INT. J. CONTROL, 1978, VOL. 28, NO. 2, 295–306

Second-order convergent algorithms for the steady-state Riccati equation†

BRIAN D. O. ANDERSON‡

Iterative algorithms with second-order convergence properties are studied for finding the limiting solution of the steady-state Riccati equation of linear quadratic control. It is shown that known algorithms fit into one of two broad classes. Various connections are described between known algorithms.

1. Introduction

We consider the standard limiting Riccati equation of linear-quadratic control theory

$$PF+F'P-PGR^{-1}G'P+Q=0 \tag{1}$$

As is well known, this equation arises in considering the problem of minimizing a performance index $\int_0^\infty (x'Qx+u'Ru)\,dt$, $Q=Q'\geqslant 0$, $R=R'>0$ associated with the linear system $\dot{x}=Fx+Gu$. In case $[F, G]$ is stabilizable and $[F, Q^{1/2}]$ detectable, there exists a stabilizing solution $\bar{P}$ of (1), i.e. a solution such that $F-GR^{-1}G'\bar{P}$ has eigenvalues in $\mathrm{Re}\,[s]<0$. Furthermore, $\bar{P}\geqslant 0$. Our task is to discuss second-order methods for the solution of (1), i.e. methods which generate a sequence $P_k\rightarrow\bar{P}$ with $\|P_k-\bar{P}\|\leqslant K\|P_{k-1}-\bar{P}\|^2$ for some K and all k.

It turns out that there appear to be two abstractly different second-order methods, each of which has appeared in more than one form. We shall term these the doubling algorithm class, and the Newton–Raphson class. The nomenclature is slightly unfortunate, in that one concrete implementation of the doubling algorithm approach does involve Newton–Raphson iterations which are not concrete examples of the class we term Newton–Raphson. However, the names are appropriate in view of the nomenclature attaching to various concrete implementations already existing.

In § 2 we describe various concrete doubling algorithms, including one which at first glance does not appear to have this property, and in § 3 we describe two concrete implementations of the Newton–Raphson algorithm class which have hitherto not apparently been recognized as connected.

There are other Riccati equations than (1) which arise in system theory. The material of the following sections applies to these *mutatis mutandis*.

2. Doubling algorithms

To understand doubling algorithms, it is easier to begin with the discrete-time limiting Riccati equation, viz.

$$P=F'PF+Q-FPG[R+G'PG]^{-1}G'PF' \tag{2}$$

Received 23 February 1977 ; revision received 28 June 1977.
† Work supported by the Australian Research Grants Committee.
‡ Department of Electrical Engineering, University of Newcastle, New South Wales, 2308, Australia.

with $Q=Q'\geqslant 0$, $R=R'>0$. It is known that with appropriate conditions (viz. $[F, G]$ stabilizable and $[F, Q^{1/2}]$ detectable), one can obtain the stabilizing solution $\bar{P}$ as $\lim_{k\to\infty} P_k$ where

$$\left.\begin{aligned} P_{k+1} &= F'P_kF+Q-FP_kG[R+G'P_kG]^{-1}G'P_kF' \\ P_0 &= 0 \end{aligned}\right\} \tag{3}$$

In doubling, one seeks to compute not every P_k, but rather the sequence $P_0, P_1, P_2, P_4, P_8, \ldots, P_{2^k}, \ldots$. The basic idea of doubling algorithms (Davison and Maki 1973, Friedlander *et al.* 1976) is to provide equations for updating from $P_{2^{k-1}}$ to P_{2^k}. The update equations are, with $\gamma_k=P_{2^k}$,

$$\alpha_{k+1}=\alpha_k(I+\beta_k\gamma_k)^{-1}\alpha_k' \tag{4 a}$$

$$\beta_{k+1}=\beta_k+\alpha_k(I+\beta_k\gamma_k)^{-1}\beta_k\alpha_k' \tag{4 b}$$

$$\gamma_{k+1}=\gamma_k+\alpha_k'\gamma_k(I+\beta_k\gamma_k)^{-1}\alpha_k \tag{4 c}$$

with

$$\alpha_0=F, \quad \beta_0=GR^{-1}G', \quad \gamma_0=Q \tag{5}$$

[In case β_k, γ_k are non-singular, more symmetric forms of (4) are possible; also, in the usual situation where for arbitrary $P_0\geqslant 0$ in (3), $P_k\to\bar{P}$, we can replace γ_0 by an arbitrary non-negative matrix.] One can, of course, provide a brute force proof that the γ_k are as claimed. However, it is instructive to consider how these equations come about. Here, we shall briefly comment on three ways of thinking about (4).

The first way depends on two observations. The first observation is that the solution of a Riccati equation can be obtained in terms of the solution of a linear equation (Hamiltonian equation) of twice the dimension. The second observation is that any linear equation of the form $X_{k+1}=\Phi X_k$, X_0 given, has an immediately obtainable doubling solution; because $\Phi^{2^k}=\Phi^{2^{k-1}}.\Phi^{2^{k-1}}$ (time-invariance is of course crucial here) a doubling algorithm is available for the transition matrix of the equation. Combining these two observations, one sees immediately that, in principle, the derivation of a Riccati equation doubling algorithm should be possible. In practice, it is straightforward; the three quantities α_k, β_k, γ_k together determine the value of Φ^{2^k}, and conversely, where Φ is the system matrix of the Hamiltonian equation associated with (2). The details are given in Appendix 1, since such a derivation does not appear to exist in the literature.

The second way of thinking about the doubling algorithm is by scattering theory concepts, where the idea is an old one (van de Hulst 1963, Hansen 1969, 1971).

Imagine a block of infinite extent in two dimensions and bounded by two parallel planes in the third orthogonal direction. Waves are incident on both faces of the block, are transmitted through the block, and are reflected or transmitted outwards from each face. The scattering matrix is a 2×2 matrix which relates the inward or incident waves at the faces to the outward waves at the faces. It is clear that if two blocks are brought into contact, a single scattering matrix should describe the aggregated block. This scattering

matrix can be computed from the scattering matrices of the individual blocks by an operation known as the star product.

Now imagine that blocks are of uniform material with a block of unit thickness having scattering matrix S_1. If a block of twice unit thickness has scattering matrix S_2, then $S_2 = S_1 * S_1$, on regarding the block as an aggregate of two unit thickness blocks. More generally, a block of thickness j units is an aggregate of a block of thickness $(j-1)$ units and a block of unit thickness, so that $S_j = S_{j-1} * S_1$.

Now let us permit the entries of S to be themselves matrices. It turns out (Friedlander *et al.* 1976) that if the entries of S_1 are identified with certain expressions involving F, G, Q, R and P_1 in (3), then the entries of S_2 involve P_2 and other quantities, the entries of S_3 involve P_3, etc., because the $*$ operation yielding S_j from S_{j-1} and S_1 in effect involves the recursion (3).

However, we observe from the physical aspects of the situation that $S_4 = S_2 * S_2$, $S_8 = S_4 * S_4$, and so on. Thus the $*$ formulae may be used to generate at step k the quantity S_{2^k}. As just observed S_{2^k} contains P_{2^k}. In fact S_{2^k} is given in effect by α_k, β_k and γ_k, and the formulae (4) mimic the star multiplication which yields $S_{2^{k+1}}$ from S_{2^k} (Friedlander *et al.* 1976).

The fact that these formulae could be applied outside the scattering theory constraint to control and filtering problems is one of the novel contributions of Friedlander *et al.* (1976).

A third way of thinking about (4) depends on relating Riccati equation solutions with non-zero initial conditions to solutions with zero initial conditions, and coupling such relations with the underlying time-invariance, see Bierman and Sidhu (1976) for details. (It has been known for some time that the solution of a continuous-time Riccati equation for arbitrary initial condition has a nice representation in terms of that boundary condition and various other computable quantities; this idea has been developed in a systems context by Lainiotis (1975). It is no real surprise that the long-known result for continuous-time equations should carry over to discrete-time.) More precisely, we have the following general property. Consider (3) with initial condition Π. Denote the corresponding solution by $P_k(\Pi)$. There exist two other related equations for quantities $\Theta_k(\Pi)$ and $\Psi_k(\Pi)$. Together the equations for P_k, Θ_k and Ψ_k have the following crucial property: $P_k(\Pi_1)$, $\Theta_k(\Pi_1)$ and $\Psi_k(\Pi_1)$ for some $\Pi = \Pi_1$ can be computed simply from $P_k(\Pi)$, $\Theta_k(\Pi)$, $\Psi_k(\Pi)$ and Π_1; in particular, iteration of the P_k, Θ_k and Ψ_k equations with the new initial condition is not required.

This leads to a doubling algorithm of Bierman and Sidhu (1976) in the following way. Denote $P_k(\Pi = 0)$, $\Theta_k(\Pi = 0)$ and $\Psi_k(\Pi = 0)$ by P_{k0}, Θ_{k0}, Ψ_{k0}.

(*a*) Obtain P_{10}, Θ_{10}, Ψ_{10}.

(*b*) Obtain $P_1(\Pi = P_{10})$, $\Theta_1(\Pi = P_{10})$, $\Psi_1(\Pi = P_{10})$. By the time-invariance of the P equation, $P_1(\Pi = P_1) = P_2(\Pi = 0) = P_{20}$. The structure of the Θ_k and Ψ_k equations also ensures $\Theta_1(\Pi = P_1) = \Theta_{20}$ and $\Psi_1(\Pi = P_1) = \Psi_{20}$.

(*c*) Using P_{20}, Θ_{20} and Ψ_{20} we obtain $P_2(\Pi = P_{20})$, $\Theta_2(\Pi = P_{20})$ $\Psi_2(\Pi = P_{20})$. By time-invariance, these quantities are P_{40}, Θ_{40} and Ψ_{40}.

The procedures continues, and in particular the interval doubling continues. Equations (4) are the update equations.

Now let us turn to continuous-time doubling algorithms. Equations can be found Lainiotis (1975) and Ljung *et al.* (1976). An equivalent form of the equations is provided by (4) in which now α_0, β_0, γ_0 have connotations quite different to those possessed in the discrete-time case, as we now explain.

Associated with (1) is the Hamiltonian system.

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} = \begin{bmatrix} F & -GR^{-1}G' \\ -Q & -F' \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} \quad \begin{bmatrix} X(0) \\ Y(0) \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix} \tag{6}$$

with the property that $Y(t)X^{-1}(t) = P(t)$ for $t < 0$, $P(\cdot)$ being the transient solution of $-\dot{P} = PF + F'P - PGR^{-1}G'P + Q$ and $P(t) \to \bar{P}$ as $t \to -\infty$.

Equivalently, one has

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} = \begin{bmatrix} -F & GR^{-1}G' \\ Q & F' \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} \quad \begin{bmatrix} X(0) \\ Y(0) \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix} \tag{7}$$

for $t \geqslant 0$, with $Y(t)X^{-1}(t) = P(t)$, where $P(t)$ is now the solution of

$$\dot{P} = PF + F'P - PGR^{-1}G'P + Q$$

with $P(t) \to \bar{P}$.

Denote the transition matrix of (7) by

$$\exp(\mathscr{H}t) = \Phi(t, 0) = \begin{bmatrix} \Phi_{11}(t, 0) & \Phi_{12}(t, 0) \\ \Phi_{21}(t, 0) & \Phi_{22}(t, 0) \end{bmatrix} \tag{8}$$

with $\mathscr{H}$ the system matrix in (7). Then, see Lainiotis (1975) and Ljung *et al.* (1976), the initializing quantities for the doubling algorithm are, for arbitrary $\Delta = 0$,

$$\alpha_0 = \Phi_{11}^{-1}(\Delta, 0) = \Phi_{11}(0, \Delta) \tag{9 a}$$

$$\beta_0 = \Phi_{11}^{-1}(\Delta, 0)\Phi_{12}(\Delta, 0) \tag{9 b}$$

$$\gamma_0 = \Phi_{21}(\Delta, 0)\Phi_{11}^{-1}(\Gamma, 0) \tag{9 c}$$

[Usually, any $\gamma_0 = \gamma_0' \geqslant 0$ may be used in lieu of (9 c).]

Why should the recursions equations for α_k, etc. be the same as for the discrete-time case? The answer is that because $\mathscr{H}$ is a (continuous-time) Hamiltonian matrix, $\exp(\mathscr{H}\Delta)$ for arbitrary real Δ is a symplectic, or discrete-time Hamiltonian matrix, i.e. there exists a discrete-time linear-quadratic problem for which the system matrix of the Hamiltonian equations is precisely $\exp(\mathscr{H}\Delta)$. (This fact can be checked via straightforward calculation set out in Appendix 2.) This means that the quantities α_k, β_k, γ_k define

$$[\exp(\mathscr{H}\Delta)]^{2^k} = \Phi(\Delta 2^k, 0)$$

and conversely ; in fact,

$$\gamma_k = \Phi_{21}(\Delta 2^k, 0)\Phi_{11}^{-1}(\Delta 2^k, 0)$$

Since $P(t) = \Phi_{21}(t, 0)\Phi_{11}^{-1}(t, 0)$, $\lim_{k\to\infty} \gamma_k = \bar{P}$. We see then that the *continuous-time doubling algorithm* of Lainiotis (1975) and Ljung *et al.* (1976) *is effectively a two-stage procedure ; the first stage requires the setting up of a discrete-time problem, the limiting solution of which is the same as that of the continuous-time problem ; the second stage comprises use of the discrete-time doubling algorithm.*

The scattering ideas which yield discrete-time doubling formulae carry over to continuous-time (Ljung *et al.* 1976). Also, the derivation of doubling formulas based on comparing Riccati equations for different initial conditions carries over, see Lainiotis (1975). In both cases, (4) and (9) are obtained with the quantities α_k, β_k and γ_k possessing the same significance as in the preceding paragraph.

Once the observations linking continuous-time doubling to discrete-time doubling is made, one might ask whether there is an easier way to set up a discrete-time problem than by computing $\exp(\mathscr{H}\Delta)$ or the equivalent. The answer is yes. In Anderson and Moore (1971) and Hitz and Anderson (1972) there is presented a procedure for passing from a continuous-time to a discrete-time linear quadratic problem different to that associated with $\exp(\mathscr{H}\Delta)$ where matrix exponentials do not have to be computed ; rather, certain rational functions of the matrices of the continuous-time problem yield the matrices of the discrete-time problem, while one may arrange for the limiting solution of the Riccati equations in the discrete and continuous problems to be the same. Almost certainly, use of this transformation, followed by use of (4) and (5), would be easier than use of (9) followed by (4). One minor difference between the two algorithms might however in some cases lead one to prefer use of (9) and (4). The iterates γ_k in this latter case provide the value of the solutions of the Riccati differential equations at $\Delta, 2\Delta, 4\Delta, \ldots$, whereas this information does not come from the other algorithm ; iterates of a discrete-time Riccati difference equation are provided, but these are not easily related to values of the differential equation solution.

There is however another algorithm for continuous-time problems again which is abstractly a doubling algorithm which may be quickest of all. This is an algorithm due to Roberts (1971), and investigated in detail by Denman and Beavers (1976).

This algorithm does not appear at first sight to be a doubling algorithm : it goes as follows. Let

$$\mathscr{H} = \begin{bmatrix} -F & GR^{-1}G' \\ Q & F' \end{bmatrix} \tag{10}$$

and form sign $\mathscr{H}$ according to the following iterative scheme : $\mathscr{H}_0 = \mathscr{H}$; $\mathscr{H}_{k+1} = \frac{1}{2}(\mathscr{H}_k + \mathscr{H}_k^{-1})$; $\operatorname{sign}\mathscr{H} = \lim_{k\to\infty} \mathscr{H}_k$. Then with

$$F = \operatorname{sign}\mathscr{H} + \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix} = \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \tag{11}$$

one has

$$\bar{P} = F_{21} F_{11}^{-1} \tag{12}$$

The sign of a matrix is a generalization of the sign of a number; if $\mathscr{H}$ is diagonal, say $\mathscr{H} = T\Lambda T^{-1}$ with $\Lambda = \text{diag}\,(\lambda_i)$, then sign $\mathscr{H} = T$ diag $(\mu_i)T^{-1}$, where $\mu_i = \text{sign}\,(\text{Re}\,\lambda_i)$, so long as Re $\lambda_i \neq 0$ for any μ_i. The algorithm for constructing sign $\mathscr{H}$ is actually a Newton–Raphson algorithm, but the overall algorithm for obtaining $\bar{P}$ is not similar to those of the next section, but is rather a doubling algorithm, as we now show.

Define

$$T_k = (\mathscr{H}_k + I)(\mathscr{H}_k - I)^{-1} \tag{13}$$

Then one can check that the recursion for $\mathscr{H}_k$ implies

$$T_{k+1} = T_k^2 \tag{14}$$

Therefore the sign calculation on $\mathscr{H}_k$ is a doubling algorithm for T_k. But actually a little more is true. Because $\mathscr{H}_0$ is a (continuous-time) Hamiltonian matrix, T_0 is a symplectic or discrete-time Hamiltonian matrix, see Appendix 2, and itself is therefore associated with a discrete-time linear-quadratic problem.

The eigenvalues of T_0 are bilinear transforms $\lambda \to (\lambda+1)(\lambda-1)^{-1}$ of those of $\mathscr{H}$, while the same relation stands between the eigenvalues of the continuous-time and discrete-time Hamiltonian matrices in the transformation of (Anderson and Moore (1971) and Hitz and Anderson (1972). Thus we might view the sign algorithm as implicitly replacing a continuous-time problem by a discrete-time one in like manner to these authors, and solving the discrete-time problem by the doubling algorithm. However, it is never necessary to obtain explicitly the matrices defining the discrete-time problem.

Note that the implementation of the recursion $\mathscr{H}_{k+1} = \frac{1}{2}(\mathscr{H}_k + \mathscr{H}_k^{-1})$ can in general proceed with an updating of three matrix equations of dimension equal to F. In fact and as shown in Appendix 3, if

$$\mathscr{H}_k = \begin{bmatrix} A_k & B_k \\ C_k & D_k \end{bmatrix} \tag{15}$$

one has $D_k = -A_k'$, $B_k = B_k'$, $C_k = C_k'$ (since the Hamiltonian property of $\mathscr{H}_0$ propagates, as is easily checked), and if A_k is non-singular,

$$A_{k+1} = \tfrac{1}{2}\{A_k + [A_k + B_k(A_k')^{-1}C_k]^{-1}\} \tag{16 a}$$

$$B_{k+1} = \tfrac{1}{2}\{B_k + [A_k + B_k(A_k')^{-1}C_k]^{-1}B_k(A_k')^{-1}\} \tag{16 b}$$

$$C_{k+1} = \tfrac{1}{2}\{C_k + (A_k')^{-1}C_k[A_k + B_k(A_k')^{-1}C_k]^{-1}\} \tag{16 c}$$

$$A_0 = -F, \quad B_0 = GR^{-1}G', \quad C_0 = Q \tag{17}$$

with

$$\bar{P} = \lim_{k \to \infty} C_k(A_k + I)^{-1} \tag{18}$$

Square-root versions can be obtained also. Suppose that $B_k = \Gamma_k \Gamma_k'$, $C_k = \Lambda_k \Lambda_k'$ and $X_k = \Lambda_k A_k^{-1} \Gamma_k$.

Then assuming all inverses exist, (16) are equivalent to

$$A_{k+1}=\tfrac{1}{2}[A_k+A_k^{-1}\,\Gamma_k(I+X_k'\,X_k)^{-1}\Gamma_k^{-1}] \tag{19 a}$$

$$\Gamma_{k+1}\Gamma_{k+1}'=\tfrac{1}{2}[\Gamma_k\Gamma_k'+A_k^{-1}\,\Gamma_k(I+X_k'\,X_k)^{-1}\Gamma_k'(A_k')^{-1}] \tag{19 b}$$

$$\Lambda_{k+1}'\,\Lambda_{k+1}=\tfrac{1}{2}[\Lambda_k'\,\Lambda_k+(A_k')^{-1}\Lambda_k'(I+X_kX_k')^{-1}\Lambda_kA_k^{-1}] \tag{19 c}$$

for which square-root formulae follows as for the discrete-time doubling formulae of Bierman and Sidhu (1976). It is evident from these formulae that rank Γ_k and rank Λ_k are monotone increasing ; in fact, under conditions of complete controllability and observability of $[F, G]$ and $[F, Q^{1/2}]$, Γ_k and Λ_k will be non-singular after dim A_k iterations.

There are, in summary, three formally different doubling algorithms for continuous-time systems : that based on calculating exp $(\mathscr{H}\Delta)$ and using a discrete-time doubling algorithm, that based on setting up a discrete-time linear-quadratic problem by rational transformation and again using a discrete-time doubling algorithm, and that based on computation of the sign of the Hamiltonian. The ordering of this list probably reflects an increasing degree of simplicity.

We also comment why the doubling algorithms are of order 2. The sign algorithm has this property, since the scheme for computing sign $\mathscr{H}$ involves a Newton–Raphson iteration, which has order 2 convergence (Luenberger 1973). The scheme based on using (4) and (9) is most easily checked to have order 2 convergence by using the following formula for the difference $\Sigma(t)$ between the stabilizing steady state solution of (1) and the solution of $\dot{P}=PF+F'P-PGR^{-1}G'P+Q$ run forward in time with $P(0)=0$:

$$\Sigma(t)=\exp(\bar{F}'t)\left[I+\bar{P}\int_0^t \exp(\bar{F}\tau)GR^{-1}G'\exp(\bar{F}'\tau)\,d\tau\right]^{-1}\bar{P} \times\exp(\bar{F}t) \tag{20}$$

(This formula is not difficult to derive from the defining equations. It is also derived in Anderson Moore (1971, § 15.3). The formula shows that $\Sigma(t)\to 0$ exponentially fast, i.e. $\|\Sigma(t)\|=0\ [\exp(-\alpha t)]$ for some $\alpha>0$, whence $\|\Sigma(2t)\|<K\|\Sigma(t)\|^2$ for some K and all $t\geqslant T$ for suitably large T.

We remark finally that Davison and Maki (1973) went a good way towards a continuous-time doubling algorithm of the type described in this section. These authors proposed computing $C=e^{\mathscr{H}\Delta}$ for some small Δ, and then evaluating $C^2, C^4, C^8, \ldots$. From a partition of these powers of C, $P(2\Delta)$, $P(4\Delta)$, $P(8\Delta)$, ... can be obtained. The method is almost certainly less computationally stable than that based on use of (4).

3. Newton–Raphson algorithms

In this section we shall restrict attention to continuous-time algorithms, the corresponding discrete-time algorithms being easy to find. We study in particular the algorithms due to Kleinman (1968), Hewer (1971), Sandell (1971), Vit (1972), Wilson (1969) and Tunnicliffe-Wilson (1972).

The Kleinman algorithm for (1) runs as follows. Let K_0 be any gain matrix such that Re $\lambda_i(F-GK_0')<0$ for all i. Define P_{i+1} as the solution of

$$P_{i+1}(F-GK_i')+(F-GK_i')'P_{i+1}=-K_iRK_i'-Q \tag{21}$$

$$K_{i+1}=P_{i+1}GR^{-1} \tag{22}$$

Then $P_i \downarrow \bar{P}$ as $i\to\infty$ provided the usual conditions for existence of $\bar{P}$ mentioned in § 1 hold. This algorithm can be viewed as resulting from use of a Newton–Raphson algorithm for solving (1). Provided the initial condition is chosen appropriately (K_0 is a stabilizing gain), convergence is guaranteed.

Tunnicliffe-Wilson's approach arises from trying to solve via a Newton–Raphson approach an equation expressing a spectral factorization property of the optimal regulator. Let $\bar{K}$ be the gain of the optimal regulator. Then one has (Anderson and Moore 1971)

$$R+G'(sI-F')^{-1}Q(sI-F)^{-1}G=[I+G'(-sI-F')^{-1}\bar{K}] \times R[I+\bar{K}'(sI-F)^{-1}G] \tag{23}$$

Given the quantities on the left side of (23), the requirement that

$$\text{Re } \lambda_i(F-G\bar{K}')<0$$

i.e. the closed-loop regulator by asymptotically stable, uniquely determines the spectral factor $I+\bar{K}'(sI-F)^{-1}G$, from which $\bar{K}$ can be found. (The matrix $\bar{P}$ can then be found by solving (21) with K_i replaced by $\bar{K}$.)

We shall assume $[F, G]$ is completely controllable rather than merely completely stabilizable, and translate (23) into an equation involving polynomial matrices. It is this equation which Tunnicliffe-Wilson tackles with a Newton–Raphson algorithm. We shall review the algorithm and then show it is effectively the same as that of Kleinman.

To rewrite (23) using polynomial matrices is easy, using matrix fraction descriptions. It is a standard result (Wolovich 1974) that for a controllable pair $[F, G]$, the set of transfer function matrices $L'(sI-F)^{-1}G$, L variable, is obtainable as the set of MFD's $B(s)A^{-1}(s)$, where $A(s)$ is a column proper polynomial matrix determined entirely by F, G, while $B(s)$ depends on L. Accordingly, after setting $Q=DD'$ for some D, we may write (23) as

$$R+A'^{-1}(-s)B'(-s)B(s)A^{-1}(s)=[I+A'^{-1}(-s)\bar{C}'(-s)] \times R[I+\bar{C}(s)A^{-1}(s)]$$

or, writing X_* for $X'(-s)$,

$$A_*RA+B_*B=[A_*+\bar{C}_*]R[A+\bar{C}] \tag{24}$$

This equation may be used to define the solution of the control problem, for when quantities on the left are known, $\bar{C}$ is uniquely defined by (24) with the simultaneous requirement that det $[A+\bar{C}]$ have all zeros in Re $[s]<0$.

Let C_i be the ith approximation to $\bar{C}$. Then a Newton–Raphson algorithm yields, as easy calculations show,

$$B_*B=-C_{i*}RC_i-C_{i+1*}R(A+C_i)-(A_*+C_{i*})RC_{i+1} \tag{25}$$

or

$$-(A_*+C_{i*})^{-1}C_{i+1*}-C_{i+1}(A+C_i)^{-1}=(A_*+C_{i*})^{-1}\times(C_{i*}RC_i+B_*B)(A+C_i)^{-1} \quad (26)$$

One may regard (25) as defining C_{i+1} as the solution of a polynomial equation† $M_*X+X_*M=N=N_*$ where $M_*=(A_*+C_{i*})R$ and $N=-B_*B-C_{i*}RC_i$ are known; if M and M_* are co-prime in an appropriate sense, this determines X uniquely. Alternatively, one can regard (26) as defining C_{i+1} from the fact that $-C_{i+1}(A+C_i)^{-1}$ is the stable part of the right side of (26). For it can be shown (Tunnicliffe-Wilson 1972) that if $\det[A-C_0]$ has all zeros in $\mathrm{Re}[s]<0$, the same is true of $\det[A+C_i]$ for all i.

To relate (21) and (26), define $\mathscr{C}_i(s)$ by $K_i'(sI-F)^{-1}G=\mathscr{C}_i(s)A^{-1}(s)$ and F_i by $F-GK_i'$. Then because

$$K_{i+1}'(sI-F_i)^{-1}G=K_{i+1}'(sI-F)^{-1}G[I+K_i'(sI-F)^{-1}G]^{-1}$$

we obtain

$$K_{i+1}'(sI-F_i)^{-1}G=\mathscr{C}_{i+1}(s)\{A(s)+\mathscr{C}_i(s)\}^{-1}$$

Now (21) and (22) together yield

$$-G'(-sI-F_i')^{-1}K_{i+1}-K_{i+1}'(sI-F_i)^{-1}G=G'(-sI-F_i)^{-1}\times[K_iRK_i'+DD'](sI-F_i)^{-1}G$$

or

$$-(A_*+\mathscr{C}_{i*})^{-1}\mathscr{C}_{i+1}-\mathscr{C}_{i+1}(A+\mathscr{C}_i)^{-1}=(A_*+\mathscr{C}_{i*})^{-1}\times[\mathscr{C}_{i*}R\mathscr{C}_i+B_*B](A+\mathscr{C}_i) \quad (27)$$

Comparison of (26) and (27) shows that if $C_0=\mathscr{C}_0$, then $C_i=\mathscr{C}_i$ for all i, thus the two algorithms are seen to be, at least abstractly, the same.

On the surface, use of the Kleinman algorithm appears nowhere near as appealing as the use of a doubling algorithm for finding the quantities $\bar{P}$ and $\bar{K}$, since the solving of a linear matrix equation can be a substantial task, and this is required at each iteration of the Kleinman algorithm. The fact however that the algorithm is equivalent to the use of (25) changes the picture. In case the problem involves a scalar control, (25) becomes especially attractive, since the coefficients of C_{i+1} are easily obtained by inversion of a matrix of dimension equal to the dimension of the state vector. Matters are however more complicated in the case of vector controls.

Finally, one might well ask whether there is necessarily a connection between the doubling algorithms of § 2 and the Newton–Raphson algorithms of this section. It would appear not. If one examines the discrete-time version of the Kleinman algorithm (Hewer 1971), quite different iterative schemes are obtained with the iterates possessing different meanings. In the former case, one must update three matrix equations of dimension equal to the state vector; in the latter, one must solve a linear matrix equation of

† Such equations are studied by Barnett (1971) and Anderson and Bitmead (1977).

the same dimension as the state vector at each iteration. Again, in the former case, one obtains a subsequence of the sequence of iterates of the transient discrete-time Riccati equation, which, if started with zero initial condition, converges monotonically upwards to the limiting solution; by contrast, the discrete-time Kleinman algorithm obtains a monotone decreasing sequence. Despite these differences, it would seem rash to claim that the two classes of algorithm must be abstractly distinct.

4. Concluding remarks

We have identified two classes of second-order algorithms for solving the steady-state Riccati equation of linear-quadratic control. In the process we have attempted to motivate three distinct continuous-time doubling algorithms as flowing from the obvious doubling algorithm for obtaining the powers A^{2^k} from a square matrix A. We have also shown that the Hamiltonian sign algorithm is one of two true integration-free doubling algorithms, and would suggest that it is not entirely appropriate to use the term 'integration-free' for a 'recursive' algorithm where integration or computation of a matrix exponential is required to obtain the initial iterates.

As far as Newton–Raphson based algorithms are concerned, we have shown that the algorithms of Kleinman and Tunnicliffe-Wilson essentially execute the same calculations, the first in terms of the matrices of a state variable realization, the second in terms of the polynomial matrices of a matrix fraction description, with in both cases the same transfer function being involved.

Appendix 1

Doubling algorithm derivation from the Hamiltonian equation

From the equation

$$P_{k+1} = F'P_kF + Q - FP_kG[R + G'P_kG]^{-1}G'P_kF'$$

we have by rearrangement, and assuming that F is non-singular

$$\begin{aligned} P_{k+1} &= \{QF^{-1} + [F' + QF^{-1}GR^{-1}G']P_k\}\{F^{-1} + F^{-1}GR^{-1}G'P_k\}^{-1} \\ &= [C + DP_k][A + BP_k]^{-1} \end{aligned} \tag{A 1}$$

with obvious definitions of A, B, C, D. Consider also the linear equation

$$\begin{bmatrix} Y_{k+1} \\ Z_{k+1} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} Y_k \\ Z_k \end{bmatrix} \tag{A 2}$$

Equation (A 1) shows that if Y_k, Z_k are such that $Z_kY_k^{-1} = P_k$, then one has $Z_{k+1}Y_{k+1}^{-1} = P_{k+1}$ and generally $Z_mY_m^{-1} = P_m$.

Let Φ be the transition matrix in (A 2). Then a doubling procedure leads to Φ^{2^k} and

$$\begin{bmatrix} Y_{2^k} \\ Z_{2^k} \end{bmatrix} = \Phi^{2^k} \begin{bmatrix} Y_0 \\ Z_0 \end{bmatrix} = \Phi^{2^k} \begin{bmatrix} I \\ 0 \end{bmatrix}$$

the value of Y_0, Z_0 being chosen to ensure that $P_0 = 0$.

Now one can verify by an easy induction argument that

$$\Phi^{2^k} = \begin{bmatrix} \alpha_k^{-1} & \alpha_k^{-1}\beta_k \\ \gamma_k\alpha_k^{-1} & \alpha_k' + \gamma_k\alpha_k^{-1}\beta_k \end{bmatrix}$$

with α_k, β_k, γ_k as in (4) and (5), so then $Z_{2^k}Y_{2^k}^{-1} = \gamma_k = P_{2^k}$. The case of F singular can be obtained by a technical limiting argument.

Appendix 2

Symplectic matrices derivable from a continuous-time Hamiltonian matrix

Suppose that

$$\mathscr{H} = \begin{bmatrix} F & -GR^{-1}G' \\ -Q & -F' \end{bmatrix}$$

We will show that $\exp(\mathscr{H}\Delta)$ for arbitrary $\Delta > 0$ and $(\mathscr{H}+I)(\mathscr{H}-I)^{-1}$ are symplectic. [A matrix Z is termed sympectic if $Z'JZ = J$, where

$$J = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix}$$

and symplectic matrices arise in discrete-time linear-quadratic optimization problems in the way that Hamiltonian matrices arise in continuous-time linear-quadratic optimization problems.]

Using the fact that $J\mathscr{H} = -\mathscr{H}'J$ and $J^{-1} = -J$ we have

$$\begin{aligned} J^{-1}\exp(\mathscr{H}'\Delta)J\exp(\mathscr{H}\Delta) &= \exp(J^{-1}\mathscr{H}'J\Delta)\exp(\mathscr{H}\Delta) \\ &= \exp(-J\mathscr{H}'J\Delta)\exp(\mathscr{H}\Delta) \\ &= \exp(J^2\mathscr{H}\Delta)\exp(\mathscr{H}\Delta) \\ &= \exp(-\mathscr{H}\Delta)\exp(\mathscr{H}\Delta) \\ &= I \end{aligned}$$

Also

$$\begin{aligned} (\mathscr{H}'+I)J(\mathscr{H}+I) &= \mathscr{H}'J\mathscr{H} + \mathscr{H}'J + J\mathscr{H} + J \\ &= \mathscr{H}'J\mathscr{H} - J\mathscr{H} - \mathscr{H}'J + J \\ &= (\mathscr{H}'-I)J(\mathscr{H}-I) \end{aligned}$$

Therefore

$$(\mathscr{H}'-I)^{-1}(\mathscr{H}'+I)J(\mathscr{H}+I)(\mathscr{H}-I)^{-1} = J$$

Appendix 3

The iteration for the sign of a Hamiltonian matrix

$\mathscr{H}$ is Hamiltonian if and only if $J\mathscr{H} = -\mathscr{H}'J$, where

$$J = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix}$$

It is trivial to observe that if $\mathscr{H}$ is Hamiltonian, so must $\mathscr{H}^{-1}$ be, and then $\frac{1}{2}(\mathscr{H}+\mathscr{H}^{-1})$. Therefore, the Hamiltonian nature of $\mathscr{H}_0$ propagates in the sign algorithm $\mathscr{H}_{k+1}=\frac{1}{2}(\mathscr{H}_k+\mathscr{H}_k^{-1})$. Also, with

$$\mathscr{H}_k=\begin{bmatrix} A_k & B_k \\ C_k & -A_k' \end{bmatrix}$$

and with A_k non-singular, the matrix inversion lemma gives

$$\mathscr{H}_k^{-1}=\begin{bmatrix} [A_k+B_k(A_k')^{-1}C_k]^{-1} & [A_k+B_k(A_k')^{-1}C_k]^{-1}B_k(A_k')^{-1} \\ (A_k')^{-1}C_k[A_k+B_k(A_k')^{-1}C_k]^{-1} & -[A_k'+C_k'(A_k)^{-1}B_k']^{-1} \end{bmatrix}$$

The formulae (16) are immediate.

References

Anderson, B. D. O., and Bitmead, R., 1977, *Int. J. Control*, **26**, 235.

Anderson, B. D. O., and Moore, J. B., 1971, *Linear Optimal Control* (New Jersey : Prentice-Hall).

Barnett, S., 1971, *Matrices in Control Theory* (London : Van Nostrand Reinhold Company).

Bierman, G. J., and Sidhu, G. S., 1976, Tech. Memo. 33–799, Jet Propulsion Laboratory, Pasadena, California.

Davison, E. J., and Maki, M. C., 1973, *I.E.E.E. Trans. autom. Control*, **18**, 71.

Denman, E. D., and Beavers, A. N., Jr., 1976, *Appl. Math. Comp.*, **2**, 63.

Friedlander, B., Kailath, T., and Ljung, L., 1976, *J. Franklin Inst.*, **301**, 71.

Hansen, J. E., 1969, *Astrophys. J.*, **155**, 565 ; 1971, *J. atmos. Sci.*, **28**, 120.

Hewer, G. A., 1971, *I.E.E.E. Trans. autom. Control*, **16**, 382.

Hitz, K. L., and Anderson, B. D. O., 1972, *Proc. Instn elect. Engrs*, **119**, 1402.

Van de Hulst, H. C., 1963, ' A New Look at Multiple Scattering ', Goddard Institute for Space Studies, N.A.S.A. New York.

Kleinman, D. L., 1968, *I.E.E.E. Trans. autom. Control*, **13**, 114.

Lainiotis, D. G., 1975, *Proc. I.E.E.E. Dec. Control Conf.*, p. 736.

Luenberger, D. G., 1973, *Introduction to Linear and Nonlinear Programming* (London : Addison–Wesley Publishing Co.).

Ljung, L., Kailath, T., and Friedlander, B., 1976, *Proc. I.E.E.E.*, **64**, 131.

Roberts, J. D., 1971, CUED/B–Control/TR–13 Report, Cambridge University.

Sandell, N. R., 1974, *I.E.E.E. Trans. autom. Control*, **19**, 254.

Tunnicliffe-Wilson, G., 1972, *SIAM J. appl. Math.*, **23**, 420.

Vit, K., 1972, *I.E.E.E. Trans. autom. Control*, **17**, 258.

Wilson, G., 1969, *SIAM J. numer. Analysis*, **6**, 1.

Wolovich, W. A., 1974, *Linear Multivariable Systems* (New York : Springer–Verlag).