The special structure of linear system models in second-order form was considered. Computational advantage was sought in solving Riccati equations related to these models. Special computational advantage was realized in the velocity feedback control problem. Controllability and observability tests for second-order models were derived directly in terms of the system model matrices. These tests had the additional advantage of enabling one to test individual model modes.

Some topics of continuing research in these areas will include:

1. Iterative refinement procedure in the continuous-time formulation that does not require explicit inversion of the R matrix.

2. Further development of condition estimates for the Riccati problem, especially for the case $E \neq I$ and considering other forms for the perturbation $\delta F$. 

94
3. Further exploitation of the special structure of models in second-order form in all types of control and filtering computations.

We note here that all further research in the area of numerically solving algebraic Riccati equations can benefit from the availability of RICPACK.
REFERENCES


100


Appendix A

SOFTWARE DESCRIPTION
The purpose of this appendix is to provide more detailed information on the software package RICPACK than is appropriate for the main body of the thesis. However, it is not a complete documentation package for RICPACK. First there will be a discussion of the hierarchy of the software routines employed in the package. Then a sample terminal session for a particular example problem is illustrated. Finally, some appropriate software listings are included.

Figure A.1 illustrates the hierarchy of the software routines. That is, the routines on the upper levels employ the routines of the lower levels. At the lowest level we have the basic matrix manipulation routines like add, subtract, multiply, etc., and some simple combinations of these basic operations. The next level consists of standard routines for linear equations, eigenvalues and singular value decomposition (SVD). Most routines in this level are from LINPACK or EISPACK, or are slight modifications to routines from LINPACK and EISPACK. Subroutines that are modified have the modifications noted in the comment documentation included in the subroutine. A list of the Level 0 and I routines is given in Table A.1. The SVD routine is listed separately with the BLAS routines that it requires from LINPACK, as it is the only routine to require the BLAS and could be modified to eliminate said BLAS. No further documentation of these routines is provided in this appendix.
<table>
<thead>
<tr>
<th>LEVEL 0</th>
<th>LEVEL 1</th>
<th>LEVEL 2</th>
<th>LEVEL 3</th>
<th>MAIN PROGRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SINGULAR VALUE DECOMPOSITION</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EIGENVALUES</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EISPACK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>LINPACK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>LINEAR EQUATIONS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RICPACK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RESID</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RESOL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SESEPST</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>LIPPOS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>LIPORD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RICCOR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RICITN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RICGAIN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RICBALG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RICBALM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RICBALI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SCHUR FORM ORDERING</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>COMPRESSED PENCIL (CHPES)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PENCIL WITH R^2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>LYAPUNOV SOLUTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HARD BALANCING</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>FEEDBACK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>COORDINATE BALANCING</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NEWTON ITERATION</td>
</tr>
</tbody>
</table>

**FIGURE A.1.** Hierarchy of subroutines in software package RICPACK.
### TABLE A.1

Subroutine List for Levels 0 and 1

<table>
<thead>
<tr>
<th>Level 0</th>
<th>Level 1</th>
<th>DSVDC</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCORBK</td>
<td>BALANC</td>
<td>MLINEQ</td>
</tr>
<tr>
<td>DINRM</td>
<td>ORTHES</td>
<td></td>
</tr>
<tr>
<td>MADD</td>
<td>DGEFAM</td>
<td>QZHESW</td>
</tr>
<tr>
<td>MMUL</td>
<td>DGESLM</td>
<td>QZITW</td>
</tr>
<tr>
<td>MOUT</td>
<td>DGEFAM</td>
<td>QZVAL</td>
</tr>
<tr>
<td>MQF</td>
<td>DGESLM</td>
<td>QZVAL</td>
</tr>
<tr>
<td>MQFWO</td>
<td>DSTSLV</td>
<td>REBAKB</td>
</tr>
<tr>
<td>MSQLE</td>
<td>ELMHES</td>
<td>REDUCE</td>
</tr>
<tr>
<td>MSUB</td>
<td>GIV</td>
<td>REDUC2</td>
</tr>
<tr>
<td>MULA</td>
<td>GRADBK</td>
<td>ROTC</td>
</tr>
<tr>
<td>MULB</td>
<td>GRADEQ</td>
<td>ROTR</td>
</tr>
<tr>
<td>MULWOA</td>
<td>HQR</td>
<td>SCALBK</td>
</tr>
<tr>
<td>MULWOB</td>
<td>HQRORT</td>
<td>SCALEG</td>
</tr>
<tr>
<td>PERMUT</td>
<td>IMTQL2</td>
<td>SYMSLV</td>
</tr>
<tr>
<td>SAVE</td>
<td>LINEQ</td>
<td>TRED2</td>
</tr>
<tr>
<td>SEQUIV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRNATA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRNATB</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Subroutines at Levels 2 and 3 perform more specialized tasks. The task title is given in Figure A.1, and the main subroutine performing the task is shown in parenthesis. The major comment documentation from these main subroutines, as it appears in the software, is given in this appendix. This documentation can be consulted for more details on the purpose of each routine, or for a list of subroutines that each of these main subroutines may call.

Of course, the main driver program is at the highest level (4), and a complete listing is provided. This main program would be rewritten, or at least extensively modified for most applications. This driver was written as a research tool and as such performs calculations not relevant to many analysis and design applications. A higher level language would be more appropriate to interface the Level 0 through Level 3 subroutines with a larger CACSD package and perform the necessary input and output.

A sample output listing from an actual terminal session to solve one particular problem is given in Figure A.2. The problem is Example 1, unbalanced case, for ε = .0001. The session illustrates most of the features and options of RICPACK. The computer prompt for data is the ">" sign. The listing is self-explanatory and will not be elaborated further.

The main program listing and heading documentation comment listings for the main subroutines of Levels 2 and 3 follow Figure A.2.
NWC TP 6521

@XQT RICPACK.

GENERALIZED ALGEBRAIC MATRIX RICCATI EQUATION SOLVER

ENTER SYSTEM ORDER 'N'; = FOR CONTINUOUS TIME PROBLEMS, - FOR DISCRETE
MAXIMUM ORDER = 20

> 2

N = 2

ENTER NUMBER OF SYSTEM INPUTS 'M'
AND NUMBER OF SYSTEM OUTPUTS "L"

> 1, 1

M = 1  L = 1

ENTER FLAG FOR DESIRED SOLUTION:
-1 FOR STABILIZING SOLUTION
0 FOR ANY SOLUTION
+ 1 FOR DESTABILIZING SOLUTION

> -1

IORD = -1

ENTER BALANCING FLAG: 0 FOR WARD BALANCING
1 FOR CO-ORDINATE BALANCING
2 FOR NO BALANCING

> 2

IBAL = 2

NO BALANCING ATTEMPTED

DO YOU WISH TO ITERATE FOR ROBUSTNESS RECOVERY (Y OR N)

> N

KFLAG = N

DO YOU WISH TO ENTER AN "E" MATRIX (Y OR N)

DEFAULT IS E = IDENTITY MATRIX

FIGURE A.2 Sample terminal session output listing.

108
EFLAG = N

USING DEFAULT: "E" = IDENTITY

ENTER THE 2 X 2 SYSTEM MATRIX "A" BY ROWS

> 1.0
> 0 -2

THE "A" MATRIX IS:

1.0000000000000000 0.0000000000000000
0.0000000000000000 -2.0000000000000000

ENTER THE 2 X 1 INPUT MATRIX "B" BY ROWS

> 0.0001
> 0

THE "B" MATRIX IS:

0.0000000000000000-003
0.0000000000000000

ENTER THE 1 X 2 OUTPUT MATRIX "C" BY ROWS

> 1,1

THE "C" MATRIX IS:

1.0000000000000000 1.0000000000000000

DO YOU WISH TO ENTER A CONTROL WEIGHTING MATRIX "R" (Y OR N)
DEFAULT IS R = IDENTITY MATRIX

> N

RFLAG = N

USING DEFAULT: "R" = IDENTITY

DO YOU WISH TO ENTER AN OUTPUT WEIGHTING MATRIX "Q" (Y OR N)

> N

FIGURE A.2 Sample terminal session output listing (continued).
QFLAG = N

USING DEFAULT: "Q" = IDENTITY

DO YOU WISH TO ENTER A STATE/INPUT CROSS-WEIGHTING MATRIX "S" (Y OR N)
DEFAULT IS S = ZERO MATRIX

>N

SFLAG = N

USING DEFAULT: "S" = ZERO MATRIX

THE CLOSED LOOP EIGENVALUES FOR THE STABILIZING RICCATI SOLUTION ARE:

(FOR CONTINUOUS TIME)
-1.000000000000000000 .000000000000000000
-2.000000000000000000 .000000000000000000

THE STABILIZING RICCATI SOLUTION IS:

200000000.512931183 3333333332798460259
.3333333332871560127 .249999999722222220

X1N = .20000000 + 001 H1N = .20000000 + 001 K(Z11) = .19402850 + 009

KB(X) = .30000000 + 001 KA(X) = .50000000-008

SEPEST = .20000000 + 001 SEP = .20000000 + 001

DO YOU WISH A RESIDUAL CALCULATION (Y OR N)

>Y

RSFLG = Y

RESIDUAL 1-NORM/SOLUTION 1-NORM = .129311838731697373-009

DO YOU WANT TO SEE THE RESIDUAL MATRIX (Y OR N)

>Y

RMFLG = Y

FIGURE A.2 Sample terminal session output listing (continued).
THE RESIDUAL MATRIX, BY ROWS:

```
-258623678237199783-001  -320515168225621316-010
-320515168225621316-010   .789603050697242101-017
```

DO YOU WANT TO TRY ITERATIVE IMPROVEMENT (Y OR N)

> Y

NFLAG = Y

ENTER MAXIMUM NUMBER OF NEWTON ITERATIONS

> 5

MAX. NO. OF ITERATIONS = 5

CONVERGED AFTER 2 ITERATIONS OF NEWTONS METHOD

CONVERGENCE CRITERIA 1

THE Refined STABILIZING SOLUTION IS:

```
200000000.499999999 .33333333277777778
.33333333277777778  .249999999722222223
```

THE CLOSED LOOP EIGENVALUES FOR THE STABILIZING RICCATI SOLUTION ARE:

```
-1.00000000499999998 .000000000000000000
-2.000000000000000000
```

RESIDUAL 1-NORM/SOLUTION 1-NORM = .4656612862345640-017

DO YOU WANT TO SEE THE RESIDUAL MATRIX (Y OR N)

> Y

RMFLG = Y

THE RESIDUAL MATRIX, BY ROWS:

```
.931322574615478516-009  .867361737988403547-018
.173472347597680709-017  -.23744271489084994-017
```

THE GAIN MATRIX IS:

```
20000.000049999998  .33333333277777779-004
```

FIGURE A.2 Sample terminal session output listing (continued).
THIS IS AN INTERACTIVE MAIN DRIVER PROGRAM FOR THE SOFTWARE
PACKAGE RICPACK. THIS DRIVER IS FOR SOLVING THE CONTINUOUS-
TIME OR DISCRETE-TIME GENERALIZED OPTIMAL REGULATOR PROBLEM.
THE PROGRAM PROMPTS FOR ALL NECESSARY INPUT. FOR THE PROBLEM
SPECIFICATIONS AND DETAILS ON THE SOLUTION METHOD, SEE

REF.: ARNOLD, W.F., "ON THE NUMERICAL SOLUTION OF
ALGEBRAIC MATRIX RICCATI EQUATIONS," PHD THESIS, USC,
DECEMBER 1983.

HISTORY:
THIS DRIVER WAS WRITTEN BY W.F. ARNOLD, NAVAL WEAPONS CENTER,
CODE 35104, CHINA LAKE, CA 93555, AS PART OF THE SOFTWARE
PACKAGE RICPACK, SEPTEMBER 1983.

SUBROUTINES CALLED:
BALANC, BALCOR, BCMRK, CMPRS, ELMHES, FBGAIN, HQR, LYPNCD,
MADD, MLINEQ, MOUT, MQFWO, MSCALE, MSUB, MULB, NEWT, RESID,
RICSOL, RINV, SAVE, SEPEST, SEQUIV, TRNATB

*****FUNCTION SUBPROGRAMS:
DOUBLE PRECISION DINRM
INTEGER IND(40), I, IBAL, INFO, IORD, J, L, L1, L2, L3, M, MAXIT,
X N, NN, NNPJ, NNPNI, NOUT, NPI, NPJ, NP1, NR, NRD, NRT
CHARACTER CFLAG, EFLAG, KFLAG, NFLAG, QFLAG, RDFLG, RFLAG,
X RMFLG, RSFLG, SFLAG
DOUBLE PRECISION A(20, 20), AS(20, 20), B(20, 20), C(20, 20),
X CQC(20, 20), E(20, 20), F(40, 40), G(40, 40), Q(20, 20),
X R(20, 20), RS(20, 20), S(20, 20), U(60, 60), WK(60, 20),
X Z(40, 40), ALFI(40), ALFR(40), BETA(60), CPERM(40),
X CSCALE(40), CA, CB, CL1N, CL1N, COND, C1N, DG, DGI, DGN,
X RSD, RTOL, SEP, SR, TEMP, X1N
LOGICAL TYPE
DATA NR, NRD, NRT / 20, 40, 60 /
RTOL = 1.0D + 06
INFO = 0
NOUT = 6
TYPE = .TRUE.
CFLAG = 'Y'
WRITE (NOUT, 800) NR

READ IN PROBLEM PARAMETERS

READ (5, *) N
WRITE (NOUT, 802) N
IF (N.GT.0) GOTO 10
TYPE = .FALSE.
N = - N
10 CONTINUE
IF (N.LE.NR .AND. N.NE.0) GOTO 20

112
WRITE (NOUT, 804)
STOP
20 CONTINUE
WRITE (NOUT, 806)
READ (5, *) M, L
WRITE (NOUT, 808) M, L
IF (M.EQ.0) WRITE (NOUT, 810)
IF (L.NE.0) GOTO 30
WRITE (NOUT, 812)
STOP
30 CONTINUE
WRITE (NOUT, 814)
READ (5, *) IORD
WRITE (NOUT, 816) IORD
IF (M.EQ.0) IORD = -1
WRITE (NOUT, 818)
READ (5, *) IBAL
WRITE (NOUT, 820) IBAL
IF (IBAL.NE.0 .AND. IBAL.NE.1) WRITE (NOUT, 822)
WRITE (NOUT, 824)
READ (5, 834) KFLAG
WRITE (NOUT, 826) KFLAG
IF (KFLAG.NE.'Y') GOTO 40
WRITE (NOUT, 828)
READ (5, *) DGI
WRITE (NOUT, 830) DGI
40 CONTINUE
NN = 2 * N
NNPM = NN + M
IF (IBAL.EQ.1) GOTO 50

C C READ IN PROBLEM MODEL MATRICES C
C
WRITE (NOUT, 832)
READ (5, 834) EFLAG
WRITE (NOUT, 836) EFLAG
IF (EFLAG.EQ.'Y') GOTO 60
50 CONTINUE
WRITE (NOUT, 838)
GOTO 80
60 CONTINUE
WRITE (NOUT, 840) N, N
DO 70 I = 1, N
   READ (5, *) (E(I, J), J = 1, N)
70 CONTINUE
WRITE (NOUT, 842)
CALL MOUT(NOUT, NR, N, N, E)
80 CONTINUE
WRITE (NOUT, 844) N, N
DO 90 I = 1, N
90 CONTINUE
READ (5, *) (A(I, J), J = 1, N)
90 CONTINUE
    CALL SAVE(NR, NR, N, N, A, AS)
    WRITE (NOUT, 846)
    CALL MOUT(NOUT, NR, N, N, A)
    IF (M.EQ.0) GOTO 110
    WRITE (NOUT, 848) N, M
    DO 100 I = 1, N
        READ (5, *) (B(I, J), J = 1, M)
    100 CONTINUE
    WRITE (NOUT, 850)
    CALL MOUT(NOUT, NR, N, M, B)
110 CONTINUE
    WRITE (NOUT, 852) L, N
    DO 120 I = 1, L
        READ (5, *) (C(I, J), J = 1, N)
    120 CONTINUE
    WRITE (NOUT, 854)
    CALL MOUT(NOUT, NR, L, N, C)
    CALL SAVE(NR, NR, L, N, C, COC)
    IF (M.EQ.0) GOTO 180
    WRITE (NOUT, 856)
    READ (5, 834) RFLAG
    WRITE (NOUT, 858) RFLAG
    IF (RFLAG.EQ.'Y') GOTO 130
    WRITE (NOUT, 860) GOTO 180
130 CONTINUE
    WRITE (NOUT, 862)
    READ (5, 834) RDFLG
    WRITE (NOUT, 864) RDFLG
    IF (RDFLG.NE.'Y') GOTO 160
    WRITE (NOUT, 866) M
    DO 150 J = 1, M
        DO 140 I = 1, M
            R(I, J) = 0.0D0
        140 CONTINUE
    150 CONTINUE
    READ (5, *) (R(I, I), I = 1, M)
    WRITE (NOUT, 868)
    CALL MOUT(NOUT, NR, M, M, R)
    GOTO 180
160 CONTINUE
    WRITE (NOUT, 870) M, M
    DO 170 I = 1, M
        READ (5, *) (R(I, J), J = 1, M)
    170 CONTINUE
    WRITE (NOUT, 868)
    CALL MOUT(NOUT, NR, M, M, R)
180 CONTINUE
IF (KFLAG.EQ.'Y') GOTO 230
WRITE (NOUT, 872)
READ (5, 834) QFLAG
WRITE (NOUT, 874) QFLAG
IF (QFLAG.EQ.'Y') GOTO 190
WRITE (NOUT, 876)
GOTO 210
190 CONTINUE
WRITE (NOUT, 878) L, L
DO 200 I = 1, L
   READ (5, *) (Q(I, J), J = 1, L)
200 CONTINUE
WRITE (NOUT, 880)
CALL MOUT(NOUT, NR, L, L, Q)
210 CONTINUE

* FORM THE MATRIX PRODUCT CT*Q*C AND STORE IN CQC *

CALL TRNATB(NR, NRD, L, N, C, G)
IF (QFLAG.NE.'Y') GOTO 220
CALL MULB(NR, NR, L, L, N, Q, CQC, CPERM)
220 CONTINUE
CALL MULB(NRD, NR, N, L, N, G, CQC, CPERM)
GOTO 320

* FORM ROBUSTNESS RECOVERY TERM *

230 CONTINUE
ITER = 1
WRITE (NOUT, 882)
READ (5, 834) QFLAG
WRITE (NOUT, 874) QFLAG
IF (QFLAG.EQ.'Y') GOTO 240
WRITE (NOUT, 884)
GOTO 260
240 CONTINUE
WRITE (NOUT, 886) N, N
DO 250 I = 1, N
   READ (5, *) (Q(I, J), J = 1, N)
250 CONTINUE
WRITE (NOUT, 880)
CALL MOUT(NOUT, NR, N, N, Q)
260 CONTINUE
CALL TRNATB(NR, NRD, L, N, C, G)
CALL MULB(NRD, NR, N, L, N, G, CQC, CPERM)
CALL SAVE(NR, NRT, N, N, CQC, WK)
CALL MSCALE(NR, N, N, N, DGI, CQC)
DG = DGI
IF (QFLAG.EQ.'Y') GOTO 280
DO 270 I = 1, N
CQC(I, I) = CQC(I, I) + 1.0D0
270 CONTINUE
GOTO 290
280 CONTINUE
CALL MADD(NR, NR, NR, N, N, CQC, Q, CQC)
290 CONTINUE
DO 310 I = 1, N
   DO 300 J = 1, N
      WK(J*N, I) = WK(J, I)
300 CONTINUE
310 CONTINUE
320 CONTINUE
   IF (M.EQ.0) GOTO 350
   WRITE (NOUT, 888)
   READ (5, 834) SFLAG
   WRITE (NOUT, 890) SFLAG
   IF (SFLAG.EQ.'Y') GOTO 330
   WRITE (NOUT, 892)
   GOTO 350
330 CONTINUE
   WRITE (NOUT, 894) N, M
   DO 340 I = 1, M
      READ (5, 0) (S(I, J), J = 1, M)
340 CONTINUE
   WRITE (NOUT, 896)
   CALL MOUT(NOUT, MR, N, M, S)
350 CONTINUE
C
C CALCULATE CO-ORDINATE BALANCING TRANSFORMATION IF REQUESTED
C
   IF (IBAL.NE.1) GOTO 360
   CALL BALCOR(NR, NRD, L, M, A, B, C, CQC, S, E, Q, Z, ALFI,
      ALFR, BETA, IND, INFO, SFLAG, TYPE)
   IF (INFO.EQ.0) GOTO 360
   WRITE (NOUT, 898)
   IF (INFO.EQ.1) WRITE (NOUT, 900)
   IF (INFO.EQ.-1) WRITE (NOUT, 902)
   IF (INFO.EQ.-2) WRITE (NOUT, 904)
   WRITE (NOUT, 906)
   READ (5, 834) RSFLG
   IF (RSFLG.NE.'Y') STOP
   IBAL = 0
360 CONTINUE
C
C TAKE INVERSE OF R MATRIX AND STORE IN Q
C
   IF (RFLAG.NE.'Y') GOTO 420
   IF (RDFLG.NE.'Y') GOTO 380
   DO 370 I = 1, M
      Q(I, I) = 1.0D0 / R(I, I)
370 CONTINUE
370 CONTINUE
    GOTO 420
380 CONTINUE
   CALL SAVE(NR, NRD, M, M, R, G)
   DO 400 J = 1, M
      DO 390 I = 1, M
         Q(I, J) = 0.0DO
   Q(J, J) = 1.0DO
   390 CONTINUE
   400 CONTINUE
   CALL MLINEQ(NRD, NR, M, M, Q, COND, IND, CPERM)
   WRITE (NOUT, 908) COND
   IF (COND.LT.RTOL) GOTO 410
   WRITE (NOUT, 910)
   READ (5, 834) CFLAG
   IF (CFLAG.NE.'Y') GOTO 430
   410 CONTINUE
   WRITE (NOUT, 912)
   420 CONTINUE
C
C SET UP MATRIX PENCIL FOR CONTINUOUS-TIME PROBLEM USING
C R INVERSE
C
   IF (SFLAG.EQ.'Y') CALL
      X SEQUIV(NR, NRD, M, M, A, B, CQC, S, Q, F, G, RDFLG, RFLAG)
   CALL RINV(NR, NRD, N, NN, M, E, A, B, CQC, Q, G, F, RS, CPERM,
      X RDFLG, RFLAG, EFLAG, IBAL, TYPE)
   GOTO 440
430 CONTINUE
C
C SET UP MATRIX PENCIL FOR CONTINUOUS-TIME PROBLEM USING
C COMPRESSION TECHNIQUE WHEN R IS ILL-CONDITIONED WITH
C RESPECT TO INVERSION
C
   WRITE (NOUT, 914)
   IF (.NOT.TYPE) CFLAG = 'Y'
   KFLAG = 'N'
   CALL CMPRS(NR, NRD, NRT, N, NN, NNPM, M, E, A, B, CQC, R, S, G,
      X F, U, WK, CPERM, CSCALE, BETA, EFLAG, SFLAG, IBAL, INFO)
   IF (INFO.NE.0) GOTO 720
440 CONTINUE
C
C CONVERT PENCIL TO DISCRETE-TIME PENCIL IF DISCRETE PROBLEM

NP1 = N + 1
   DO 460 J = NP1, NN
      DO 450 I = 1, NN
         TEMP = G(I, J)
         G(I, J) = F(I, J)
   450 CONTINUE
   460 CONTINUE
F(I, J) = - TEMP

450 CONTINUE
460 CONTINUE
470 CONTINUE
   IF (IBAL.EQ.1) CALL BCORBK(NR, NRD, M, N, A, B, CQC, S, Z)
C
   SAVE PENCIL IF ITERATING FOR ROBUSTNESS RECOVERY
C
   IF (KFLAG.NE.'Y') GOTO 500
   CALL SAVE(NRD, NRT, NN, NN, G, U)
DO 490 I = 1, N
   NPI = N + I
   NNPI = NN + I
DO 480 J = 1, N
   NPJ = N + J
   NNPJ = NN + J
   U(NWPJ, I) = F(J, I)
   U(NNPJ, NPI) = F(J, NPI)
   U(J, NNPI) = F(NPJ, I)
   U(NPJ, NNPI) = F(NPJ, NPI)
480 CONTINUE
490 CONTINUE
500 CONTINUE
C
   COMPUTE THE RICCATI SOLUTION
C
   CALL RICSOL(NR, NRD, NN, N, G, F, E, Z, ALFR, ALFI, BETA,
X       CPERM, CSSCALE, IND, IORD, IBAL, TYPE, EFLAG)
   IF (IND(1).NE.0) GOTO 690
   INFO = IND(2)
   IF (CPERM(1).EQ.1.OD + 20) CALL SAVE(NRD, NRD, N, N, Z, F)
   IF (IBAL.NE.1) GOTO 530
C
   ESTIMATE CONDITION OF CO-ORDINATE BALANCING TRANSFORMATION
   WITH RESPECT TO INVERSION
C
   DO 520 J = 1, N
      DO 510 I = 1, N
         G(I, J) = 0.0D0
510     CONTINUE
      G(J, J) = 1.0D0
520 CONTINUE
   CALL SAVE(NR, NRD, N, N, E, Z)
   CALL MLINEQ(NRD, NRD, N, N, Z, G, COND, IND, CSSCALE)
   WRITE (NOUT, 916) COND
530 CONTINUE
C
   OUTPUT THE SOLUTION
C
   WRITE (NOUT, 918)
IF (.NOT.TYPE) GOTO 560
WRITE (NOUT, 920)
DO 550 I = 1, NN
   IF (ALFR(I).GE.0.0D0) GOTO 550
   IF (BETA(I).NE.0.0D0) GOTO 540
   WRITE (NOUT, 922)
   GOTO 550
540 CONTINUE
   ALFR(I) = ALFR(I) / BETA(I)
   ALFI(I) = ALFI(I) / BETA(I)
   WRITE (NOUT, *) ALFR(I), ALFI(I)
550 CONTINUE
GOTO 580
560 CONTINUE
WRITE (NOUT, 924)
DO 570 I = 1, NN
   IF (BETA(I).EQ.0.0D0) GOTO 570
   IF (CDABS(ALFR(I)).GE.BETA(I)) GOTO 570
   ALFR(I) = ALFR(I) / BETA(I)
   ALFI(I) = ALFI(I) / BETA(I)
   WRITE (NOUT, *) ALFR(I), ALFI(I)
570 CONTINUE
580 CONTINUE
IF (INFO.NE.0) GOTO 700
IF (CPERM(I).EQ.1.0D + 20) GOTO 710
IF (M.EQ.0) WRITE (NOUT, 926)
IF (IORD.EQ.-1) WRITE (NOUT, 928)
IF (IORD.EQ.0) WRITE (NOUT, 930)
IF (IORD.EQ.1) WRITE (NOUT, 932)
CALL MOUT(NOUT, NR, N, N, F)

COMPUTE CONDITION ESTIMATES
X1N = D1NRM(NRD, N, N, F)
CALL SAVE(NR, NR, N, N, CQC, C)
   IF (SFLAG.EQ.'Y') CALL
   X SEQUIV(NR, NRD, M, N, AS, B, C, S, Q, Z, G, RDLFG, RFLAG)
   X CSR = D1NRM(NR, N, N, C)
   CALL FBIAGN(NR, NRD, NRD, N, M, A, B, E, R, Q, S, F, Z, G,
   X CSR, IND, EFLAG, RDLFG, RFLAG, SFLAG, TYPE)
   WRITE (NOUT, 934) X1N, C1N, CPERM(I)
   CALL MULB(NR, NRD, M, N, N, B, Z, CSR)
   CALL MSUB(NR, NRD, NRD, N, N, A, Z, Z)
   IF (TYPE) GOTO 600
   CL1N = D1NRM(NRD, N, N, Z)
   CALL TRNAT(NRD, N, Z)
   CL1N = D1NRM(NRD, N, N, Z)
   CALL BALANC(NRD, N, Z, L1, L2, CSR)
   CALL ELMHES(NRD, N, L1, L2, Z, IND)
   CALL HQR(NRD, N, L1, L2, Z, ALFR, ALFI, L3)

119
IF (L3.NE.0) WRITE (NOUT, 936)
SR = 0.0D0
DO 590 I = 1, N
SR = DMAX1(SR, DSQRT(ALFR(I)*ALFR(I) + ALFI(I)*ALFI(I)))
590 CONTINUE
TEMP = 1.0D0 - CL1N * CLT1N
IF (TEMP.LE.0.0D0) TEMP = 1.0D0 - SR * SR
CA = DFLOAT(N) * (C1N / X1N + 2.0D0 * CL1N * CLT1N) / TEMP
WRITE (NOUT, 938) CA, SR, CL1N, CLT1N
GOTO 620
600 CONTINUE
L1 = 0
L2 = 1
CALL LYPCMD(NRD, NRD, N, Z, F, G, ALFR, ALFI, BETA, L1, L2)
CALL SEPEST(NRD, N, Z, G, SEP, L1)
IF (L1.NE.0) WRITE (NOUT, 940)
TEMP = DABS(ALFR(1))
DO 610 I = 2, N
TEMP = DMIN1(TEMP, DABS(ALFR(I)))
610 CONTINUE
CB = (C1N / X1N + 2.0D0 * D1NRM(NR, N, N, AS) + X1N *
X D1NRM(NR, N, N, RS)) / SEP
CA = C1N / (X1N * SEP)
TEMP = 2.0D0 * TEMP
WRITE (NOUT, 942) CB, CA, TEMP, SEP
620 CONTINUE
C C SET UP FOR ANOTHER ROBUSTNESS RECOVERY ITERATION C C IF (KFLAG.NE.'Y') GOTO 680
WRITE (NOUT, 944)
READ (5, 834) KFLAG
WRITE (NOUT, 826) KFLAG
IF (KFLAG.NE.'Y') GOTO 650
WRITE (NOUT, 946)
READ (5, *) DGN
WRITE (NOUT, 830) DGN
IF (IBAL.EQ.1 .AND. ITER.EQ.1).CALL XMQFWO(NRT, NR, N, WK, E, CPERM)
ITER = ITER + 1
DG = DG - DGN
DO 640 I = 1, N
NPI = N + I
NNPI = NN + I
DO 630 J = 1, N
NPJ = N + J
NNPJ = NN + J
U(NPJ, I) = U(NPJ, I) + DG * WK(J, I)
F(J, I) = U(NNPJ, I)
F(J, NPI) = U(NNPJ, NPI)
630 CONTINUE
640 CONTINUE
F(NPJ, I) = U(J, NMPI)
F(NPJ, MPI) = U(NPJ, NMPI)

630 CONTINUE
640 CONTINUE
   CALL SAVE(NRT, NRD, NM, NN, U, G)
   DG = DGN
   GOTO 500
650 CONTINUE
   DG = DG - DGI
   DO 670 I = 1, N
       DO 660 J = 1, N
           CQC(J, I) = CQC(J, I) + DG * WK(J+N, I)
       660 CONTINUE
   670 CONTINUE
680 CONTINUE
   IF (CFLAG.NE.'Y') STOP
   C
   COMPUTE THE RESIDUAL
   C
   WRITE (NOUT, 948)
   READ (5, 834) RSFLG
   WRITE (NOUT, 950) RSFLG
   IF (RSFLG.NE.'Y') GOTO 730
   WRITE (NOUT, 952) RSD
   WRITE (NOUT, 954)
   READ (5, 834) RMFLG
   WRITE (NOUT, 956) RMFLG
   IF (RMFLG.NE.'Y') GOTO 730
   WRITE (NOUT, 958)
   CALL MOUT(NOUT, NR, N, N, RS)
   GOTO 730
690 CONTINUE
   WRITE (NOUT, 960)
   STOP
700 CONTINUE
   WRITE (NOUT, 962)
   STOP
710 CONTINUE
   WRITE (NOUT, 964)
   CALL MOUT(NOUT, NRD, N, N, F)
   STOP
720 CONTINUE
   WRITE (NOUT, 966)
   STOP
730 CONTINUE
   IF (IORD.NE.-1) STOP
   C

121
APPLY NEWTON'S METHOD FOR ITERATIVE IMPROVEMENT

WRITE (NOUT, 968)
READ (5, 834) NFLAG
WRITE (NOUT, 970) NFLAG
IF (NFLAG.NE.'Y') GOTO 750
WRITE (NOUT, 972)
READ (5, *) MAXIT
WRITE (NOUT, 974) MAXIT
CALL SAVE(NRD, NRT, N, N, F, U)
CALL NEWT(NR, NR, NRD, NRT, N, M, E, A, B, CQC, R, S, Q, RS,
X U, G, F, Z, ALFR, ALFI, BETA, IND, RTOL, EFLAG, RFLAG,
X RDFLG, SFLAG, SEP, TYPE, MAXIT, INFO, NOUT)
IF (INFO.NE.0) GOTO 760
WRITE (NOUT, 976) MAXIT
WRITE (NOUT, 978) IND(1)
CALL RESID(NR, NR, NRD, NRT, N, M, E, A, B, CQC, R, S, Q, RS,
X U, G, Z, CPERM, IND, RTOL, EFLAG, RFLAG, RDFLG, SFLAG,
X RSD, TYPE, NOUT)
WRITE (NOUT, 980)
CALL MOUT(NOUT, NRT, N, N, U)
DO 740 I = 1, N
WRITE (NOUT, 982) ALFR(I), ALFI(I)
740 CONTINUE
WRITE (NOUT, 952) RSD
CALL SAVE(NRT, NRD, N, N, U, F)
WRITE (NOUT, 954)
READ (5, 834) RMFLG
WRITE (NOUT, 956) RMFLG
IF (RMFLG.NE.'Y') GOTO 750
WRITE (NOUT, 958)
CALL MOUT(NOUT, NR, N, N, RS)
750 CONTINUE

CALCULATE OPTIMAL FEEDBACK GAIN MATRIX

CALL FBGAIN(NR, NRD, NRD, N, M, A, B, E, R, Q, S, F, Z, G,
X CPERM, IND, EFLAG, RDFLG, RFLAG, SFLAG, TYPE)
IF (CPERM(1).GT.RTOL) WRITE (NOUT, 984) CPERM(1)
WRITE (NOUT, 982)
CALL MOUT(NOUT, NRD, M, N, Z)
STOP

760 CONTINUE
IF (INFO.EQ.-3) GOTO 770
WRITE (NOUT, 986) INFO
CALL MOUT(HOUT, NRT, N, N, U)
WRITE (NOUT, 988) BETA(1)
CALL RESID(NR, NR, NRD, NRT, N, M, E, A, B, CQC, R, S, Q, RS,
X U, G, Z, CPERM, IND, RTOL, EFLAG, RFLAG, RDFLG, SFLAG,
X RSD, TYPE, NOUT)
GOTO 740
770 CONTINUE
WRITE (NOUT, 990)
STOP
800 FORMAT(/' GENERALIZED ALGEBRAIC MATRIX RICCATI EQUATION SOLVER',
X //, ' ENTER SYSTEM ORDER "N": + FOR CONTINUOUS TIME PROBLEM',
X ' - FOR DISCRETE', /, ' MAXIMUM ORDER =', I3, /)
802 FORMAT (/' N =', I3/)
804 FORMAT (/' ORDER EXCEEDS MAXIMUM *** EXECUTION TERMINATED', /)
806 FORMAT (/' ENTER NUMBER OF SYSTEM INPUTS "M": ', /,
X ' AND NUMBER OF SYSTEM OUTPUTS "L": ', /)
808 FORMAT (/' M =', I3, 5X, ' L =', I3/)
810 FORMAT (/' SYSTEM HAS NO INPUTS, RICCATI EQUATION DEGENERATES',
X ' TO A LYAPUNOV EQUATION', /)
812 FORMAT (/' SYSTEM HAS NO OUTPUT *** EXECUTION TERMINATED', /)
814 FORMAT (/' ENTER FLAG FOR DESIRED SOLUTION: ', /,
X ' -1 FOR STABILIZING SOLUTION', /, ' 0 FOR ANY SOLUTION', /,
X ' +1 FOR DESTABILIZING SOLUTION', /)
816 FORMAT (/' IORD =', I2/)
818 FORMAT (/' ENTER BALANCING FLAG: 0 FOR WARD BALANCING', /,
X ' 1 FOR CO-ORDINATE BALANCING', /,
X ' 2 FOR NO BALANCING', /)
820 FORMAT (/' IBAL =', I2/)
822 FORMAT (/' NO BALANCING ATTEMPTED', /)
824 FORMAT
X (' DO YOU WISH TO ITERATE FOR ROBUSTNESS RECOVERY',
X ' (Y OR N)', /)
826 FORMAT (/' KFLAG = ', A1/)
828 FORMAT (/' ENTER INITIAL GAMMA PARAMETER', /)
830 FORMAT (/' DG = ', D26.18/)
832 FORMAT (/' DO YOU WISH TO ENTER AN "E" MATRIX (Y OR N)', /,
X ' DEFAULT IS E = IDENTITY MATRIX', /)
834 FORMAT (1A1)
836 FORMAT (/' EFLAG = ', A1/)
838 FORMAT (/' USING DEFAULT: "E" = IDENTITY', /)
840 FORMAT (/' ENTER THE ', I3, ' X ', I3, ' MATRIX "E" BY ROWS', /)
842 FORMAT (/' THE "E" MATRIX IS:', /)
844 FORMAT
X (' ENTER THE ', I3, ' X ', I3, ' SYSTEM MATRIX "A" BY ROWS', /)
846 FORMAT (/' THE "A" MATRIX IS:', /)
848 FORMAT
X (' ENTER THE ', I3, ' X ', I3, ' INPUT MATRIX "B" BY ROWS', /)
850 FORMAT (/' THE "B" MATRIX IS:', /)
852 FORMAT
X (' ENTER THE ', I3, ' X ', I3, ' OUTPUT MATRIX "C" BY ROWS', /)
854 FORMAT (/' THE "C" MATRIX IS:', /)
856 FORMAT (/' DO YOU WISH TO ENTER A CONTROL WEIGHTING MATRIX "R"',
X ' (Y OR N)', /, ' DEFAULT IS R = IDENTITY MATRIX', /)
858 FORMAT (/' RFLAG = ', A1/)
NWC TP 6521

860 FORMAT (/' USING DEFAULT: "R" = IDENTITY'/)
862 FORMAT (/' DO YOU WISH TO ENTER A DIAGONAL "R" MATRIX (Y OR N)' /)
864 FORMAT (/' RDFLG = ', A1/)
866 FORMAT (/' ENTER THE ', I3, ' DIAGONAL ELEMENTS AS A ROW'/)
868 FORMAT (/' THE "R" MATRIX IS:' /)
870 FORMAT (/' ENTER THE ', I3, ' X ', I3, ' INPUT WEIGHTING MATRIX',
X ' "R" BY ROWS'/)
872 FORMAT (/' DO YOU WISH TO ENTER AN OUTPUT WEIGHTING MATRIX "Q"',
X ' (Y OR N)', '/ ', ' DEFAULT IS Q = IDENTITY MATRIX'/)
874 FORMAT (/' QFLAG = ', A1/)
876 FORMAT (/' USING DEFAULT: "Q" = IDENTITY'/)
878 FORMAT (/' ENTER THE ', I3, ' X ', I3, ' OUTPUT WEIGHTING MATRIX "Q"',
X ' BY ROWS'/)
880 FORMAT (/' THE "Q" MATRIX IS:' /)
882 FORMAT (/' DO YOU WISH TO ENTER A STATE WEIGHTING MATRIX "Q"',
X ' (Y OR N)', '/ ', ' DEFAULT IS Q = IDENTITY MATRIX'/)
884 FORMAT (/' USING DEFAULT: "Q" = IDENTITY'/)
886 FORMAT (/' ENTER THE ', I3, ' X ', I3, ' STATE WEIGHTING MATRIX "Q"',
X ' BY ROWS'/)
888 FORMAT (/' DO YOU WISH TO ENTER A STATE/INPUT CROSS-WEIGHTING',
X ' MATRIX "S" (Y OR N)', '/ ', ' DEFAULT IS S = ZERO MATRIX'/)
890 FORMAT (/' SFLAG = ', A1/)
892 FORMAT (/' USING DEFAULT: "S" = ZERO MATRIX'/)
894 FORMAT (/' ENTER THE ', I3, ' X ', I3, ' CROSS-WEIGHTING MATRIX "S"',
X ' BY ROWS'/)
896 FORMAT (/' THE "S" MATRIX IS:' /)
898 FORMAT (/' CO-ORDINATE BALANCING UNSUCCESSFUL BECAUSE:' /)
900 FORMAT (/' CAN NOT COMPUTE SOLUTION TO LYAPUNOV EQUATION'/)
902 FORMAT (/' CONTROLLABILITY GRAMMIAN NOT NUMERICALLY P.D.'/)
904 FORMAT (/' CONVERGENCE FAILURE IN EIGENVALUE ROUTINE'/)
906 FORMAT (/' DO YOU WISH TO CONTINUE WITH WARD BALANCING (Y OR N)' /)
908 FORMAT X (/' INVERSION CONDITION ESTIMATE FOR "R" MATRIX = ', D26.18, '/)
910 FORMAT (/' DO YOU WISH TO CONTINUE USING "R" INVERSE (Y OR N)' /)
912 FORMAT (/' PROCEEDING WITH SOLUTION USING "R" INVERSE'/)
914 FORMAT (/' PROCEEDING WITH COMPRESSION TECHNIQUE'/)
916 FORMAT (/' A CONDITION ESTIMATE FOR THE CO-ORDINATE BALANCING',
X ' TRANSFORMATION IS:', /, D26.18 /)
918 FORMAT X (/' THE CLOSED LOOP EIGENVALUES FOR THE STABILIZING RICCATI',
X ' SOLUTION ARE:' /)
920 FORMAT (/' (FOR CONTINUOUS TIME)' /)
922 FORMAT (/' INFINITE EIGENVALUE'/)
924 FORMAT (/' (FOR DISCRETE TIME)' /)
926 FORMAT (/' THE LYAPUNOV SOLUTION IS:' /)
928 FORMAT (/' THE STABILIZING RICCATI SOLUTION IS:' /)
930 FORMAT (/' A RICCATI SOLUTION IS:' /)
932 FORMAT (/' THE DESTABILIZING RICCATI SOLUTION IS'/)
934 FORMAT 
X (/' X1N = ', D15.8, 5X, 'C1N = ', D15.8, 5X, 'K(Z11) = ', D15.8 /)
NWC TP 6521

936 FORMAT ('ERROR IN HQR/')
938 FORMAT ('KA(X) =', D15.8, 10X, 'SR =', D15.8, 10X, 'C1N =', D15.8)
940 FORMAT ('ERROR IN SEPEST/')
942 FORMAT ('KB(X) =', D15.8, 10X, 'KA(X) =', D15.8, 10X, 'SEPEST =', D15.8)
944 FORMAT ('DO YOU WISH TO ENTER ANOTHER GAMMA (Y OR N)/')
946 FORMAT ('ENTER NEW VALUE FOR GAMMA/')
948 FORMAT ('DO YOU WISH A RESIDUAL CALCULATION (Y OR N)/')
950 FORMAT ('RSFLG = ', A1)
952 FORMAT ('RESIDUAL 1-NORM/SOLUTION 1-NORM = ', D26.18)
954 FORMAT ('DO YOU WANT TO SEE THE RESIDUAL MATRIX (Y OR N)/')
956 FORMAT ('RMFLG = ', A1)
958 FORMAT ('THE RESIDUAL MATRIX, BY ROWS:/')
960 FORMAT ('MORE THAT 50 ITERATIONS REQUIRED BY QZITW', '/

962 FORMAT ('CONVERGENCE FAILURE IN ORDER ROUTINE', '/

964 FORMAT ('SCHUR VECTOR MATRIX SINGULAR TO WORKING PRECISION', '/

966 FORMAT ('COMPRESSION FAILURE *** EXECUTION TERMINATED/')
968 FORMAT ('DO YOU WANT TO TRY ITERATIVE IMPROVEMENT (Y OR N)/')
970 FORMAT ('NFLAG = ', A1)
972 FORMAT ('ENTER MAXIMUM NUMBER OF NEWTON ITERATIONS/')
974 FORMAT ('MAX. NO. OF ITERATIONS = ', I4)
976 FORMAT ('CONVERGED AFTER', I3, ' ITERATIONS OF NEWTONS METHOD/')
978 FORMAT ('CONVERGENCE CRITERIA', I2)
980 FORMAT ('THE REFINED STABILIZING SOLUTION IS:/')
982 FORMAT ('THE GAIN MATRIX IS:/')
984 FORMAT ('R + GT*X*G ILL-CONDITIONED WRT INVERSION', '/

986 FORMAT ('NEWTON ITERATION FAILED, INFO = ', I3, /

988 FORMAT ('THE 1-NORM OF THE ERROR IS: ', D26.18)
990 FORMAT ('SORRY, PROGRAM NOT ABLE TO PERFORM NEWTON ITERATION', /

END
SUBROUTINE BALCOR (NR, NRZ, L, M, N, A, B, C, CQC, S, T, WK, Z, WK1, WK2, 
    WK3, IPVT, ISTAB, SFLAG, TYPE)

    *****PARAMETERS:
    INTEGER NR, NRZ, L, M, N, IPVT(N), ISTAB
    CHARACTER SFLAG
    DOUBLE PRECISION A(NR, N), B(NR, M), C(NR, N), CQC(NR, N), S(NR, M), 
    T(NR, N), WK(NR, N), Z(NRZ, N), WK1(N), WK2(N), WK3(N)
    LOGICAL TYPE

    *****LOCAL VARIABLES:
    INTEGER I, J, NPI, NPF

    *****FORTRAN FUNCTIONS:
    NONE.

    *****SUBROUTINES CALLED:
    BCORBK, BLCRDC, BLCRDD, MMUL, MQFWO, MULB, TRNATB

    *****PURPOSE:
    GIVEN THE MODEL MATRICES A, B AND C FOR A FIRST ORDER LINEAR
    MODEL IN STATE SPACE FORM, THIS SUBROUTINE CALCULATES A
    BALANCING TRANSFORMATION, T, SUCH THAT IF T WAS APPLIED TO THE
    MODEL AS A CHANGE OF COORDINATES, THE REACHABILITY AND
    OBSERVABILITY GRAMMIIANS WOULD BE EQUAL AND DIAGONAL. HOWEVER,
    T IS ACTUALLY APPLIED TO THE MODEL MATRICES IN A SPECIAL WAY
    FOR SOLUTION OF THE OPTIMAL REGULATOR PROBLEM BY RICPACK.

    REF.: ARNOLD, W.F., "ON THE NUMERICAL SOLUTION OF
    ALGEBRAIC MATRIX RICCATI EQUATIONS," PHD THESIS, USC,
    DECEMBER 1983.

    *****PARAMETER DESCRIPTION:

    ON INPUT:

    NR INTEGER
    ROW DIMENSION OF THE ARRAYS CONTAINING THE A, B, 
    C, CQC, S, T AND WK MATRICES AS DECLARED IN THE
    MAIN CALLING PROGRAM DIMENSION STATEMENT;

    NRZ INTEGER
    ROW DIMENSION OF THE ARRAY CONTAINING THE Z MATRIX
    AS DECLARED IN THE MAIN CALLING PROGRAM DIMENSION
    STATEMENT;

    L INTEGER

    126
ROW DIMENSION OF THE C MATRIX;

M INTEGER
COLUMN DIMENSION OF THE B AND S MATRICES;

N INTEGER
ORDER OF THE SQUARE MATRICES A, CQC AND T
COLUMN DIMENSION OF THE C MATRIX;

A REAL(NR,N)
MODEL SYSTEM MATRIX, ALTERED BY THIS ROUTINE;

B REAL(NR,M)
MODEL INPUT MATRIX;

C REAL(NR,N)
MODEL OUTPUT MATRIX, ALTERED BY THIS ROUTINE;

CQC REAL(NR,N)
MATRIX PRODUCT CT*Q*C WHERE T DENOTES MATRIX
TRANSPOSE, ALTERED BY THIS ROUTINE;

S REAL(NR,M)
STATE - INPUT CROSS-WEIGHTING MATRIX, ALTERED BY
THIS ROUTINE;

WK REAL(NR,N)
SCRATCH ARRAY OF SIZE AT LEAST N BY N;

WK1, WK2, WK3
REAL(N)
WORKING VECTORS OF SIZE AT LEAST N;

IPVT INTEGER(N)
WORKING VECTOR OF SIZE AT LEAST N;

SFLAG CHARACTER
FLAG SET TO 'Y' IF S IS OTHER THAN THE ZERO MATRIX;

TYPE LOGICAL
= .TRUE. FOR A CONTINUOUS-TIME MODEL
= .FALSE. FOR A DISCRETE-TIME MODEL.

ON OUTPUT:

A CONTAINS THE MATRIX PRODUCT F*T;

CQC CONTAINS THE MATRIX PRODUCT (C*T)-TRANSPOSE*Q*C*T;
**CONTAINS THE MATRIX PRODUCT T-TRANSPOSE*S;**

**REAL(NR,N)**

**CONTAINS THE BALANCING TRANSFORMATION, T;**

**REAL(NRZ,N)**

2N BY (N+M) MATRIX WITH THE ORIGINAL A MATRIX STORED
IN THE UPPER LEFT N BY N BLOCK, CQC IN THE LOWER LEFT
N BY N BLOCK, S IN THE UPPER RIGHT N BY M BLOCK AND
B IN THE LOWER RIGHT N BY M BLOCK;

**INTEGER**

**ERROR FLAG WITH THE FOLLOWING MEANINGS**

- **ZERO,** NORMAL RETURN
- **NONZERO,** A BALANCING TRANSFORMATION COULD NOT BE
  CALCULATED AND THE A, CQC AND S MATRICES ARE
  RETURNED UNALTERED.

**ALGORITHM NOTES:**

NONE.

**HISTORY:**

THIS SUBROUTINE WAS WRITTEN BY W.F. ARNOLD, NAVAL WEAPONS
CENTER, CODE 35104, CHINA LAKE, CA 93555, AS PART OF THE
SOFTWARE PACKAGE RICPACK, SEPTEMBER 1983.
SUBROUTINE RICSOL(NR,NRD,NN,N,G,F,E,Z,ALFR,ALFI,BETA,CPERM, 
X CSSCALE,IND,IORD,IBAL,TYPE,EFLAG)

****PARAMETERS:
INTEGER NR,NRD,NN,N,IND(2NN),IORD,IBAL
CHARACTER EFLAG
DOUBLE PRECISION G(NRD,NN),F(NRD,NN),E(NR,N),Z(NRD,NN), 
X ALFR(2NN),ALFI(2NN),BETA(2NN),CPERM(2NN),CSSCALE(2NN)
LOGICAL TYPE

****LOCAL VARIABLES:
INTEGER I,IERR,IFAIL,IGH,J,LOWNPJ
DOUBLE PRECISION COND,EPSEPS1
LOGICAL MATZ

****FORTRAN FUNCTIONS:
NONE.

****SUBROUTINES CALLED:
BALGBK, BALGEN, MLINEQ, MULB, ORDER, QZHESW, QZITW, QZVAL

****PURPOSE:
GIVEN THE 2N BY 2N MATRIX PENCIL

\[ \lambda \mathbf{F} - \mathbf{G} \]

THIS SUBROUTINE FINDS THE ORTHOGONAL MATRIX \( \mathbf{Z} \) SUCH THAT

\[ \mathbf{Q} \cdot (\lambda \mathbf{F} - \mathbf{G}) \cdot \mathbf{Z} \]

IS IN GENERALIZED ORDERED REAL SCHUR FORM. FURTHERMORE, THE
UPPER LEFT N BY N BLOCK OF THE TRANSFORMED PENCIL CONTAINS
THE EIGENVALUES SPECIFIED BY THE PARAMETER IORD. THE
SUBROUTINE THEN SOLVES THE LINEAR SYSTEM

\[ \mathbf{X} \cdot \mathbf{E} \cdot \mathbf{Z}_{11} = \mathbf{Z}_{21} \]

FOR \( \mathbf{X} \), WHERE \( \mathbf{Z}_{11} \) AND \( \mathbf{Z}_{21} \) ARE THE UPPER AND LOWER LEFT N BY N
BLOCKS OF \( \mathbf{Z} \).

REF.: ARNOLD, W.F., "ON THE NUMERICAL SOLUTION OF
ALGEBRAIC MATRIX RICCATI EQUATIONS," PHD THESIS, USC,
DECEMBER 1983.

****PARAMETER DESCRIPTION:

ON INPUT:
NR INTEGER
ROW DIMENSION OF THE ARRAY CONTAINING THE MATRIX E
AS DECLARED IN THE MAIN CALLING PROGRAM DIMENSION
STATEMENT;

NRD INTEGER
ROW DIMENSION OF THE ARRAYS CONTAINING THE MATRICES
G, F AND Z AS DECLARED IN THE MAIN CALLING PROGRAM
DIMENSION STATEMENT;

NN INTEGER
ORDER OF THE SQUARE MATRICES G AND F;

N INTEGER
ORDER OF THE SQUARE MATRIX E;

G REAL(NRD,NN)
PENCIL MATRIX CORRESPONDING TO THE GENERALIZED RICCATI
PROBLEM, WHOSE CONTENTS ARE ALTERED BY THIS ROUTINE;

F REAL(NRD,NN)
PENCIL MATRIX CORRESPONDING TO THE GENERALIZED RICCATI
PROBLEM, WHOSE CONTENTS ARE ALTERED BY THIS ROUTINE;

E REAL(NR,N)
DESCRIPTOR MATRIX OR CO-ORDINATE BALANCING MATRIX
AS SPECIFIED BY THE PARAMETER IBAL;

CPERM REAL(NN)
WORKING VECTOR OF SIZE AT LEAST NN;

CScale REAL(NN)
WORKING VECTOR OF SIZE AT LEAST NN;

IND INTEGER(NN)
WORKING VECTOR OF SIZE AT LEAST NN;

IORD INTEGER
PARAMETER SPECIFYING THE SPECTRUM OF THE UPPER LEFT N
BY N BLOCK OF THE ORDERED REAL SCHUR FORM AS FOLLOWS:
I = 1 GENERALIZED EIGENVALUES WHOSE MAGNITUDE IS LESS
THAN UNITY
I = 0 ANY ORDER
I = -1 GENERALIZED EIGENVALUES WHOSE REAL PARTS ARE
LESS THAN ZERO;

IBAL INTEGER
PARAMETER SPECIFYING THE BALANCING BEING EMPLOYED AS
FOLLOWS:
= 0  WARD BALANCING AND E IS A DESCRIPTOR MATRIX
= 1  CO-ORDINATE BALANCING AND E IS THE BALANCING
    TRANSFORMATION
= 2  NO BALANCING AND E IS A DESCRIPTOR MATRIX;

TYPE  LOGICAL
= .TRUE.  FOR CONTINUOUS-TIME SYSTEM
= .FALSE.  FOR DISCRETE-TIME SYSTEM;

EFLAG  CHARACTER
FLAG SET TO 'Y' IF E IS A DESCRIPTOR MATRIX THAT IS
OTHER THAN THE IDENTITY MATRIX.

ON OUTPUT:

F  CONTAINS THE SOLUTION MATRIX X COMPUTED AS SHOWN
    ABOVE;

Z  REAL(NRD,NN)
CONTAINS THE MATRIX PRODUCT
  ( E  0 ) ( Z11  Z12 )
  ( 0  I ) ( Z21  Z22 )
WHERE Z IS THE ORTHOGONAL TRANSFORMATION MATRIX
DESCRIBED ABOVE;

ALFR  REAL(NN)
REAL PARTS OF THE DIAGONAL ELEMENTS THAT WOULD RESULT
IF THE Q AND Z TRANSFORMATIONS WERE APPLIED TO THE
G MATRIX SUCH THAT IT WOULD BE REDUCED COMPLETELY TO
TRIANGULAR FORM AND THE DIAGONAL ELEMENTS OF THE
TRANSFORMED F MATRIX (ALSO TRIANGULAR) WOULD BE REAL
AND POSITIVE;

ALFI  REAL(NN)
IMAGINARY PARTS OF THE DIAGONAL ELEMENTS THAT WOULD
RESULT IF THE Q AND THE Z TRANSFORMATIONS WERE
APPLIED TO THE G MATRIX SUCH THAT IT WOULD BE REDUCED
COMPLETELY TO TRIANGULAR FORM AND THE DIAGONAL;
ELEMENTS OF THE TRANSFORMED F MATRIX (ALSO TRIANGULAR)
WOULD BE REAL AND POSITIVE. NONZERO VALUES OCCUR IN
PAIRS; THE FIRST MEMBER IS POSITIVE AND THE SECOND
MEMBER IS NEGATIVE;

BETA  REAL(NN)
REAL NONNEGATIVE DIAGONAL ELEMENTS OF F THAT WOULD
RESULT IF G WERE REDUCED COMPLETELY TO TRANGULAR
FORM; THE GENERALIZED EIGENVALUES ARE THEN GIVEN BY
THE RATIOS ((ALFR + I*ALFI)/BETA);

CPERM(1)
CONDITION ESTIMATE OF E*Z11 WITH RESPECT TO INVERSION;

IND(1) ERROR FLAG AS FOLLOWS
= 0  INDICATES NORMAL RETURN
= NONZERO IF MORE THAN 50 ITERATIONS WERE REQUIRED TO
DETERMINE THE DIAGONAL BLOCKS FOR THE QUASITRIANGULAR
FORM;

IND(2) ERROR FLAG AS FOLLOWS
= 0  INDICATES NORMAL RETURN
= 1  INDICATES ATTEMPTED REORDERING FAILED.

*****ALGORITHM NOTES:
NONE.

*****HISTORY:
THIS SUBROUTINE WAS WRITTEN BY W.F. ARNOLD, NAVAL WEAPONS
CENTER, CODE 35104, CHINA LAKE, CA 93555, AS PART OF THE
SOFTWARE PACKAGE RICPACK, SEPTEMBER 1983.
SUBROUTINE NEWT(NR,NRR, NRW, NRX, N, M, E, A, B, CQC, R, S, RI, RSDM, X, W1,
W2, W3, ALFR, ALFI, WK1, IPVT, RTOL, EFLAG, RFLAG,
RDFLG, SFLAG, SEP, TYPE, MAXIT, INFO, NOUT)

*****PARAMETERS:
INTEGER NR, NRR, NRW, NRX, N, M, IPVT(N), MAXIT, INFO, NOUT
CHARACTER EFLAG, RFLAG, RDFLG, SFLAG
DOUBLE PRECISION E(NR,N), A(NR,N), B(NR,M), CQC(NR,N), R(NR,M),
S(NR,M), RI(NR,M), RSDM(NRR,N), X(NRX,N),
W1(NRW,N), W2(NRW,N), W3(NRW,N), ALFR(N),
ALFI(N), WK1(N), RTOL, SEP
LOGICAL TYPE

*****LOCAL VARIABLES:
INTEGER I, IER1, IER2, ITER, J
DOUBLE PRECISION DPN1, EPS, E1N, H1N, TOL, T1, T2, T3, X1N

*****FORTRAN FUNCTIONS:
DOUBLE PRECISION DSQRT

*****FUNCTION SUBPROGRAMS:
DOUBLE PRECISION D1NRM

*****SUBROUTINES CALLED:
FBGAIN, LYPCND, LYPDSD, MADD, MMUL, MQF, MScale, MSUB, MUL, SEPEST, TRNATB

------------------------------------------------------------------------------------------------

*****PURPOSE:
GIVEN THE MODEL MATRICES FOR THE CONTINUOUS- OR DISCRETE-TIME
OPTIMAL REGULATOR PROBLEM THAT RESULTS IN A GENERALIZED
ALGEBRAIC RICCATI EQUATION (GARE), AND AN INITIAL GUESS FOR
THE SOLUTION TO THE GARE, THIS SUBROUTINE APPLIES A NEWTON
TYPE ITERATIVE REFINEMENT PROCEDURE. TO GUARANTEE
CONVERGENCE, THE INITIAL GUESS MUST STABILIZE THE CLOSED LOOP
SYSTEM MATRIX E**-1*(A - B*K), WHERE

K = (R**-1)*(B*E + ST)  \hspace{1cm} \text{CONTINUOUS}

K = ((R + B*E*B)**-1)*(B*E*A + ST)  \hspace{1cm} \text{DISCRETE}

AND T DENOTES MATRIX TRANSPOSE.

REF.: ARNOLD, W.F., "ON THE NUMERICAL SOLUTION OF
ALGEBRAIC MATRIX RICCATI EQUATIONS," PHD THESIS, USC,
DECEMBER 1983.

*****PARAMETER DESCRIPTION:
ON INPUT:

NR INTEGER
ROW DIMENSION OF THE ARRAYS CONTAINING THE MATRICES E, A, B, CQC, R, S AND RI AS DECLARED IN THE MAIN CALLING PROGRAM DIMENSION STATEMENT;

NRR INTEGER
ROW DIMENSION OF THE ARRAY CONTAINING THE MATRIX RSDM AS DECLARED IN THE MAIN CALLING PROGRAM DIMENSION STATEMENT;

NRW INTEGER
ROW DIMENSION OF THE ARRAYS CONTAINING THE MATRICES W1, W2 AND W3 AS DECLARED IN THE MAIN CALLING PROGRAM DIMENSION STATEMENT;

NRX INTEGER
ROW DIMENSION OF THE ARRAY CONTAINING THE MATRIX X AS DECLARED IN THE MAIN CALLING PROGRAM DIMENSION STATEMENT;

N INTEGER
ORDER OF THE SQUARE MATRICES E, A, CQC, RSDM AND X ROW DIMENSION OF THE MATRICES B AND S;

M INTEGER
ORDER OF THE SQUARE MATRICES R AND RI COLUMN DIMENSION OF THE MATRICES B AND S;

E REAL(NR,N)
MODEL DESCRIPTOR MATRIX;

A REAL(NR,N)
MODEL SYSTEM MATRIX;

B REAL(NR,M)
MODEL INPUT MATRIX;

CQC REAL(NR,N)
MATRIX PRODUCT CT\*Q\*C WHERE T DENOTES MATRIX TRANSPOSE;

R REAL(NR,M)
CONTROL WEIGHTING MATRIX;

S REAL(NR,M)
STATE - INPUT CROSS-WEIGHTING MATRIX;
RI       REAL(NR,M) 
        INVERSE OF THE CONTROL WEIGHTING MATRIX;

X        REAL(NRX,N) 
        INITIAL GUESS FOR RICCATI SOLUTION THAT MUST 
        STABILIZE THE CLOSED LOOP SYSTEM MATRIX;

W1,W2,W3 REAL(NRW,N) 
        SCRATCH ARRAYS OF SIZE AT LEAST N BY N;

WK1      REAL(N) 
        WORK VECTOR OF LENGTH AT LEAST N;

IPVT     INTEGER(N) 
        WORK VECTOR OF LENGTH AT LEAST N;

RTOL     REAL 
        TOLERANCE ON THE CONDITION ESTIMATE OF R+B'T*X*B 
        WITH RESPECT TO INVERSION (DISCRETE PROBLEM). 
        ERROR RETURN IF THIS TOLERANCE IS EXCEEDED;

EFLAG    CHARACTER 
        FLAG SET TO 'Y' IF E IS OTHER THAN THE IDENTITY 
        MATRIX;

RFLAG    CHARACTER 
        FLAG SET TO 'Y' IF R IS OTHER THAN THE IDENTITY 
        MATRIX;

RDFLG    CHARACTER 
        FLAG SET TO 'Y' IF R IS A DIAGONAL MATRIX;

SFLAG    CHARACTER 
        FLAG SET TO 'Y' IF S IS OTHER THAN THE ZERO MATRIX;

TYPE     LOGICAL 
        = .TRUE. FOR CONTINUOUS-TIME SYSTEM 
        = .FALSE. FOR DISCRETE-TIME SYSTEM;

MAXIT    INTEGER 
        MAXIMUM NUMBER OF NEWTON ITERATIONS PERMITTED;

NOUT     INTEGER 
        UNIT NUMBER OF OUTPUT DEVICE FOR ERROR WARNING 
        MESSAGES.

ON OUTPUT:
C RSDM REAL(NRR,N)
DIFFERENCE MATRIX BETWEEN SOLUTIONS AT LAST TWO
ITERATIONS;

C X
REFINED RICCATI SOLUTION MATRIX;

C ALFR REAL(N)
REAL PARTS OF THE CLOSED LOOP SYSTEM EIGENVALUES
UNDER THE OPTIMAL FEEDBACK FOR THE RICCATI SOLUTION
AT THE LAST ITERATION;

C ALFI REAL(N)
IMAGINARY PARTS OF THE CLOSED LOOP SYSTEM
EIGENVALUES UNDER THE OPTIMAL FEEDBACK FOR THE
RICCATI SOLUTION AT THE LAST ITERATION;

C WK1(1) 1-NORM OF THE RSDM MATRIX;

C IPVT(1) CONVERGENCE CRITERIA INDICATOR;

C SEP
ESTIMATE OF THE SEPARATION OF THE CLOSED LOOP
SPECTRUM AT THE LAST ITERATION (CONTINUOUS PROBLEM);

C MAXIT NUMBER OF ITERATIONS PERFORMED;

C INFO INTEGER
ERROR FLAG WITH THE FOLLOWING MEANINGS
= -1 NO CONVERGENCE
= -2 INITIAL GUESS NOT STABILIZING
= -3 E MATRIX NOT IDENTITY
= -4 INDICATES A FAILURE OF THE QR ALGORITHM TO
DETERMINE THE EIGENVALUES IN SOLVING THE
LYAPUNOV EQUATION
= -5 CONDITION ESTIMATE OF R+B*T*X*B WITH RESPECT TO
INVERSION EXCEEDS THE TOLERANCE VALUE RTOL.

****ALGORITHM NOTES:
The algorithm currently employed is based on the Bartels-
Stewart algorithm for Lyapunov equations and is valid for the
case E = identity only at this time.

****HISTORY:
This subroutine was written by W.F. Arnold, Naval Weapons
Center, Code 35104, China Lake, CA 93555, as part of the
software package RICPACK, September 1983.
SUBROUTINE ORDER (A,B,Z,NMAX,N,EPS,IFAIL,TYPE,IFIRST,IND)
C
****PARAMETERS:
INTEGER NMAX,N,IFAIL,IFIRST,IND(N)
DOUBLE PRECISION A(NMAX,N),B(NMAX,N),Z(NMAX,N),EPS
LOGICAL TYPE
C
****LOCAL VARIABLES:
INTEGER I,II,III,IS,ISTEP,K,L,LS,LS1,LS2,L1,L2,NUM
C
****FORTRAN FUNCTIONS:
DOUBLE PRECISION DABS
C
****SUBROUTINES CALLED:
EXCHQZ
C
-------------------------------

*****PURPOSE:
GIVEN THE UPPER TRIANGULAR MATRIX B AND UPPER HESSENBERG
MATRIX A WITH 1X1 OR 2X2 DIAGONAL BLOCKS, THIS SUBROUTINE
REORDERS THE DIAGONAL BLOCKS ALONG WITH THE GENERALIZED
EIGENVALUES CORRESPONDING TO THE REGULAR MATRIX PENCIL
A - LAMBDA*B BY CONSTRUCTING ROW AND COLUMN EQUIVALENCE
TRANSFORMATIONS QT AND ZT. THE COLUMN TRANSFORMATIONS ARE
THEN APPLIED TO THE MATRIX Z.
REF.: VAN DOOREN, P., A GENERALIZED EIGENVALUE APPROACH FOR
SOLVING RICCATI EQUATIONS, SIAM J. SCI. STAT. COMPUT.,
VOL. 2, NO. 2, JUNE 1981, 121-135.

*****PARAMETER DESCRIPTION:
ON INPUT:
C
NMAX
INTEGER
ROW DIMENSION OF THE ARRAYS CONTAINING A,B,Z AS
DECLARED IN THE MAIN CALLING PROGRAM DIMENSION
STATEMENT;
C
N
INTEGER
ORDER OF THE MATRICES A,B,Z;
C
A
REAL(NMAX,N)
UPPER HESSENBERG MATRIX WITH 1X1 OR 2X2 DIAGONAL
BLOCKS. ELEMENTS OUTSIDE THE UPPER HESSENBERG
STRUCTURE ARE ARBITRARY;
C
B
REAL(NMAX,N)
UPPER TRIANGULAR MATRIX. ELEMENTS OUTSIDE THE
UPPER TRIANGULAR STRUCTURE ARE ARBITRARY;

EPS REAL
REQUIRED ABSOLUTE ACCURACY OF THE RESULTS.
NORMALLY EQUAL TO THE MACHINE PRECISION;

TYPE LOGICAL
CONTROL PARAMETER THAT SPECIFIES THE REGIONS OF THE
THE COMPLEX PLANE THAT THE GENERALIZED EIGENVALUES
ARE ORDERED BY. TO CONTROL THE REGION THAT APPEARS
FIRST, SEE IFIRST BELOW
= .TRUE. GENERALIZED EIGENVALUES ARE ORDERED BY
REGION INSIDE THE COMPLEX LEFT HALF PLANE OR
OUTSIDE THIS REGION
= .FALSE. GENERALIZED EIGENVALUES ORDERED BY REGION
INSIDE THE UNIT CIRCLE OR OUTSIDE THIS REGION;

IFIRST INTEGER
CONTROL PARAMETER THAT SPECIFIES WHICH OF THE
REGIONS SPECIFIED BY TYPE(SEE ABOVE) APPEARS FIRST
(I.E. IN THE UPPER LEFT NXN BLOCK)
= -1 INSIDE REGION APPEARS FIRST
= +1 OUTSIDE REGION APPEARS FIRST
IFIRST IS ALTERED BY THIS SUBROUTINE;

IND INTEGER(N)
WORKING ARRAY THAT IS ALTERED BY THIS SUBROUTINE.

ON OUTPUT:
A,B UPPER HESSENBERG MATRIX, UPPER TRIANGULAR MATRIX
REORDERED AS SPECIFIED BY TYPE AND IFIRST(ABOVE);
Z REAL(NMAX,N)
THIS ARRAY IS OVERWRITTEN BY THE PRODUCT OF THE
CONTENTS OF THE ARRAY Z(UPON ENTRY INTO THIS
SUBROUTINE), AND THE COLUMN TRANSFORMATIONS ZT
(CALCULATED BY THIS SUBROUTINE);
IFAIL INTEGER
ERROR FLAG
= 1 INDICATES ATTEMPTED REORDERING FAILED
= 0 NORMAL RETURN.

*****ALGORITHM NOTES:
NONE.

*****HISTORY:
ORIGINAL VERSION THAT SORTED BY UNIT CIRCLE REGION OF COMPLEX
SUBROUTINE LYPCND(NF,NH,N,F,H,Z,WR, WI, WK, IER1, IER2)

****PARAMETERS:
INTEGER NF, NH, N, IER1, IER2
DOUBLE PRECISION F(NF,N), H(NH,N), Z(NF,N), WR(N), WI(N), WK(N)

****LOCAL VARIABLES:
INTEGER LOW, IGH

****SUBROUTINES CALLED:
ORTHE, ORTRAN, HQRORT, MQFWO(MULWOA), SYMSLV(LINEQ, DGECOM, DGESLM)
TRNATA

*****PURPOSE:
THIS SUBROUTINE SOLVES THE CONTINUOUS TIME LYAPUNOV EQUATION

\[
T \quad F^*X + X^*F + H = 0 .
\]

BY THE BARTELS-STEWART ALGORITHM (SEE REF.(1)).

*****PARAMETER DESCRIPTION:
ON INPUT:

NF, NH  ROW DIMENSIONS OF THE ARRAYS CONTAINING F
        (AND Z), H, RESPECTIVELY, AS DECLARED IN
        MAIN CALLING PROGRAM DIMENSION STATEMENT;

N  ORDER OF THE MATRICES F AND H;

F  N X N (REAL) MATRIX;

H  N X N SYMMETRIC MATRIX;

IER1  INTEGER VARIABLE; NORMALLY SET IER1 TO 0;
      IF IER1 IS SET TO A NON-ZERO INTEGER, THE
      REDUCTION OF F TO REAL SCHUR FORM IS;
      SKIPPED AND THE ARRAYS F AND Z ARE ASSUMED
      TO CONTAIN THE REAL SCHUR FORM AND
      ACCOMPANYING ORTHOGONAL MATRIX THUS
      PERMITTING MORE EFFICIENT SOLUTION OF
      SEVERAL EQUATIONS WITH DIFFERENT CONSTANT
      TERMS H;

IER2  INTEGER VARIABLE; NORMALLY SET IER2 TO 0;
      IF ONLY A REAL SCHUR FORM OF F AND
ASSOCIATED ORTHOGONAL SIMILARITY MATRIX Z
ARE DESIRED SET IER2 TO A NON-ZERO
INTEGER.

ON OUTPUT:

H
N X N ARRAY CONTAINING THE (SYMMETRIC)
SOLUTION X OF THE LYAPUNOV EQUATION;

F
N X N ARRAY CONTAINING IN ITS UPPER
TRIANGLE AND FIRST SUBDIAGONAL A REAL
SCHUR FORM OF F;

Z
N X N ARRAY CONTAINING, ON OUTPUT, THE
ORTHOGONAL MATRIX THAT REDUCES F TO REAL
SCHUR FORM;

WR
REAL SCRATCH VECTOR OF LENGTH N; ON OUTPUT
(WR(I),I=1,N) CONTAINS THE REAL PARTS OF
THE EIGENVALUES OF F AND THUS CAN BE USED
TO TEST THE STABILITY OF F;

WI
REAL SCRATCH VECTOR OF LENGTH N; ON OUTPUT
CONTAINS THE IMAGINARY PARTS OF THE
EIGENVALUES OF F;

WK
REAL SCRATCH VECTOR OF LENGTH N;

IER1
=0 FOR NORMAL RETURN (IF =0 ON INPUT),
=J IF THE J-TH EIGENVALUE HAS NOT BEEN
DETERMINED IN THE QR ALGORITHM (IF =0 ON
INPUT).

*****ALGORITHM NOTES:
IT IS ASSUMED THAT F HAS NO EIGENVALUES WHICH SUM TO ZERO (THIS
CAN BE CHECKED FROM THE ARRAY WR). THIS IS SUFFICIENT TO
GUARANTEE A UNIQUE SOLUTION.
IF, MOREOVER, F IS STABLE THEN X IS NONNEGATIVE DEFINITE.

*****REFERENCES:
(1) BARTELS, R.H., AND G.W. STEWART, SOLUTION
OF THE MATRIX EQUATION AX + XB = C,
ALGORITHM 432, COMM. ACM, 15(1972),820-826.

*****HISTORY:
WRITTEN BY ALAN J. LAUB (DEP'T. OF EE-SYSTEMS, U. OF SOUTHERN
CALIF., LOS ANGELES, CA 90089, PH.: (213) 743-5535) SEP. 1977
SUBROUTINE LYPDSD(NF,NH,N,F,H,Z,WR,WI,WK,U,IDIM,IER1,IER2)

****PARAMETERS:
INTEGER NF,NH,N,IER1,IER2,IDIM(N)
DOUBLE PRECISION F(NF,N),U(NF,N),H(NH,N),Z(NF,N),WR(N),WI(N),
X
WK(N)

****LOCAL VARIABLES:
INTEGER LOW,IGH,KIN,KOUT

****SUBROUTINES CALLED:
ORTHES,ORTRAN,HQRORT,MQFWO(MULWOA),DSTSLV(LINEQ,DDCOMP,DSOLVE)
TRNATA

*****PURPOSE:
THIS SUBROUTINE SOLVES THE DISCRETE TIME LYAPUNOV EQUATION

\[ T F^* X F - X = H. \]

BY A MODIFICATION OF THE BARTELS-STEWART ALGORITHM (SEE REFS. (1) AND (2)).

*****PARAMETER DESCRIPTION:
ON INPUT:

(AND Z,U), AND H, RESPECTIVELY, AS DECLARED IN THE CALLING PROGRAM DIMENSION STATEMENT;

N ORDER OF THE MATRICES F AND C;

F N X N (REAL) MATRIX;

H N X N SYMMETRIC MATRIX;

IER1 INTEGER VARIABLE; NORMALLY SET IER1 TO 0;
IF IER1 IS SET TO A NON-ZERO INTEGER, THE REDUCTION OF F TO REAL SCHUR FORM IS, SKIPPED AND THE ARRAYS F AND Z ARE ASSUMED TO CONTAIN THE REAL SCHUR FORM AND ACCOMPANING ORTHOGONAL MATRIX THUS PERMITTING MORE EFFICIENT SOLUTION OF SEVERAL EQUATIONS WITH DIFFERENT CONSTANT TERMS H;

IER2 INTEGER VARIABLE; NORMALLY SET IER2 TO 0;
IF ONLY A REAL SCHUR FORM OF F AND
ASSOCIATED ORTHOGONAL SIMILARITY MATRIX Z
ARE DESIRED SET IER2 TO A NON-ZERO
INTEGER.

ON OUTPUT:

H N X N ARRAY CONTAINING THE (SYMMETRIC)
SOLUTION X OF THE LYAPUNOV EQUATION;

F N X N ARRAY CONTAINING IN ITS UPPER
TRIANGLE AND FIRST SUBDIAGONAL A REAL
SCHUR FORM OF F;

Z N X N ARRAY CONTAINING, ON OUTPUT, THE
ORTHOGONAL MATRIX THAT REDUCES F TO REAL
SCHUR FORM;

WR REAL SCRATCH VECTOR OF LENGTH N; ON OUTPUT
(WR(I),I=1,N) CONTAINS THE REAL PARTS OF
THE EIGENVALUES OF F AND THUS CAN BE USED
TO TEST THE STABILITY OF F;

WI REAL SCRATCH VECTOR OF LENGTH N; ON OUTPUT
CONTAINS THE IMAGINARY PART OF THE
EIGENVALUES OF F;

WK REAL SCRATCH VECTOR OF LENGTH N;

U N X N REAL SCRATCH ARRAY;

IDIM INTEGER SCRATCH VECTOR OF LENGTH N;

IER1 =0 FOR NORMAL RETURN (IF =0 ON INPUT),
=J IF THE J-TH EIGENVALUE HAS NOT BEEN
DETERMINED IN THE QR ALGORITHM (IF =0 ON
INPUT).

****ALGORITHM NOTES:
IT IS ASSUMED THAT F HAS NO EIGENVALUES WITH PRODUCT EQUAL TO
ZERO (THIS CAN BE CHECKED FROM THE ARRAY WR). THIS IS
SUFFICIENT TO GUARANTEE A UNIQUE SOLUTION.
IF, MOREOVER, F IS STABLE THEN X IS NONNEGATIVE DEFINITE.

****REFERENCES:
(1) BARTELS, R.H., AND G.W. STEWART, SOLUTION
OF THE MATRIX EQUATION AX + XB = C,
ALGORITHM 432, COMM. ACM, 15(1972), 820-826.

(2) BARRAUD,A.Y., A NUMERICAL ALGORITHM TO SOLVE A XA-X=Q,
SUBROUTINE SEPEST(NR,N,T,Q,SEP,INFO)

*****PARAMETERS:
INTEGER NR,N,INFO
DOUBLE PRECISION T(NR,N),Q(NR,N),SEP

*****LOCAL VARIABLES:
INTEGER IPVT(4),I,II,IM1,J,JP1,K1,K2,K1M1,L1,L1M1,L1MK1,X
M,ND,NM1
DOUBLE PRECISION A(4,4),VEC(4),Z(4),AIN,RCOND,S,TEMP,T1N

*****FORTRAN FUNCTIONS:
DOUBLE PRECISION DABS,DMAX1,DMIN1,ISIGN

*****FUNCTION SUBPROGRAMS:
DOUBLE PRECISION D1NRM

*****SUBROUTINES CALLED:
DGEOM, DGESLM, MSCALE, SYMSLV

-------------------------------------------------------------------

*****PURPOSE:
GIVEN A QUASITRIANGULAR MATRIX T AND A SYMMETRIC MATRIX Q,
THIS SUBROUTINE COMPUTES

SEP = MIN( 1-NORM( T-TRANSPOSE*Q + Q*T)/ 1-NORM(Q))

REF.: BARTELS, R.H. AND G.W. STEWART, "SOLUTION OF THE MATRIX
EQUATION A*X + X*B = C," COMM. OF THE ACM, VOL. 15,
PP. 820-826, 1972.
CLINE, A.K., MOLER, C.B., STEWART, G.W. AND J.H. WILKINSON,
"AN ESTIMATE OF THE CONDITION NUMBER OF A MATRIX," SIAM J. OF

*****PARAMETER DESCRIPTION:

ON INPUT:

NR INTEGER
ROW DIMENSION OF THE ARRAYS CONTAINING THE MATRICES
T AND Q AS DECLARED IN THE MAIN CALLING PROGRAM
DIMENSION STATEMENT;

N INTEGER
ORDER OF THE SQUARE MATRICES Q AND T;

T REAL(NR,N)
QUASITRIANGULAR INPUT MATRIX.
ON OUTPUT:

SEP
REAL
AN ESTIMATE OF THE QUANTITY SPECIFIED ABOVE;

Q
REAL(NR,N)
THE MATRIX THAT MINIMIZES THE QUANTITY SEP;

INFO
INTEGER
ERROR FLAG WITH THE FOLLOWING MEANINGS
= 0 INDICATES NORMAL RETURN
= (N+1)*L+K INDICATES THE L-TH AND K-TH EIGENVALUES
OF T FORM A +/- PAIR, SO SEP IS EQUAL TO ZERO.

*****ALGORITHM NOTES:

T
LET PHI(Y) = T*Y + Y*T . Q IS OBTAINED BY INVERSE ITERATION ON
T
PHI*PHI . THE STARTING VALUE OF Q IS CHOSEN AS FOLLOWS:
PARTITION ALL MATRICES CONFORMALLY WITH T. Q IS CHOSEN TO
SATISFY
T
T
T*Y + Y*T = B, T *Q + Q*T = Y/1-NORM(Y).

*****HISTORY:

THIS SUBROUTINE IS A MODIFIED VERSION OF THE SUBROUTINE OF THE
SAME NAME WRITTEN BY RALPH BEYERS, 2/82 REF.: BEYERS, R.,
"HAMILTONIAN AND SYMPLECTIC ALGORITHMS FOR THE ALGEBRAIC
RICCATI EQUATION," PHD THESIS, CORNELL UNIVERSITY, PP.289-295,
JANUARY 1983. THE MODIFICATIONS WERE MADE BY W.F. ARNOLD,
NAVAL WEAPONS CENTER, CODE 35104, CHINA LAKE, CA 93555, AS
IN THE MAIN CALLING PROGRAM DIMENSION STATEMENT:

NRW INTEGER
ROW DIMENSION OF THE ARRAYS CONTAINING FB AND W AS DECLARED IN THE MAIN PROGRAM DIMENSION STATEMENT;

N INTEGER
ORDER OF THE SQUARE MATRICES A, E, AND X
ROW DIMENSION OF THE MATRICES B, S, AND FB;

M INTEGER
ORDER OF THE SQUARE MATRICES R AND RI
COLUMN DIMENSION OF THE MATRICES B AND S;

A REAL(NR,N)
MODEL SYSTEM MATRIX;

B REAL(NR,M)
MODEL INPUT MATRIX;

E REAL(NR,N)
MODEL DESCRIPTOR MATRIX;

R REAL(NR,M)
INPUT WEIGHTING MATRIX;

RI REAL(NR,M)
INVERSE OF THE INPUT WEIGHTING MATRIX;

S REAL(NR,M)
STATE - INPUT CROSS-WEIGHTING MATRIX;

X REAL(NRX,N)
ALGEBRAIC RICCATI EQUATION SOLUTION MATRIX;

W REAL(NRW,N)
SCRATCH ARRAY OF SIZE AT LEAST N BY N;

WK REAL(N)
WORKING VECTOR OF LENGTH AT LEAST N;

IPVT INTEGER(M)
WORKING VECTOR OF LENGTH AT LEAST M;

EFLAG CHARACTER
FLAG SET TO 'Y' IF E IS OTHER THAN THE IDENTITY MATRIX;

RDFLG CHARACTER
FLAG SET TO 'Y' IF R IS A DIAGONAL MATRIX;

RFLAG CHARACTER
FLAG SET TO 'Y' IF R IS OTHER THAN THE IDENTITY MATRIX;

SFLAG CHARACTER
FLAG SET TO 'Y' IF S IS OTHER THAN THE ZERO MATRIX;

TYPE LOGICAL
= .TRUE. FOR CONTINUOUS-TIME SYSTEM
= .FALSE. FOR DISCRETE-TIME SYSTEM.

ON OUTPUT:

FB REAL(NRW,N)
OPTIMAL FEEDBACK GAIN MATRIX AS DESCRIBED ABOVE;

WK(1) ESTIMATED CONDITION NUMBER OF R+BT*X*B WITH RESPECT TO INVERSION (DISCRETE PROBLEM).

*****ALGORITHM NOTES:
NONE.

*****HISTORY:
SUBROUTINE CMPRS(NR, NRD, NRT, N, NN, NNPM, M, E, A, CQC, R, S, G, F, U, X, WK, WK1, WK2, WK3, EFLAG, SFLAG, IBAL, INFO)

*****PARAMETERS:
INTEGER NR, NRD, NRT, N, NN, NNPM, M, IBAL, INFO
CHARACTER EFLAG, SFLAG
DOUBLE PRECISION E(NR,M), A(NR,M), B(NR,M), CQC(NR,N), X, R(NR,M), S(NR,M), G(NRD,NN), F(NRD,NN), U(NRT,NNPM), X, WK(NRT,M), WK1(M), WK2(M), WK3(NNPM)

*****LOCAL VARIABLES:
INTEGER I, J, JOB, K, NNPI, NPI, NPJ, NPK

*****FORTRAN FUNCTIONS:
NONE.

*****SUBROUTINES CALLED:
DSVDC

---------------------------------------------------------------------

*****PURPOSE:
THIS SUBROUTINE EMPLOYS THE SINGULAR VALUE DECOMPOSITION TO
DETERMINE AN ORTHOGONAL MATRIX U, (2N+M) BY (2N+M), SUCH THAT

\[
\begin{pmatrix}
U_{11} & U_{12} \\
U_{21} & U_{22}
\end{pmatrix}
\begin{pmatrix}
B & 0 \\
0 & 0
\end{pmatrix}
= \begin{pmatrix}
0 & 0 \\
0 & 0
\end{pmatrix}
\]

AND THEN FORMS THE MATRIX PENCIL

\[
\lambda U_{11} - (U_{11}^*(STBT)) + U_{12}^*(STAT)
= \lambda F - G
\]

WHERE T DENOTES MATRIX TRANSPOSE.
THIS PENCIL CAN THEN BE USED FOR SOLVING THE CONTINUOUS-TIME
GARE.

REF.: ARNOLD, W.F., "ON THE NUMERICAL SOLUTION OF
ALGEBRAIC MATRIX RICCATI EQUATIONS," PHD THESIS, USC,
DECEMBER 1983.

*****PARAMETER DESCRIPTION:
ON INPUT:

NR INTEGER
ROW DIMENSION OF THE ARRAYS CONTAINING THE MATRICES E, A, B, CQC, R AND S AS DECLARED IN THE MAIN CALLING PROGRAM DIMENSION STATEMENT;

NRD INTEGER
ROW DIMENSION OF THE ARRAYS CONTAINING THE MATRICES G AND F AS DECLARED IN THE MAIN CALLING PROGRAM DIMENSION STATEMENT;

NRT INTEGER
ROW DIMENSION OF THE ARRAYS CONTAINING THE MATRICES U AND WK AS DECLARED IN THE MAIN CALLING PROGRAM DIMENSION STATEMENT;

N INTEGER
ORDER OF THE SQUARE MATRICES E, A AND CQC
ROW DIMENSION OF THE MATRICES B AND S;

NN INTEGER
ORDER OF THE SQUARE MATRICES G AND F;

NNPM INTEGER
= NN + M;

M INTEGER
ORDER OF THE SQUARE MATRIX R
COLUMN DIMENSION OF THE MATRICES B AND S;

E REAL(NR,N)
MODEL DESCRIPTOR MATRIX;

A REAL(NR,N)
MODEL SYSTEM MATRIX;

B REAL(NR,M)
MODEL INPUT MATRIX;

CQC REAL(NR,N)
MATRIX PRODUCT CT*Q*C WHERE T DENOTES MATRIX TRANPOSE;

R REAL(NR,M)
CONTROL WEIGHTING MATRIX;

S REAL(NR,M)
STATE - INPUT CROSS-WEIGHTING MATRIX;

WK REAL(NRT,M)
SCRATCH ARRAY OF SIZE AT LEAST (NN+M) BY M;

WK1,WK2 REAL(M)
WORKING VECTORS OF SIZE AT LEAST M;

WK3 REAL(NNPM)
WORKING VECTOR OF SIZE AT LEAST NN+M;

EFLAG CHARACTER
FLAG SET TO 'Y' IF E IS OTHER THAN THE IDENTITY
MATRIX;

SFLAG CHARACTER
FLAG SET TO 'Y' IF S IS OTHER THAN THE ZERO MATRIX;

IBAL INTEGER
PARAMETER SET TO 1 IF CO-ORDINATE BALANCING IS
BEING USED.

ON OUTPUT:

G REAL(NRD,NN)
MATRIX OF THE COMPRESSED PENCIL AS DEFINED ABOVE;

F REAL(NRD,NN)
MATRIX OF THE COMPRESSED PENCIL AS DEFINED ABOVE;

U REAL(NRT,NNPM)
ORTHOGONAL COMPRESSION MATRIX AS DEFINED ABOVE;

INFO INTEGER
ERROR FLAG WITH MEANING AS FOLLOWS

= 0 NORMAL RETURN
= NONZERO IF SINGULAR VALUE DECOMPOSITION COULD
NOT BE CALCULATED.

*****ALGORITHM NOTES:
NONE.

*****HISTORY:
THIS SUBROUTINE WAS WRITTEN BY W.F. ARNOLD, NAVAL WEAPONS
CENTER, CODE 35104, CHINA LAKE, CA 93555, AS PART OF THE
SOFTWARE PACKAGE RICPACK, SEPTEMBER 1983.

----------------------------------

152
SUBROUTINE RINV(NR, NRD, N, NN, M, E, A, B, CQC, RI, G, F, WK1, WRK, RDFLG, RFLAGE, RFLAGEFLAG, IBAL, TYPE)

C
C *PARAMETERS:
INTEGER NR, NRD, N, NN, M, IBAL
CHARACTER RDFLG, RFLAG, EFLAG
DOUBLE PRECISION E(NR, N), A(NR, M), B(NR, M), CQC(NR, N), RI(NR, M),
G(NRD, NN), F(NRD, NN), WK1(NR, N), WRK(N)
LOGICAL TYPE
C
C LOCAL VARIABLES:
INTEGER I, J, K, NPI, NPJ
C
C FORTRAN FUNCTIONS:
NONE.
C
C SUBROUTINES CALLED:
MULB, TRNATB
C
C PURPOSE:
THIS SUBROUTINE FORMS THE MATRIX PENCIL:

\[
\begin{pmatrix} E & 0 \\ \lambda & F \end{pmatrix} 
\begin{pmatrix} A & -BR^*BT \\ 0 & ET \end{pmatrix} 
\begin{pmatrix} -CQC & -AT \\ \end{pmatrix} 
\]

WHERE T DENOTES MATRIX TRANSPOSE.
THIS SUBROUTINE IS USEFUL IN SOLVING THE CONTINUOUS-TIME GARE.
C
C REF.: ARNOLD, W.F., "ON THE NUMERICAL SOLUTION OF
ALGEBRAIC MATRIX RICCATI EQUATIONS," PHD THESIS, USC,
DECEMBER 1983.

C
C PARAMETER DESCRIPTION:
C
ON INPUT:
C
NR INTEGER
ROW DIMENSION OF THE ARRAYS CONTAINING THE MATRICES E, A, B, CQC, RI AND WK1 AS DECLARED IN THE MAIN CALLING PROGRAM DIMENSION STATEMENT;
C
NRD INTEGER
ROW DIMENSION OF THE ARRAYS CONTAINING THE MATRICES F AND G AS DECLARED IN THE MAIN CALLING PROGRAM
DIMENSION STATEMENT;

N INTEGER
ORDER OF THE SQUARE MATRICES E, A AND CQC
ROW DIMENSION OF THE MATRIX B;

NN INTEGER
SIZE OF THE MATRIX PENCIL;

M INTEGER
ORDER OF THE SQUARE MATRIX RI
COLUMN DIMENSION OF THE MATRIX B;

E REAL(NR,N)
MODEL DESCRIPTOR MATRIX;

A REAL(NR,N)
= A - B*RI*ST IN THE GENERALIZED CASE;

B REAL(NR,M)
MODEL INPUT MATRIX;

CQC REAL(NR,N)
= CT*QC - S*RI*ST IN THE GENERALIZED CASE;

RI REAL(NR,M)
INVERSE OF THE CONTROL WEIGHTING MATRIX;

WK1 REAL(NR,N)
SCRATCH ARRAY OF SIZE AT LEAST M BY N;

WRK REAL(N)
WORKING VECTOR OF SIZE AT LEAST N;

RDFLG CHARACTER
FLAG SET TO 'Y' IF RI IS A DIAGONAL MATRIX;

RFLAG CHARACTER
FLAG SET TO 'Y' IF RI IS OTHER THAN THE IDENTITY MATRIX;

EFLAG CHARACTER
FLAG SET TO 'Y' IF E IS OTHER THAN THE IDENTITY MATRIX;

IBAL INTEGER
= 1 IF CO-ORDINATE BALANCING IS BEING USED;

TYPE LOGICAL
= .TRUE. FOR CONTINUOUS-TIME SYSTEM
= .FALSE. FOR DISCRETE-TIME SYSTEM.

ON OUTPUT:

G REAL(NRD,NN)
PENCIL MATRIX AS DEFINED ABOVE;

F REAL(NRD,NN)
PENCIL MATRIX AS DEFINED ABOVE.

****ALGORITHM NOTES:
NONE.

****HISTORY:
SUBROUTINE RESID(NR, NRR, NRW, NRX, N, M, E, A, B, CQC, R, S, RI, RSDM, X, W1, X W2, WK, IPVTR, RTOL, EFLAG, RFLAG, RDFLG, SFLAG, RSD, X TYPE, NOUT)

C

*****PARAMETERS:
INTEGER NR, NRR, NRW, NRX, N, M, IPVTR(N), NOUT
CHARACTER EFLAG, RFLAG, RDFLG, SFLAG
DOUBLE PRECISION E(NR, N), A(NR, N), B(NR, N), CQC(NR, N), R(NR, M), X S(NR, M), RI(NR, M), RSDM(NRR, N), X(NRX, N), W1(NRW, N), W2(NRW, N), X WK(N), RTOL, RSD
LOGICAL TYPE

C

*****LOCAL VARIABLES:
INTEGER I
DOUBLE PRECISION COND

C

*****FORTRAN FUNCTIONS:
NONE.

C

*****FUNCTION SUBPROGRAMS:
DOUBLE PRECISION D1NRM

C

*****SUBROUTINES CALLED:
MADD, MLINEQ, MMUL, MQF, MSUB, MULA, TRNATB

C

*****PURPOSE:
THIS SUBROUTINE CALCULATES THE RESIDUAL MATRIX AND ITS 1-NORM
FOR THE GARE AS FOLLOWS:

CONTINUOUS:

RSDM = AT*X*E + ET*X*A - (BT*X*E+ST)*RI*(BT*X*E+ST) + CQC

DISCRETE:

RSDM = AT*X*A - ET*X*E + CQC
- (BT*X*A+ST)*((R+BT*X*B)**-1)*(BT*X*A+ST)

WHERE T DENOTES MATRIX TRANSPOSE.

REF.: ARNOLD, W.F., "ON THE NUMERICAL SOLUTION OF
ALGEBRAIC MATRIX RICCATI EQUATIONS," PHD THESIS, USC,
DECEMBER 1983.

C

*****PARAMETER DESCRIPTION:
ON INPUT:

```
NR INTEGER
ROW DIMENSION OF THE ARRAYS CONTAINING THE MATRICES
E, A, B, CQC, R, RI AND S AS DECLARED IN THE MAIN
CALLING PROGRAM DIMENSION STATEMENT;

NRR INTEGER
ROW DIMENSION OF THE ARRAY CONTAINING THE MATRIX
RSDM AS DECLARED IN THE MAIN CALLING PROGRAM
DIMENSION STATEMENT;

NRW INTEGER
ROW DIMENSION OF THE ARRAYS CONTAINING THE MATRICES
W1 AND W2 AS DECLARED IN THE MAIN CALLING PROGRAM
DIMENSION STATEMENT;

NRX INTEGER
ROW DIMENSION OF THE ARRAY CONTAINING THE MATRIX X
AS DECLARED IN THE MAIN CALLING PROGRAM DIMENSION
STATEMENT;

N INTEGER
ORDER OF THE SQUARE MATRICES E, A, CQC, RSDM AND X
ROW DIMENSION OF THE MATRICES B AND S;

M INTEGER
ORDER OF THE SQUARE MATRICES R AND RI
COLUMN DIMENSION OF THE MATRICES B AND S;

E REAL(NR,N)
MODEL DESCRIPTOR MATRIX;

A REAL(NR,N)
MODEL SYSTEM MATRIX;

B REAL(NR,M)
MODEL INPUT MATRIX;

CQC REAL(NR,N)
MATRIX PRODUCT CT*C WHERE T DENOTES MATRIX
TRANSPOSE;

R REAL(NR,M)
CONTROL WEIGHTING MATRIX;

S REAL(NR,M)
STATE - INPUT CROSS-WEIGHTING MATRIX;
```
RI REAL(NR,N)
INVERSE OF THE CONTROL WEIGHTING MATRIX;

X REAL(NRX,N)
SOLUTION MATRIX FOR THE GARE WHOSE RESIDUAL IS TO BE
DETERMINED;

W1,W2 REAL(NRW,N)
SCRATCH ARRAYS OF SIZE AT LEAST N BY N;

WK REAL(N)
WORK VECTOR OF LENGTH AT LEAST N;

IPVT INTEGER(N)
WORK VECTOR OF LENGTH AT LEAST N;

RTOL REAL
TOLERANCE ON THE CONDITION ESTIMATE OF R+BT*X*B WITH
RESPECT TO INVERSION (DISCRETE PROBLEM). IF THIS
TOLERANCE IS EXCEEDED AN ERROR MESSAGE IS PRINTED
AND AN ERROR RETURN IS MADE;

EFLAG CHARACTER
FLAG SET TO 'Y' IF E IS OTHER THAN THE IDENTITY
MATRIX;

RFLAG CHARACTER
FLAG SET TO 'Y' IF R IS OTHER THAN THE IDENTITY
MATRIX;

RDFLG CHARACTER
FLAG SET TO 'Y' IF R IS A DIAGONAL MATRIX;

SFLAG CHARACTER
FLAG SET TO 'Y' IF S IS OTHER THAN THE ZERO MATRIX;

TYPE LOGICAL
= .TRUE. FOR CONTINUOUS-TIME SYSTEM
= .FALSE. FOR DISCRETE-TIME SYSTEM;

NOUT INTEGER
UNIT NUMBER OF OUTPUT DEVICE FOR ERROR WARNING
MESSAGES.

ON OUTPUT:
RSDM REAL(NRR,N)
THE RESIDUAL MATRIX CALCULATED AS INDICATED ABOVE:
IPVT(1) ERROR FLAG AS FOLLOWS
 = 0 NORMAL RETURN
 = 1 THE CONDITION ESTIMATE OF R*BT*X*B WITH RESPECT TO INVERSION EXCEEDS THE TOLERANCE RTOL;

RSD REAL
THE 1-NORM OF THE RESIDUAL MATRIX DIVIDED BY THE 1-NORM OF THE SOLUTION MATRIX.

****ALGORITHM NOTES:
NONE.

****HISTORY:
SUBROUTINE BALGEN (N, MA, A, MB, B, LOW, IGH, CSSCALE, CPERM, WK)

*****PARAMETERS:
INTEGER N, MA, MB, LOW, IGH
DOUBLE PRECISION A(MA, N), B(MB, N), CSSCALE(N), CPERM(N), WK(N, 6)

*****LOCAL VARIABLES:
NONE.

*****FUNCTIONS:
NONE.

*****SUBROUTINES CALLED:
REDUCE, SCALEG, GRADEQ

*****PURPOSE:
THIS SUBROUTINE BALANCES THE MATRICES A AND B TO IMPROVE THE
ACCURACY OF COMPUTING THE EIGENSYSTEM OF THE GENERALIZED
EIGENPROBLEM A*X = (LAMBDA)*B*X. THE ALGORITHM IS SPECIFICAL-
LY DESIGNED TO PRECEDE QZ TYPE ALGORITHMS, BUT IMPROVED
PERFORMANCE IS EXPECTED FROM MOST EIGENSYSTEM SOLVERS.
REF.: WARD, R. C., BALANCING THE GENERALIZED EIGENVALUE
PROBLEM, SIAM J. SCI. STAT. COMPUT., VOL. 2, NO. 2, JUNE 1981,
141-152.

*****PARAMETER DESCRIPTION:
ON INPUT:

MA, MB    INTEGER
ROW DIMENSIONS OF THE ARRAYS CONTAINING MATRICES
A AND B RESPECTIVELY, AS DECLARED IN THE MAIN
CALLING PROGRAM DIMENSION STATEMENT;

N    INTEGER
ORDER OF THE MATRICES A AND B;

A    REAL(MA, N)
CONTAINS THE A MATRIX OF THE GENERALIZED
EIGENPROBLEM DEFINED ABOVE;

B    REAL(MB, N)
CONTAINS THE B MATRIX OF THE GENERALIZED
EIGENPROBLEM DEFINED ABOVE;

WK    REAL(N, 6)
WORK ARRAY THAT MUST CONTAIN AT LEAST 6*N STORAGE
LOCATIONS. WK IS ALTERED BY THIS SUBROUTINE.

ON OUTPUT:

A, B  CONTAIN THE BALANCED A AND B MATRICES;

LOW  INTEGER
BEGINNING INDEX OF THE SUBMATRICES OF A AND B
CONTAINING THE NON-ISOLATED EIGENVALUES;

IGH  INTEGER
ENDING INDEX OF THE SUBMATRICES OF A AND B
CONTAINING THE NON-ISOLATED EIGENVALUES. IF
IGH = 1 (LOW = 1 ALSO), THE A AND B MATRICES HAVE
BEEN PERMUTED INTO UPPER TRIANGULAR FORM AND HAVE
NOT BEEN BALANCED;

CSCALE REAL(N)
CONTAINS THE EXPONENTS OF THE COLUMN SCALING FACTORS
IN ITS LOW THROUGH IGH LOCATIONS AND THE REDUCING
COLUMN PERMUTATIONS IN ITS FIRST LOW-1 AND ITS
IGH+1 THROUGH N LOCATIONS;

CPERM REAL(N)
CONTAINS THE COLUMN PERMUTATIONS APPLIED IN GRADING
THE A AND B SUBMATRICES IN ITS LOW THROUGH IGH
LOCATIONS;

WK  CONTAINS THE EXPONENTS OF THE ROW SCALING FACTORS
IN ITS LOW THROUGH IGH LOCATIONS, THE REDUCING ROW
PERMUTATIONS IN ITS FIRST LOW-1 AND ITS IGH+1
THROUGH N LOCATIONS, AND THE ROW PERMUTATIONS
APPLIED IN GRADING THE A AND B SUBMATRICES IN ITS
N+LOW THROUGH N+IGH LOCATIONS.

*****ALGORITHM NOTES:
NONE.

*****HISTORY:
WRITTEN BY R. C. WARD........
SUBROUTINE BALGBK (N,MZ,Z,M,LOW,IGH,CSCALE,CPERM)

*****PARAMETERS:
INTEGER N,MZ,M,LOW,IGH
DOUBLE PRECISION Z(MZ,N),CSCALE(N),CPERM(N)

*****LOCAL VARIABLES:
NONE.

*****FUNCTIONS:
NONE.

*****SUBROUTINES CALLED:
GRADBK, SCALBK

*****PURPOSE:
THIS SUBROUTINE BACK TRANSFORMS THE EIGENVECTORS OF A
GENERALIZED EIGENVALUE PROBLEM A*X = (LAMBDA)*B*X, THAT WAS
BALANCED BY SUBROUTINE BALGEN, TO THOSE OF THE ORIGINAL
PROBLEM.
REF.: WARD, R. C., BALANCING THE GENERALIZED EIGENVALUE
PROBLEM, SIAM J. SCI. STAT. COMPUT., VOL. 2, NO. 2, JUNE 1981,
141-152.

*****PARAMETER DESCRIPTION:
ON INPUT:
MZ INTEGER
ROW DIMENSION OF THE ARRAY Z AS SPECIFIED IN THE
MAIN CALLING PROGRAM DIMENSION STATEMENT;
N INTEGER
ORDER OF THE MATRICES A AND B IN THE EIGENPROBLEM;
M INTEGER
SPECIFIES THE NUMBER OF EIGENVECTORS TO BE TRANSFORMED;
Z REAL(MZ,N)
CONTAINS THE EIGENVECTORS TO BE TRANSFORMED;
LOW INTEGER
SPECIFIES THE BEGINNING INDEX OF THE SUBMATRICES OF
A AND B WHICH WERE BALANCED;
IGH INTEGER
SPECIFIES THE ENDING INDEX OF THE SUBMATRICES OF A AND B WHICH WERE BALANCED;

CScale REAL(N) CONTAINS THE REDUCING COLUMN PERMUTATIONS AND SCALING INFORMATION AS RETURNED FROM BALGEN;

CPerm REAL(N) CONTAINS IN ITS LOW THROUGH IGH LOCATIONS THE COLUMN PERMUTATIONS APPLIED IN GRADING THE A AND B SUBMATRICES AS RETURNED FROM BALGEN.

ON OUTPUT:

Z CONTAINS THE TRANSFORMED EIGENVECTORS.

*****ALGORITHM NOTES:
NONE.

*****HISTORY:
WRITTEN BY R. C. WARD.......
INITIAL DISTRIBUTION

2 Naval Air Systems Command (AIR-00D4)
3 Chief of Naval Research, Arlington (ONR-411-MA, Holland)
2 Naval Sea Systems Command (SEA-09B312)
1 Commander in Chief, U.S. Pacific Fleet (Code 325)
1 Commander, Third Fleet, Pearl Harbor
1 Commander, Seventh Fleet, San Francisco
3 Naval Ship Weapon Systems Engineering Station, Port Hueneme
    Code 5711, Repository (2)
    Code 5712 (1)
1 Naval War College, Newport
1 Air Force Institute of Technology, Wright-Patterson Air Force Base (C. Houpis)
12 Defense Technical Information Center
  1 Langley Research Center (NASA), Hampton (E. Armstrong)
  1 Alphatech, Inc., Burlington, MA (Dr. N. R. Sandell, Jr.)
  1 Cornell University, Ithaca, NY (Department of Computer Science (C. Van Loan)
  1 Honeywell, Inc., Systems and Research Center, Minneapolis, MN (Dr. N. A. Lehtomaki)
  1 Integrated Systems Incorporated, Palo Alto, CA (Dr. R. A. Walker)
  1 Litton Systems, Inc., Woodland Hills, CA (MS 67/35, Dr. R. W. Bass)
  1 McGill University, Montreal, Canada (Department of Computer Science, C. Paige)
  1 New Mexico Engineering Research Institute, Albuquerque, NM (Department of Computer Science, C. Moler)
  1 Northern Illinois University, Dekalb, IL, (Department of Mathematical Sciences, R. Byers)
  1 Rice University, Houston, TX (Department of Mathematical Sciences, J. E. Dennis, Jr.)
  1 Systems Control Technology, Inc., Palo Alto, CA (Dr. A. Emami-Naeini)
  1 University of California, Lawrence Livermore National Laboratory, Livermore, CA (Dr. C. J. Herget)
  5 University of California, Santa Barbara, Santa Barbara, CA (Department of Electrical & Computer Engineering, A. J. Laub)
3 University of Southern California, Los Angeles, CA (Department of Electrical Engineering-Systems,
  L. M. Silverman)
1 University of Wisconsin, Madison, WI (Department of Mathematics, D. Russell)