REFERENCES

1. Athan, M., and Peter L. Falb, Optimal Control : An Introduction to the Theory and Its Applications, Mc. Graw - Hill Book Company, New York, 1966.

2. Anderson, Brian D.O., and John B. Moore, Linear Optimal Control, Plentice - Hall, Inc., Engle wood, cliffs, New Jersey, 1971.

3. Cohn, N., Power - System Interconnections Control of Generation and Power Flow, Leeds & Northrup Company, North Wales, Pa., 1968.

4. Cohn, N., A Step - by - Step Analysis of Load - Frequency Control, Leeds & Northrup Company, Philadenphia, Pa., 1956.

5. Elgerd, Olle I., Electric Energy Systems Theory : An Introduction, Mc Graw - Hill Book Company, New York, 1971.

6. Elgerd, Olle I., and Charles E. Fosha, Megawatt - Frequency Control of Multi area Electric Energy Systems, IEEE Trans on Power Apparatus and Systems, Vol. PAS - 89, No. 4, April, 1970, pp. 556 - 562.

7. Fosha, Charles E., and Olle I. Elgerd, The Megawatt - Frequency Control Problem: A New Approach Via Optimal Control Theory, IEEE Trans. on Power Apparatus and Systems, Vol. PAS - 89, No.4, April, 1970, pp. 563 - 577.

8. Kirchmayer, Leon K., Economic Control of Inter-connected Systems, John Wiley & sons, Inc., New York, 1959.

9. Miniesy, Samir M., and E.V. Bohn, Optimum Load - Frequency Continuous Control with Unknown Deterministic Power Demand, IEEE Trans. on Power Apparatus and Systems, Vol. PAS - 91 No. 5, Sept/Oct; 1972, pp. 1910 - 1915.

APPENDIX

APPENDIX A

Controllability and Observability

Given a system described by

$$\dot{\bar{X}} = A\bar{X} + B\bar{U}$$

$$\bar{Y} = C\bar{X} .$$

It is desired to find controllability and observability of this system.

Controllability. A system is said to be controllable if it is possible to drive any state of the system to the origin in a finite time, by a suitable control energy U. Let G be the n x nm matrix, defined by the relation

$$G \triangleq \left[B \quad AB \quad A^2B \; --- \; A^{n-1} B \right] .$$

Then, the system is completely controllable if and only if the rank of G is n, that is

$$\text{rank} = n$$

or, equivalently, if and only if there is a set of n linearly independent column vectors of G. In particular, if B is an n x 1 matrix, that is if the control is scalar - valued, then the system is completely controllable if and only if the n x n matrix G is nonsingular, i.e.,

$$\det G \neq 0 .$$

Observability. A system is said to be observable if every initial state $\bar{X}(0)$ can be exactly determined from measurements of the output $\bar{Y}$ over a finite interval of time $0 \leqslant t \leqslant t_f$. Let H be an n x nm matrix, determined by the relation

$$H \triangleq \left[C^T \;\; A^T C^T - - - (A^T)^{n-1} \;\; C^T \right] .$$

Then, the system is completely observable if and only if the rank of H is n or, equivalently, if and only if there is n linearly independent column vectors of H. In particular, if C is a 1 x n matrix, that is if the output is scalar - valued, then the system is completely observable if and only if the n x n matrix H is nonsingular.

System Condition

The necessary condition required for the system in this research is that, the system must be completely controllable. This means that, since B is n x 1 matrix, the determinant of $G \neq 0$. The requirement of controllability guarantees that the minimum cost functional is finite.[1] From the matrix

$$G = \left[B \;\; AB \;\; A^2B - - - \;\; A^{n-1} B \right] .$$

Substituting in the numerical values of A and B, then

$$G = \begin{bmatrix} 0 & 0 & 0 & 3.84615 \times 5 \times 12.5 \times 3.33333 \\ 0 & 0 & 12.5 \times 3.33333 \times 3.84615 & 3.84615 \left[-3.33333 \times 12.5 - 3.33333 \times 12.5 \,(12.5 + 3.84615)\right] \\ 0 & 12 \times 3.84615 & -12.5 \times 3.84615(12.5 + 3.84615) & 3.84615 \left[12.5^2(12.5 + 3.84615) + 12.5 + 3.84615^2\right] \\ 3.84615 & -3.84615 & 3.84615^3 & -3.84615^4 \end{bmatrix} .$$

$$\text{Det} = 3.84615 \times 12.5 \times 3.84615 \times 3.33333 \times 12.5 \times 3.84615 \times 3.84615 \times 5 \times 12.5 \times 3.33333$$

$$= 23{,}744{,}418.3949$$

$$\neq 0 .$$

Thus, the system is completely controllable.

APPENDIX B

Derivation of the Optimal Controller $\hat{\bar{U}}$

Problem statement. Given the linear system

$$\dot{\bar{X}} = A\bar{X} + B\bar{U} \tag{1}$$

$$\bar{Y} = C\bar{X} \, . \tag{2}$$

The cost functional

$$J = \frac{1}{2} \int_0^T (\bar{X}^T Q \bar{X} + \bar{U}^T R \bar{U}) \, dt \tag{3}$$

where

A	=	n x n	matrix
B	=	n x r	matrix
C	=	m x n	matrix
$\bar{X}$	=	n x 1	vector
$\bar{U}$	=	r x 1	vector
Q	=	n x n	positive semidefinite matrix
R	=	r x r	positive definite matrix.

Find the optimal controller $\hat{\bar{U}}$, i.e., the controller which will drive the system (1) so as to minimize the cost functional J (3).

Derivation of $\hat{\bar{U}}$. The Hamiltonian is of the following form :

$$H = \frac{1}{2}\left[\bar{X}^T Q \bar{X}\right] + \frac{1}{2}\left[\bar{U}^T R \bar{U}\right] + \bar{P}^T\left[A\bar{X} + B\bar{U}\right]$$

where $\bar{P}$ is the costate vector.[1] For an optimal controller $\hat{\bar{U}}$, $\bar{P}$ and $\bar{X}$ must be the solution of

$$\dot{\bar{X}} = \frac{\partial H}{\partial \bar{P}} , \quad \dot{\bar{P}} = -\frac{\partial H}{\partial \bar{X}} \, .$$

Thus, $$\dot{\bar{X}} = A\bar{X} + B\bar{U} \; ; \; \dot{\bar{P}} = -Q\bar{X} - A^T\bar{P} . \qquad (4)$$

It is also necessary that

$$\frac{\partial H}{\partial \bar{U}} = 0 = R\bar{U} + B^T\bar{P} \qquad (5)$$

then $$\hat{\bar{U}} = -R^{-1} B^T\bar{P} . \qquad (6)$$

The requirement that R is positive definite guarantees the existence of R^{-1}.

The optimal controller must minimize the Hamiltonian. The necessary condition $\frac{\partial H}{\partial \bar{U}} = 0$ yields only an extremum of H with respect to $\bar{U}$. In order that the extremum of H to be a minimum with respect to $\bar{U}$, the matrix $\frac{\partial^2 H}{\partial^2 \bar{U}}$ must be positive definite. But, from eq. (5) it is seen that

$$\frac{\partial^2 H}{\partial^2 \bar{U}} = R$$

and hence since R is positive definite, it follows that the control $\hat{\bar{U}}$ given by eq. (6) does indeed minimize H, and hence H is minimal. It was shown that the costate vector $\bar{P}$ and the state vector $\bar{X}$ are related by an equation of the form

$$\bar{P} = K\bar{X} \qquad (7)$$

where, K is an n x n time - varying matrix.[1] Differentiating eq. (7) with respect to time results in

$$\dot{\bar{P}} = \dot{K}\bar{X} + K\dot{\bar{X}} . \qquad (8)$$

Substituting $\hat{\bar{U}}$ of eq. (6) in eq.(4) yields

$$\dot{\bar{X}} = A\bar{X} - BR^{-1}B^TK\bar{X} . \qquad (9)$$

Substituting $\dot{\bar{X}}$ of eq. (9) in eq. (8) then

$$\dot{\bar{P}} = \left[\dot{K} + KA - KBR^{-1}B^{T}K\right]\bar{X} . \qquad (10)$$

Substituting $\bar{P} = K\bar{X}$ in the second eq. (4) and then substituting in eq. (10) yields

$$-Q\bar{X} - A^{T}K\bar{X} = \left[\dot{K} + KA - KBR^{-1}B^{T}K\right]\bar{X} .$$

Rearranging, finally

$$\left[\dot{K} + KA - KBR^{-1}B^{T}K + A^{T}K + Q\right]\bar{X} = 0 .$$

Thus,
$$\dot{K} + KA - KBR^{-1}B^{T}K + A^{T}K + Q = 0 \qquad (11)$$

which is the matrix Riccati equation.

For a time - invariant system and for final time $T = \infty$, K is the solution of the nonlinear matrix algebraic equation :

$$-A^{T}K - KA + KBR^{-1}B^{T}K - Q = 0 . \qquad (12)$$

Once the steady state solution of the matrix Riccati equation (12) is found, $\hat{\bar{U}}$ may be calculated from eq. (6) with $\bar{P} = K\bar{X}$ that is,

$$\hat{\bar{U}} = -R^{-1}B^{T}K\bar{X} .$$

This is the desired optimal controller which will drive the system in such a way that the cost functional J is minimized.

APPENDIX C

Analog Computer Program

For simulation with an analog computer, programs are written as shown in Figs. 1, 2 and 3. Fig. 1 is the program for simulation of the conventional control system which does not include effects of speed - governor dead band. Fig. 2 includes speed - governor dead band. Fig. 3 is the program for simulation of the optimal control system. Analog computer used in the study is YEW analog computer.

It should be noted that, all three programs are written from the block diagram of Fig. 11, Fig. 19 and from the optimal control system respectively. These programs are written with full representation of each transfer function by each element of analog computer. Thus, in actual simulation, may drop some unnecessary summers in the same loop. But, this may lead to an incorrect sign when plotting the curves, so care must be taken in this case. Figs. 1, 2 and 3 are shown as follows :

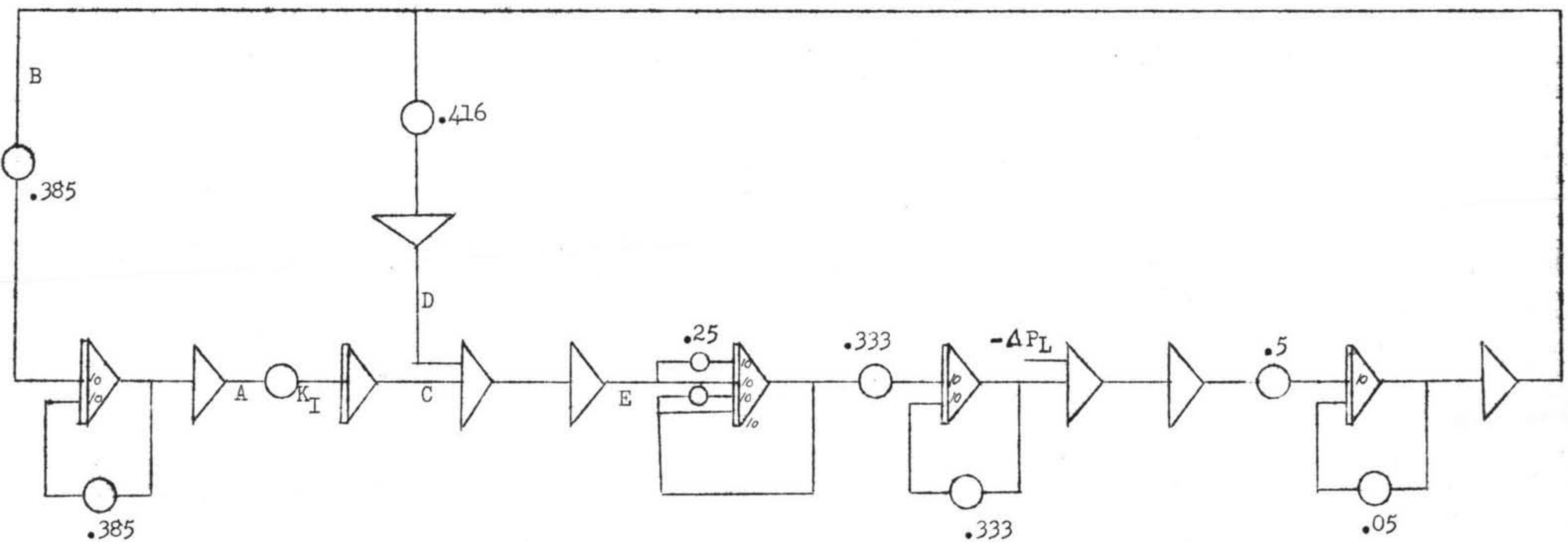


Figure 1. Program for simulation of the conventional control system without speed - governor dead band.

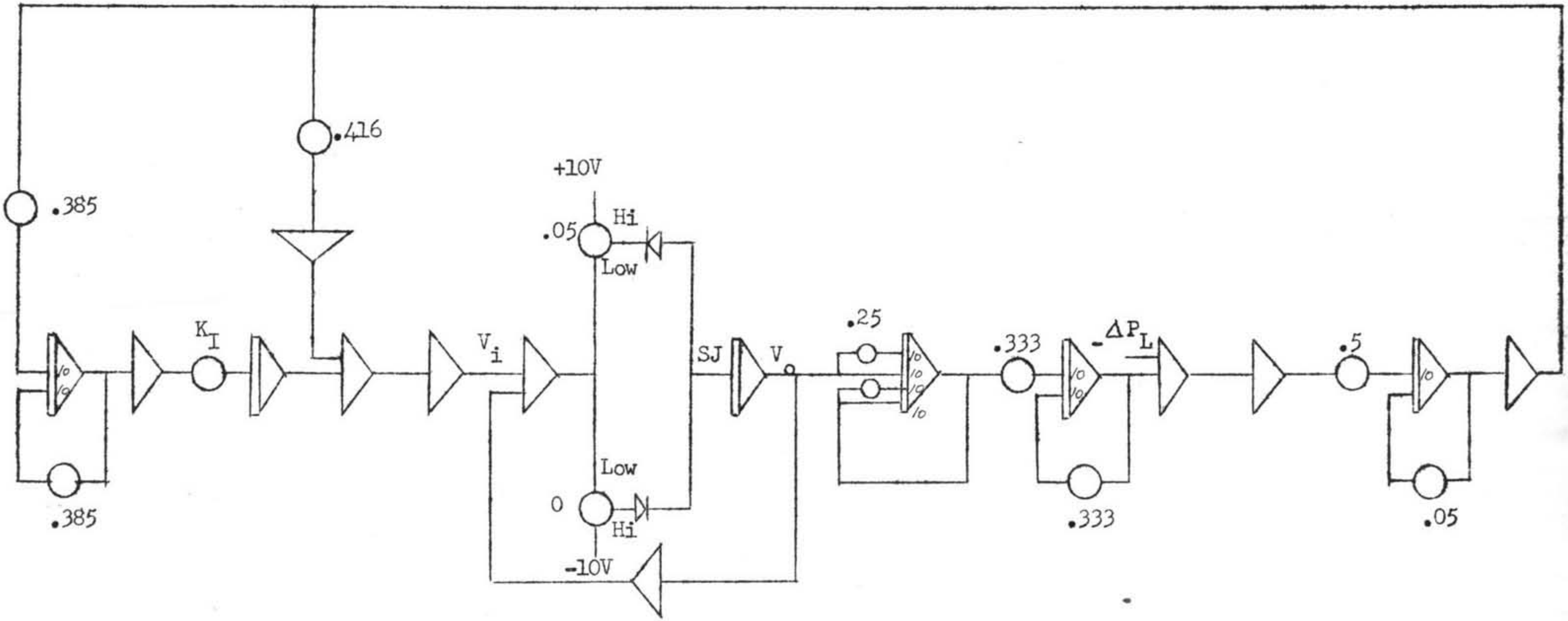


Figure 2. Program for simulation of the conventional control system with speed - governor dead band.

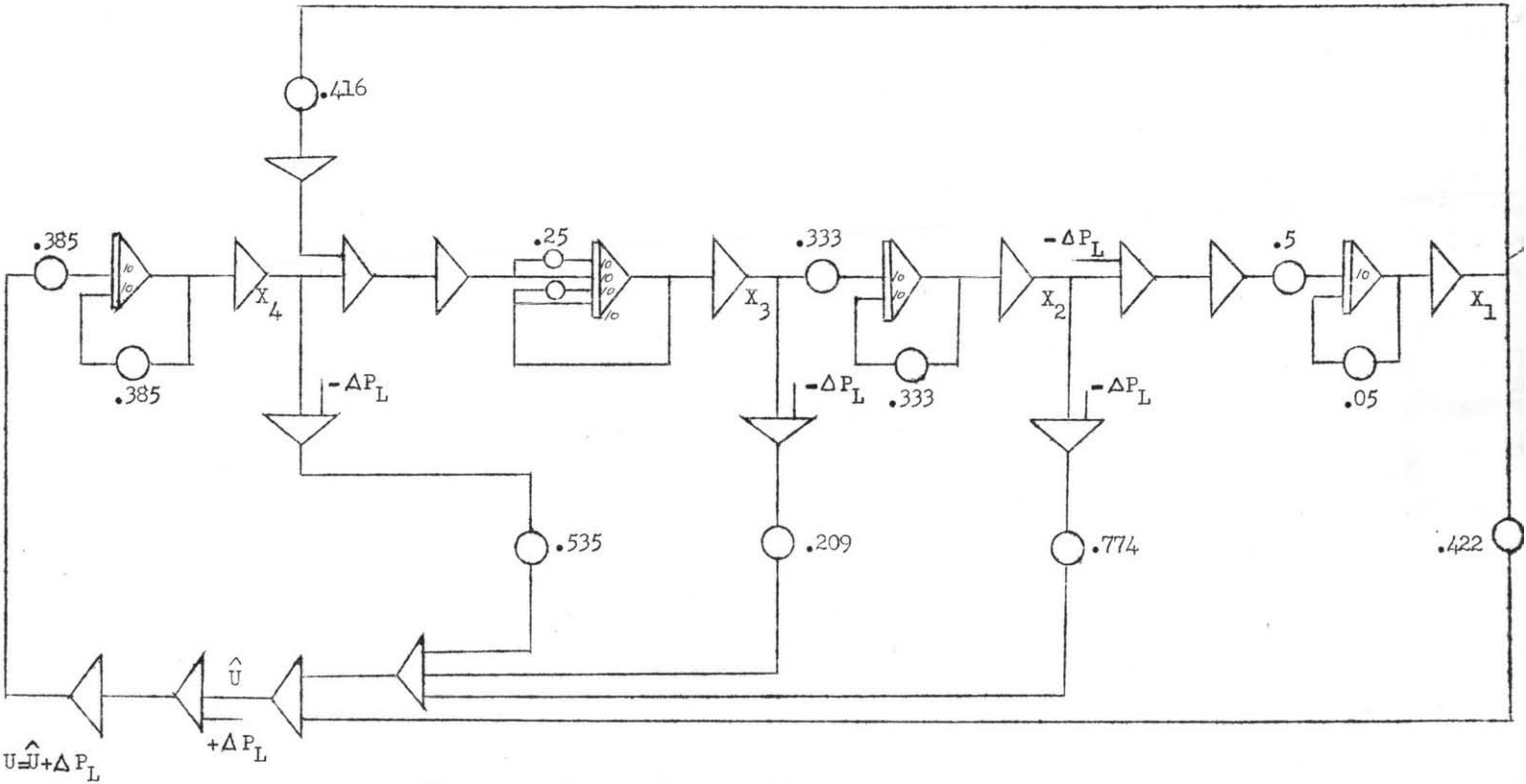


Figure 3. Program for simulation of the optimal control system without speed - governor dead band.

APPENDIX D

Solving the Riccati Equation

By using iterative method developed by Hitz, K.L., and B.D.O. Anderson the Riccati equation can be solved.[2] In this method, four new matrices are formed. Then, a new equation for iteration is written. The new equation is composed of the four new matrices. By means of a digital computer, the iteration is carried on until the accuracy is within the prescribed limit, in this case within .001. After obtaining the result from the iteration, the solution of the Riccati equation can also be obtained. The procedure is as follows : The Riccati equation to be solved is

$$KA + A^TK - KBR^{-1}B^TK + Q = 0 . \qquad (1)$$

Let

$$E = (I - A)^{-1} (I + A)$$

$$F = 2(I - A)^{-2} B$$

$$G = R + B^T (I - A^T)^{-1} Q (I - A)^{-1}B$$

and

$$H = Q (I - A)^{-1}B \text{ ; where } I = \text{identity matrix.}$$

The iterative equation is

$$\phi_{i+1} = E^T\phi_iE - \left[E^T\phi_iF + H\right] \left[G + F^T\phi_iF\right]^{-1} \left[E^T\phi_iF + H\right]^T + Q$$

where $\phi_o = 0$.

Then, the solution of equation (1) is

$$K = 2 \left[I - A^T\right]^{-1} \phi \left[I - A\right]^{-1} \qquad (3)$$

where

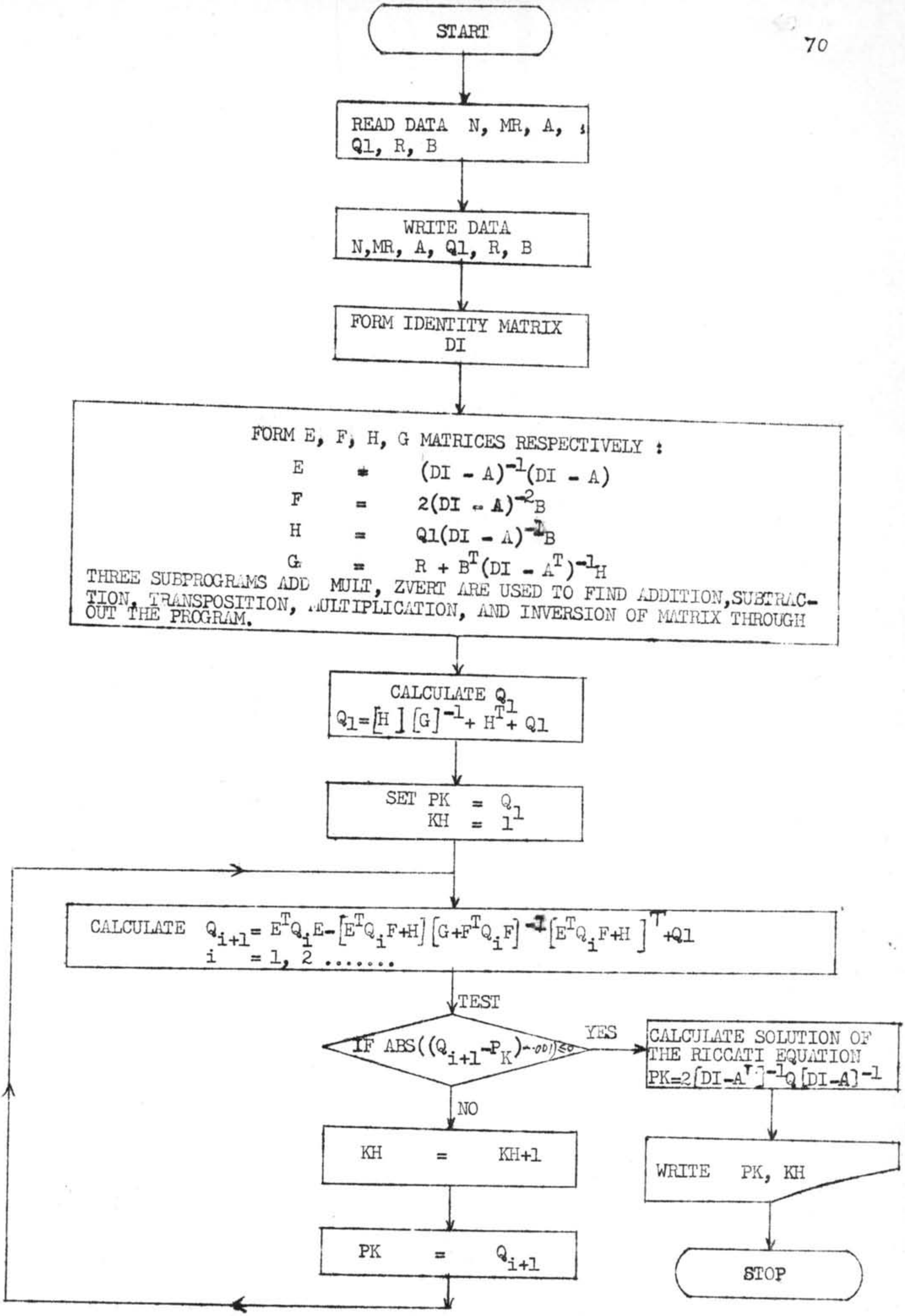
$$\phi = \lim_{i \to \infty} \phi_i .$$

In the computer program, some variables are represented by others to avoid confusion. These are

Q	for	Ø
Q1	"	Q
PK	"	K
DI	"	I
MR	"	r
N	"	n

where n, r are the dimensions of the square matrices A and R respectively.

The program is written with one main program and three subprograms. The three subprograms are used for transposition and multiplication, addition and subtraction, and for inversion of matrices. These subprograms named MULT, ADD, and ZVERT respectively. Matrices E, F, G and H are formed in main program by the helps of these subprograms. After substituting these four matrices into the iterative equation, the iteration are, then, carried on until the accuracy is not less than .001. This means that the difference between the elements of matrix Q just calculated and those corresponding elements previously calculated is not greater than .001. Obviously, after obtaining Q or Ø, K matrix can be calculated easily from equation (3). The flow chart is shown as follows :


START
READ DATA N, MR, A, Q1, R, B
WRITE DATA N,MR, A, Q1, R, B
FORM IDENTITY MATRIX DI
FORM E, F, H, G MATRICES RESPECTIVELY :
E = (DI - A)^-1(DI - A)
F = 2(DI - A)^-2 B
H = Q1(DI - A)^-1 B
G = R + B^T(DI - A^T)^-1 H
THREE SUBPROGRAMS ADD MULT, ZVERT ARE USED TO FIND ADDITION, SUBTRACTION, TRANSPOSITION, MULTIPLICATION, AND INVERSION OF MATRIX THROUGHOUT THE PROGRAM.
CALCULATE Q1
Q1=[H] [G]^-1 + H^T + Q1
SET PK = Q1
KH = 1
CALCULATE Q_i+1 = E^T Q_i E - [E^T Q_i F+H] [G+F^T Q_i F]^-1 [E^T Q_i F+H]^T +Q1
i = 1, 2 ....
TEST
IF ABS((Q_i+1 - P_K) - .001) ≤ 0
YES
CALCULATE SOLUTION OF THE RICCATI EQUATION
PK=2[DI-A^T]^-1 Q [DI-A]^-1
NO
KH = KH+1
PK = Q_i+1
WRITE PK, KH
STOP

DOS FORTRAN IV 360N-FO-479 3-8

OPTIONS IN EFFECT

LOAD =4

DECK NO

LIST YES

LISTX NO

EBCDIC

```
C      PROGRAM FOR SOLVING RICCATI EQUATION
C      USING ITERATIVE METHOD
C      DEVELOPED BY HITZ,K.L.,AND B.D.O.ANDERSON
C      ************************
       COMMON /B1/A1,DMX,A2,IR
       COMMON/B3/A3
       COMMON/B4/DI,A,KK
       DIMENSION A(9,9),Q1(9,9),R(2,2),B(9,9),E(9,9),F(9,9)
       DIMENSION G(2,2),H(9,9),Q(9,9),DI(9,9),A1(9,9)
       DIMENSION A2(9,9),A3(9,9),A4(9,9),A5(9,9),A6(9,9)
       DIMENSION PK(9,9)
C      N AND MR REPRESENT FOR N AND R THE DIMENSION
C      OF MATRIX
       READ(1,101) N,MR
   101 FORMAT (2I2)
       READ(1,100) ((A(I,J),J=1,N),I=1,N)
   100 FORMAT(8F8.5)
       READ(1,100)((Q1(I,J),J=1,N),I=1,N)
       READ(1,100)((R(I,J),J=1,MR),I=1,MR)
       READ(1,100)((B(I,J),J=1,MR),I=1,N)
       WRITE(8,589)
   589 FORMAT (//12X,'CHECK DATA INPUT')
       WRITE(8,599)N,MR
   599 FORMAT(//17X,'N =',I2,5X,'MR =',I2)
       WRITE(8,588)
   588 FORMAT(//12X,'A(N,N)-MATRIX ')
       WRITE(8,300)((A(I,J),J=1,N),I=1,N)
   300 FORMAT(//(/12X,4F10.5))
       WRITE(8,587)
   587 FORMAT(//12X,'Q(N,N)-MATRIX ')
       WRITE(8,300)((Q1(I,J),J=1,N),I=1,N)
       WRITE(8,586)
   586 FORMAT (//12X,'R(MR,MR)-MATRIX ')
       WRITE(8,301)((R(I,J),J=1,MR),I=1,MR)
   301 FORMAT(//(/12X,F10.5))
       WRITE(8,585)
   585 FORMAT(//12X,'B(N,MR)-MATRIX ')
       WRITE(8,301)((B(I,J),J=1,MR),I=1,N)
C      FORM IDENTITY MATRIX DI
       DO10I=1,N
       DO10J=1,N
       IF(I.NE.J)GOTO11
       DI(I,J)=1.
       GOTO10
    11 DI(I,J)=0.
    10 CONTINUE
C      FORM E-MATRIX
       KK=1
       CALL ADD (N,N)
       CALL ZVERT(N)
       IF(DMX.EQ.0.)GOTO99
       KK=2
       CALL ADD(N,N)
       IR=1
```

```
      CALL MULT(N,N,N)
      DO12 I=1,N
      DO12 J=1,N
   12 E(I,J)=A3(I,J)
C     FORM F-MATRIX
      DO14I=1,N
      DO14 J=1,N
   14 A2(I,J)=A1(I,J)
      CALL MULT(N,N,N)
      DO15 I=1,N
      DO15J=1,N
   15 A4(I,J)=A1(I,J)
      DO16 I=1,N
      DO16 J=1,N
      A1(I,J)=A3(I,J)
   16 A2(I,J)=B(I,J)
      CALL MULT(N,N,MR)
      DO17 I=1,N
      DO17J=1,MR
   17 F(I,J)=2*A3(I,J)
C     FORM H-MATRIX
      DO18 I=1,N
      DO18 J=1,N
      A1(I,J)=O1(I,J)
   18 A2(I,J)=A4(I,J)
      CALL MULT(N,N,N)
      DO19 I=1,N
      DO19J=1,N
      A1(I,J)=A3(I,J)
   19 A2(I,J)=B(I,J)
      CALL MULT(N,N,MR)
      DO20 I=1,N
      DO20J=1,MR
   20 H(I,J)=A3(I,J)
C     FORM G-MATRIX
      DO30 I=1,N
      DO30 J=1,N
   30 A1(I,J)=A(I,J)
      IR=2
       CALL MULT(N,N,N)
      DO22 I=1,N
      DO22 J=1,N
   22 A(I,J)=A1(I,J)
      KK=1
      CALL ADD(N,N)
      CALL ZVERT(N)
      IF(DMX.EQ.0.)GOTO99
      DO23 I=1,N
      DO23 J=1,N
   23 A5(I,J)=A1(I,J)
      DO24 I=1,N
      DO24 J=1,MR
   24 A2(I,J)=H(I,J)
      IR=1
```

```
      CALL MULT(N,N,MR)
      DO25 I=1,N
      DO25 J=1,MR
   25 A1(I,J)=B(I,J)
      IR=2
      CALL MULT(N,N,N)
      DO26 I=1,N
      DO26 J=1,MR
   26 A2(I,J)=A3(I,J)
      IR=1
      CALL MULT(MR,N,MR)
      DO27 I=1,MR
      DO27 J=1,MR
      DI(I,J)=R(I,J)
   27 A(I,J)=A3(I,J)
      KK=2
      CALL ADD(MR,MR)
      DO28 I=1,MR
      DO28J=1,MR
   28 G(I,J)=A2(I,J)
C     CALCULATE  QI+1,I=0  WHICH SUBSTITUTE Q0=0
      DO31 I=1,MR
      DO31 J=1,MR
   31 A1(I,J)=G(I,J)
      CALL ZVERT(MR)
      DO81 I=1,N
      DO81 J=1,MR
      A2(I,J)=A1(I,J)
   81 A1(I,J)=-H(I,J)
      CALL MULT(N,MR,MR)
      DO32 I=1,N
      DO32J=1,N
   32 A1(I,J)=H(I,J)
      IR=2
      CALL MULT(N,N,N)
      DO33 I=1,N
      DO33 J=1,N
      A2(I,J)=A1(I,J)
   33 A1(I,J)=A3(I,J)
      IR=1
      CALL MULT(N,MR,N)
      DO34 I=1,N
      DO34J=1,N
      DI(I,J)=A3(I,J)
   34 A(I,J)=Q1(I,J)
      CALL ADD(N,N)
      DO35 I=1,N
      DO35 J=1,N
   35 Q(I,J)=A2(I,J)
C     SET PK-MATRIX EQUAL TO Q(I,J) AND SET KH=1
      KH=1
      DO 76 I=1,N
      DO 76J=1,N
   76 PK(I,J)=Q(I,J)
```

```
C      CALCULATE QI+1
    29 DO36 I=1,N
       DO36 J=1,N
       A1(I,J)=F(I,J)
    36 A2(I,J)=Q(I,J)
       IR=2
       CALL MULT(N,N,N)
       IR=1
       CALL MULT(MR,N,N)
       DO37 I=1,N
       DO37 J=1,N
       A1(I,J)=A3(I,J)
    37 A2(I,J)=F(I,J)
       CALL MULT(MR,N,MR)
       DO38 I=1,MR
       DO38 J=1,MR
       DI(I,J)=A3(I,J)
    38 A(I,J)=G(I,J)
       CALL ADD(MR,MR)
       DO39 I=1,MR
       DO39 J=1,MR
    39 A1(I,J)=A2(I,J)
       CALL ZVERT(MR)
       DO40 I=1,N
       DO40 J=1,N
       A6(I,J)=A1(I,J)
       A1(I,J)=E(I,J)
    40 A2(I,J) =Q(I,J)
       IR=2
       CALL MULT(N,N,N)
       IR=1
       CALL MULT(N,N,N)
       DO41 I=1,N
       DO41 J=1,N
       A1(I,J)=A3(I,J)
    41 A2(I,J)=F(I,J)
       CALL MULT(N,N,MR)
       DO42 I=1,N
       DO42 J=1,MR
       DI(I,J)=A3(I,J)
    42 A(I,J)=-H(I,J)
       KK=1
       CALL ADD(N,MR)
       DO43 I=1,N
       DO43 J=1,N
    43 A2(I,J)=A6(I,J)
       CALL MULT(N,MR,MR)
       IR=2
       CALL MULT(N,N,N)
       DO44 I=1,N
       DO44 J=1,N
       A2(I,J)=A1(I,J)
    44 A1(I,J)=A3(I,J)
       IR=1
```

```
      CALL MULT(N,MR,N)
      DO45 I=1,N
      DO45 J=1,N
   45 A6(I,J)=-A3(I,J)
      DO46 I=1,N
      DO46 J=1,N
      A1(I,J)=E(I,J)
   46 A2(I,J)=Q(I,J)
      IR=2
      CALL MULT(N,N,N)
      IR=1
      CALL MULT(N,N,N)
      DO47 I=1,N
      DO47 J=1,N
      A1(I,J)=A3(I,J)
   47 A2(I,J)=E(I,J)
      CALL MULT(N,N,N)
      DO48 I=1,N
      DO48 J=1,N
      DI(I,J)=A3(I,J)
   48 A(I,J)=A6(I,J)
      KK=2
      CALL ADD(N,N)
      DO49 I=1,N
      DO49 J=1,N
      DI(I,J)=A2(I,J)
   49 A(I,J)=Q1(I,J)
      CALL ADD(N,N)
      DO50 I=1,N
      DO50 J=1,N
   50 Q(I,J)=A2(I,J)
C     CHECK IF ((QI+1-PK)-.001) LESS THAN 0,IF NOT ADD 1 TO
C     KH AND SET PK=QI+1 THEN RECALCULATE QI+1,IF THE LOGIC
C     IS TRUE WRITE KH=NO OF ITERATIONS
      DO70 I=1,N
      DO70 J=1,N
      IF(ABS(Q(I,J)-PK(I,J))-.001)70,70,71
   70 CONTINUE
      GOTO72
   71 KH=KH+1
      DO73I=1,N
      DO73J=1,N
   73 PK(I,J)=Q(I,J)
      GOTO29
C     CALCULATE PK=SOLUTION OF THE RICCATI EQUATION
   72 WRITE(8,600)
  600 FORMAT('1'//12X,'SOLUTION OF THE RICCATI EQUATION')
      DO51 I=1,N
      DO51 J=1,N
      A1(I,J)=A5(I,J)
   51 A2(I,J)=Q(I,J)
      CALL MULT(N,N,N)
      DO52 I=1,N
      DO52 J=1,N
```

```
      A1(I,J)=2.*A3(I,J)
   52 A2(I,J)=A4(I,J)
      CALL MULT(N,N,N)
      DO53 I=1,N
      DO53 J=1,N
   53 Q(I,J)=A3(I,J)
      WRITE(8,501)
  501 FORMAT(//12X,'K-MATRIX IS ')
      WRITE(8,300)((Q(I,J),J=1,N),I=1,N)
      WRITE(8,500)KH
  500 FORMAT(//12X,'NO OF ITERATIONS =',I3)
      GOTO80
   99 WRITE(8,202)
  202 FORMAT(//12X,'NO SOLUTION,THE MATRIX IS SINGULAR')
   80 STOP
      END
```

```
C        SUBROUTINE FOR FINDING MATRIX ADDITION
C        AND SUBTRACTION
         SUBROUTINE ADD(KN,KM)
         COMMON/B1/A1,DMX,A2,IR
         COMMON/B4/D1,F1,K1
         DIMENSION D1(9,9),F1(9,9),A1(9,9),A2(9,9)
         IF(K1.EQ.2)GOTO13
         DO12 I=1,KN
         DO12 J=1,KM
      12 A1(I,J)=D1(I,J)-F1(I,J)
         RETURN
      13 DO14 I=1,KN
         DO14 J=1,KM
      14 A2(I,J)=D1(I,J)+F1(I,J)
         RETURN
         END
```

```
C     SUBROUTINE FOR FINDING MATRIX TRANSPOSITION
C     AND MULTIPLICATION
      SUBROUTINE MULT(KX,KY,KZ)
      COMMON/B1/T,CKZ,A2,IR
      COMMON/B3/CZ
      DIMENSION T(9,9),A2(9,9),CZ(9,9)
      IF(IR.EQ.2)GOTO27
      P=0.
      DO5 I=1,KX
      DO5J=1,KZ
      DO6 L=1,KY
    6 P=P+T(I,L)*A2(L,J)
      CZ(I,J)=P
    5 P=0.
      RETURN
   27 I=KX-1
      DO20 K=1,I
      M=K+1
      DO20 L=M,KX
      TA=T(K,L)
      T(K,L)=T(L,K)
   20 T(L,K)=TA
      RETURN
      END
```

```
C      SUBROUTINE FOR FINDING MATRIX INVERSION
       SUBROUTINE ZVERT(NC)
       COMMON/B1/A,AMX,A2,IR
       DIMENSION B(9),CR(9,9),A(9,9),DD(9),IVT(9)
       IF(NC.NE.1)GOTO42
       A(1,1)=1/A(1,1)
       RETURN
    42 DO50 I=1,NC
    50 B(I)=0.
       KK=1
       DO55 ID=1,NC
       DO30 IM=1,NC
    30 DD(IM)=ABS(A(IM,1))
    15 AMX=DD(1)
       IDX=1
       DO10 L=2,NC
       IF(AMX.GE.DD(L))GOTO10
       AMX=DD(L)
       IDX=L
    10 CONTINUE
       IVT(KK)=IDX
       IF(AMX.EO.0.)GOTO93
       IF(KK.EQ.1)GOTO44
       N=KK-1
       DO25 J=1,N
       IF(IVT(KK)-IVT(J))25,18,25
    25 CONTINUE
       GOTO44
    18 DD(IDX)=0.
       GOTO15
    44 KK=KK+1
       B(IDX)=1.
       AMR=A(IDX,1)
       DO34 J=1,NC
    34 A(IDX,J)=A(IDX,J)/AMR
       B(IDX)=B(IDX)/AMR
       DO22 IP=1,NC
       IF(IP.EQ.IDX)GOTO22
       AL=-A(IP,1)
       B(IP)=B(IDX)*AL+B(IP)
       DO23 IE=1,NC
    23 A(IP,IE)=A(IDX,IE)*AL+A(IP,IE)
    22 CONTINUE
       DO55 I=1,NC
       JJ=NC-1
       DO14 J=1,JJ
    14 A(I,J)=A(I,J+1)
       A(I,NC)=B(I)
    55 B(I)=0.
       DO40 IV=1,NC
       IR=IVT(IV)
       DO40 JV=1,NC
    40 CR(IV,JV)=A(IR,JV)
       DO41 JZ=1,NC
```

```
      IC=IVT(JZ)
      DO41IZ=1,NC
      A(IZ,IC)=CR(IZ,JZ)
   41 CONTINUE
   93 RETURN
      END
```

```
// EXEC

          CHECK DATA INPUT

              N = 4        MR = 1

          A(N,N)-MATRIX

            -0.05000    5.00000    0.0        0.0
             0.0       -3.33333    3.33333    0.0
            -5.20833    0.0      -12.50000   12.50000
             0.0        0.0        0.0       -3.84615

          Q(N,N)-MATRIX

             1.00000    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

          R(MR,MR)-MATRIX

             1.00000

          B(N,MR)-MATRIX

             0.0
             0.0
             0.0
             3.84615
```

SOLUTION OF THE RICCATI EQUATION

K-MATRIX IS

0.41118	0.34949	0.07492	0.10978
0.34949	0.43422	0.10487	0.20128
0.07492	0.10487	0.02622	0.05427
0.10978	0.20128	0.05427	0.13911

NO OF ITERATIONS = 17

13.49.56,DURATION 00.06.20

VITA

Name :	Tongyou Chulapinyo
Education :	Degree of Bachelor of Electrical Engineering, College of Engineering, Chulalongkorn University, Bangkok, Thailand, 1968.
Position and office:	Area unit engineer, Transformer Section, Branch Bangkapi, Metropolitan Electricity Authority, Bangkok, Thailand.