AN EMBEDDING APPROACH FOR SOLVING RICCATI EQUATIONS

GERALD CHARLES T. GH. D’ANS

UNIVERSITY OF ILLINOIS – URBANA, ILLINOIS

"This document has been approved for public release and sale; its distribution is unlimited"
AN EMBEDDING APPROACH FOR SOLVING RICCATI EQUATIONS

by

GERARD CHARLES Y. GH. D'ANS

This work was supported in part by the Joint Services Electronics Program (U.S. Army, U.S. Navy and U.S. Air Force) under contract DAAB 07-67-C-0199; and in part by Air Force 68-1679.

Reproduction in whole or in part is permitted for any purpose of the United States Government.

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

The author expresses sincere thanks to Dr. P. V. Kokotović, his advisor, for his excellent guidance and his patience during the course of this research. He also appreciates the helpful contributions of Dr. J. B. Cruz, Dr. W. R. Perkins and Dr. J. Medanić. He further thanks his associate M. S. Jamshidi for testing the computer subroutines.

Finally, the author wishes to acknowledge the support of the Coordinated Science Laboratory. Special thanks are due to Mrs. Sherry Kallembach for her superb typing of the manuscript.
# TABLE OF CONTENTS

1. INTRODUCTION ............................................................... 1
   1.1 State Regulator Problem .............................................. 1
   1.2 Kalman-Bucy Filter .................................................... 2
   1.3 Statement of the problem ............................................ 3

2. EMBEDDING APPROACH FOR SOLVING RICCATI EQUATIONS ............ 5
   2.1 Concept of embedding .................................................. 5
   2.2 Embedding for the exact solution .................................... 6
   2.3 Embedding for an approximate solution ............................. 7
   2.4 Performance Analysis of the approximate solution .............. 12

3. EXAMPLES ..................................................................... 14
   3.1 An R-C cable .............................................................. 14
   3.2 A more coupled system ............................................... 19
   3.3 A strip winding mill .................................................... 22

4. COMPUTER ROUTINES ..................................................... 24
   4.1 Standard Method ....................................................... 24
   4.2 Exact Embedding Method .............................................. 25
   4.3 Approximate Embedding Method ...................................... 28

5. CONCLUSION ................................................................. 31

6. LIST OF REFERENCES ..................................................... 32

APPENDICES ................................................................. 33
LIST OF FIGURES

Figure | Page
--- | ---
1. Linear system with optimal control | 34
2. Kalman-Bucy filter | 35
3. Scheme for the R-C cable | 36
4. R-C model of the transmission cable | 37
5. Performances of different methods for example 1 | 38
6. Effect of coupling on K matrix for example 1 | 39
7. Performances of different methods for example 2 | 40
8. Constant speeding winding process | 41
9. Flow chart of UJERIC | 42
10. Flow chart of SECM | 43
11. Interaction of the subroutines called by INTEGRE | 44
12. Flow chart for KPRIMC and K1COMP | 45
13. Flow chart for the use of KPRIMC and K1COMP | 46
1. INTRODUCTION

The Riccati equation needs to be solved in the control theory and the estimation theory. Most frequently, this need arises in the state regulator and the Kalman-Bucy filter design.

No analytical solution of the Riccati equation has been found except in the one-dimensional case. For higher orders, one must use computer techniques to obtain a numerical solution. The conventional approaches fail to be satisfactory for many applications of reasonably high dimensions. The present work gives a way to avoid that kind of difficulty.

1.1 State Regulator Problem

Consider the linear dynamical system

\[ \dot{x} = Ax + Bu \]  

where A and B are \( n \times n \) and \( n \times m \) matrices and \( x \) and \( u \) are \( n \)- and \( m \)-vectors, respectively.

The optimal control \( u^* \) must minimize

\[ J = \frac{1}{2} \int_{t_0}^{T} (x'Qx + u'Ru) dt + \frac{1}{2} x'Tx \]  

where \( Q, R \) and \( T \) are \( n \times n \), \( m \times m \) and \( n \times n \) positive definite and symmetric matrices. The matrix \( Q \) can be positive semi-definite.

The control \( u^* \) is known to be

\[ u^* = -R^{-1} B'Kx \]  

where \( K \) satisfies the Riccati equation.
The block diagram for the optimum system is given in fig. 1. (See Appendix A.)

1.2 Kalman-Bucy Filter

Consider the system in Eq. 1.1 modified to include an additive white noise input \( v(t) \) with

\[
E\{v(t) v'(\tau)\} = \delta(t-\tau) \quad (1.5)
\]

The observation of the state is made through a measurement matrix \( C \) and is corrupted by an additive white noise vector \( w(t) \) with

\[
E\{w(t) w'(\tau)\} = \delta(t-\tau) \quad (1.6)
\]

The system equations are:

\[
\dot{x} = Ax + B(u+v) \quad x(t_0) = a \quad (1.7)
\]
\[
y = Cx + w \quad (1.8)
\]

We assume the hypotheses of observability and controllability are fulfilled. The minimum mean-squared estimate (mmse) of \( x(t) \) from the noisy observation is given by

\[
\dot{x} = A\hat{x} + Bu + K(t)C'R^{-1}[y-C\hat{x}] \quad \hat{x}(t_0) = E[a] \quad (1.9)
\]

where \( K \) satisfies
\[ \dot{K} = AK + KA' - KC'R^{-1}CK + BQ \]  

\[ K(t_0) = E[[x(t_0) - \hat{x}(t_0)][x(t_0) - \hat{x}(t_0)]]' \]  

Eqn. 1.10 is clearly the dual of eqn. 1.4. The same technique can be used for solving each of them. A block diagram of the system is seen in fig. 2 (see appendix A).

1.3 Statement of the problem

In eqns. 1.4 and 1.10, a nonlinear matrix differential equation must be integrated with respect to time. The initial value is given in the Kalman-Bucy problem.

In the state regulator problem, the final value is available which leads to a backward integration. A case of special interest is to find K at a very large time assuming all other matrices time-invariant. That can be used to find the Wiener filter from eqn. 1.10 [1].

Two conventional methods are currently used to determine K(\infty). The standard method is to keep on integrating eqn. 1.4 or eqn. 1.10 until K comes to a steady state value [2]. That can be satisfactory if a properly varying integration step is chosen, which is often the main difficulty: how to conciliate the speed of the procedure with the stability in the integration.

A second approach consists of solving eqn. 1.4 and eqn. 1.10 with \( \dot{K} = 0 \). As easily noticed, the resulting matrix equation is algebraic and nonlinear. A Newton procedure can therefore be applied [3], for which a good first guess is decisive. The main goal of this work is to find a better technique of computing such an initial guess.
The conventional methods are difficult to apply when the system analyzed is of a high dimension and when $K$ slowly converges to the steady state solution.

One approach (section 2.3) reduces the dimensionality of the problem by working with decoupled systems.

The second (section 2.2) avoids the convergence problem. The latter is done by an integration between fixed limits. Both of them are applications of the embedding principle.
2. EMBEDDING APPROACH FOR SOLVING RICCATI EQUATIONS

2.1 Concept of embedding

In an embedding procedure, a problem \( \xi \) and its solution \( \eta \) can be considered as elements of a family \( X \) containing similar problems and of the family \( Y \) containing the corresponding solutions, respectively. The latter can be done in eqn. 1.4 if \( A, S \) or \( Q \) are functions of \( \varepsilon \). \( \varepsilon \) can index the elements of \( X \) and \( Y \) so that \( K_\varepsilon \in Y \) is the solution of eqn. 1.4 and \( A_\varepsilon = A(\varepsilon), S_\varepsilon = S(\varepsilon) \) and \( Q_\varepsilon = Q(\varepsilon) \). The families \( X \) and \( Y \) are called the problem space \( X \) and the solution space \( Y \), respectively. Each problem \( x \in X \) and its solution \( y \in Y \) are denoted by \( (x,y) \in X \times Y \). Moreover, there should be a group \( G \) from \( Y \) into itself such that, for any \( (x_1, y_1) \) and \( (x_2, y_2) \in X \times Y \), there is a \( g_{12} \in G \) with

\[
y_1 = g_{12}y_2
\]

That property can be phrased by noting that \( Y \) is the orbit of any \( y \in Y \), since \( Y = \{gy : g \in G\} \) for any \( y \in Y \). The inverse of a transformation \( g \) exists, iff \( g \) is one-to-one and onto. This must be the case here to ensure the uniqueness of the pair \( (x,y) \) through group transformations.

In section 2.2, the inverse \( g^{-1} \) of \( g \) will be an integration from \( \varepsilon=1 \) to \( \varepsilon=0 \). In section 2.3, \( g^{-1} \) will be an expansion at \( \varepsilon=1 \), the value of which is considered at \( \varepsilon=0 \). The composition property of groups will also ensure uniqueness of any pair \( (x,y) \in X \times Y \).

If there is a pair \( (x_0, y_0) \) such that \( y_0 \) is easily obtained, then any \( (x,y) \in X \times Y \) can be obtained by a group transformation and there is a \( g \in G \) such that \( y = g^{-1}y_0 \).
In an embedding approach, \( g \) must be given an exact expression and a satisfactory approximation \( g^* \) of \( g \) must be computed. \( g^* \) generally differs from \( g \) for two reasons.

A theoretical approximation can first be made, e.g., by voluntarily truncating an exact power series expansion of \( K_\varepsilon \) in terms of \( \varepsilon \), as we will see in section 2.3.

Second, because approximative methods have to be used to obtain a numerical solution for \( g \). For example, an integration performed with a Runge Kutta routine introduces errors, which can also be brought by round-off approximations. This is the case for the method of section 2.2.

The quality of the approximation \( g^* \) can be measured by comparing the values of functions depending on the solutions \( \eta \) and \( g^* y_0 \). In section 2.2 and 2.3, let \( \eta, x_0 \) and \( g^* \) be \( K_1, K_0 \) and \( g^*_1 \), respectively. The function is the maximum of the expression (1.2) over all initial values of the system (1.1) with modulus 1.

2.2 Embedding for the exact solution

Consider the optimization problem in section 1.1. One can express the time-invariant matrices \( A, B, Q \) and \( R \) as the values of some adequate function of \( \varepsilon \) at \( \varepsilon = 1 \).

The Riccati matrix, \( K \) as defined by eqn. 1.4 or eqn. 1.10 depends on \( \varepsilon \) and \( t \). Let for all \( \varepsilon \in [\varepsilon_0, \varepsilon_1] \), \( A, B, Q \) and \( R \) be continuously differentiable functions of \( \varepsilon \). The unique positive definite solution \( K \) of eqn. 1.4 will exist if there is a complete controllability in the state regulator problem. This assumption is understood from now on. This resulting solution is a continuously differentiable function of \( \varepsilon \) for
Suppose that this function can be easily computed at \( \epsilon_0 \). The solution \( K = K(\epsilon) \) of eqn. 1.4 can be obtained for all \( \epsilon \in [\epsilon_0, \epsilon_1] \) from the following differential equation

\[
\frac{dK}{d\epsilon} (A - SK) + (A - SK)' \frac{dK}{d\epsilon} = -K\alpha - \alpha' K \zeta K \kappa - \sigma
\]  

(2.1)

where

\[
\alpha = \frac{dA}{d\epsilon}, \quad \zeta = \frac{dS}{d\epsilon}, \quad \text{and} \quad \sigma = \frac{dQ}{d\epsilon}
\]

We can use \( K(\epsilon_0) \) as an initial condition for the integration of eqn. 2.1. The eqn. 2.1 is obtained by differentiating eqn. 1.4 with respect to \( \epsilon \).

Since the \( K \) matrix is symmetric, to solve eqn. 2.1 is equivalent to find the solution of a linear system of \( n(n+1)/2 \) algebraic equations at each integration step using the updated value of the \( K \) matrix in the second member.

### 2.3 Embedding for an approximate solution

In this case, \( A, B, Q \) and \( R \) expressed in terms of \( \epsilon \) and have the following special structure

\[
A = \begin{pmatrix} A_1 & \epsilon A_{12} \\ \epsilon A_{21} & A_2 \end{pmatrix}, \quad B = \begin{pmatrix} B_1 & \epsilon B_{12} \\ \epsilon B_{21} & B_2 \end{pmatrix}, \quad Q = \begin{pmatrix} Q_1 & \epsilon Q_{12} \\ \epsilon Q_{21} & Q_2 \end{pmatrix}
\]

(2.2)
The terminal cost, if any, is the quadratic form of:

\[
T = \begin{pmatrix}
T_1 & \varepsilon T_{12} \\
\varepsilon T_{21} & T_2
\end{pmatrix}
\]

where \( A_1, A_2, A_{12}, A_{21} \) are \( n_1 \times n_1, n_2 \times n_2, n_1 \times n_2 \) and \( n_2 \times n_1 \) matrices, \( n_1 + n_2 = n \), respectively.

\( B_1, B_2, B_{12}, B_{21} \) are \( n_1 \times m_1, n_2 \times m_2, n_1 \times m_2 \) and \( n_2 \times m_1 \) matrices, \( m_1 + m_2 = m \), respectively.

The above matrices are introduced in

\[
\dot{K} = -KA - A'K + KSK - Q \quad K(T,\varepsilon) = T
\]

with \( S = BR^{-1}B' \). In its asymptotic form, 2.3 reduces to

\[
KA + A'K - KSK + Q = 0
\]

Again the matrix \( K \) depends on \( \varepsilon \) and \( t \). Eqn. 2.4 is a system of \( v = \frac{n}{2}(n+1) \) nonlinear algebraic equations. As is easily seen for \( \varepsilon = 0 \), the system splits in two disconnected and simpler subsystems.

Unfortunately, such an approximation may be far from optimum for \( \varepsilon = 1 \) and the system may even be unstable. Assuming that the solution \( K(t,\varepsilon) \) of eqn. 2.4 is analytical in \( \varepsilon \) for any \( t \in [t_0, T] \), we define \( M \) as a truncated Mac-Laurin series expansion of the optimum gain \( K(t,\varepsilon) \)

\[
M = K(t,0) + \varepsilon \frac{\partial K(t,0)}{\partial \varepsilon} + \ldots + \frac{\varepsilon^m}{m!} \frac{\partial^m K(t,0)}{\partial \varepsilon^m}
\]
The next pages are devoted to the computation of $K(t,0)$ and the derivatives $\frac{\partial^m K(t,0)}{\partial \epsilon^m}$.

Partitioning $K$ as follows

$$
\begin{align*}
K_1 + \epsilon \frac{\partial K_1(t,0)}{\partial \epsilon} + \frac{\epsilon^2}{2} \frac{\partial^2 K_1(t,0)}{\partial \epsilon^2} + \ldots &+ K_{12} + \epsilon \frac{\partial K_{12}(t,0)}{\partial \epsilon} + \frac{\epsilon^2}{2} \frac{\partial^2 K_{12}(t,0)}{\partial \epsilon^2} + \ldots \\
K_{21} + \epsilon \frac{\partial K_{21}(t,0)}{\partial \epsilon} + \frac{\epsilon^2}{2} \frac{\partial^2 K_{21}(t,0)}{\partial \epsilon^2} + \ldots &+ K_2 + \epsilon \frac{\partial K_2(t,0)}{\partial \epsilon} + \frac{\epsilon^2}{2} \frac{\partial^2 K_2(t,0)}{\partial \epsilon^2} + \ldots
\end{align*}
$$

and letting $\epsilon = 0$ yield

$$
\begin{align*}
\dot{K}_1 &= -K_1 A_1 - A_1' K_1 + K_1 S_1 K_1 - Q_1 + K_{12} S_2 K_{12} \\
K_1(T,0) &= T_1 \quad (2.6) \\
\dot{K}_2 &= -K_2 A_2 - A_2' K_2 + K_2 S_2 K_2 - Q_2 + K_{12} S_1 K_{12} \\
K_2(T,0) &= T_2 \quad (2.7) \\
\dot{K}_{12} &= -K_{12} G_2 - G_1' K_{12} \quad K_{12}(T,0) = 0 \quad (2.8)
\end{align*}
$$

where $S_i = B_i R_i^{-1} B'_i$ and $G_i = A_i - S_i K_i$ $i = 1, 2$.

Eqn. 2.8 is homogeneous with final values being zero. Thus, $K_{12} = 0$ for all $t$ and the first term $K(t,0)$ of the series of eqn. 2.5 is

$$
K = \begin{pmatrix} K_1 & 0 \\ 0 & K_2 \end{pmatrix}
$$

where $K_1 = K_1(t,0)$ and $K_2 = K_2(t,0)$ are obtained from decoupled solutions.
by
\[ \dot{K}_1 = -K_1 A_1 - A_1' K_1 + K_1 S_1 K_1 - Q_1 \] (2.9)
\[ K_1(T,0) = T_1 \]
\[ \dot{K}_2 = -K_2 A_2 - A_2' K_2 + K_2 S_2 K_2 - Q_2 \] (2.10)
\[ K_2(T,0) = T_2 \]

With the controllability hypotheses assumed, those equations reduce to
\[ K_1 A_1 + A_1' K_1 - K_1 S_1 K_1 + Q_1 = 0 \] (2.11)
\[ K_2 A_2 + A_2' K_2 - K_2 S_2 K_2 + Q_2 = 0 \] (2.12)

in the time invariant case. These two problems are similar to the original one, but with dimensionalities reduced approximately by a factor of four if \( n_1 \sim n_2 \).

The computation continues in a sequence of steps. At step \( j \), \( j = 1, 2, \ldots, m \), the term \( \frac{\partial^j K(t,0)}{\partial \varepsilon^j} = K^j \) is obtained. (From now on, derivatives with respect to \( \varepsilon \) will be denoted by superscripts, e.g., \( K^1, K^2, \ldots, A^1, B^1, S^1, S^2 \), etc....)

Theorem 1 below summarizes the procedure, showing that there are only two different forms of steps \( j \) for \( j = 2i \) and \( j = 2i+1 \). The proof of the theorem and expressions for \( F_{1}^{2i-1}, F_{2}^{2i-1} \) and \( F_{12}^{2i} \) are given in appendix C.

Theorem 1

(a) when \( j = 2i \), at \( \varepsilon = 0 \)
\[ K_{2i} = \frac{\partial^{2i} K(t,0)}{\partial \varepsilon^{2i}} = \begin{pmatrix} K_{12}^{2i} & 0 \\ 0 & K_{22}^{2i} \end{pmatrix} \]

\( i = 0,1,2, \ldots \)

That is, \( K_{12}^{2i} = 0 \) and the \( n_1 \times n_1 \) and \( n_2 \times n_2 \) symmetric matrices \( K_{12}^{2i} \) and \( K_{22}^{2i} \) are obtained from eqns. 2.11 and 2.12 for \( i = 0 \). For \( i > 0 \), they are obtained from the decoupled linear equations

\[ K_{12}^{2i-1} = -K_{12}^{2i} G_2^{2i-1} - G_1^{2i} K_{12}^{2i-1} \]

\[ K_{22}^{2i-1} = -K_{22}^{2i} G_2^{2i-1} - G_2^{2i} K_{22}^{2i-1} \]

\[ K_{12}^{2i-1} (T,0) = 0 \]

\[ K_{22}^{2i-1} (T,0) = 0. \] (2.13) (2.14)

where \( G_1 = A_1 - S_1 K_1 \) and \( G_2 = A_2 - S_2 K_2 \). \( F_1^{2i-1} \) and \( F_2^{2i-1} \) do not depend on \( K_1^n \) for \( n = 2i, 2i+1, \ldots \). For the time-invariant problem, we have to solve

\[ K_{12}^{2i} G_1 + G_1^{2i} K_{12}^{2i} = -F_1^{2i-1} \]

\[ K_{22}^{2i} G_2 + G_2^{2i} K_{22}^{2i} = -F_2^{2i-1} \] (2.15)

(b) when \( j = 2i+1 \), at \( \varepsilon = 0 \)

\[ K_{2i+1} = \frac{\partial^{2i+1} K(t,0)}{\partial \varepsilon^{2i+1}} = \begin{pmatrix} 0 & K_{12}^{2i+1} \\ (K_{12}')^{2i+1} & 0 \end{pmatrix} \]

\( i = 0,1, \ldots \) i.e., \( K_{12}^{2i+1} = 0 \) and \( K_{22}^{2i+1} = 0. \)
The $n_1 \times n_2$ matrix $K_{12}^{2i+1}$ is obtained from the linear equation

$$K_{12}^{2i+1} = K_{12}^{2i+1} G_2 + G_1 K_{12}^{2i+1} - F_{12}^{2i}$$

$$K_{12}^{2i+1} (T, 0) = 0$$

(2.17)

where $F_{12}^{2i}$ does not depend on $K_{12}^n$ for $n = 2i+1, 2i+2, \ldots$. For the time-invariant problem eqn. 2.17 yields

$$K_{12}^{2i+1} G_2 + G_1 K_{12}^{2i+1} = -F_{12}^{2i}$$

(2.18)

An examination of eqns. 2.15, 2.16 and 2.18 shows that the homogeneous parts are the same for all $i$'s. Therefore, the calculations of the series 2.5 become especially expedient for the time-invariant problem, since only three inversions of matrices are necessary to obtain $\frac{\partial^i K(\varepsilon, 0)}{\partial \varepsilon^i}$ for all $i$'s and those matrices do not exceed $\frac{\sqrt{2}}{2}$ in dimensionality.

2.4 Performance Analysis of the approximate solution

In this section, the question of quality of approximation is examined. Let $J_K(\varepsilon)$ represent the performance index in eqn. 1.2 obtained using the optimal controller in eqn. 1.3. The system for this case is

$$\dot{x} = (A - SK)x$$

(2.19)

Let $J_M(\varepsilon)$ be the performance index in eqn. 1.3 obtained using the matrix $M$ of eqn. 2.5 as an approximation to $K$. The system for this case is

$$\dot{x} = (A - SM)x$$

(2.20)
The performances of the two systems are evaluated by comparing the Mac-Laurin series expansions of $J_M$ and $J_K$. In appendix D, the following theorem is proved.

**Theorem 2**

Let $J_K(\varepsilon)$ and $J_M(\varepsilon)$ be defined as above, with $M$ as the $m$-th-order polynomial of eqn. 2.5. Then the first $2m+2$ terms of the Mac-Laurin series of $J_M(\varepsilon)$ are equal to the corresponding $2m+2$ terms of the Mac Laurin series for $J_K(\varepsilon)$, i.e.,

$$\left. \frac{d^i J_M}{d\varepsilon^i} \right|_{\varepsilon=0} = \left. \frac{d^i J_K}{d\varepsilon^i} \right|_{\varepsilon=0}$$

for $i = 0, 1, \ldots, 2m+1$

Thus the $m$-th-order approximation $M$ of the optimal gain matrix $K$ results in a $(2m+1)$-th-order approximation $J_M$ of the optimum performance $J_K$.

For example, with $m = 2$, that is, by solving one eqn. 2.15, one eqn. 2.16 and one eqn. 2.18, $J_M$ approximates $J_K$ to the fifth order in $\varepsilon$. 
3. EXAMPLES

3.1 An R-C cable

The problem is to design a feedback control which forces the voltage along a cable used for the transmission of electrical power to zero. The nominal voltage and power are 220 kV and 500 MW, respectively. A complete computation according to the safety rules* yields the following linear characteristics for the cable:

- Linear resistance: 0.0283 ohm/km.
- Linear reactance: 0.0262 ohm/km.
- Linear admittance: \( \frac{1}{5300} \) siemens/km.

One may neglect the reactance and study an R-C model of the cable. The linear capacitance computed for the linear admittance at 50 Hz is:

\[ C = 0.6 \, \mu \text{F/km}. \]

The equivalent impedance for the rest of the network is 0.01 p.u., resistive for simplification:

1 p.u. of resistance = 96.6 Ω

The power delivered to the resistive load is assumed to be 250 MW. The equivalent resistance is: \( R_L = 195 \, \Omega \).

Suppose that one can measure the voltage at the points given in fig. 3. The equivalent model is built with \( \pi \) cells in fig. 4. The values

*According to Belgian Standard - See ref. [4].
of the components are:

\[
\begin{align*}
R_1 &= 0.283 \, \Omega \\
C_1 &= 3 \, \mu F \\
R_2 &= 0.142 \, \Omega \\
C_2 &= 4.5 \, \mu F \\
R_3 &= 0.283 \, \Omega \\
C_3 &= 4.5 \, \mu F \\
R_4 &= 0.283 \, \Omega \\
C_4 &= 6 \, \mu F \\
R_5 &= 0.182 \, \Omega \\
C_6 &= 1.5 \, \mu F \\
R_6 &= 0.966 \, \Omega \\
R_L &= 195 \, \Omega
\end{align*}
\]

The equations of the system in the normal form with the \( e_i \)'s, \( i = s, \ldots, 6 \), as state variables, are described in [8]. These are:

\[
\begin{align*}
C_i \frac{de_i}{dt} &= \frac{e_{i+1} - e_i}{R_i} + \frac{e_{i-1} - e_i}{R_{i-1}} \\
C_1 \frac{de_1}{dt} &= \frac{e_2 - e_1}{R_1} + \frac{E - e_1}{R_s} \\
C_5 \frac{de_5}{dt} &= -\frac{e_6}{R_L} - \frac{e_5 - e_6}{R_5}
\end{align*}
\]

The matrix \( A \) contains only terms of the form \( a_{ii}, a_{i-1,i}, a_{i+1,i} \) with

\[
\begin{align*}
a_{ii} &= \frac{1}{C_i} \left( \frac{1}{R_i} + \frac{1}{R_{i-1}} \right) \\
a_{i+1,i} &= \frac{1}{R_i C_{i+1}} \\
A_{i,i+1} &= \frac{1}{R_i C_i}
\end{align*}
\]
The matrix $B$ is a column with the first element being $\frac{1}{R_s C_1}$ and the other ones zero.

The elements of the matrices $A$ and $B$ are given below:

in sec$^{-1}$:

\[
\begin{array}{cccc}
  a_{11} & = & -1.525 \times 10^6 & \\
  a_{12} & = & 1.18 \times 10^6 & \\
  a_{21} & = & 0.81 \times 10^6 & \\
  a_{22} & = & -2.43 \times 10^6 & \\
  a_{23} & = & 1.62 \times 10^6 & \\
  a_{32} & = & 1.62 \times 10^6 & \\
  a_{33} & = & -2.43 \times 10^6 & \\
  a_{34} & = & 0.81 \times 10^6 & \\
  a_{43} & = & 0.59 \times 10^6 & \\
  a_{44} & = & -1.18 \times 10^6 & \\
  a_{45} & = & 0.59 \times 10^6 & \\
  a_{54} & = & 0.81 \times 10^6 & \\
  a_{55} & = & -2.43 \times 10^6 & \\
  a_{56} & = & 1.62 \times 10^6 & \\
  a_{65} & = & 4.86 \times 10^6 & \\
  a_{66} & = & -4.86 \times 10^6 & \\
\end{array}
\]

$b_1 = 0.345 \times 10^6$ with $b_i = 0$ \(i = 2, \ldots, 6\).

A performance index of the form (1.2) is used without a terminal cost ($F = 0$). $Q = 10^6 I_6$ and $R$ is $10^6$, scalar. Clearly, one can reduce $A$, $B$, $Q$ and $R$ by a factor of $10^6$ in

\[-A'K - KA + KBR^{-1}B'K - Q = 0\] (3.1)

without any change for the solution $K$.

The time-invariant solution obtained by the standard method is:

\[
\begin{bmatrix}
  0.796 & 0.931 & 0.870 & 1.11 & 0.790 & 0.263 \\
  0.931 & 1.755 & 1.677 & 2.176 & 1.567 & 0.522 \\
  0.870 & 1.677 & 1.911 & 2.495 & 1.821 & 0.608 \\
  1.11 & 2.176 & 2.495 & 4.365 & 3.338 & 1.123 \\
  0.790 & 1.567 & 1.821 & 3.338 & 3.268 & 1.134 \\
  0.263 & 0.522 & 0.608 & 1.123 & 1.134 & 0.480
\end{bmatrix}
\]
The zeroth-order approximation for decoupled subsystems is given by partitioning $A$ in $3 \times 3$ submatrices.

$$K_1 = \begin{bmatrix}
0.530 & 0.402 & 0.257 \\
0.402 & 0.709 & 0.468 \\
0.247 & 0.468 & 0.516
\end{bmatrix}$$

$$K_2 = \begin{bmatrix}
1.139 & 1.043 & 0.360 \\
1.043 & 1.657 & 0.599 \\
0.360 & 0.599 & 0.303
\end{bmatrix}$$

The approximate solution $\xi 2.5$ obtained for $m = 2$ is:

$$M = \begin{bmatrix}
0.614 & 0.566 & 0.444 & 0.334 & 0.247 & 0.083 \\
0.566 & 1.029 & 0.833 & 0.639 & 0.486 & 0.163 \\
0.444 & 0.833 & 0.930 & 0.712 & 0.564 & 0.190 \\
0.334 & 0.639 & 0.712 & 3.087 & 3.176 & 1.144 \\
0.247 & 0.486 & 0.564 & 3.176 & 4.489 & 1.750 \\
0.083 & 0.163 & 0.190 & 1.144 & 1.750 & 0.818
\end{bmatrix}$$

The procedure of section 2.2 can be implemented by expressing $A = A(\xi)$ for $\xi = 1$ as follows:

$$A(\xi) = \begin{bmatrix}
1.18\xi & 0 & 0 & 0 & 0 & 0 \\
-2.43 & 1.62\xi & 0 & 0 & 0 & 0 \\
1.62\xi & -2.43 & 0.81\xi & 0 & 0 & 0 \\
0 & 0 & 0.59\xi & -1.18 & 0.59\xi & 0 \\
0 & 0 & 0.81\xi & -2.43 & 1.62\xi & 0 \\
0 & 0 & 0 & 0 & 4.86 & -4.86
\end{bmatrix}$$
\[ B(\varepsilon) = \begin{pmatrix} 0.345 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \]

\[ Q(\varepsilon) = I_6 \]

\[ R(\varepsilon) = 1. \]

The solution \( K(\varepsilon) \) of eqn. 3.1 for \( \varepsilon=0 \) is a 6x6 diagonal matrix whose elements are computed below.

S is diagonal with all zero elements except \( S_{11} = b_1^2 \).

The first element \( k_{11} \) of \( K(\varepsilon) \) for \( \varepsilon = 0 \) satisfies

\[ -0.690 k_{11} + b_1^2 k_{11}^2 - 1 = 0 \quad (3.2) \]

The others can be computed from

\[ -2a_{ii} k_{ii} - 1 = 0 \quad (3.3) \]

Note that \( K(\varepsilon) \), for \( \varepsilon = 0 \), must be positive definite and real.

To ensure that, all the six subsystems are disconnected at \( \varepsilon = 0 \) and completely controllable. In the case of the last five subsystems, where no control of the state is available, the asymptotic stability is required only. \( K(0) \) is found to be:

\[ K(0) = \text{diag} (0.299, 0.206, 0.206, 0.424, 0.206, 0.103) \]

Integrating eqn 2.1 with 250 steps from \( \varepsilon=0 \) to \( \varepsilon=1 \), we obtain the exact solution without any significant change.

Fig. 5 gives a comparison between the performance indices defined in eqn. 1.2 for \( T = \infty \) obtained by using the M approximation of \( K \) and \( m = 0,1,2 \).
The initial condition for the state vector is

\[ x'_0 = (3, 2, 1, 0, 1, 2) \]

Fig. 6 presents \( k_{11}(\varepsilon) \), \( k_{12}(\varepsilon) \) and \( k_{16}(\varepsilon) \) obtained by the method of section 2.2. This can be interpreted as a measure of the interaction between subsystems, when \( \varepsilon \) increases.

### 3.2 A more coupled system

Let

\[
A = \begin{bmatrix}
2 & 1 & 2 & 2 & 1.6 & 1.6 & 1.6 \\
3 & 1 & 2 & 2 & 2.4 & 0.8 & 1.6 \\
2 & 3 & 2 & 1 & 0.8 & 2.4 & 1.6 \\
1 & 1 & 1 & 1 & 1.6 & 1.6 & 0.8 \\
1.6 & 0.8 & 1.6 & 0.8 & 0 & -1 & -1 \\
1.6 & 0.8 & 1.6 & 1.6 & 5 & 5 & 0 \\
0.8 & 1.6 & 0 & 0.8 & 0 & 2 & 1
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
2 & 3 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 2 & 5 \\
0 & 0 & 0 & 0.5 & -0.7 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

\[
Q = \text{diag (10, 10, 10, 10, 10, 10, 10)}
\]

and

\[
R = \text{diag (2, 1, 1, 1, 10)}
\]

Note that most of the elements of \( A \) are of the same order of magnitude, which does not suggest a suitable decoupling. Assume that the
following partitioning is made for $\varepsilon = 0.8$.

$$
A_1 = \begin{bmatrix}
2 & 1 & 2 & 2 \\
3 & 1 & 2 & 2 \\
2 & 3 & 2 & 1 \\
1 & 1 & 1 & 1
\end{bmatrix}
\quad A_{12} = \begin{bmatrix}
2 & 2 & 2 \\
3 & 1 & 2 \\
1 & 3 & 2 \\
2 & 2 & 1
\end{bmatrix}
$$

$$
A_{21} = \begin{bmatrix}
2 & 1 & 2 & 1 \\
2 & 1 & 2 & 2 \\
1 & 2 & 0 & 1
\end{bmatrix}
\quad A_2 = \begin{bmatrix}
0 & -1 & -1 \\
5 & 5 & 0 \\
0 & 2 & 1
\end{bmatrix}
$$

If $\varepsilon = 0$, the solutions of the ten nonlinear scalar equations of eqn. 2.9 and six nonlinear scalar equations of eqn. 2.10 are, respectively

$$
K_1 = \begin{bmatrix}
1.28 & -0.11 & 0.78 & 0.30 \\
-0.11 & 4.10 & 1.91 & -2.96 \\
0.78 & 1.91 & 8.07 & -5.11 \\
0.30 & -2.96 & -5.11 & 10.43
\end{bmatrix}
$$

$$
K_2 = \begin{bmatrix}
2.12 & 3.03 & 1.06 \\
3.03 & 12.00 & 6.55 \\
1.06 & 6.55 & 15.23
\end{bmatrix}
$$

The solution of the twelve linear scalar equations of 2.18 for $j = 2i+1 = 1$ is

$$
K_{12}^1 = \begin{bmatrix}
0.67 & 1.60 & 0.22 \\
0.22 & 0.35 & 3.02 \\
-0.66 & -3.20 & -14.87 \\
2.39 & 6.56 & 13.41
\end{bmatrix}
$$
and the solutions of the ten linear scalar equations of 2.15 and the six linear equations of 2.16 are, respectively:

\[
K_1^2 = \begin{bmatrix}
0.46 & 0.09 & 1.04 & -0.16 \\
0.09 & 0.81 & -3.50 & 3.59 \\
1.04 & -3.50 & 37.86 & -31.16 \\
-0.16 & 3.59 & -31.16 & 25.60
\end{bmatrix}
\]

\[
K_2^2 = \begin{bmatrix}
-0.07 & -1.35 & 0.97 \\
-1.35 & -7.87 & -5.79 \\
0.97 & -5.79 & 32.19
\end{bmatrix}
\]

From \(K_1, K_2, K_1^1, K_1^2\) and \(K_2^2\), form \(M\) for \(\epsilon = 0.8\).

\[
M = \begin{bmatrix}
1.43 & -0.08 & 1.11 & 0.25 & 0.53 & 1.28 & 0.17 \\
-0.08 & 4.36 & 0.79 & -1.81 & 0.18 & 0.28 & 2.42 \\
1.11 & 0.79 & 20.18 & -15.08 & -0.52 & -2.56 & -11.89 \\
0.25 & -1.81 & -15.08 & 18.63 & 1.91 & 5.25 & 10.72 \\
0.53 & 0.18 & -0.52 & 1.91 & 2.01 & 2.60 & 1.37 \\
1.28 & 0.28 & -2.56 & 5.25 & 2.60 & 9.48 & 4.69 \\
0.17 & 2.42 & -11.89 & 10.72 & 1.37 & 4.69 & 25.53
\end{bmatrix}
\]

The standard method would give as an optimal \(K\) for \(\epsilon = 0.8\) results:

\[
K = \begin{bmatrix}
1.38 & -0.20 & 1.59 & -0.26 & 0.41 & 1.13 & -0.51 \\
-0.20 & 4.34 & 0.46 & -1.75 & 0.10 & -0.48 & 2.59 \\
1.59 & 0.46 & 31.09 & -24.58 & -1.10 & 1.97 & -21.67 \\
-0.26 & -1.75 & -24.58 & 26.82 & 2.25 & 1.10 & 18.74 \\
0.41 & 0.10 & -1.10 & 2.25 & 2.08 & 2.34 & 1.26 \\
1.13 & -0.48 & 1.97 & 1.10 & 2.34 & 9.58 & 0.66 \\
-0.51 & 2.59 & -21.67 & 18.74 & 1.26 & 0.66 & 23.95
\end{bmatrix}
\]
obtained from 28 coupled nonlinear equations of the form of eqn. 2.3.

In Fig. 7, performances achieved with zeroth-, first- and second-order approximations of $K$ are compared with the optimal performance $J_K$ for different $\varepsilon$ in the interval $[0, 1.1]$ and for an arbitrarily chosen initial condition. Observe that, despite substantial differences in some elements of $M$ and $K$, $J_M$ for $m = 2$ remains within 15% of the optimum $J_K$ for the range of $\varepsilon$ considered.

3.3 A strip winding mill

Fig. 8 gives a short description of a strip winding mill, for which a tension regulator problem is designed. The state variables are the tension $x_2$ and the angular velocity $x_1$. The coil diameter $\rho$ which slowly varies from $\rho = 1$ to $\rho = 3$ strongly influences the system dynamics. For example, the coil moment of inertia increases 81 times during the process. Therefore, if one adopts a quadratic performance index to optimize this system, the feedback matrix must depend on $\rho$. Numerical values of a particular plant [5] are used to clarify the computational procedure,

\[
A(\varepsilon) = \begin{pmatrix} -0.04\varepsilon & -0.4\varepsilon \\ 261 & -0.36 \end{pmatrix}, \quad B(\varepsilon) = \begin{pmatrix} 0.09\varepsilon \\ 0 \end{pmatrix}
\]

\[
Q = \begin{pmatrix} 0 & 0 \\ 0 & 500 \end{pmatrix}, \quad R = 1
\]

where $\varepsilon = \frac{1}{\rho^2}$ and therefore $\varepsilon \in \left[\frac{1}{9}, 1\right]$. 
The Riccati equation (1.4) becomes, when $\dot{K} = 0$,

\[
\begin{align*}
&b_1 k_{11}^2 - 2a_{11} k_{11} - 2a_{21} k_{12} = 0 \\
&b_1 k_{11} k_{12} - a_{12} k_{11} - a_{22} k_{12} - a_{11} k_{12} - a_{21} k_{22} = 0 \quad (3.4) \\
&b_1 k_{12}^2 - 2a_{12} k_{12} - 2a_{22} k_{22} - q_{22} = 0
\end{align*}
\]

where $a_{11}, a_{12}, \ldots, b_1, q_{22}, k_{11}, k_{12}, k_{22}$ are the obvious elements of $A, B, Q$ and $K$. The solution of eqn. 2.1 in section 2.2 is equivalent to

\[
\begin{bmatrix}
\frac{dk_{11}}{d\varepsilon} \\
\frac{dk_{12}}{d\varepsilon} \\
\frac{dk_{22}}{d\varepsilon}
\end{bmatrix} =
\begin{bmatrix}
e + 0.2e^2 k_{11} & -6500 & 0 \\
5e + 0.1e^2 k_{12} & 4.5 + 0.5e + 0.1e^2 k_{11} - 3250 \\
0 & 10e + 0.2e^2 k_{12} & 9
\end{bmatrix}^{-1}
\begin{bmatrix}
k_{11} + 0.2e k_{11}^2 \\
0.5k_{12} + 5k_{11} \\
0.2e k_{11} k_{12}
\end{bmatrix}
\]

(3.5)

Thus, at each integration step in the method of section 2.2, the procedure requires the solution for $\frac{dK}{d\varepsilon}$ of eqn. 2.1. This is equivalent to perform the operation of eqn. 3.5. This operation is simpler than solving the system 3.4 for different $\varepsilon$. Results are available in ref. [5].
4.1 Standard Method

The subroutine UJERIC has been developed for solving eqn. 1.4. A description of UJERIC is given for the time-invariant case. A more general routine is also available.

Description and use of UJERIC

UJERIC (A,B,Q,R,RICC,ACR,S,VECT,N,M,TO,TF,LL,RICCN,RAC,JOI,ERR)

Refer to eqn. 1.4 for the meaning of the arguments given below.

- **A**: nxn matrix A
- **B**: nxm matrix B
- **Q**: nxn matrix Q
- **R**: m-dimensional vector whose components are the diagonal elements of R
- **RICC**: nxn Riccati matrix K
- **S**: nxn matrix \( S = BR^{-1}B' \)
- **ACR, RICCN, RAC**: nxn work matrices
- **VECT**: n-dimensional work vector
- **N, M**: n,m as defined for eqn. 1.4
- **TO, TF**: initial and final time of integration
- **LL**: number of integration steps between TO and TF
- **JOI**: RICC is printed every (JOI+1)th steps with the time at which it has been computed
- **ERR**: if the largest element of K is smaller in modulus than ERR, the routine stops integrating. That test is performed on RICC, when it is being printed.
An integration by the Runge Kutta method is performed in UJERIC. Moreover the following subroutines must be included in the program:

- **RICCATI**: computing one step of integration
- **DOTK**: computing $K$
- **SCOMP**: computing $S$ from $B$ and $R$
- **IMPMAT**: printing matrices

$A, B, Q, R$, and $RICC = K(TF)$ must be read before calling UJERIC. $N, M, TO, TF, LL, JOI$ and $ERR$ are defined in the arguments of UJERIC, when it is called.

A flow chart of UJERIC is given in fig. 9a. Fig. 9b gives a survey of the interactions between the subroutines called by UJERIC.

### 4.2 Exact Embedding Method

Eqn. 2.1 is solved by a routine of the type INTEGRE. The second member is supposed to contain only $-K\alpha \alpha' K$.

**Description and use of INTEGRE**

\[
\text{INTEGRE}(A, S, RIC, RICN, RICE, RICSTOR, SECV, Q, AROND, E0, EF, AMAT, VECMAT, N, M, LL, JOI, AE).
\]

The arguments have the following meaning in eqn. 2.1:

- **$A$**: $n \times n$ matrix $A$
- **$RIC$**: $n \times n$ matrix $K$
- **$S$**: $n \times n$ matrix $S = BR^{-1}B'$
- **$E0, EF$**: initial and final value of the $\varepsilon$-integration
- **$N, M$**: usual definition of $n$ and $m$
LL, JOI : same as in UJERIC (see section 4.1)
AE : nxn matrix α
RICN, RICE, RICSTOR, AROND are nxn work matrices with AROND = A-SK
SECV, VECMAT : nnl-dimensional work vectors with nnl = \( \frac{n}{2}(n+1) \)
AMAT : nnlXnnl work matrix

A, RIC and S must be read or computed before calling INTEGRE and specific values must be assigned in the list of arguments for E0, EF, N, M, LL and JOI.

Moreover, the following subroutines must be included in the program:

- **STEPINT** (computes one step of integration by Runge Kutta)
- **DKE** (computes \( \frac{dK}{de} \))
- **CAROND** (computes A-SK from A, S and K)
- **MATCOEF** (computes the nnlXnnl matrix of coefficients for the linear system in eqn. 2.1)
- **SOLRES** (solves Ax = b for x)
- **IMPMAT** (prints matrices)
- **SECM** (see below)

Depending on the second member, minor changes are introduced in 3 steps:

1. First the routine SECM is modified as follows.

   This routine computes the second member of eqn. 2.1 denoted SM. It stores the upper triangular part of the symmetric nxn matrix SM into the nnl-dimensional vector SECM by the following rule:
\[
\text{SEC}_V \left[ \frac{(2n-i+1)\frac{i-1}{2} + (j-i+1)}{2} \right] = SM(i,j) \quad \text{for } j \geq i \text{ and } j, i \leq n
\]

This is done by the procedure of fig. 10.

2. If any new array X is needed besides A, B and RIC, X is included after JOI in the arguments of INTEGRE, after E in the arguments of STEPINT and after N in the arguments of DKE. It should also be mentioned in the list of SECM.

A statement DIMENSION X(1) must be added in the dimension field of STEPINT, INTEGRE, DKE and SECM.

3. Any matrix or vector X in eqn. 2.1 varying with \( \epsilon \), except RIC, must be computed by a subroutine

\[XCOMP(X,E), \ E \text{ standing for } \epsilon\]

These subroutines will be called after the statements 8001 and 8002 and one statement after 8004 in a listing available for interested readers.*

If S depends on \( \epsilon \), apply step 3 above, otherwise compute S by SCOMP before calling INTEGRE.

Fig. 11 describes the interaction between the subroutines called by INTEGRE.

The output of the subroutine is \( K(\epsilon) \) printed at a frequency adjustable by the values of JOI, LL, EO and EF.

*Available at CSL FORTRAN 60 Master.
4.3 Approximate Embedding Method

Computation of $K_{12}^1$

The equation 2.2 is solved for $K_{12}^1$ by the subroutine KPRIMC. It is, however, necessary first to compute $K_1$ and $K_2$ for the decoupled system. In this case, $B_{12}^1$, $B_{21}^1$ and $Q_{12}^1$ are assumed to be zero. As seen further, minor corrections are needed to treat other cases.

Description and use of KPRIMC

KPRIMC(D,EM,VECMAT,S1,S2,RI1,RI2,A1,A2,C1,C2,ACOM,VECSECM,PRIMK,N,NU)

Refer to eqn. 2.2 and 2.18 with $i=0$ for the meaning of the arguments given below:

- $RI_1$ : $n_1 \times n_1$ $K_1$ matrix
- $RI_2$ : $n_2 \times n_2$ $K_2$ matrix
- $S_1$ : $n_1 \times n_1$ matrix defined as $B_{11}^{-1}B_{11}'$
- $S_2$ : $n_2 \times n_2$ matrix defined as $B_{22}^{-1}B_{22}'$
- $A_1$ : $n_1 \times n_1$ matrix $A_1$
- $A_2$ : $n_2 \times n_2$ matrix $A_2$
- $C_1$ : $n_1 \times n_2$ matrix $A_{12}^{-1}$
- $C_2$ : $n_2 \times n_1$ matrix $A_{21}^{-1}$
- $N,NU$ : $n,n_1$ as defined in section 2.3
- VECSECN,VECMAT : $n_1 \times n_2$ - work vector
- ACOM : $n_1 n_2 \times n_1 n_2$ work matrix
- EM and D : $n_1 \times n_1$ matrix $G_1$ and $n_2 \times n_2$ matrix $G_2$
- PRIMK : $n_1 \times n_2$ matrix $K_{12}^1$
RI1, RI2, S1, S2, A1, A2, C1, C2 must be read before calling KPRIMC. N and NU are defined in the list of arguments.

If $F_{12}^0$ has a more general structure, minor changes must be made in KPRIMC and AIDE subroutines.

AIDE computes $n_1 \times n_2$ matrix $F_{12}^0$ and stores it in VECSECM with the following index correspondence

\[
\text{VECSECM}[(i-1)n_2 + j] = \text{VECSECM}(\text{JOAN})
\]

\[= F_{12}^0(i,j) \quad i=1,\ldots,n_1, j=1,\ldots,n_2/\]

JOAN must be present in the arguments of AIDE, when AIDE is called in KPRIMC at its usual place.

If any new array X is needed for AIDE, a statement of the type

\[
\text{DIMENSION X}(1)
\]

must be inserted in AIDE and KPRIMC

\[\text{Computation of } K_1^2\]

K1COMP is the subroutine used to solve eqns. 2.15 and 2.16 for $K_1^2$ and $K_2^2$, respectively.

\[\text{Description and use of K1COMP}\]

\[
\text{K1COMP}(A1, RI1, S1, S2, C2, PRIMK, AMAT, SECM, N1, N2, VECTINC, AROND)
\]

A1, RI1, S1, S2, C2, PRIMK and N1 have the same meaning as in KPRIMC.

\[
\text{AMAT} : \text{n1xnn1 work matrix}
\]

\[
\text{SECM, VECTINC} : \text{n1 - work vectors}
\]

\[
\text{AROND} : n_1 \times n_1 \text{ matrix } A_1 - S_1 K_1
\]
The output is the vector VECTINC giving the upper triangular part of $K_1^2$ by the following index correspondence:

$$\text{VECTINC}[j+(i-1)n_1] = K_1^2(i,j)$$

$A_1$, $R_{i1}$, $S_1$, $S_2$, $C_2$, and $PRIMK$ are read before calling $K_1COMP$. $N_1$ and $N_2$, with the obvious definition, are assigned values in the arguments of $K_1COMP$ being called. Moreover, $CAROND$, $CALSECM$, $MATCOEF$ and $SOLRES$ must be present in the program.

For the computation of $K_2^2$, just permute 1 and 2 indices and instead of $PRIMK$ introduce its transpose $PRIMKT$. Change the dimensions of $AMAT$, $SECM$, $VECTINC$ and $AROND$ accordingly. For example, permute $n_1$ and $n_2$.

A flow chart shows to use $KPRIMC$ and $K1COMP$ to compute $M$. (fig. 13.)
5. CONCLUSION

Two embedding methods have been presented for the solution of the Riccati equation. They require less computer time than the conventional methods used so far.

The method of section 2.3 introduces a new $\varepsilon$ variable, in terms of which $K$ is expanded in MacLaurin series around $\varepsilon_0$. All derivatives $\frac{d^mK}{d\varepsilon^m}$ at $\varepsilon_0$ are obtained from decoupled equations. The performance index associated to such an expansion of the $n$th order in $\varepsilon$ is accurate up to the $(2n+1)\text{th}$ order. This property along with the reduction of dimensionality strongly pleads in favor of this method.

For $T = \infty$ in eqn. 1.4, the terms $\frac{d^{2i}K}{d\varepsilon^{2i}}$ and $\frac{d^{2i+1}K}{d\varepsilon^{2i+1}}$ are the solutions of linear systems of algebraic equations which differ only by the second members for successive $i$'s.

On the other hand, the embedding method in section 2.2, using an $\varepsilon$-integration with fixed limits of $\varepsilon$, has a speed independent on the system analyzed, since there is no convergence difficulty involved. Moreover, the method of section 2.2 can be used to compute the $K$ matrix for a given range of a slowly varying parameter as in example 3. Another application is the selection of $Q$ and $R$ to ensure a good response for the system [6]. One can construct $Q$ as a function of $\varepsilon$ and use eqn. 2.1 where $\alpha$ and $\zeta$ are zero. A satisfactory $K$ can then be selected from the family $K(\varepsilon)$. 
LIST OF REFERENCES


APPENDIX A.
(figures 1 to 13)
Figure 1. Linear system with optimal control.
Figure 2. Kalman Bucy filter
Figure 3. Scheme for the R-C cable.
Figure 4. R-C model of the transmission cable
Figure 5. Performances of different methods for example 1.
Figure 6. Effect of coupling on K matrix for example 1.
Figure 7. Performances of different methods for example 2.
Figure 8. Constant speeding winding process.
Figure 9. Flow chart of UJERIC.
Figure 10. Flow chart of SECM.
Figure 11. Interaction of the subroutines called by INTEGRE.
Figure 12. Flow chart for KPRIMC and KICOMP.
Figure 13. Flow chart for the use of KPRIMC and KICOMP.
APPENDIX B

Computation of $\frac{1}{2} \int_{t_0}^{T} (x'Qx + u'Ru) dt$

Assume $u = -R^{-1}B'Fx$ for the system $\dot{x} = Ax + Bu$. $F$ is not necessarily the Riccati matrix as defined by eqn. 1.4. The normal equation of the system with the feedback control is

$$\dot{x} = (A-SF)x \quad x(t_0) = x_0$$  \hspace{1cm} \text{B.1}

Call

$$\frac{1}{2} \int_{t_0}^{T} (u'Ru + x'Qx) dt = \frac{1}{2} x'(t)P(t)x(t)$$

By differentiating eqn. B.1 with respect to $t$, one obtains

$$\dot{P} = -(A-SF)' - P(A-SF) - FSF - Q$$  \hspace{1cm} \text{B.2}

where $P(T) = 0$. The performance index is

$$J = \frac{1}{2} x'_0 P(t_0)x_0$$  \hspace{1cm} \text{B.3}

For the time-invariant case, eqn. B.3 holds where $P(t_0)$ satisfies

$$(A-SF)'P + P(A-SF) = -FSF - Q$$  \hspace{1cm} \text{B.4}

provided the system B.1 is stable.
APPENDIX C

Proof of theorem 1.

Differentiate eqn. 2.2 with respect to \( \epsilon \) \( j \) times and let \( \epsilon = 0 \)

\[
\dot{K}_j = -K^j - A'K - j(K^{j-1}A + A'K^{j-1})
\]

\[
+ \frac{\partial^j KSK}{\partial \epsilon^j} \bigg|_{\epsilon=0} - Q^j \text{ K(T,0) = 0 } \quad \text{C.1}
\]

Denote by 'd' any square matrix of the form

\[
\begin{pmatrix}
    d_1 & 0 \\
    0 & d_2
\end{pmatrix}
\]

where \( d_1 \) and \( d_2 \) are submatrices and denote by 'a' any square matrix of the form

\[
\begin{pmatrix}
    0 & a_1 \\
    a_2 & 0
\end{pmatrix}
\]

Notice that \( ad = a, \ da = a, \ ad = d \)

and \( dd = d \). Note also that, at \( \epsilon = 0 \), \( A = d, \ A^1 = a, \ A^2 = 0, \ S = d, \)
\( S^1 = a, S^2 = d, Q = d, Q^1 = a \) and \( Q^2 = 0 \). From eqns. 2.9 and 2.10,
\( K = d \). Then, for \( j = 1 \), all terms in equations C.1 which do not depend on
\( K^j \) are of the type 'a'. Hence, the equations for \( K_1^1 \) and \( K_2^1 \) are homogeneous
so that \( K_1^1 \) and \( K_2^1 \) must be zero for all \( t \), since they are zero at time \( t = T \).
\( K^1 \) is thus of the type 'a'.

The same check can be repeated for \( j = 2 \), and it is easily
verified that the term independent on \( K^j \) is 'd' so that \( K_{12}^j = 0 \) for the
same reason. Then assuming \( K^{2i-1} \) is 'a', the independent term of eqn. C.1 for \( 2i \) differentiations is 'd', hence \( K_{12}^{2i} = 0 \). Similarly for one more differentiation, except that the independent term is 'a', hence \( K_{1}^{2i+1} \) and \( K_{2}^{2i+1} = 0 \) which completes the proof.

Expressions for the independent terms are:

\[
\begin{align*}
F_{12}^0 &= A_{21}^1 K_2 + K_1 A_{12}^1 - K_1 S_{12}^1 K_2 + Q_{12} \\
F_{11}^1 &= 2G_{21}^1 K_{21} + 2K_{12}^1 G_{21}^1 - K_1 S_{12}^1 K_2 + 2K_{12}^1 S_{21}^1 K_2 \\
F_{22}^1 &= 2G_{12}^1 K_{12} + 2K_{21}^1 G_{12}^1 - K_2 S_{21}^1 K_2 + 2K_{21}^1 S_{12}^1 K_1 \\
F_{12}^2 &= 3G_{21}^1 K_2^2 + 3K_{12}^1 G_{12}^1 - 3K_{12}^1 S_{21}^1 K_2 - 6K_{12}^1 S_{12}^1 K_{12} - 3K_{12}^1 S_{12}^1 K_{12} \\
G_{12}^1 &= A_{12}^1 - S_{12}^1 K_2 - S_1 K_{12}^1 \\
G_{21}^1 &= A_{21}^1 - S_{21}^1 K_1 - S_2 K_{21}^1 \\
S_{1}^1 &= 2B_{12}^1 R_{21}^1 B_{12} \\
S_{2}^1 &= 2B_{21}^1 R_{12}^1 B_{21} \\
K_{12}^1 &= K_{21}^1 \\
S_{21}^1 &= S_{21}^1 \\
S_{12}^1 &= B_{12}^1 R_{21}^1 B_{2} + B_{1} R_{1}^1 B_{21} \\
S_{12}^1 &= A_{12}^m B_{2} + B_{1} R_{1}^1 B_{21} \\
\end{align*}
\]

The notation \( A_{m}^m \) must be understood as the matrix obtained by:
1. differentiating $A^m$ times with respect to $\varepsilon$

2. setting $\varepsilon$ to zero

3. extracting the submatrices from $A^m$ partitioned as

$$A^m = \begin{pmatrix} A_{11}^m & A_{12}^m \\ A_{21}^m & A_{22}^m \end{pmatrix}$$

4. transposing $A^m$ as expressed in eqn. C.2, if needed. For example,

$$S'_{12} = B_{21}^1 R_{11}^{-1} B_{1}^1' + B_{22}^1 R_{22}^{-1} B_{12}^1$$
APPENDIX D

Proof of theorem 2

The optimum performance $J_K$ is

$$J_K = \frac{1}{2} \langle x, Kx \rangle \bigg|_{t=t_0}$$

and the performance $J_M$ of the system in eqn. 2.20 is

$$J_M = \frac{1}{2} \langle x, P_x \rangle \bigg|_{t=t_0}$$

as shown in appendix B. $P = P(t, \epsilon)$ is the solution of the linear equation

$$\dot{P} = -P(A-SM) - (A'-MS)P - MSM - Q \quad \text{(D.1)}$$

with $P(T, \epsilon) = 0$.

To verify eqn. 2.21, it is sufficient to prove that, at $\epsilon = 0$,

the matrix

$$\Gamma = P - K \quad \text{(D.2)}$$

and its first $2m+1$ derivatives

$$\Gamma^i = \left. \frac{\partial^i \Gamma(t, \epsilon)}{\partial \epsilon} \right|_{\epsilon=0} \quad i = 1, 2, \ldots, 2m+1$$

are zero, for all $t \epsilon [t_0, T]$.

From eqns. 1.4, 2.5, D.1 and D.2

$$\Gamma = -\Gamma G - C'\Gamma - \epsilon^{m+1} (\Gamma S \Lambda + \Lambda S \Gamma)$$

$$- \epsilon^{m+2} \Lambda S \Lambda \quad \Gamma(t, \epsilon) = 0$$

$$\text{D.3}$$
where \( G = A - SK \) and 
\[
\Lambda = \sum_{i=m+1}^{\infty} \frac{\epsilon^{i-m-1}}{i!} K^i.
\]

Clearly, for \( \epsilon = 0 \), eqn. D.3 yields \( \Gamma^0(t,0) = 0 \).

Proceeding by induction, assume that \( \Gamma^j(t,0) = 0 \) for 
\( j = 0, 1, \ldots, i \). Differentiating eqn. D.3 \( i+1 \) times, it is seen that the 

differential equation for \( \Gamma^{i+1} \) is homogeneous, provided \( i \leq 2m+1 \). 

Since \( \Gamma^{i+1}(T,0) = 0 \), \( \Gamma^{i+1}(t,0) = 0 \). Hence \( \Gamma^i(t,0) = 0 \) for 
\( i = 0, 1, \ldots, 2m+1 \), which proves eqn. 2.21.
ERRATUM
Mr Jerome Fox, Research Coordinator
Polytechnic Institute of Brooklyn
55 Johnson St (Should be 333 Jay St)
Brooklyn, N.Y. 11201

OMIT
Mr Morton M. Pavane, Chief
AFSC Scientific & Tech. Liaison Office
26 Federal Plaza, Suite 1313
New York, New York 10007

ADDENDUM
Dept of Electrical Engineering
Rice University
Houston, Texas 77001

School of Engineering & Applied Science
University of Virginia
Charlottesville, Virginia 22903

Dept of Electrical Engineering
College of Engineering & Technology
Ohio University
Athens, Ohio 45701

Project Mac
Document Room
Massachusetts Institute of Technology
545 Technology Square
Cambridge, Massachusetts 02139

Lehigh University
Dept of Electrical Engineering
Bethlehem, Pennsylvania 18015

Commander Test Command (TCDT-)
Defense Atomic Support Agency
Sandia Base
Albuquerque, New Mexico 87115

Materials Center Reading Room 13-2137
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Professor James A. Cadzow
Department of Electrical Engineering
State University of New York at Buffalo
Buffalo, New York 14214
The Riccati equation needs to be solved in the control theory and the estimation theory. Most frequently, this need arises in the state regulator and the Kalman-Bucy filter design.

No analytical solution of the Riccati equation has been found except in the one-dimensional case. For higher orders, one must use computer techniques to obtain a numerical solution. The conventional approaches fail to be satisfactory for many applications of reasonably high dimensions. The present work gives a way to avoid that kind of difficulty.
<table>
<thead>
<tr>
<th>KEY WORDS</th>
<th>LINK A</th>
<th>LINK B</th>
<th>LINK C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Riccati equation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e- coupling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e- embedding</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance index</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>algorithm</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Approximation of Riccati-matrix</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>