On the Solution of the Stationary Riccati Equation by Integration

Choice of Steplength

Lindahl, Sture

CODEN: LUTFD2/(TFRT-3115)/1-42/(1974)

ON THE SOLUTION OF THE STATIONARY RICCATI EQUATION BY INTEGRATION - CHOICE OF STEPLENGTH

STURE LINDAHL

Department of Automatic Control
Lund Institute of Technology
September 1974

Dokumentutgivare
Lund Institute of Technology
Dept of Automatic Control

Handläggare
06T0

Författare
Sture Lindahl

Dokumentnamn
REPORT

Dokumentbeteckning
LUTFD2/(TFRT-8115)/1-42/(1974)

Utgivningsdatum
Sept 1974

Ärendebeteckning
06T6

10T4

Dokumenttitel och undertitel

18T0

On the solution of the stationary Riccati equation by integration-choice of steplength

Referat (sammandrag)

26T0

One way to solve the stationary Riccati equation

$$0 = A^TX + XA - XBQ_2^{-1} B^TX + Q_1$$

is to integrate the associated differential equation

$$-\frac{dS}{dt} = A^TS + SA - SBQ_2^{-1} B^TS + Q_1$$

until ds/dt is zero. In order to reduce the computation time it is desirable to increase the steplength as much as possible. The numerical stability of the integration method limits the steplength. In this report the results of some numerical experiments are given. A rule of thumb for the choice of steplength is also given.

Referat skrivet av
Author

Förslag till ytterligare nyckelord
44T0

Klassifikationssystem och -klass(er)
50T0

Indextermer (ange källa)
52T0
numerical integration (Thesaurus of Engineering and Scientific Terms, Engineers Joint Council, N.Y., USA)

Omfång
42 pages

Övriga bibliografiska uppgifter
56T2

Språk
English

Sekretessuppgifter
60T0

ISSN
60T4

ISBN
60T6

Dokumentet kan erhållas från
Department of Automatic Control
Lund Institute of Technology
Box 725, S-220 07 LUND 7, Sweden

Mottagarens uppgifter
62T4

Pris
66T0

Blankett LU 11:25 1976-07

DOKUMENTDATABLAD enligt SIS 62 10 12

SIS-DB 1

ON THE SOLUTION OF THE STATIONARY RICCATI EQUATION BY INTEGRATION-CHOICE OF STEPLENGTH

STURE LINDAHL

TABLE OF CONTENTS

ON THE SOLUTION OF THE STATIONARY RICCATI EQUATION BY INTEGRATION-CHOICE OF STEPLENGTH

ABSTRACT

One way to solve the stationary Riccati equation

$$0 = A^T X + XA - XBQ_2^{-1} B^T X + Q_1$$

is to integrate the associated differential equation

$$-\frac{dS}{dt} = A^T S + SA - SBQ_2^{-1} B^T S + Q_1$$

until dS/dt is zero. In order to reduce the computation time it is desirable to increase the steplength as much as possible. The numerical stability of the integration method limits the steplength. In this report the results of some numerical experiments are given. A rule of thumb for the choice of steplength is also given.

1. INTRODUCTION

Consider the linear, time-invariant stabilizable system

$$\frac{dx}{dt} = Ax + Bu \qquad (1.1)$$

and the quadratic cost functional to be minimized

$$V = \int_0^\infty [x^T(s)\ Q_1 x(s) + u^T(s)\ Q_2 u(s)]ds \qquad (1.2)$$

where x is the n-dimensional state vector, u the r-dimensional control vector, A a n×n-matrix and B a n×r-matrix. The symmetric n×n-matrix, Q_1, is non-negative definite, and Q_2 is a positive definite symmetric r×r-matrix.

The optimal control is a linear, time-invariant feedback

$$u(t) = - L\ x(t) \qquad (1.3)$$

where

$$L = Q_2^{-1}\ B^T X \qquad (1.4)$$

and X is a symmetric non-negative, definite solution of the algebraic Riccati equation

$$A^T X + XA - XBQ_2^{-1}\ B^T X + Q_1 = 0 \qquad (1.5)$$

There are at least four different methods of solving (1.5).

Method A.

Let

$$a_i = \begin{bmatrix} b_i \\ c_i \end{bmatrix} \qquad i = 1,2,\ldots,n$$

be eigenvectors of

$$E = \begin{bmatrix} A & -BQ_2^{-1}\ B^T \\ -Q_1 & -A^T \end{bmatrix}$$

In [1] it is shown that each solution of (1.5) can be expressed as

$$X = \begin{bmatrix} c_1, & c_2, & \ldots, & c_n \end{bmatrix} \begin{bmatrix} b_1, & b_2, & \ldots, & b_n \end{bmatrix}^{-1}$$

where the inverse is assumed to exist for certain combinations of eigenvectors. Conversely, if $\begin{bmatrix} b_1, & b_2, & \ldots, & b_n \end{bmatrix}$ is non-singular, then

$$X = \begin{bmatrix} c_1, & c_2, & \ldots, & c_n \end{bmatrix} \begin{bmatrix} b_1, & b_2, & \ldots, & b_n \end{bmatrix}^{-1}$$

satisfies (1.5)

<u>Method B.</u>

Consider

$$J = \frac{1}{2}\, x^T(t_1)\ Q_o x(t_1) + \frac{1}{2} \int_{t_o}^{t_1} [x^T(s)Q_1 x(s) + u^T(s)Q_2 u(s)]ds \qquad (1.6)$$

The optimal control is a linear feedback

$$u(t) = -\ Q_2^{-1}\ B^T S(t)\ x(t) \qquad (1.7)$$

where S(t) is a solution of the Riccati equation

$$-\frac{dS}{dt} = A^T S + SA - SBQ_2^{-1}\ B^T S + Q_1 \qquad (1.8)$$

$$S(t_1) = Q_o \qquad (1.9)$$

Let $\pi(t;\, Q_o,\, t_1)$ be the solution of (1.8) with the initial condition

$$\pi(t_1;\, Q_o,\, t_1) = Q_o \tag{1.10}$$

Using the fact that the system is stabilizable, it can be shown [1] that

$$\lim_{t\to-\infty} \pi(t;\, 0,\, t_1) = \bar{S} \tag{1.11}$$

exists. $\bar{S}$ is a non-negative, definite solution of (1.5). Using the fundamental matrix approach

$$\pi(t;\, Q_o,\, t_1) = [\Sigma_{21}(t;\, t_1) + \Sigma_{22}(t;\, t_1)Q_o] \cdot$$

$$[\Sigma_{11}(t;\, t_1) + \Sigma_{12}(t;\, t_1)Q_o]^{-1} \tag{1.12}$$

where

$$\Sigma(t,\, t_1) = \begin{bmatrix} \Sigma_{11}(t;\, t_1) & \Sigma_{12}(t;\, t_1) \\ \Sigma_{21}(t;\, t_1) & \Sigma_{22}(t;\, t_1) \end{bmatrix} \tag{1.13}$$

and

$$\frac{d}{dt}\, \Sigma(t;\, t_1) = E\, \Sigma(t;\, t_1) \tag{1.14}$$

$\pi(t;\, 0,\, t_1)$ is computed for increasing value of t_1-t until a stationary solution is reached with desired accuracy.

<u>Method C.</u>

The optimal regulator problem can also be solved by straightforward integration of (1.8) until a stationary solution is reached with desired accuracy. In [2] a fourth-order Runge-

Kutta method was used.

Method D (Kleinman's method).

Kleinman [5] has proposed that Newton's method should be used in order to solve (1.5). In each iteration we have to solve the Lyapunov equation

$$A_K^T V_k + V_k A_k + Q_1 + L_k^T Q_2 L_k = 0$$

where

$$A_k = A - BL_k$$

and

$$L_k = Q_2^{-1} B^T V_{k-1}$$

The author has no experience with this method.

Choice of method.

Using method A we have to compute all eigenvalues and all eigenvectors of a $2n\times 2n$-matrix, invert a $n\times n$-matrix and compute a couple of $n\times n$-matrix products. For high-order systems the eigenvalue problem can cause numerical difficulties, especially if there are multiple eigenvalues.

Using method B we have to compute the transition matrix for (1.14), and for high-order systems this may lead to computational difficulties.

Method C is computationally simpler than method A and method B since it can be shown [1] that the numerical integration is a stable procedure. The disadvantage of method

C is that it is time consuming. In order to reduce the computation time it is desirable to increase the steplength as much as possible. The numerical stability of the integration method used limits the steplength. The results of some numerical experiments with (1.8) is given and a rule of thumb for the steplength is also given.

2. CHOICE OF STEPLENGTH

Consider the matrix differential equation

$$-\frac{dS}{dt} = A^T S + SA - SBQ_2^{-1}B^T S + Q_1 \tag{2.1}$$

$$S(t_1) = Q_o \tag{2.2}$$

Our problem is to determine the maximum possible steplength, h_{max}.

Let S_1 denote the <u>exact</u> solution of (2.1) and S_2 an <u>approximate</u> solution of (2.1) obtained by numerical integration. Introducing $\Delta S = S_1 - S_2$ we can derive the following differential equation for ΔS

$$-\frac{d}{dt}\Delta S = (A-BQ_2^{-1}\ B^T S_1)^T \Delta S + \Delta S(A-BQ_2^{-1}\ B^T S_1) + \Delta SBQ_2^{-1}\ B^T\ \Delta S \tag{2.3}$$

with the initial condition

$$\Delta S(t_1) = 0 \tag{2.4}$$

Equation (2.3) can also be written as

$$-\frac{d}{dt}\Delta S = F^T(t)\Delta S + \Delta SF(t) + \Delta SBQ_2^{-1}\ B^T\ \Delta S \tag{2.5}$$

where

$$F(t) = [A - BQ_2^{-1} B^T S_1(t)] \tag{2.6}$$

Introducing $z = (\Delta s_{11}, \Delta s_{21}, \ldots, \Delta s_{nn})^T$ equation (2.5) can be written as

$$\frac{dz}{dt} = G(t)z + O(||z||^2) \tag{2.7}$$

The eigenvalues $\mu_k(t)$ of $G(t)$ are given by [3]

$$\mu_k(t) = \lambda_i(t) + \lambda_j(t) \quad i,j = 1,2,\ldots,n \tag{2.8}$$

where $\lambda_i(t)$ is an eigenvalue of $F(t)$. To obtain numerical stability for equation (2.1) it seems reasonable to use a steplength which makes

$$\frac{dz}{dt} = G(t)z \tag{2.9}$$

numerically stable for all t.

This means that we obtain the following rule of thumb

$$h_{max} \leq \hat{h}$$

where

$$\mu_i(t)\hat{h} \in D_s$$

and D_s is the domain of stability for the integration method used.

If we use a fourth-order Runge-Kutta method to integrate

$$\frac{dy}{dt} = My \tag{2.10}$$

it is known [4] that this integration procedure is stable if and only if

$$h\,\rho_i \in \Omega \qquad \forall i \qquad i = 1,2,\ldots N$$

where ρ_i is an eigenvalue of M and

$$\Omega = \{\alpha \in C; |\ 1 + \alpha + \alpha^2/2 + \alpha^3/6 + \alpha^4/24\ | < 1\}$$

3. NUMERICAL EXPERIMENTS

In this section the results of some numerical experiments are given.

Example 1

The system is

$$\frac{dx}{dt} = u \tag{3.1}$$

and the cost functional is

$$V = \int_0^\infty [100x^2(s) + u^2(s)]ds \tag{3.2}$$

It is possible to calculate an expression for S(t) and L(t)

$$S(t) = 10(1-e^{-20t})/(1+e^{-20t}) \tag{3.3}$$

$$L(t) = 10(1-e^{-20t})/(1+e^{-20t}) \tag{3.4}$$

It is also possible to calculate an expression for $\lambda(t)$

$$\lambda(t) = -10(1-e^{-20t})/(1+e^{-20t}) \tag{3.5}$$

The maximum possible steplength is $\hat{h} = 0.139$.

From table 3.1 we can see that if $h \leqslant 0.136$ the initial error is reduced to zero after 30 steps. If $h = 0.139$ (the predicted maximum steplength) then the initial error is not reduced. If $h > 0.140$ the initial error is increased.

From table 3.2 we can see the result if we start with $Q_o = 0$. If $h \leqslant 0.136$ the solution approaches the correct solution. If $h \geqslant 0.140$ the difference equations defined by the fourth-order Runge-Kutta integration procedure seems to have another stationary solution than the stationary solution to the differential equation.

h	Number of steps (k)			
	0	10	20	30
0.100	10.000010	10.000000	10.000000	10.000000
0.120	10.000010	10.000000	10.000000	10.000000
0.124	10.000010	10.000000	10.000000	10.000000
0.128	10.000010	10.000000	10.000000	10.000000
0.132	10.000010	10.000000	10.000000	10.000000
0.136	10.000010	10.000003	10.000001	10.000000
0.138	10.000010	10.000006	10.000004	10.000002
0.139	10.000010	10.000010	10.000010	10.000010
0.140	10.000010	10.000010	10.000010	10.000010
0.160	10.000010	10.004057	10.900070	11.015269
0.180	10.000010	10.673110	11.235236	10.261799
0.200	10.000010	-7.8973007	-6.1129192	-6.3156540

Table 3.1. $S(k \cdot h)$ for different steplengths and Q_o in the neighbourhood of the stationary solution, $X = 10$.

h	Number of steps (k)			
	0	10	20	30
0.100	0.0000000	9.9998834	9.9999999	9.9999999
0.120	0.0000000	9.9888766	9.9999672	10.000000
0.124	0.0000000	9.9656295	9.9996639	9.9999968
0.128	0.0000000	9.8871715	9.9962034	9.9998761
0.132	0.0000000	9.6105378	9.9494184	9.9942882
0.136	0.0000000	8.7672235	9.2072164	9.5158461
0.140	0.0000000	7.4291753	7.4157795	7.4152622
0.160	0.0000000	3.7616103	3.7614541	3.7614541
0.180	0.0000000	-1.4746107	-1.4666215	-1.4667521
0.200	0.0000000	$-1.5139580 \cdot 10^{-5}$	$-2.4889708 \cdot 10^{-3}$	-0.40397096

Table 3.2. $S(k \cdot h)$ in Example 1 for different steplengths and $Q_o = 0$.

Example 2

The system is

$$\frac{dx_1}{dt} = -x_1 + u_1$$

$$\frac{dx_2}{dt} = x_2 + u_2$$

and the cost functional is

$$V = \int_0^\infty [0.002001x_1^2(s) + u_1^2(s) + u_2^2(s)]ds$$

The eigenvalues λ_1 and λ_2 corresponding to the stationary solution of the Riccati equation are

$\lambda_1 = -1.001$ and $\lambda_2 = -1.0$. The maximum steplength is $\hat{h} = 1.39$.

From table 3.3 and 3.4 we can see that if $h \geq 1.40$ the numerical solution becomes indefinite, but if $h < 1.39$ the numerical solution remains positive definite. We can also see that the numerical solution approaches the exact solution very slowly if the steplength is chosen near the predicted maximum steplength.

h	Number of steps (k)			
	0	10	20	30
1.00	0.0000000	0.99998273	1.0000000	1.0000000
1.20	0.0000000	0.99694269	0.99999069	0.99999999
1.24	0.0000000	0.98989057	0.99989801	0.99999901
1.28	0.0000000	0.96597843	0.99884576	0.99996089
1.32	0.0000000	0.88463093	0.98674634	0.99847821
1.36	0.0000000	0.60857983	0.84758890	0.94077613
1.38	0.0000000	0.27983353	0.48308406	0.62986818
1.39	0.0000000	0.02348535	0.04673550	0.06973800
1.40	0.0000000	-0.03238645	-0.07622321	-0.13631858
1.60	0.0000000	-63.571238	-62.552227	-62.551731

Table 3.3. 1000 $s_{11}(k \cdot h)$ in Example 2 for different steplengths and $Q_o = \mathrm{diag}(0.0,\ 1.9999)$.

h	Number of steps (k)			
	0	10	20	30
1.00	1.9999000	2.0000000	2.0000000	2.0000000
1.20	1.9999000	1.9999997	2.0000000	2.0000000
1.24	1.9999000	1.9999991	2.0000000	2.0000000
1.28	1.9999000	1.9999968	2.0000000	2.0000000
1.32	1.9999000	1.9999891	1.9999989	2.0000000
1.36	1.9999000	1.9999629	1.9999863	1.9999950
1.38	1.9999000	1.9999318	1.9999536	1.9999685
1.39	1.9999000	1.9999078	1.9999149	1.9999216
1.40	1.9999000	1.9998753	1.9998443	1.9998055
1.60	1.9999000	1.9524459	1.3759230	1.3761453

Table 3.4. $s_{22}(k \cdot h)$ in Example 2 for different steplengths and $Q_o = \mathrm{diag}(0.0,\ 1.9999)$.

From table 3.5 and 3.6 we can see the result if $Q_o = I$. The closed loop eigenvalues are then $\lambda_1(t_1) = -2.0$ and $\lambda_2(t_1) = 0.0$. The maximum steplength is now predicted to be ≈ 0.7 for $t = t_1$ and to be 1.39 when we reach the stationary value of the Riccati equation. If $h \geq 1.2$ the numerical solution "explodes", but if $h = 1.0$ the numerical solution approaches the correct solution. From table 3.5 and table 3.6 it can be seen that the choice of initial value (Q_o) can reduce the maximum steplength.

From table 3.7 and 3.8 we can see that if $Q_0 = 0$ and h is sufficiently small the numerical solution approaches a positive, semi-definite solution of the algebraic Riccati equation. The closed loop eigenvalues are $\lambda_1 = -1.001$ and $\lambda_2 = 1.0$, i.e. the closed loop system is unstable.

h	Number of steps (k)			
	0	10	20	30
1.00	1000·0000	0.88122385	0.99999795	1.0000000
1.20	1000·0000	$1.72 \cdot 10^{38}$	$1.85 \cdot 10^{38}$	$1.99 \cdot 10^{38}$
1.24	1000·0000	$3.83 \cdot 10^{28}$	$1.39 \cdot 10^{34}$	$5.08 \cdot 10^{39}$

Table 3.5. $1000 s_{11}$ (kh) in Example 2 for different steplengths and $Q_o = I$.

h	Number of steps (k)			
	0	10	20	30
1.00	1.0000000	1.9999883	2.0000000	2.0000000
1.20	1.0000000	1.9988877	1.9999967	2.0000000
1.24	1.0000000	1.9965630	1.9999664	1.9999997

Table 3.6. $s_{22}(k \cdot h)$ in Example 2 for different steplengths and $Q_o = I$.

h	Nümber of steps (k)			
	0	10	20	30
1.00	0.0000000	0.99998273	1.0000000	1.0000000
1.20	0.0000000	0.99694269	0.99999069	0.99999999
1.40	0.0000000	-0.32386451	0.76223211	-1.3631858
1.60	0.0000000	-635.71238	-625.52227	-625.51731

Table 3.7. $1000 s_{11}(k \cdot h)$ in Example 2 for different steplengths and $Q_o = 0$.

h	Number of steps (k)			
	0	10	20	30
1.00	0.0000000	0.0000000	0.0000000	0.0000000
1.20	0.0000000	0.0000000	0.0000000	0.0000000
1.40	0.0000000	0.0000000	0.0000000	0.0000000

Table 3.8. $s_{22}(k \cdot h)$ in Example 2 for different steplengths and $Q_o = 0$.

From table 3.9 and 3.10 we can see that if $Q_o = 10 \cdot I$ the steplength must be chosen to be ≈ 0.15 in order to obtain numerical stability.

From table 3.11 and 3.12 we can see that it is possible to start with $Q_o = diag(0.0, 1.0)$ and use a steplength near the maximum possible and obtain numerical stability.

h	Number of steps (k)			
	0	100	200	300
0.10	10000·000	1.0000034	1.0000001	1.0000001
0.15	10000·000	1.0000001	1.0000001	1.0000001
0.20	10000·000	$-3.44\cdot10^{35}$	$-1.05\cdot10^{27}$	$-2.93\cdot10^{36}$

Table 3.9. 1000 $s_{11}(k\cdot h)$ in Example 2 for different steplengths and $Q_o = 10\cdot I$.

h	Number of steps (k)			
	0	100	200	300
0.10	10.000000	2.0000001	2.0000001	2.0000001
0.15	10.000000	2.0000001	2.0000001	2.0000001
0.20	10.000000	2.0000000	2.0000000	2.0000000

Table 3.10. $s_{22}(k \cdot h)$ in Example 2 for different steplengths and $Q_o = 10 \cdot I$.

h	Number of steps (k)			
	0	10	20	30
1.00	0.0000000	0.99998273	1.0000000	1.0000000
1.20	0.0000000	0.99694269	0.99999069	0.99999999
1.24	0.0000000	0.98989057	0.99989801	0.99999901
1.28	0.0000000	0.96597843	0.99884576	0.99996089
1.32	0.0000000	0.88463093	0.98674634	0.99847821
1.36	0.0000000	0.60857983	0.84758890	0.94077613
1.38	0.0000000	0.27983353	0.48308406	0.62986818
1.39	0.0000000	0.02348535	0.04673550	0.06973800
1.40	0.0000000	-0.32386451	-0.76223211	-1.3631858
1.60	0.0000000	-635.71238	-625.52227	-625.51731

Table 3.11. $1000 s_{11}(k \cdot h)$ in Example 2 for different steplengths and $Q_o = \mathrm{diag}(0.0, 1.0)$.

h	Number of steps (k)			
	0	10	20	30
1.00	1.0000000	1.9999883	2.0000000	2.0000000
1.20	1.0000000	1.998877	1.9999967	2.0000000
1.24	1.0000000	1.9965630	1.9999664	1.9999997
1.28	1.0000000	1.9887173	1.9996204	1.9999877
1.32	1.0000000	1.9610539	1.9949420	1.9994289
1.36	1.0000000	1.8767226	1.9207219	1.9515850
1.38	1.0000000	1.8100455	1.8193716	1.8219726
1.39	1.0000000	1.7756506	1.7767811	1.7769033
1.40	1.0000000	1.7429177	1.7415780	1.7415264
1.60	1.0000000	1.3761611	1.3761454	1.3761454

Table 3.12. $s_{22}(k \cdot h)$ in Example 2 for different steplengths and $Q_o = \mathrm{diag}(0.0, 1.0)$.

Example 3.

The A-matrix is given in table 3.13 and the B-matrix is given in table 3.14. The matrices in the cost functional are defined in table 3.15, 3.16 and 3.17.

The absolute largest eigenvalue to $(A-BQ_2^{-1}\ B^TX)$ is $\lambda_{max} =$ $= -0.86907$, and we predict the maximum possible steplength $\hat{h} = 1.60$. From table 3.18 and 3.19 we can see that if $h \leqslant 1.64$ the method is numerically stable. If the steplength is increased to 1.66 and larger, the numerical solution becomes indefinite as can be seen from table 3.20 and 3.21.

A-MATRIX

COLUMN NR	1	2	3	4	5
ROW NR 1	.00000000	.00000000	1.0000000	.00000000	.00000000
ROW NR 2	.00000000	.00000000	.00000000	1.0000000	.00000000
ROW NR 3	-.13170980-04	-.96791798-06	.20002711-03	.54515979-07	.46031430-04
ROW NR 4	-.24601647-03	-.18255061-03	.15007541-05	.25000602-01	-.18774520-03
ROW NR 5	.78567932-01	.58067810-01	-.62180548-03	-.34535808-03	-.31832823
ROW NR 6	.28879373	-.49754370-01	.63869415-03	-.12290437-02	.19315409
ROW NR 7	.00000000	.00000000	.00000000	.00000000	.00000000
ROW NR 8	.00000000	.00000000	.00000000	.00000000	.00000000
ROW NR 9	.00000000	.00000000	.00000000	.00000000	.00000000
ROW NR 10	.00000000	.00000000	.00000000	.00000000	.00000000

COLUMN NR	6	7	8	9	10
ROW NR 1	.00000000	.00000000	.00000000	.00000000	.00000000
ROW NR 2	.00000000	.00000000	.00000000	.00000000	.00000000
ROW NR 3	.14145960-04	.00000000	.00000000	.49862601-04	.00000000
ROW NR 4	.12249854-02	.00000000	.00000000	.00000000	.23873246-02
ROW NR 5	-.67394390-01	.31415920	.00000000	.00000000	.00000000
ROW NR 6	-.47811935	.00000000	.26179933	.00000000	.00000000
ROW NR 7	.00000000	-.25000000	.00000000	.00000000	.00000000
ROW NR 8	.00000000	.00000000	-.33333333	.00000000	.00000000
ROW NR 9	.00000000	.00000000	.00000000	-.52958364-02	.00000000
ROW NR 10	.00000000	.00000000	.00000000	.00000000	-.82862453

Table 3.13. The A-matrix in Example 3.

B-MATRIX

COLUMN NR	1	2	3	4	5
ROW NR 1	.00000000	.00000000	.00000000	.00000000	.00000000
ROW NR 2	.00000000	.00000000	.00000000	.00000000	.00000000
ROW NR 3	.00000000	.00000000	.69889413-04	.00000000	.00000000
ROW NR 4	.00000000	.00000000	.00000000	-.15915497-02	.00000000
ROW NR 5	.00000000	.00000000	.00000000	.00000000	.00000000
ROW NR 6	.00000000	.00000000	.00000000	.00000000	.00000000
ROW NR 7	.25000000	.00000000	.00000000	.00000000	.00000000
ROW NR 8	.00000000	.33333333	.00000000	.00000000	.00000000
ROW NR 9	.00000000	.00000000	-.65320781-02	.00000000	.50344827-02
ROW NR 10	.00000000	.00000000	.00000000	.82862454	.00000000

Table 3.14. The B-matrix in Example 3.

MATRIX Q0

COLUMN NR	1	2	3	4	5
ROW NR 1	379.82376	-43.451472	34608.372	727.89941	5.8865554
ROW NR 2	-43.451471	110.31614	3774.3282	1525.7137	1.2761667
ROW NR 3	34608.371	3774.3284	9166720.6	-103378.96	1154.8761
ROW NR 4	727.89954	1525.7137	-103378.98	301072.06	171.96108
ROW NR 5	5.8865548	1.2761670	1154.8761	171.96107	1.8481911
ROW NR 6	2.7113116	2.9714576	-153.81176	678.83426	.50985337
ROW NR 7	4.7754853	.96221570	920.80695	124.32339	.89868520
ROW NR 8	1.6680177	1.2289739	-87.778200	341.46571	.29917379
ROW NR 9	48.529049	35.200892	40288.950	-57.274871	5.2857703
ROW NR 10	2.1447715	3.8590539	-296.69852	824.96145	.47013877

COLUMN NR	6	7	8	9	10
ROW NR 1	2.7113146	4.7754852	1.6680177	48.529049	2.1447713
ROW NR 2	2.9714602	.96221569	1.2289739	35.200894	3.8590539
ROW NR 3	-153.81192	920.80695	-87.778200	40288.951	-296.69848
ROW NR 4	678.83495	124.32339	341.46571	-57.275035	824.96145
ROW NR 5	.50985406	.89868520	.29917379	5.2857701	.47013875
ROW NR 6	2.5550682	.27412874	1.0424168	.14953485	1.8614350
ROW NR 7	.27412875	2.4004240	.17594776	4.1992211	.33977994
ROW NR 8	1.0424168	.17594776	1.7797400	-.14854220-01	.93719392
ROW NR 9	.14953410	4.1992211	-.14854216-01	338.62543	-.26860848
ROW NR 10	1.8614363	.33977994	.93719392	-.26860881	2.8505125

Table 3.15. The Q_o-matrix in Example 3.

MATRIX Q1

COLUMN NR	1	2	3	4	5
ROW NR 1	1.0000000	.00000000	.00000000	.00000000	.00000000
ROW NR 2	.00000000	1.0000000	.00000000	.00000000	.00000000
ROW NR 3	.00000000	.00000000	1.0000000	.00000000	.00000000
ROW NR 4	.00000000	.00000000	.00000000	1.0000000	.00000000
ROW NR 5	.00000000	.00000000	.00000000	.00000000	1.0000000
ROW NR 6	.00000000	.00000000	.00000000	.00000000	.00000000
ROW NR 7	.00000000	.00000000	.00000000	.00000000	.00000000
ROW NR 8	.00000000	.00000000	.00000000	.00000000	.00000000
ROW NR 9	.00000000	.00000000	.00000000	.00000000	.00000000
ROW NR 10	.00000000	.00000000	.00000000	.00000000	.00000000

COLUMN NR	6	7	8	9	10
ROW NR 1	.00000000	.00000000	.00000000	.00000000	.00000000
ROW NR 2	.00000000	.00000000	.00000000	.00000000	.00000000
ROW NR 3	.00000000	.00000000	.00000000	.00000000	.00000000
ROW NR 4	.00000000	.00000000	.00000000	.00000000	.00000000
ROW NR 5	.00000000	.00000000	.00000000	.00000000	.00000000
ROW NR 6	1.0000000	.00000000	.00000000	.00000000	.00000000
ROW NR 7	.00000000	1.0000000	.00000000	.00000000	.00000000
ROW NR 8	.00000000	.00000000	1.0000000	.00000000	.00000000
ROW NR 9	.00000000	.00000000	.00000000	1.0000000	.00000000
ROW NR 10	.00000000	.00000000	.00000000	.00000000	1.0000000

Table 3.16. The Q_1-matrix in Example 3.

MATRIX Q2

COLUMN NR	1	2	3	4	5
ROW NR 1	1.0000000	.00000000	.00000000	.00000000	.00000000
ROW NR 2	.00000000	1.0000000	.00000000	.00000000	.00000000
ROW NR 3	.00000000	.00000000	10.000000	.00000000	.00000000
ROW NR 4	.00000000	.00000000	.00000000	10.000000	.00000000
ROW NR 5	.00000000	.00000000	.00000000	.00000000	10.000000

Table 3.17. The Q_2-matrix in Example 3.

RESULT AFTER 571 RUNGE-KUTTA STEP
****

COMPUTED S-MATRIX

COLUMN NR	1	2	3	4	5
ROW NR 1	380.00371	-43.455657	34614.816	727.53316	5.8871521
ROW NR 2	-43.455655	110.31786	3775.5042	1525.7415	1.2763391
ROW NR 3	34614.815	3775.5044	9168047.6	-103376.58	1155.0512
ROW NR 4	727.53265	1525.7405	-103376.55	301073.06	171.96207
ROW NR 5	5.8871520	1.2763401	1155.0512	171.96212	1.8482147
ROW NR 6	2.7105601	2.9715248	-153.79704	678.83614	.50985655
ROW NR 7	4.7759351	.96235238	920.94452	124.32421	.89870375
ROW NR 8	1.6676458	1.2290048	-87.773803	341.46676	.29917508
ROW NR 9	48.479482	35.205842	40290.324	-57.130109	5.2860472
ROW NR 10	2.1437742	3.8591233	-296.69558	824.96413	.47014098

COLUMN NR	6	7	8	9	10
ROW NR 1	2.7105675	4.7759351	1.6676459	48.479484	2.1437752
ROW NR 2	2.9715292	.96235236	1.2290048	35.205845	3.8591253
ROW NR 3	-153.79722	920.94451	-87.773802	40290.324	-296.69563
ROW NR 4	678.83715	124.32421	341.46676	-57.130075	824.96413
ROW NR 5	.50985763	.89870375	.29917508	5.2860467	.47014108
ROW NR 6	2.5550723	.27413143	1.0424189	.14985874	1.8614401
ROW NR 7	.27413144	2.4004387	.17594880	4.1994457	.33978178
ROW NR 8	1.0424189	.17594880	1.7797410	-.14701636-01	.93719672
ROW NR 9	.14985750	4.1994456	-.14701627-01	338.64982	-.26822516
ROW NR 10	1.8614420	.33978178	.93719672	-.26822509	2.8505197

Table 3.18. The result after 571 steps with h = 1.62.

RESULT AFTER 583 RUNGE-KUTTA STEP
****

COMPUTED S-MATRIX

COLUMN NR	1	2	3	4	5
ROW NR 1	380.00378	-43.455654	34614.824	727.53297	5.8871530
ROW NR 2	-43.455651	110.31785	3775.5042	1525.7417	1.2763390
ROW NR 3	34614.823	3775.5044	9168047.7	-103376.56	1155.0512
ROW NR 4	727.53294	1525.7407	-103376.55	301073.07	171.96206
ROW NR 5	5.8871534	1.2763395	1155.0511	171.96212	1.8482148
ROW NR 6	2.7105621	2.9715264	-153.79712	678.83656	.50985672
ROW NR 7	4.7759357	.96235228	920.94453	124.32420	.89870378
ROW NR 8	1.6676459	1.2290047	-87.773778	341.46677	.29917514
ROW NR 9	48.479460	35.205842	40290.324	-57.130030	5.2860472
ROW NR 10	2.1437747	3.8591238	-296.69556	824.96415	.47014100

COLUMN NR	6	7	8	9	10
ROW NR 1	2.7105674	4.7759356	1.6676459	48.479467	2.1437747
ROW NR 2	2.9715292	.96235226	1.2290047	35.205844	3.8591256
ROW NR 3	-153.79717	920.94453	-87.773777	40290.324	-296.69557
ROW NR 4	678.83717	124.32420	341.46677	-57.129989	824.96414
ROW NR 5	.50985768	.89870378	.29917514	5.2860464	.47014112
ROW NR 6	2.5550733	.27413147	1.0424192	.14985863	1.8614411
ROW NR 7	.27413147	2.4004387	.17594882	4.1994457	.33978176
ROW NR 8	1.0424192	.17594883	1.7797412	-.14701564-01	.93719677
ROW NR 9	.14985767	4.1994457	-.14701557-01	338.64982	-.26822494
ROW NR 10	1.8614425	.33978176	.93719676	-.26822487	2.8505196

Table 3.19. The result after 583 steps with h = 1.64.

h = 1.66

RESULT AFTER 561 RUNGE-KUTTA STEP
****

COMPUTED S-MATRIX

COLUMN NR	1	2	3	4	5
ROW NR 1	379.98904	-43.477512	34616.792	722.32851	5.8841974
ROW NR 2	-43.477511	110.28535	3778.4478	1517.9916	1.2719391
ROW NR 3	34616.792	3778.4475	9167781.5	-102675.48	1155.4492
ROW NR 4	722.32714	1517.9914	-102675.43	299226.97	170.91398
ROW NR 5	5.8841959	1.2719399	1155.4492	170.91408	1.8476198
ROW NR 6	2.6988028	2.9540215	-152.21338	674.66656	.50748926
ROW NR 7	4.7737986	.95917080	921.23232	123.56635	.89827353
ROW NR 8	1.6617216	1.2201830	-86.975751	339.36538	.29798213
ROW NR 9	48.482037	35.209652	40289.980	-56.223082	5.2865621
ROW NR 10	1.8720191	3.4544652	-260.08796	728.56958	.41541511

COLUMN NR	6	7	8	9	10
ROW NR 1	2.6988120	4.7737986	1.6617216	48.482034	1.8720216
ROW NR 2	2.9540243	.95917079	1.2201830	35.209655	3.4544656
ROW NR 3	-152.21366	921.23232	-86.975750	40289.980	-260.08807
ROW NR 4	674.66740	123.56635	339.36538	-56.223239	728.56961
ROW NR 5	.50749050	.89827354	.29798213	5.2865616	.41541531
ROW NR 6	2.5456546	.27241967	1.0376727	.15190771	1.6437156
ROW NR 7	.27241968	2.4001276	.17508617	4.1998180	.30021058
ROW NR 8	1.0376727	.17508617	1.7773491	-.13669204-01	.82747251
ROW NR 9	.15190618	4.1998180	-.13669199-01	338.64938	-.22087148
ROW NR 10	1.6437172	.30021058	.82747250	-.22087176	-2.1826091

Table 3.20. The result after 561 steps with h = 1.66.

h = 1.80

RESULT AFTER 576 RUNGE-KUTTA STEP
****

COMPUTED S-MATRIX

COLUMN NR	1	2	3	4	5
ROW NR 1	379.98091	-43.489766	34617.913	719.40854	5.8825412
ROW NR 2	-43.489765	110.26708	3780.0997	1513.6442	1.2694711
ROW NR 3	34617.914	3780.0994	9167633.1	-102282.21	1155.6726
ROW NR 4	719.40681	1513.6438	-102282.17	298191.31	170.32601
ROW NR 5	5.8825406	1.2694720	1155.6725	170.32608	1.8472860
ROW NR 6	2.6922080	2.9442018	-151.32523	672.32744	.50616114
ROW NR 7	4.7726009	.95738623	921.39388	123.14120	.89803217
ROW NR 8	1.6583979	1.2152344	-86.528098	338.18651	.29731289
ROW NR 9	48.483422	35.211788	40289.789	-55.714382	5.2868513
ROW NR 10	1.7195753	3.2274677	-239.55274	674.49618	.38471612

COLUMN NR	6	7	8	9	10
ROW NR 1	2.6922168	4.7726009	1.6583980	48.483428	1.7195785
ROW NR 2	2.9442048	.95738622	1.2152344	35.211791	3.2274684
ROW NR 3	-151.32538	921.39388	-86.528098	40289.789	-239.55283
ROW NR 4	672.32819	123.14120	338.18651	-55.714482	674.49620
ROW NR 5	.50616249	.89803217	.29731289	5.2868516	.38471626
ROW NR 6	2.5403713	.27145937	1.0350101	.15305683	1.5215810
ROW NR 7	.27145938	2.3999531	.17460223	4.2000270	.27801275
ROW NR 8	1.0350100	.17460223	1.7760073	-.13090144-01	.76592171
ROW NR 9	.15305520	4.2000270	-.13090142-01	338.64915	-.19430832
ROW NR 10	1.5215824	.27801274	.76592170	-.19430852	-5.0059819

Table 3.21. The result after 576 steps with h = 1.80.

It has been argued that numerical instability could be detected by inspecting the quantity

$$\delta(k) = \max_{ij} \{|s_{ij}(k) - s_{ji}(k)| \ / \ |s_{ij}(k)|\}$$

and in table 3.22 we show this quantity for different steplengths. In table 3.22 we also show another quantity

$$\rho(k) = \max_{i,j} \{| \frac{ds_{ij}(k)}{dt} | \ / \ |s_{ij}(k)$$

In this example this quantity is more sensitive to numerical instability than δ.

Steplength	$\max_{ij}\{\lvert s_{ij}-s_{ji}\rvert/\lvert s_{ij}\rvert\}$	$\max_{ij}\{\lvert\frac{ds_{ij}}{dt}\rvert/\lvert s_{ij}\rvert\}$
1.62	$0.829\cdot10^{-5}$	$0.497\cdot10^{-7}$
1.64	$0.645\cdot10^{-5}$	$3.19\cdot10^{-7}$
1.66	$1.00\cdot10^{-5}$	9.57
1.80	$1.05\cdot10^{-5}$	5.56

Table 3.22. Two test quantities used to detect numerical instability.

4. CONCLUSIONS

The numerical integration of the Riccati equation is known to be a stable procedure. The numerical stability of the integration method used determines the upper limit of the steplength. The following rule of thumb can be given for the choice of steplength, h_{max}.

Let $\lambda_i(t)$ be the eigenvalues of

$$(A-BQ_2^{-1} B^T S(t))$$

and

$$\mu_k(t) = \lambda_i(t) + \lambda_j(t) \qquad i,j = 1,2,\ldots,n$$

The maximum steplength is then given by

$$h_{max} \leq \hat{h}$$

where $\hat{h}$ is given by

$$\hat{h} = \min_h \{h | h\, \mu_k(t) \in D_s, \quad \forall k,\ t\}$$

and D_s is the domain of numerical stability of the integration method used.

A necessary condition to satisfy this rule is that

$$\hat{h}\, \bar{\mu}_k \in D_s$$

where

$$\bar{\mu}_k = \bar{\lambda}_i + \bar{\lambda}_j \qquad i,j = 1,2,\ldots,n$$

and $\bar{\lambda}_i$ is an eigenvalue of

For some

$(A-BQ_2^{-1}\ B^TX)$

For some choices of Q_o may be a sufficient condition too. It has also been proven that an improper choice of Q_o may lead to a considerable reduction of the steplength. It has been proven as well that the choice $Q_o = 0$ may lead to a limiting value of $S(t)$ such that the closed loop system is unstable.

5. REFERENCES

[1] Mårtensson, K.: "On the Matrix Riccati Equation", Information Sciences 3, pp. 17-49, March 1971.

[2] Mårtensson, K.: "Linear Quadratic Control Pachage, Part I - The Continous Problem", Report 6802, Lund Institute of Technology, Division of Automatic Control.

[3] Gantmacher, F. R.: "The Theory of Matrices", Chelsea, New York, 1959.

[4] Gear, C. W.: "Numerical Initial Value Problems in Ordinary Differential Equations", Prentice Hall, Englewood Cliffs, New York, 1971.

[5] Kleinman, D. L.: "On an Iterative Technique for Riccati Equation Computations", IEEE Trans. on Automatic Control, Correspondence pp. 114-115, February 1968.