

10-66-212
106771
P. 26

STOCHASTIC MODEL OF THE NASA/MSFC GROUND FACILITY
FOR LARGE SPACE STRUCTURES WITH UNCERTAIN PARAMETERS
- THE MAXIMUM ENTROPY APPROACH

REPORT

by

Dr. Wei-Shen Hsia
Department of Mathematics
The University of Alabama
Tuscaloosa, Alabama 35487
(205) 348-5153

NASA Grant Number : NAG8-081
Grant Period : 10-20-86 to 10-19-87

(NASA-CR-181489) STOCHASTIC MODEL OF THE N88-12343
NASA/MSFC GROUND FACILITY FOR LARGE SPACE
STRUCTURES WITH UNCERTAIN PARAMETERS: THE
MAXIMUM ENTROPY APPROACH (Alabama Univ.) Unclas
26 p Avail: NTIS HC A03/MF A01 CSCL 12B G3/66 0106771

TABLE OF CONTENTS

	pages
1. Introduction	1
2. Maximum Entropy Modelling	3
3. Computation Algorithm	7
4. Computation Results of ME Design	9
5. Conclusion	13
6. References	15
7. Appendix I *** Computer Program for ME Design	16

1. INTRODUCTION

The National Aeronautics and Space Administration and the Department of Defense are actively involved in the development of a validated technology data base in the areas of control/structures inter-action, deployment dynamics and system performance for Large Space Structures (LSS). In the Control System Division of the System Dynamics Laboratory of the NASA/MSFC, a Ground Facility (GF), in which the dynamics and control system concepts being considered for LSS applications can be verified, has been designed and built under Dr. Henry Waites' supervision [8]. The viability and versatility of this MSFC LSS ground test facility was recognized by the U. S. Air Force Wright Aeronautical Laboratory as a site for their Vibration Control of Space Structures (VCOSS) testing.

One of the important aspects of the GF is to verify the analytical model for the control system design. The procedure is to describe the control system mathematically as well as possible, then to perform tests on the control system, and finally to factor those results into the mathematical model.

However, development of a "correct" mathematical model of a system is still an art. In constructing large order

structural models, various errors, such as modelling errors, parameter errors, improperly modeled uncertainties, and errors due to linearization of non-linear effect, create a great challenging task of determining "best" models for a dynamic system. It is recognized that it is conceivable that better performance will be anticipated when uncertainties are modeled through stochastic multiplicative and additive noise terms. Optimal control strategies generated under all possible parameter variations will definitely create more robust control systems, under controllability and observability conditions, than those generated by the usual approaches [2]. To avoid ad hoc assumptions regarding "a priori" statistics, Hyland [2,3,4] used the maximum entropy principle to determine a priori probability assignment induced from available data. A main advantage of maximum entropy approach is that it sacrifices as little near-nominal performance as possible while securing performance insensitivity over the likely range of modelling errors.

In this report, we design a stochastic control model of the NASA/MSFC Ground Facility for LSS control verification through the maximum entropy principle adopted in Hyland's method [2,3,4]. Using ORACLS, a computer program is implemented for this purpose. Four models are then tested. Results are presented in this report.

2. MAXIMUM ENTROPY MODELLING

Consider a linear system :

$$\begin{aligned}\dot{X} &= AX + BU + \omega_1 \\ Y &= CX + \omega_2\end{aligned}\tag{1}$$

where

$$X \in R^n, U \in R^m, Y \in R^l, A \in R^{n \times n}, B \in R^{n \times m}, C \in R^{l \times m},$$

and

$$SD(\omega_1, \omega_2) = (v_1, v_2).$$

We seek to determine a dynamic compensator

$$\begin{aligned}\dot{Z} &= A_c Z + FY \\ U &= -KZ\end{aligned}\tag{2}$$

where $Z \in R^n$, $A_c \in R^{n \times n}$, $F \in R^{n \times l}$ and $K \in R^{m \times n}$ that minimizes the Quadratic Cost Function :

$$J = \int_0^{\infty} (X^T R_1 X + U^T R_2 U) dt\tag{3}$$

where R_1 and R_2 are penalty matrices. The maximum entropy (ME) design approach [1,2,3,4,5] is used to minimize J in the presence of parameter uncertainties.

In most instances, the actual system dynamics differ from the nominal model by an error distribution matrix. The basic premise of ME error modelling is that the magnitude of the error is a white-noise process $\alpha(t)$. Assuming there are p uncorrelated error sources, the system dynamic matrices

become :

$$A_{\text{actual}} = A + \sum_{i=1}^p \alpha_i(t) A_i \quad (4)$$

with the B_{actual} and C_{actual} matrices taking similar forms.

For the simplicity and in order to get a good inside look at the ME design technique, we assume there is only one error distribution matrix A_1 in the system. Under these assumption, the necessary conditions for optimality of the Quadratic Cost Function can be derived after the system dynamics are presented by means of stochastic differential equations. The resulting equations take the form of two Riccati equations and two Lyapunov equations, all coupled by the stochastic parameters [6]. That is, we need to solve four nonnegative-definite P , Q , \hat{P} and \hat{Q} such that

$$\begin{aligned} P A_S + A_S^T P + A_1^T P A_1 - P_S^T R_2^{-1} P_S + R_1 + A_1^T \hat{P} A_1 &= 0 \\ A_S Q + Q A_S^T + A_1 Q A_1^T - Q_S V_2^{-1} Q_S^T + V_1 + A_1 \hat{Q} A_1^T &= 0 \\ \hat{P} A_{Q_S} + A_{Q_S}^T \hat{P} + P_S^T R_2^{-1} P_S &= 0 \\ A_{P_S} \hat{Q} + \hat{Q} A_{P_S}^T + Q_S V_2^{-1} Q_S^T &= 0 \end{aligned} \quad (5)$$

where

$$\begin{aligned} A_S &\triangleq A + \frac{1}{2} A_1^2, \quad P_S \triangleq B^T P, \quad Q_S \triangleq Q C^T, \\ A_{Q_S} &\triangleq A_S - Q_S V_S^{-1} C, \quad A_{P_S} \triangleq A_S - B R_2^{-1} B^T P. \end{aligned}$$

The compensator matrices then take on the following forms,

$$\begin{aligned}
A_C &= A_S - Q_S V_2^{-1} C - B R_2^{-1} P_S \\
F &= Q_S V_2^{-1} \\
K &= R_2^{-1} P_S.
\end{aligned} \tag{6}$$

Unfortunately, the covariance matrices V_1 and V_2 of the Wiener processes ω_1 and ω_2 , respectively, in (1) are usually not known. However, we developed a method of estimating those two import matrices as follows.

Consider the system

$$\dot{X} = AX + BU + \omega_1. \tag{7}$$

(7) can be rewritten as

$$dx^i = \left(\sum_j A_j^i x^j + \sum_k B_k^i U^k \right) dt + d\omega_1^i, \quad i = 1, \dots, n.$$

Let $r_1^{ij} = E[X^i X^j]$ and $r_1^{ij} = E[X^i X^j]$. By Ito's rule, we have

$$\begin{aligned}
d(X^i X^j) &= (dx^i) X^j + X^i (dx^j) + (dx^i) (dx^j) \\
&= \left(\sum_k A_k^i X^k X^j + \sum_l B_l^i U^l X^j \right) dt \\
&\quad + \left(\sum_k A_k^j X^k X^i + \sum_l B_l^j U^l X^i \right) dt \\
&\quad + X^j d\omega_1^i + X^i d\omega_1^j + (d\omega_1^i) (d\omega_1^j).
\end{aligned} \tag{8}$$

With the assumption that X and ω_1 are uncorrelated, (8) becomes

$$\dot{r}^{ij} = \sum_k A_k^i r^{kj} + \sum_k A_k^j r^{ki} + \sum_l B_l^i q^{lj} + \sum_l B_l^j q^{lj} + V_1^{ij}$$

or

$$V_1^{ij} = \sum_k A_k^i r^{kj} + \sum_k A_k^j r^{ki} + \sum_l B_l^i q^{lj} + \sum_l B_l^j q^{lj} - \dot{r}^{ij}, \quad (9)$$

where $V_1^{ij} dt = E[d\omega_1^i d\omega_1^j]$ and $q^{ij} = E[U^i X^j]$.

If we assume, in addition, that X and U are uncorrelated, then we can drop the terms involving q^{ij} in (9). And (9) becomes

$$V_1^{ij} = \sum_k A_k^i r^{kj} + \sum_k A_k^j r^{ki} - \dot{r}^{ij} \quad (10)$$

for $i, j = 1, 2, \dots, n$.

Therefore, if r and \dot{r} can be estimated, then the covariance matrix V_1 can be estimated through (10).

Estimation of V_2 for ω_2 in the equation $Y = CX + \omega_2$ is a much easier job. We simply use the standard statistics technique to estimate V_2 by $E[(Y - CX)(Y - CX)^T]$.

3. COMPUTATION ALGORITHM

In this report, we treat all four equations in (5) together as a single Riccati equation. This approach is different from the one proposed by Gruzen [6] in which each iteration involves solving the first two equations of (5) as Riccati equations and then solving the last two equations of (5) as Lyapunov equations.

We can rewrite (5) as following:

$$\begin{aligned}
 PA_S + A_S^T P - P^T B R_2^{-1} B^T P + A_1^T (P + (A_1^{-1})^T R_1 A_1^{-1} + \hat{P}) A_1 &= 0 \\
 QA_S^T + A_S Q - QC^T V_2^{-1} C Q + A_1 (Q + A_1^{-1} V_1 (A_1^{-1})^T + \hat{Q}) A_1^T &= 0 \\
 \hat{P} A_{QS} + A_{QS}^T \hat{P} - \hat{P}^T \theta R_2^{-1} \theta^T \hat{P} + P_S^T R_2^{-1} P_S &= 0 \\
 \hat{Q} A_{PS}^T + A_{PS} \hat{Q} - \hat{Q} \theta V_2^{-1} \theta^T \hat{Q} + Q_S V_1^{-1} Q_S^T &= 0
 \end{aligned} \tag{11}$$

where matrix θ indicates zero matrix with appropriate dimension. In a more concise form, we have:

$$P^* A^* + (A^*)^T P^* - (P^*)^T B^* (R^*)^{-1} (B^*)^T P^* + (H^*)^T Q^* H^* = 0 \tag{12}$$

where

$$\begin{aligned}
 P^* &= \begin{bmatrix} P^T & & & 0 \\ & Q & & \\ & & \hat{P} & \\ 0 & & & \hat{Q} \end{bmatrix}, & R^* &= \begin{bmatrix} R_2 & & & 0 \\ & V_2 & & \\ & & R_2 & \\ 0 & & & V_2 \end{bmatrix}, \\
 A^* &= \begin{bmatrix} A_S & & & 0 \\ & A_S^T & & \\ & & A_S & \\ 0 & & & A_{PS}^T \end{bmatrix}, & B^* &= \begin{bmatrix} B & & 0 \\ & C^T & \\ 0 & & \theta \end{bmatrix},
 \end{aligned}$$

$$H^* = \begin{bmatrix} A_1 & & & 0 \\ & A_1^T & & \\ & & P_S & \\ 0 & & & Q_S^T \end{bmatrix}$$

and

$$Q^* = \begin{bmatrix} P + (A_1^{-1})^T R_1 A_1^{-1} + \hat{P} & & & 0 \\ & Q + A_1^{-1} V_1 (A_1^{-1})^T + \hat{Q} & & \\ & & R_2^{-1} & \\ 0 & & & V_2^{-1} \end{bmatrix} .$$

Note that (12) does not exactly match the standard algebraic Riccati equation form:

$$PA + A^T P - PBR_2^{-1} B^T P + R_1 = 0. \quad (13)$$

Because there are unknown parameters in the last term $H^{*T} Q^* H^*$ in (12). This character affects the iteration scheme significantly. The constant term of the Riccati equation (12) includes P^* matrix. Consequently, the equation must be iterated through several times, updating P^* solution in the constant term each time until it converges to a solution. The iteration strategy is illustrated in Figure 1.

The convergence criterion used in our program is when

$$\| P^{*(n)} - P^{*(n-1)} \| < \epsilon, \text{ where } \epsilon \text{ is a preset tolerance.}$$

The software package ORACLS [7] provides a control system design and analysis environment. This package provides subroutines such as basic matrix manipulations (addition, subtraction, multiplication, transpose, etc.) and Riccati solver. The design algorithm is implemented in FORTRAN (see Appendix).

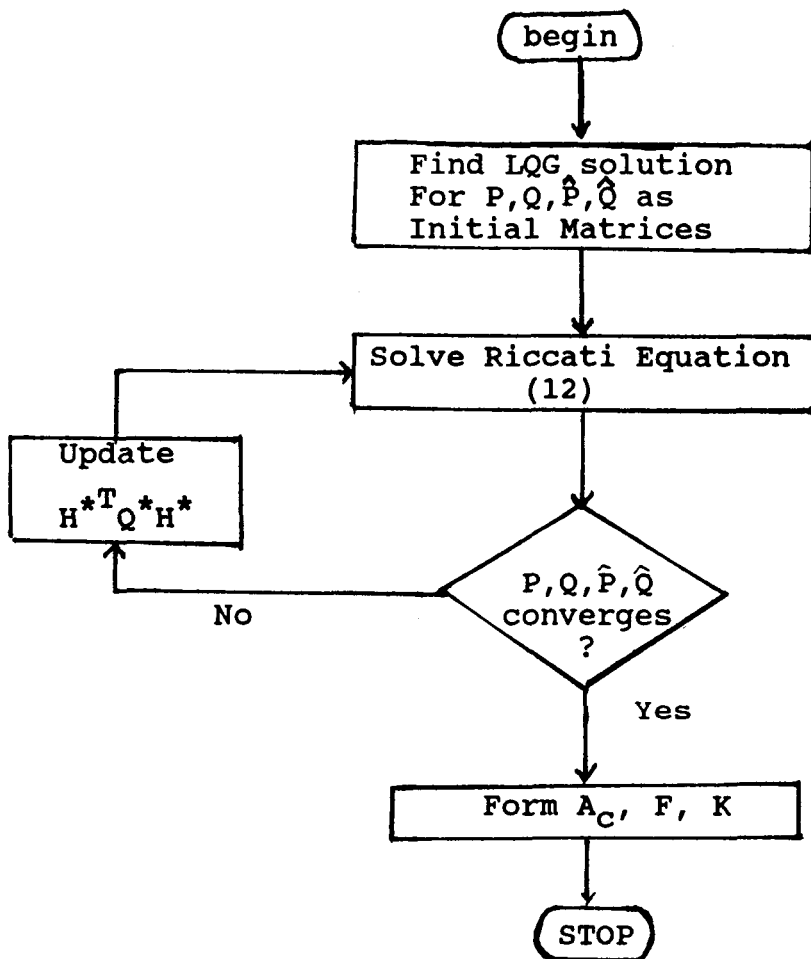


Figure 1

4. COMPUTATION RESULTS OF ME DESIGN

In this section, we applied the ME design algorithm to the MSFC Ground Test Facility in which dynamics and control system concepts being considered for LSS applications can be verified [8].

There are 50 modes in the system model. For the purpose of testing the algorithm and the FORTRAN program, we only consider stochastic models with only one mode. Mode 8 is chosen for this purpose. We also assume there is only one error distribution matrix of A in the system.

Therefore, the stochastic model concerned in this section is

$$\begin{aligned}\dot{X}(t) &= (A + \alpha_1(t)A_1)X(t) + BU(t) + \omega_1(t) \\ Y(t) &= CX(t) + \omega_2(t)\end{aligned}\tag{14}$$

Data collected from an analytical model in four different settings have been provided by Dr. Henry Waites. Using those data, we designed four settings and through which we can determine corresponding compensator matrices. In all of those four settings, we choose the modal damping $\xi_8 = 0.5\%$,

$$R_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \text{and} \quad R_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The dynamic matrices of those four models are:

Model 1 (EXP5VL Sept. 24. 1986): $\omega_8^2 = 17.44$

$$A = \begin{bmatrix} 0 & 1 \\ -17.44 & -0.04176 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -0.00154 & 0 & 0 & 0.01 & 0.000176 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 \\ 0 & 0.1 \\ 0 & 0.000176 \end{bmatrix}.$$

Model 2 (EXP6UL Oct. 1. 1986): $\omega_8^2 = 17.44$

$$A = \begin{bmatrix} 0 & 1 \\ -17.44 & -0.04176 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0.0016 & 0 & 0 & -0.01036 & -0.0004875 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 \\ 0 & -0.01036 \\ 0 & -0.0004875 \end{bmatrix}.$$

Model 3 (EMVL Dec. 29. 1986): $\omega_8^2 = 19.41$

$$A = \begin{bmatrix} 0 & 1 \\ -19.41 & -0.04176 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0.0001 & 0 & 0.002 & -0.0082 & 0 \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & 0.002 \\ 0 & -0.082 \\ 0 & 0 \end{bmatrix}.$$

Model 4 (EMFVLL Jan. 20. 1987): $\omega_8^2 = 14.4$

$$A = \begin{bmatrix} 0 & 1 \\ -14.4 & -0.03796 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0.00031890 & 0 & 0 & 0.001617 & 0 \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & 0 \\ 0 & 0.0016171 \\ 0 & 0 \end{bmatrix}.$$

As pointed out by Gruzen [6], we can scale the position coordinate by the modal frequency ω_8 , the first equation of (14) is transformed into an equivalent representation:

$$\frac{d\tilde{X}}{dt} = \begin{bmatrix} 0 & \omega_8 \\ -\omega_8 & -2\xi\omega_8 \end{bmatrix} \tilde{X} + BU, \quad (16)$$

where $\tilde{X} = \begin{bmatrix} x\omega_8 \\ \dot{x} \end{bmatrix}$ with the transformation matrix $T = \begin{bmatrix} \omega_8 & 0 \\ 0 & 1 \end{bmatrix}$.

Therefore, we can assume the uncertainty distribution matrix for the last four models takes the following form:

$$A_1 = \begin{bmatrix} 0 & 1 \\ -1 & -0.01 \end{bmatrix}.$$

The compensation matrices resulted from the algorithm are summerized as following.

Model	1	2
P	3.91 D+04 -2.50 D-03 -2.50 D-03 3.91 D+04	3.63 D+04 -2.50 D-03 -2.50 D-03 3.63 D+04
Q	4.01 D+02 -2.00 D+00 -2.00 D+00 4.01 D+02	3.72 D+04 -1.85 D+02 -1.86 D+02 3.72 D+04
P	1.47 D-06 4.88 D-09 4.83 D-09 1.95 D-11	1.58 D-06 5.26 D-09 5.26 D-09 2.10 D-11
Q	0 0 0 0	0 0 0 0
A _C	-5.0 D-01 2.39 D-03 1.07 D-06 -6.51 D+00	-5.0 D-01 2.39 D-03 1.15 D-06 -6.50 D+00
F	-6.02 D+01 0 6.88 D+00 0 3.91 D+02 7.05 D-02	5.81D+01 0 -1.77D+01 0 -3.76D+02 -1.81D+01
K	3.85 D-06 -6.02 D+01 0 0 0 0 -2.50 D-05 3.91 D+02 -4.40 D-07 6.88 D+00	-4.0 D-06 5.81 D+01 0 0 0 0 2.59 D-05 -3.76 D+02 1.22 D-06 -1.77 D+01

Model	3				4			
P	5.61 D+04	-2.50 D-03			1.47 D+06	-2.50 D-03		
	-2.50 D-03	5.61 D+04			-2.50 D-03	1.47 D+06		
Q	5.61 D+04	-2.81 D+02			1.53 D+06	-7.65 D+03		
	-2.81 D+02	5.61 D+04			-7.65 D+03	1.53 D+06		
\tilde{P}	5.61 D+04	-2.50 D-03			1.47 D+06	-2.50 D-03		
	-2.50 D-03	5.61 D+04			-2.50 D-03	1.47 D+06		
\tilde{Q}	0	0			0	0		
	0	0			0	0		
A_C	-5.0 D-01	2.27 D-03			-5.0 D-01	2.64 D-03		
	7.85 D-07	-6.50 D+00			2.58 D-08	1.47 D+00		
F	5.61 D+00	1.12 D+02	0		4.70 D+02	0	0	
	0	-4.60 D+02	0		0	2.38 D+03	0	
K	-2.50 D-07	5.61 D+00			-7.97 D-07	4.70 D+02		
	0	0			0	0		
	-5.00 D-06	1.12 D+02			0	0		
	2.05 D-05	-4.60 D+02			-4.04 D-06	2.38 D+03		
	0	0			0	0		

Figure 2

5. CONCLUSION

In general, the major issues relevant to the control of flexible space structures are "robustness" with respect to both parameter modelling errors and truncation of higher order modes. Several methods have been developed recently to deal with those problems. Among them, the maximum entropy and optimal projection (MEOP) method developed by Hyland and Bernstein specifically for the flexible structure control problems seems very promising.

In this report, we examined the ME portion of the design method. Using ORACLS, we implemented a computer program for ME method. Four small scaled models are then tested and the resulted compensation matrices are given.

The extension of this project, naturally, would be to test the OP portion of the design method and then combine those two programs to have a complete MEOP design tool.

REFERENCES

1. D.C. Hyland, "Opimal Regulation of Structural Systems with Uncertain Parameters," MIT, Lincoln Laboratory, TR-551, Feb. 1981, DDC# AD-A099111/7.
2. _____, "Structural Modelling and Control Design Under Incomplete Parameter Information: The Maximum Entropy Approach," Modelling, Analysis, and Optimization Issues for Large Space Structures, NASA CP-2258, 1983, pp. 73-96.
3. _____, "Maximum Entropy Stochastic Approach to Control Design for Uncertain Structural Systems," American Control Conference, Arlington, VA. June, 1982.
4. _____, "Minimum Information Modelling of Structural Systems with Uncertain Parameters," Proceedings of the Workshop on Applications of Distributed System Theory to the Control of Large Space Structures, JPL, Pasadena, CA. July, 1982.
5. _____, "Application of the Maximum Entropy/Optimal Projection Control Design Approach for Large Space Structures," Large Space Antenna Systems Technology Conference, NASA Langley, December, 1984.
6. A. Gruzen, "Robust Reduced-order Control of Flexible Structures," The Charles Stark Draper Laboratory, Inc. Cambridge, Mass., March, 1986, CSDL-T-900.
7. E.S. Armstong, "A Design System for Linear Multivariate Control," Marcell Dekker, Inc., New York, 1980.
8. H.B. Waites, S.M. Seltzer, and D.K. Tollison, "NASA/MSFC Ground Experiment for Large Space Structure Control Verification," NASA TM-86496, December, 1984.

APPENDIX 1

COMPUTER PROGRAM FOR ME DESIGN

```

C*****
C*  DRIVER FOR THE MAXIMUM ENTROPY DESIGNER
C*  INPUTTAPE = 5      OUTPUTTAPE = 6
C*****
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION P1(64),A1(64),B1(64),R1(64),Q1(64),H1(64),F1(64),
1         DUMMY(2100),P(4),Q(4),PHAT(4),QHAT(4),E1(4)
      DIMENSION NP1(2),NA1(2),NB1(2),NR1(2),NQ1(2),NH1(2),NF1(2),
1         IOP(3),NE1(2)
      DIMENSION P2(64),A2(64),B2(64),R2(64),Q2(64),H2(64),F2(64),
1         D21(64),D22(64),D23(64),D24(64),D25(64),D26(64)
      DIMENSION NP2(2),NA2(2),NB2(2),NR2(2),NQ2(2),NH2(2),NF2(2),
1         ND21(2),ND22(2),ND23(2),ND24(2),ND25(2),ND26(2)
      DIMENSION A5(4),A51(4),Q5(4),P5(4),Q55(4),P55(4),C5(6),B5(10),
1         V51(4),V52(9),R51(4),R52(25),V152(9),R152(25),B3(256),
1         R3(256),TH3(96),TH4(64),TH5(96),TH6(96),TH7(192),
1         TH8(64),H3(256),TA51(4),TA5(4),A31(4),A32(4),TA6(6),
1         TC5(6),A33(4),TB5(10),TA7(10),A34(4),Z1(12),Z2(8),
1         Z5(128),X1(16),X21(8),X2(16),X31(12),X3(16),X5(32),
1         X6(48),X7(64),X8(128),Q31(4),Q32(4),Z6(20),Z7(20),
1         Z8(15),Z9(27),X9(24),X10(24),X11(60),X12(36),X13(45),
1         X14(48),X15(108),X16(144),X17(48),Q3(256),F3(256),
1         P3(256),X4(16),X18(192),A3(256),TP4(256),TP5(256),
1         PP(4),PPH(4),QQ(4),QQH(4),R15(64),TM1(4),F(6),TM2(4),
1         TM3(10),TM4(10),K(10),E2(4),ACC(4),TRN(4),TRNI(4),AC(4)
      DIMENSION NA5(2),NA51(2),NQ5(2),NP5(2),NQ55(2),NP55(2),NC5(2),
1         NB5(2),NV51(2),NV52(2),NR51(2),NR52(2),NVI52(2),
1         NRI52(2),NB3(2),NR3(2),NTH3(2),NTH4(2),NTH5(2),NTH6(2),
1         NTH7(2),NTH8(2),NH3(2),NTA51(2),NTA5(2),NA31(2),NA32(2),
1         NTA6(2),NTC5(2),NA33(2),NTB5(2),NTA7(2),NA34(2),NZ1(2),
1         NZ2(2),NZ5(2),NX1(2),NX2(2),NX21(2),NX31(2),NX3(2),
1         NX5(2),NX6(2),NX7(2),NX8(2),NQ31(2),NQ32(2),NZ6(2),
1         NZ7(2),NZ8(2),NZ9(2),NX9(2),NX10(2),NX11(2),NX12(2),
1         NX13(2),NX14(2),NX15(2),NX16(2),NX17(2),NX18(2),
1         NQ3(2),NF3(2),NP3(2),NX4(2),NA3(2),NTP4(2),NTP5(2),
1         NP(2),NQ(2),NPHAT(2),NQHAT(2),NPP(2),NPPH(2),NQQ(2),
1         NQQH(2),NR15(2),NTM1(2),NF(2),NTM2(2),NTM3(2),NTM4(2),
1         NK(2),NE2(2),NACC(2),NTRN(2),NTRNI(2),NAC(2)
      INTEGER  IERR, ITE
      REAL SCLE,EPSI
      LOGICAL IDENT,DISC,FNULL
C
C  INPUT HOLLERITH DATA FOR TITLE OF OUTPUT
      CALL RDTITL
C
C  TO OBTAIN INITIAL P AND Q BY SOLVING THE ASSOCIATED RCT EQA
C  FROM LQG METHOD
C
C  INPUT COEFFICIENT MATRICES FOR THE INITIAL LQG SYSTEM
C
      CALL READ(4,A5,NA5,A51,NA51,C5,NC5,B5,NB5)
      CALL READ(4,V51,NV51,V52,NV52,R51,NR51,R52,NR52)
      CALL READ(4,A1,NA1,B1,NB1,R1,NR1,Q1,NQ1)
      CALL READ(2,D22,ND22,R15,NR15)
      CALL READ(4,B3,NB3,R3,NR3,TH3,NTH3,TRN,NTRN)
      DO 10 I=1,2
      NVI52(I)=3
      NRI52(I)=5
      NH1(I)=8
      NQ2(I)=8

```

ORIGINAL PAGE IS
OF POOR QUALITY

RCT00010
RCT00020
RCT00030
RCT00040
RCT00050
RCT00060
RCT00070
RCT00080
RCT00090
RCT00100
RCT00110
RCT00120
RCT00130
RCT00140
RCT00150
RCT00160
RCT00170
RCT00180
RCT00190
RCT00200
RCT00210
RCT00220
RCT00230
RCT00240
RCT00250
RCT00260
RCT00270
RCT00280
RCT00290
RCT00300
RCT00310
RCT00320
RCT00330
RCT00340
RCT00350
RCT00360
RCT00370
RCT00380
RCT00390
RCT00400
RCT00410
RCT00420
RCT00430
RCT00440
RCT00450
RCT00460
RCT00470
RCT00480
RCT00490
RCT00500
RCT00510
RCT00520
RCT00530
RCT00540
RCT00550
RCT00560
RCT00570
RCT00580
RCT00590
RCT00600

	ND23(I)=8	RCT00610
	NTH4(I)=8	RCT00620
	NP2(I)=8	RCT00630
	NA2(I)=8	RCT00640
	NB2(I)=8	RCT00650
	NH2(I)=8	RCT00660
	NF2(I)=8	RCT00670
	ND21(I)=8	RCT00680
	ND24(I)=8	RCT00690
	ND25(I)=8	RCT00700
	ND26(I)=8	RCT00710
	NP5(I)=2	RCT00720
	NP55(I)=2	RCT00730
	NQ5(I)=2	RCT00740
	NQ55(I)=2	RCT00750
	NTRNI(I)=2	RCT00760
10	CONTINUE	RCT00770
	CALL UNITY(VI52,NVI52)	RCT00780
	CALL GAUSEL(3,3,V52,3,VI52,IERR)	RCT00790
	CALL UNITY(RI52,NRI52)	RCT00800
	CALL GAUSEL(5,5,R52,5,RI52,IERR)	RCT00810
	CALL EQUATE(R1,NR1,R2,NR2)	RCT00820
	CALL UNITY(Q2,NQ2)	RCT00830
	CALL GAUSEL(8,8,R1,8,Q2,IERR)	RCT00840
	CALL UNITY(D23,ND23)	RCT00850
	CALL GAUSEL(8,8,R15,8,D23,IERR)	RCT00860
	CALL PRNT(VI52,NVI52,4HVI52,1)	RCT00870
	CALL PRNT(RI52,NRI52,4HRI52,1)	RCT00880
	CALL PRNT(R2,NR2,4H R2,1)	RCT00890
	CALL PRNT(Q2,NQ2,4H Q2,1)	RCT00900
	CALL PRNT(D23,ND23,4H D23,1)	RCT00910
	CALL UNITY(TRNI,NTRNI)	RCT00920
	CALL GAUSEL(2,2,TRN,2,TRNI,IERR)	RCT00930
	CALL PRNT(TRNI,NTRNI,4HTRNI,1)	RCT00940
	EPSI=0.001	RCT00950
	DIFF=100.0	RCT00960
C		RCT00970
C	CHECK IF A IS ASYMPTOTICALLY STABLE BY CSTAB	RCT00980
C		RCT00990
	IOP(1)=0	RCT01000
	IOP(2)=0	RCT01010
	IOP(3)=0	RCT01020
	SCLE=1.0	RCT01030
	CALL CSTAB(A1,NA1,B1,NB1,F1,NF1,IOP,SCLE,DUMMY)	RCT01040
C		RCT01050
C	READY TO CALL SUBROUTINE RICNWT	RCT01060
	IDENT=.TRUE.	RCT01070
	DISC=.FALSE.	RCT01080
	FNULL=.FALSE.	RCT01090
	DO 50 I=1,550	RCT01100
	DUMMY(I)=0.0	RCT01110
50	CONTINUE	RCT01120
	CALL RICNWT(A1,NA1,B1,NB1,H1,NH1,Q1,NQ1,R1,NR1,F1,NF1,P1,	RCT01130
1	NP1,IOP,IDENT,DISC,FNULL,DUMMY)	RCT01140
C		RCT01150
C	BEGINNING OF THE STEP 2	RCT01160
C		RCT01170
	PRINT *, ' '	RCT01180
C		RCT01190
C		RCT01200

```

C   CREATE MATRIX D21
C
      DO 20 I=1,64
      D21(I)=0.0
      D26(I)=0.0
20  CONTINUE
      D21(1)  =P1(19)
      D21(9)  =P1(27)
      D21(2)  =P1(20)
      D21(10) =P1(28)
      D21(19) =P1(1)
      D21(27) =P1(9)
      D21(20) =P1(2)
      D21(28) =P1(10)
C
C   CREAT MATRIX B2
C
      CALL NULL(B2,NB2)
C
C   CREATE MATRIX A2
C
      CALL TRANP(D22,ND22,D24,ND24)
      CALL MULT(D23,ND23,D24,ND24,D25,ND25)
      CALL MULT(D22,ND22,D25,ND25,D24,ND24)
      CALL MULT(D21,ND21,D24,ND24,D25,ND25)
      CALL SUBT(A1,NA1,D25,ND25,A2,NA2)
C
C   CREATE MATRIX H2)
C
      D26(1)=P1(1)
      D26(2)=P1(2)
      D26(9)=P1(9)
      D26(10)=P1(10)
      D26(22) =P1(19)
      D26(30) =P1(27)
      D26(23) =P1(20)
      D26(31) =P1(28)
      CALL TRANP(B1,NB1,D22,ND22)
      CALL MULT(D22,ND22,D26,ND26,H2,NH2)
C
C   CHECK IF A2 IS ASYMPTICALLY STABLE BY CSTAB
C
      CALL CSTAB(A2,NA2,B2,NB2,F2,NF2,IOP,SCLE,DUMMY)
C
C   READY TO CALL RICNWT TO FIND P2
C
      IDENT=.FALSE.
      DO 60 I=1,550
      DUMMY(I)=0.0
60  CONTINUE
      CALL RICNWT(A2,NA2,B2,NB2,H2,NH2,Q2,NQ2,R2,NR2,F2,NF2,P2,NP2,
1IOP,IDENT,DISC,FNULL,DUMMY)
      ITE=0
C
C   END OF SEARCHING INITIAL MATRICES BY LQG
C
      PRINT *, ' '
      PRINT *, ' '
      PRINT *, '*** STARTING LQG SOLUTIONS ARE : '
C

```

```

RCT01210
RCT01220
RCT01230
RCT01240
RCT01250
RCT01260
RCT01270
RCT01280
RCT01290
RCT01300
RCT01310
RCT01320
RCT01330
RCT01340
RCT01350
RCT01360
RCT01370
RCT01380
RCT01390
RCT01400
RCT01410
RCT01420
RCT01430
RCT01440
RCT01450
RCT01460
RCT01470
RCT01480
RCT01490
RCT01500
RCT01510
RCT01520
RCT01530
RCT01540
RCT01550
RCT01560
RCT01570
RCT01580
RCT01590
RCT01600
RCT01610
RCT01620
RCT01630
RCT01640
RCT01650
RCT01660
RCT01670
RCT01680
RCT01690
RCT01700
RCT01710
RCT01720
RCT01730
RCT01740
RCT01750
RCT01760
RCT01770
RCT01780
RCT01790
RCT01800

```

20

C
C
C

START ITERATIVE ALGORITHM

```

P5(1)=P1(1)
P5(2)=P1(2)
P5(3)=P1(9)
P5(4)=P1(10)
P55(1)=P2(1)
P55(2)=P2(2)
P55(3)=P2(9)
P55(4)=P2(10)
Q5(1)=P1(19)
Q5(2)=P1(20)
Q5(3)=P1(27)
Q5(4)=P1(28)
Q55(1)=P2(19)
Q55(2)=P2(20)
Q55(3)=P2(27)
Q55(4)=P2(28)
CALL EQUATE(P5,NP5,P,NP)
CALL EQUATE(P55,NP55,PHAT,NPHAT)
CALL EQUATE(Q5,NQ5,Q,NQ)
CALL EQUATE(Q55,NQ55,QHAT,NQHAT)
CALL EQUATE(P5,NP5,PP,NPP)
CALL EQUATE(P55,NP55,PPH,NPPH)
CALL EQUATE(Q5,NQ5,QQ,NQQ)
CALL EQUATE(Q55,NQ55,QQH,NQQH)
PRINT *, ' '
CALL PRNT(P,NP,4H P,1)
PRINT *, ' '
CALL PRNT(Q,NQ,4H Q,1)
PRINT *, ' '
CALL PRNT(PHAT,NPHAT,4H PHAT,1)
PRINT *, ' '
CALL PRNT(QHAT,NQHAT,4H QHAT,1)

```

C
C
C

CREATE H*

```

NTH4(1)=8
NTH4(2)=8
CALL UNITY(TH4,NTH4)
TH4(37)=P1(1)
TH4(38)=P1(2)
TH4(45)=P1(9)
TH4(46)=P1(10)
TH4(55)=P1(19)
TH4(56)=P1(20)
TH4(63)=P1(27)
TH4(64)=P1(28)

```

GO TO 110

100

```

ITE=ITE+1
PRINT *, ' '
PRINT *, ' '
PRINT *, ' '
PRINT *, ' '
PRINT *, ' '
PRINT *, ' '
PRINT *, ' ***** THIS IS ITERATION ',ITE,' *****'
PRINT *, ' '
PRINT *, ' *** NEW SOLUTIONS ARE **'

```

```

RCT01810
RCT01820
RCT01830
RCT01840
RCT01850
RCT01860
RCT01870
RCT01880
RCT01890
RCT01900
RCT01910
RCT01920
RCT01930
RCT01940
RCT01950
RCT01960
RCT01970
RCT01980
RCT01990
RCT02000
RCT02010
RCT02020
RCT02030
RCT02040
RCT02050
RCT02060
RCT02070
RCT02080
RCT02090
RCT02100
RCT02110
RCT02120
RCT02130
RCT02140
RCT02150
RCT02160
RCT02170
RCT02180
RCT02190
RCT02200
RCT02210
RCT02220
RCT02230
RCT02240
RCT02250
RCT02260
RCT02270
RCT02280
RCT02290
RCT02300
RCT02310
RCT02320
RCT02330
RCT02340
RCT02350
RCT02360
RCT02370
RCT02380
RCT02390
RCT02400

```

	CALL EQUATE(P5, NP5, P, NP)	RCT02410
	CALL EQUATE(P55, NP55, PHAT, NPHAT)	RCT02420
	CALL EQUATE(Q5, NQ5, Q, NQ)	RCT02430
	CALL EQUATE(Q55, NQ55, QHAT, NQHAT)	RCT02440
	PRINT *, ' '	RCT02450
	CALL PRNT(P, NP, 4H P, 1)	RCT02460
	PRINT *, ' '	RCT02470
	CALL PRNT(PHAT, NPHAT, 4HPHAT, 1)	RCT02480
	PRINT *, ' '	RCT02490
	CALL PRNT(Q, NQ, 4H Q, 1)	RCT02500
	PRINT *, ' '	RCT02510
	CALL PRNT(QHAT, NQHAT, 4HQHAT, 1)	RCT02520
	PRINT *, ' '	RCT02530
	PRINT *, ' '	RCT02540
	PRINT *, ' '	RCT02550
	PRINT *, ' *** THE L-2 NORM = ', DIFF, ' *****'	RCT02560
	IF (DIFF .LT. EPSI) GO TO 130	RCT02570
	PRINT *, ' '	RCT02580
C		RCT02590
C	CREATE H*	RCT02600
C		RCT02610
	NTH4(1)=8	RCT02620
	NTH4(2)=8	RCT02630
	CALL UNITY(TH4, NTH4)	RCT02640
	TH4(37)=P1(1)	RCT02650
	TH4(38)=P1(2)	RCT02660
	TH4(45)=P1(17)	RCT02670
	TH4(46)=P1(18)	RCT02680
	TH4(55)=P1(35)	RCT02690
	TH4(56)=P1(36)	RCT02700
	TH4(63)=P1(51)	RCT02710
	TH4(64)=P1(52)	RCT02720
110	CALL MULT(TH3, NTH3, TH4, NTH4, TH5, NTH5)	RCT02730
	NTH6(1)=12	RCT02740
	NTH6(2)=8	RCT02750
	CALL NULL(TH6, NTH6)	RCT02760
	CALL JUXTC(TH5, NTH5, TH6, NTH6, TH7, NTH7)	RCT02770
	NTH8(1)=4	RCT02780
	NTH8(2)=16	RCT02790
	CALL JUXT(TH7, NTH7, TH8, NTH8, H3, NH3)	RCT02800
C		RCT02810
C	CREATE MATRIX A*	RCT02820
C		RCT02830
C	FIND A31	RCT02840
C		RCT02850
	CALL EQUATE(A51, NA51, TA51, NTA51)	RCT02860
	CALL MULT(A51, NA51, TA51, NTA51, TA5, NTA5)	RCT02870
	CALL SCALE(TA5, NTA5, TA51, NTA51, 0.5)	RCT02880
	CALL ADD(A51, NA51, TA51, NTA51, A31, NA31)	RCT02890
	CALL EQUATE(A31, NA31, E1, NE1)	RCT02900
C		RCT02910
C	FIND A32	RCT02920
C		RCT02930
	CALL TRANP(A31, NA31, A32, NA32)	RCT02940
C		RCT02950
C	FIND A33	RCT02960
C		RCT02970
	CALL MULT(VI52, NVI52, C5, NC5, TA6, NTA6)	RCT02980
	CALL TRANP(C5, NC5, TC5, NTC5)	RCT02990
	CALL MULT(TC5, NTC5, TA6, NTA6, TA51, NTA51)	RCT03000

	CALL MULT(Q5,NQ5,TA51,NTA51,TA5,NTA5)	RCT03010
	CALL SUBT(A31,NA31,TA5,NTA5,A33,NA33)	RCT03020
C		RCT03030
C	FIND A34	RCT03040
C		RCT03050
	CALL TRANP(B5,NB5,TB5,NTB5)	RCT03060
	CALL MULT(TB5,NTB5,P5,NP5,TA7,NTA7)	RCT03070
	CALL MULT(RI52,NRI52,TA7,NTA7,TB5,NTB5)	RCT03080
	CALL MULT(B5,NB5,TB5,NTB5,TA51,NTA51)	RCT03090
	CALL SUBT(A31,NA31,TA51,NTA51,A34,NA34)	RCT03100
C		RCT03110
C	FIND A*	RCT03120
C		RCT03130
	NZ1(1)=2	RCT03140
	NZ1(2)=6	RCT03150
	NZ2(1)=2	RCT03160
	NZ2(2)=4	RCT03170
	NZ5(1)=8	RCT03180
	NZ5(2)=16	RCT03190
	CALL NULL(Z1,NZ1)	RCT03200
	CALL NULL(Z2,NZ2)	RCT03210
	CALL NULL(TA5,NTA5)	RCT03220
	CALL NULL(TH4,NTH4)	RCT03230
	CALL NULL(Z5,NZ5)	RCT03240
	CALL NULL(TH4,NTH4)	RCT03250
	CALL JUXTC(A31,NA31,Z1,NZ1,X1,NX1)	RCT03260
	CALL JUXTC(TA5,NTA5,A32,NA32,X21,NX21)	RCT03270
	CALL JUXTC(X21,NX21,Z2,NZ2,X2,NX2)	RCT03280
	CALL JUXTC(Z2,NZ2,A33,NA33,X31,NX31)	RCT03290
	CALL JUXTC(X31,NX31,TA5,NTA5,X3,NX3)	RCT03300
	CALL JUXTC(Z1,NZ1,A34,NA34,X4,NX4)	RCT03310
	CALL JUXTR(X1,NX1,X2,NX2,X5,NX5)	RCT03320
	CALL JUXTR(X5,NX5,X3,NX3,X6,NX6)	RCT03330
	CALL JUXTR(X6,NX6,X4,NX4,X7,NX7)	RCT03340
	CALL JUXTC(X7,NX7,TH4,NTH4,X8,NX8)	RCT03350
	CALL JUXTR(X8,NX8,Z5,NZ5,A3,NA3)	RCT03360
C		RCT03370
C	CREATE Q*	RCT03380
C		RCT03390
C	FIND Q31	RCT03400
C		RCT03410
	CALL UNITY(A32,NA32)	RCT03420
	CALL GAUSEL(2,2,A51,2,A32,IERR)	RCT03430
	CALL TRANP(A32,NA32,TA5,NTA5)	RCT03440
	CALL MULT(R51,NR51,A32,NA32,TA51,NTA51)	RCT03450
	CALL MULT(TA5,NTA5,TA51,NTA51,A31,NA31)	RCT03460
	CALL ADD(P5,NP5,A31,NA31,A33,NA33)	RCT03470
	CALL ADD(A33,NA33,P55,NP55,Q31,NQ31)	RCT03480
C		RCT03490
C	FIND Q32	RCT03500
C		RCT03510
	CALL MULT(V51,NV51,TA5,NTA5,TA51,NTA51)	RCT03520
	CALL MULT(A32,NA32,TA51,NTA51,A31,NA31)	RCT03530
	CALL ADD(Q5,NQ5,A31,NA31,A33,NA33)	RCT03540
	CALL ADD(A33,NA33,Q55,NQ55,Q32,NQ32)	RCT03550
C		RCT03560
C	FIND Q*	RCT03570
C		RCT03580
	NZ6(1)=2	RCT03590
	NZ6(2)=10	RCT03600

	NZ7(1)=5	RCT03610
	NZ7(2)=4	RCT03620
	NZ8(1)=5	RCT03630
	NZ8(2)=3	RCT03640
	NZ9(1)=3	RCT03650
	NZ9(2)=9	RCT03660
	CALL NULL(TA51,NTA51)	RCT03670
	CALL NULL(X3,NX3)	RCT03680
	CALL JUXTC(Q31,NQ31,Z6,NZ6,X9,NX9)	RCT03690
	CALL JUXTC(TA51,NTA51,Q32,NQ32,X21,NX21)	RCT03700
	CALL JUXTC(X21,NX21,X3,NX3,X10,NX10)	RCT03710
	CALL JUXTC(Z7,NZ7,RI52,NRI52,X13,NX13)	RCT03720
	CALL JUXTC(X13,NX13,Z8,NZ8,X11,NX11)	RCT03730
	CALL JUXTC(Z9,NZ9,VI52,NVI52,X12,NX12)	RCT03740
	CALL JUXTR(X9,NX9,X10,NX10,X14,NX14)	RCT03750
	CALL JUXTR(X14,NX14,X11,NX11,X15,NX15)	RCT03760
	CALL JUXTR(X15,NX15,X12,NX12,X16,NX16)	RCT03770
	NX17(1)=12	RCT03780
	NX17(2)=4	RCT03790
	CALL NULL(X17,NX17)	RCT03800
	CALL NULL(TH8,NTH8)	RCT03810
	CALL JUXTC(X16,NX16,X17,NX17,X18,NX18)	RCT03820
	CALL JUXTR(X18,NX18,TH8,NTH8,Q3,NQ3)	RCT03830
C		RCT03840
C	CHECK IF A* IS ASYMPOTICALLY STABLE BY CSTAB	RCT03850
C		RCT03860
	DO 120 I=1,256	RCT03870
	P3(I)=0.0	RCT03880
	F3(I)=0.0	RCT03890
120	CONTINUE	RCT03900
	IOP(1)=0	RCT03910
	CALL CSTAB(A3,NA3,B3,NB3,F3,NF3,IOP,SCLE,DUMMY)	RCT03920
C		RCT03930
C	READY TO CALL RICNWT	RCT03940
C		RCT03950
	IOP(1)=0	RCT03960
	CALL RICNWT(A3,NA3,B3,NB3,H3,NH3,Q3,NQ3,R3,NR3,F3,NF3,P3,NP3, 1 IOP,IDENT,DISC,FNULL,DUMMY)	RCT03970
180	P5(1)=P3(1)	RCT03980
	P5(2)=P3(2)	RCT03990
	P5(3)=P3(17)	RCT04000
	P5(4)=P3(18)	RCT04010
	P55(1)=P3(69)	RCT04020
	P55(2)=P3(70)	RCT04030
	P55(3)=P3(85)	RCT04040
	P55(4)=P3(86)	RCT04050
	Q5(1)=P3(35)	RCT04060
	Q5(2)=P3(36)	RCT04070
	Q5(3)=P3(51)	RCT04080
	Q5(4)=P3(52)	RCT04090
	Q55(1)=P3(103)	RCT04100
	Q55(2)=P3(104)	RCT04110
	Q55(3)=P3(119)	RCT04120
	Q55(4)=P3(120)	RCT04130
	IF (ITE .GT. 1)GO TO 250	RCT04140
	DIFF=0.0	RCT04150
	DO 210 I=1,4	RCT04160
	DIFF=DIFF+(P5(I)-PP(I))**2+(P55(I)-PPH(I))**2	RCT04170
	DIFF=DIFF+(Q5(I)-QQ(I))**2+(Q55(I)-QQH(I))**2	RCT04180
210	CONTINUE	RCT04190
		RCT04200

24

	GO TO 220	RCT04210
250	CALL SUBT(P3,NP3,TP4,NTP4,TP5,NTP5)	RCT04220
	DIFF =0.0	RCT04230
	DO 160 I =1,256	RCT04240
	DIFF=DIFF+TP5(1)**2	RCT04250
160	CONTINUE	RCT04260
220	DIFF=SQRT(DIFF)	RCT04270
	CALL EQUATE(P3,NP3,TP4,NTP4)	RCT04280
	IF(ITE .GT.50)GO TO 140	RCT04290
	GO TO 100	RCT04300
	CALL NORMS(256,16,16,TP5,1,RN1)	RCT04310
	CALL NORMS(256,16,16,TP5,1,RN2)	RCT04320
	CALL NORMS(256,16,16,TP5,1,RN3)	RCT04330
	DIFF=RN1	RCT04340
	IF (DIFF.LT. EPSI)GO TO 180	RCT04350
	DIFF=RN2	RCT04360
	IF (DIFF.LT. EPSI)GO TO 180	RCT04370
	DIFF=RN3	RCT04380
	IF (DIFF.LT. EPSI)GO TO 180	RCT04390
	IF(ITE .GT.15)GO TO 140	RCT04400
	GO TO 100	RCT04410
130	PRINT *, ' '	RCT04420
	PRINT *, ' ***** CONVERGES WITH TORELANCE = ',EPSI	RCT04430
	GO TO 150	RCT04440
140	PRINT *, ' ***** DIVERGES WITH OVER ',ITE,' ITERATIONS.'	RCT04450
	GO TO 164	RCT04460
150	CALL TRANP(C5,NC5,TC5,NTC5)	RCT04470
	CALL MULT(Q,NQ,TC5,NTC5,TA6,NTA6)	RCT04480
	CALL MULT(TA6,NTA6,VI52,NVI52,F,NF)	RCT04490
	CALL MULT(F,NF,C5,NC5,TM2,NTM2)	RCT04500
	CALL TRANP(B5,NB5,TM3,NTM3)	RCT04510
	CALL MULT(RI52,NRI52,TM3,NTM3,TM4,NTM4)	RCT04520
	CALL MULT(TM4,NTM4,P,NP,K,NK)	RCT04530
	CALL MULT(B5,NB5,K,NK,TM1,NTM1)	RCT04540
	CALL SUBT(E1,NE1,TM2,NTM2,E2,NE2)	RCT04550
	CALL SUBT(E2,NE2,TM1,NTM1,ACC,NACC)	RCT04560
	PRINT *, ' '	RCT04570
	PRINT *, ' '	RCT04580
	PRINT *, '***** COMPENSATOR MATRICES AFTER TRANSFORMATION'	RCT04590
	PRINT *, ' '	RCT04600
	CALL PRNT(ACC,NACC,4H ACC,1)	RCT04610
	CALL PRNT(F,NF,4H F,1)	RCT04620
	CALL PRNT(K,NK,4H K,1)	RCT04630
	PRINT *, ' '	RCT04640
	PRINT *, ' '	RCT04650
	PRINT *, '*****CPMPENSATOR MATRIX AC'	RCT04660
	CALL MULT(TRNI,NTRNI,ACC,NACC,A33,NA33)	RCT04670
	CALL MULT(A33,NA33,TRN,NTRN,AC,NAC)	RCT04680
	PRINT *, ' '	RCT04690
	CALL PRNT(AC,NAC,4H AC,1)	RCT04700
164	STOP	RCT04710
	END	RCT04720