Computing the Positive Stabilizing Solution to Algebraic Riccati Equations With an Indefinite Quadratic Term via a Recursive Method

Alexander Lanzon, *Senior Member, IEEE*, Yantao Feng, *Student Member, IEEE*, Brian D. O. Anderson, *Life Fellow, IEEE*, and Michael Rotkowitz, *Member, IEEE*

***Abstract*—An iterative algorithm to solve Algebraic Riccati Equations with an indefinite quadratic term is proposed. The global convergence and local quadratic rate of convergence of the algorithm are guaranteed and a proof is given. Numerical examples are also provided to demonstrate the superior effectiveness of the proposed algorithm when compared with methods based on finding stable invariant subspaces of Hamiltonian matrices. A game theoretic interpretation of the algorithm is also provided.**

***Index Terms*—Algebraic Riccati equation (ARE), H_∞ Riccati equations, indefinite quadratic term, iterative algorithms.**

I. Introduction

CONSIDER the following Algebraic Riccati Equation (ARE) in the variable P:

$$0 = A^T P + PA + PRP + Q \tag{1}$$

where A, Q, R are real $n \times n$ matrices with Q and R symmetric. Here, $(\cdot)^T$ denotes the transpose of $(\cdot)$. Associated with this Riccati equation is a $2n \times 2n$ Hamiltonian matrix

$$H := \begin{pmatrix} A & R \\ -Q & -A^T \end{pmatrix}.$$

Generally speaking, existing methods to solve AREs can be divided into two categories:

1) Direct: solutions of ARE (1) can be constructed via computation of an n-dimensional invariant subspace of Hamiltonian matrix H (for example using the Schur algorithm in [5]).
2) Iterative: a sequence of matrices which converge to the unique stabilizing solution of special classes of ARE (1) are constructed (for example using the Kleinman algorithm in [2]).

Several different direct methods to solve ARE (1) are given in [1], [5], [21], [22], [24], [31]–[33], [36]. However, compared with iterative methods to solve ARE (1), direct methods present computational disadvantages in some situations. For example, in example 6 in [5], the solution to an H_2-type ARE obtained by the Schur algorithm in [5] is inaccurate but the iterative solution obtained by the Kleinman algorithm in [2] is accurate to 13 digits in just 2 iterations. The reason for the failure of using the Schur algorithm in solving AREs is that the structure of the Hamiltonian matrix cannot be preserved in general during computation. There are some direct methods such as symplectic methods, e.g., the multishift algorithm (see [21], [35]), which can preserve the structure of the Hamiltonian matrix during matrix transformation and hence yield more reliable solutions of AREs. However, most structure-preserving direct methods will typically require more computational cost and hence they are not frequently used in practice, and appear not to be incorporated into standard package.

Traditionally, in H_2 control, one needs to solve AREs with $Q \geq 0$ and $R \leq 0$. In H_∞ control, one needs to solve AREs with positive semidefinite Q and sign indefinite R. In this paper, we only focus on developing an iterative algorithm to solve AREs arising in H_∞ control. Although the Kleinman algorithm in [2] has been shown to have many advantages such as convergence for any suitable initial condition, and a local quadratic rate of convergence [2], these advantages are strictly restricted to AREs arising in H_2 control where R in (1) must be negative semidefinite. It is not difficult to adjust the Kleinman method (which is an effective Newton's method) to also handle the separate case where $R \geq 0$, but still sign-indefinite R cannot be handled. So the question naturally arises: "Can one extend the Kleinman algorithm in [2] to solve AREs with a sign indefinite quadratic term, as those that arise in H_∞ control?" The answer is that an iterative algorithm with very simple initialization to solve such a class of AREs will be given in this paper, but the algorithm is not obtained by simply permitting indefinite R to occur in the Kleinman algorithm (see Example 3 in Section VII for a demonstration of this)—a different route has to be selected.

In the Kleinman algorithm, when a suitable initial condition is chosen and some necessary assumptions hold, it is proved that a series of Lyapunov equations can be recursively constructed at each iteration and positive semidefinite solutions of these Lyapunov equations converge to the stabilizing solution of the corresponding H_2-type ARE. In our proposed algorithm, an ARE with a sign indefinite quadratic term is replaced by a sequence of

Manuscript received July 14, 2006; revised July 17, 2007. Current version published November 05, 2008. This work was supported in part by ARC Discovery-Projects Grants DP0342683 and DP0664427 and National ICT Australia, Ltd. (National ICT Australia Ltd. is funded through the Australian Government's Backing Australia's Ability initiative, and in part through the Australian Research Council). Recommended by Associate Editor D. Henrion.

A. Lanzon is with the Control Systems Center, School of Electrical and Electronic Engineering, University of Manchester, Manchester M60 1QD, U.K. (e-mail: alexander.lanzon@manchester.ac.uk).

Y. Feng and B. D. O. Anderson are with the Research School of Information Sciences and Engineering, the Australian National University, Canberra ACT 0200, Australia. They are also with National ICT Australia,Canberra ACT 2601, Australia (e-mail: alex.feng@anu.edu.au; brian.anderson@anu.edu.au).

M. Rotkowitz is with the Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville VIC 3010, Australia (e-mail: mcrotk@unimelb.edu.au).

Digital Object Identifier 10.1109/TAC.2008.2006108

H_2-type AREs (each of which could be solved by the Kleinman algorithm if desired, though this need not happen), and the solution of the original ARE with a sign indefinite quadratic term is obtained by recursively solving these H_2-type AREs.

In H_2 control, typically, the Kleinman algorithm for solving H_2-type AREs with a negative semidefinite quadratic term is well suited as a "second iterative stage" refinement to achieve the limiting accuracy for the stabilizing solution of the ARE. For example, if an approximate solution is known (e.g., using the Schur method), and this is stabilizing, then 1–2 iterations are sufficient to achieve the limiting accuracy (because of the guaranteed final quadratic rate of convergence of a typical Newton algorithm). Now as noted, the Kleinman algorithm reduces a quadratic (Riccati) equation (with a negative semidefinite quadratic term) to several successive iterations of linear (Lyapunov) equations; the complexity of solving algebraic Lyapunov and Riccati equations with sound numerical methods (e.g., Schur form based reductions) is $O(n^3)$ for both. When Schur form based reductions are used to solve Lyapunov equations, the computation for such (Schur form based) reductions needs about $25n^3$ flops (see [6]), where 1 flop equals 1 addition/subtraction or 1 multiplication/division. About $7n^3$ flops are necessary to solve the reduced equation and to compute the solution. The basic method is described in [7]. For the solutions of AREs, the Schur approach of Laub [5] requires $240n^3$ flops of which $25(2n)^3$ flops are required to reduce a $2n \times 2n$ Hamiltonian matrix to real Schur form and the rest accounts for the computation of the eigenvalues and solving a linear equation of order n (i.e., $5/3n^3$ flops). Consequently, both Riccati and Lyapunov equations require $O(n^3)$ computations. Hence the advantage of iterative schemes such as the Kleinman algorithm (which will require several Lyapunov equations to be solved, typically) is not always the speed of computation, but rather, it is the numerical reliability of the computations to reach the limiting accuracy of a solution. Given a specified tolerance $\Delta = 0.1$ or $\Delta = 0.01$, when we use our algorithm to solve an ARE with a sign indefinite quadratic term, typically we need 3–5 iterations (see the random test in Example 2 of Section VII). In such situations, the flop count of our algorithm may be lower than some direct methods if we only need 3 iterations to obtain the solution of an ARE; however, in some situations where we need more than 3 iterations to solve an ARE with our algorithm, the flop count of our algorithm may be higher when compared with some direct methods. Consider a typical example to illustrate this point: suppose our proposed algorithm is used to solve an ARE with a sign-indefinite quadratic term and this ARE is reduced to a sequence of four AREs with a negative semidefinite quadratic term. Furthermore, suppose the Kleinman algorithm is used to solve these four AREs and each of them can be solved using two Lyapunov equations. Then since the flop count of solving a Lyapunov equation is $32n^3$ (see [6]), the total flop count for our algorithm to solve such an ARE with a sign-indefinite quadratic term is $256n^3$, which is higher than $240n^3$ (flop count of the Schur method).

Besides the Kleinman algorithm, there are some other iterative methods to solve AREs [16], [21], [23]–[25], [29], [31]–[33], [35], [36], [38]–[40], some of which exhibit quadratic convergence. Among iterative methods to solve ARE (1), Newton-type algorithms are typical and widely used [16], [21], [24], [25], [29], [31]–[33], [35], [36]. In fact, Newton's method can be used to solve more than just symmetric AREs like (1). It can also be used to solve non-symmetric AREs where Q and R in (1) are not necessarily symmetric [27], [28]. Besides Newton-type algorithms, there are other iterative algorithms to solve AREs with a sign indefinite quadratic term, for example the matrix sign function method (see [31], [33], [36]). However, there are also disadvantages when the matrix sign function method is used to solve AREs, for example, when the eigenvalues of the corresponding Hamiltonian matrix of a given ARE are close to the imaginary axis, this method will perform poorly or even fail (see Example 3 in Section VII). In this paper, since we will develop an iterative algorithm to solve AREs arising in H_∞ control, for the purpose of comparison we only use the iterative algorithms in [25], [35], which are Newton-type algorithms that can be used to solve the same class of AREs. However, our proposed algorithm to solve AREs is significantly different from the Newton's method algorithms in [25], [35]. The key distinguishing features between our algorithm and the Newton's method algorithms in [25], [35] are as follows:

- In [25], [35], a suitable initial condition must be chosen to run the corresponding iterative algorithms, and this may not always be straightforward to do, while in our proposed algorithm, an initial condition $P_0 = 0$ can always be simply chosen (see Section III). Also, when the Newton's method algorithms in [25], [35] are used to solve AREs, the computation cost and efficiency will depend greatly on the choice of the initial condition (see Example 5 in Section VII for an illustration of this).
- In [25], [35], an ARE is reduced to a series of Lyapunov equations, while in our proposed algorithm, an ARE with a sign indefinite quadratic term is reduced to a series of AREs with a negative semidefinite quadratic term, i.e., a series of H_2-type AREs.

Apart from these key distinguishing features mentioned above, there are some minor differences between the algorithms in [25], [35] and our proposed algorithm:

- In [25], [35], many different monotonic matrix sequences which converge to the stabilizing solution of an ARE can be constructed; but in our proposed algorithm, a unique and well-motivated (also with a game theoretic motivation) monotonic matrix sequence which converges to the stabilizing solution of an ARE is constructed.
- In [25], [35], monotonic matrix sequences converging to the stabilizing solution of an ARE are non-increasing; but in our proposed algorithm, the unique monotonic matrix sequence converging to the stabilizing solution of an ARE is non-decreasing.

The above commentary is all concerned with AREs arising in deterministic control systems (and, actually LQG and H_∞ filtering problems). Modifications of the AREs considered can arise when state-dependent and input-dependent noise enters the equations. There is some existing literature dealing with such problems, see [17]–[20], [34], [35] and again, Newton-type methods can be applied, see, e.g., [3], [26], [30], [35]. The equations in question are however not of interest to us here.

As noted above, in the work presented in this paper, we reduce the problem of solving a generic Riccati equation with a sign indefinite quadratic term to one of generating successive iterations of solutions of conventional H_2-type AREs with a negative semidefinite quadratic term (each of which is then amenable to the Kleinman algorithm). Consequently, we are reducing a Riccati equation that has no straightforwardly initialized iterative scheme for its solution to a number of successive iterations of Riccati equations, each of which can (if desired) be solved by an existing iterative scheme (e.g., the Kleinman algorithm).

The structure of the paper is as follows. Section II establishes some preliminary results which will be used in the main theorem. Section III presents the main result with a proof of global convergence and Section IV states the algorithm. Section V provides a proof for the local quadratic rate of convergence. Section VI gives a game theoretic interpretation of the proposed algorithm. Section VII presents some numerical examples that demonstrate the effectiveness of our algorithm. Section VIII offers additional technical remarks and Section IX gives some concluding remarks. The Appendix includes two examples relevant to the technical remarks.

II. Preliminary Results

We firstly introduce some notation: Let $\mathbb{R}^{n\times m}$ denote the set of $n \times m$ real matrices; $\mathbb{Z}$ denotes the set of integers with $\mathbb{Z}_{\geq a}$ denoting the set of integers greater or equal to $a \in \mathbb{R}$; $\rho[\cdot]$ denotes the spectral radius of a square matrix; $\overline{\sigma}(\cdot)$ denotes the maximum singular value of a matrix; $\operatorname{spec}(\cdot)$ denotes the spectrum of a square matrix; $\|\cdot\|$ denotes the Euclidean norm. A matrix $A \in \mathbb{R}^{n\times n}$ is said to be Hurwitz if all its eigenvalues have negative real part; A matrix $P \in \mathbb{R}^{n\times n}$ is called a stabilizing solution of ARE (1) if it satisfies ARE (1) and the matrix $A + RP$ is Hurwitz.

In this paper, we will restrict attention to the unique stabilizing solution Π for the following ARE:

$$0 = \Pi A + A^T\Pi - \Pi(B_2B_2^T - B_1B_1^T)\Pi + C^TC \qquad (2)$$

where A, B_1, B_2, C are real matrices with compatible dimensions. Note that stabilizing solutions to AREs are always unique (see [10]) when they exist, but for AREs with a sign indefinite quadratic term, the unique stabilizing solution Π may not always be positive semidefinite. Since our interest arises from AREs used for H_∞ control, in this case, in order to obtain an H_∞ controller, we need to solve AREs with a sign indefinite quadratic term and the stabilizing solutions of these AREs are also required to be positive semidefinite if such an H_∞ controller exists. So we focus on a unique stabilizing solution to (2) that happens to be also positive semidefinite when this exists. Necessary and sufficient conditions for the existence of a positive semidefinite stabilizing solution Π to (2) are given in [8]. However, these conditions are not easily computable as they involve integral operators and operator inverses.

The first lemma gathers together some straightforward computations.

Lemma 1: Given real matrices A, B_1, B_2, C with compatible dimensions, define

$$\begin{aligned} F : \mathbb{R}^{n\times n} &\longrightarrow \mathbb{R}^{n\times n} \\ P &\longmapsto PA + A^TP - P(B_2B_2^T - B_1B_1^T)P + C^TC. \end{aligned} \qquad (3)$$

Given also $P = P^T$, $Z = Z^T \in \mathbb{R}^{n\times n}$, then

$$F(P+Z) = F(P) + Z\hat{A} + \hat{A}^TZ - Z(B_2B_2^T - B_1B_1^T)Z \qquad (4)$$

where $\hat{A} = A + B_1B_1^TP - B_2B_2^TP$. Furthermore, if $P = P^T$, $Z = Z^T \in \mathbb{R}^{n\times n}$ satisfy

$$0 = Z\hat{A} + \hat{A}^TZ - ZB_2B_2^TZ + F(P) \qquad (5)$$

then

$$F(P+Z) = ZB_1B_1^TZ \qquad (6)$$

and

$$\rho[F(P+Z)] = \overline{\sigma}(B_1^TZ)^2. \qquad (7)$$

Proof: Equation (4) trivially follows via algebraic manipulations. Results (6) and (7) are simple consequences of (4). ■

The second lemma sets up some basic relationships between the stabilizing solution Π to (2) when it exists and the matrices P, Z satisfying (5).

Lemma 2: Given real matrices A, B_1, B_2, C with compatible dimensions, $P = P^T \in \mathbb{R}^{n\times n}$ and $Z = Z^T \in \mathbb{R}^{n\times n}$ satisfying (5), and a stabilizing $\Pi = \Pi^T \in \mathbb{R}^{n\times n}$ satisfying (2), and let $\check{A} = A + B_1B_1^T(P+Z) - B_2B_2^T\Pi$ and $\bar{A} = A + B_1B_1^TP - B_2B_2^T\Pi$. Then

(i) $\Pi \geq (P+Z)$ if $\bar{A}$ is Hurwitz,

(ii) $\check{A}$ is Hurwitz if $\Pi \geq (P+Z)$.

Proof: (i) Satisfaction of (5) yields satisfaction of (6) via Lemma 1. Adding (6) to (2), we have

$$\begin{aligned} F(P+Z) = \Pi A + A^T\Pi - \Pi(B_2B_2^T - B_1B_1^T)\Pi \\ + C^TC + ZB_1B_1^TZ. \end{aligned} \qquad (8)$$

Since

$$\begin{aligned} F(P+Z) = (P+Z)A + A^T(P+Z) \\ -(P+Z)(B_2B_2^T - B_1B_1^T)(P+Z) + C^TC \end{aligned} \qquad (9)$$

via (3), substituting (9) into (8) and rearranging, we have

$$0 = \Sigma\bar{A} + \bar{A}^T\Sigma + \Sigma B_2B_2^T\Sigma + (\Pi - P)B_1B_1^T(\Pi - P) \qquad (10)$$

where $\Sigma = \Pi - (P+Z)$. Note that the last two terms in (10) are positive semidefinite, so (i) holds by a standard Lyapunov equation type argument (see Lemma 3.18 in [4] for example).

(ii) Rearrange (10) as follows:

$$0 = \Sigma\check{A} + \check{A}^T\Sigma + \Sigma B_2B_2^T\Sigma + \Sigma B_1B_1^T\Sigma + ZB_1B_1^TZ. \qquad (11)$$

Now letting

$$W = \begin{pmatrix} B_2^T\Sigma \\ B_1^T\Sigma \\ B_1^TZ \end{pmatrix}.$$

Equation (11) becomes

$$0 = \Sigma\check{A} + \check{A}^T\Sigma + W^TW. \qquad (12)$$

Since $\Pi \geq (P+Z)$ and $W^TW \geq 0$, it is only required to show that $(W, \check{A})$ is detectable for the required result to follow (see again Lemma 3.19 in [4] for example). To this end, note that $(W, \check{A})$ is detectable because

$$\check{A} + LW = A + B_1B_1^T\Pi - B_2B_2^T\Pi$$

for $L = (0 \quad B_1 \quad 0)$, which is clearly Hurwitz as Π is the stabilizing solution for (2). This concludes the proof of part (ii). ■

III. Main Results

In this section, we set up the main theorem by constructing two positive semidefinite matrix series P_k and Z_k, and we also prove that the series P_k is monotonically non-decreasing and converges to the unique stabilizing solution Π (which is also positive semidefinite) of ARE (2) if such a solution exists.

Theorem 3: Given real matrices A, B_1, B_2, C with compatible dimensions such that (C, A) has no unobservable modes on the $j\omega$-axis and (A, B_2) is stabilizable, define $F : \mathbb{R}^{n\times n} \longrightarrow \mathbb{R}^{n\times n}$ as in (3). Suppose there exists a stabilizing solution Π, which is also positive semidefinite, of ARE (2). Then

(I) two square matrix series Z_k and P_k can be defined for all $k \in \mathbb{Z}_{\geq 0}$ recursively as follows:

$$P_0 = 0 \tag{13}$$

$$A_k = A + B_1B_1^TP_k - B_2B_2^TP_k \tag{14}$$

$Z_k \geq 0$ is the unique stabilizing solution of

$$0 = Z_kA_k + A_k^TZ_k - Z_kB_2B_2^TZ_k + F(P_k) \tag{15}$$

$$P_{k+1} = P_k + Z_k \tag{16}$$

(II) the two series P_k and Z_k in part (I) have the following properties:
1) $(A + B_1B_1^TP_k, B_2)$ is stabilizable $\forall k \in \mathbb{Z}_{\geq 0}$;
2) $F(P_{k+1}) = Z_kB_1B_1^TZ_k \ \forall k \in \mathbb{Z}_{\geq 0}$;
3) $A + B_1B_1^TP_k - B_2B_2^TP_{k+1}$ is Hurwitz $\forall k \in \mathbb{Z}_{\geq 0}$;
4) $\Pi \geq P_{k+1} \geq P_k \geq 0 \ \forall k \in \mathbb{Z}_{\geq 0}$.

(III) the limit

$$P_\infty := \lim_{k\to\infty} P_k$$

exists with $P_\infty \geq 0$. Furthermore, $P_\infty = \Pi$ is the unique stabilizing solution of ARE (2), which is also positive semidefinite.

Proof: We construct the series for Z_k and P_k to show results (I) and (II) together by an inductive argument. Firstly we show that (I) and (II) are true when $k = 0$. Then, given $k \in \mathbb{Z}_{\geq 0}$ where (I) and (II) are satisfied, we will show that (I) and (II) are also satisfied for $k + 1$.

• Case $k = 0$

Since $P_0 = 0$ via (13), (II 1) is trivially satisfied by assumption. Since (15) reduces to

$$0 = Z_0A + A^TZ_0 - Z_0B_2B_2^TZ_0 + C^TC \tag{17}$$

it is standard [4], [9] that there exists a unique positive semidefinite and stabilizing solution Z_0 for (17); hence $Z_0 \geq 0$. Since $P_1 = P_0 + Z_0$ via (16), then we have $F(P_1) = Z_0B_1B_1^TZ_0$ by Lemma 1 and (II 2) is satisfied. It is also standard [4], [9] that $(A - B_2B_2^TZ_0)$ is Hurwitz (since Z_0 is the stabilizing solution of (17)), hence (II 3) is satisfied on noting that $P_0 = 0$ and $P_1 = Z_0$. We can show (II 4) is satisfied by the following steps:
1) Since $Z_0 \geq 0$ and $P_1 = P_0 + Z_0$, then $P_1 \geq P_0$;
2) Since $P_0 = 0, (A+B_1B_1^TP_0-B_2B_2^T\Pi) = (A-B_2B_2^T\Pi)$ is Hurwitz (see [4] or [9]);
3) Since $(A + B_1B_1^TP_0 - B_2B_2^T\Pi)$ is Hurwitz, then $\Pi \geq (P_0 + Z_0) = P_1$ by Lemma 2

• Inductive step for $k \in \mathbb{Z}_{\geq 0}$.

We now consider an arbitrary $q \in \mathbb{Z}_{\geq 0}$, suppose that (I) and (II) are satisfied for $k = q \in \mathbb{Z}_{\geq 0}$, and show that (I) and (II) are also satisfied for $k = q+1$. Since $F(P_{q+1}) = Z_qB_1B_1^TZ_q \geq 0$ by inductive assumption (II 2), necessary and sufficient conditions for the existence of a unique positive semidefinite and stabilizing solution Z_{q+1} to (15) are (see [4], [9]):

(α) (A_{q+1}, B_2) is stabilizable;

(β) $(B_1^TZ_q, A_{q+1})$ has no unobservable modes on the $j\omega$-axis.

Since $A + B_1B_1^TP_{q+1} = A_{q+1} + B_2B_2^TP_{q+1}$ and $A + B_1B_1^TP_q - B_2B_2^TP_{q+1} = A_{q+1} - B_1B_1^TZ_q$, conditions (α) and (β) are clearly equivalent to the following two conditions, respectively:

(α1) $(A + B_1B_1^TP_{q+1}, B_2)$ is stabilizable;

(β1) $(B_1^TZ_q, A + B_1B_1^TP_q - B_2B_2^TP_{q+1})$ has no unobservable modes on the $j\omega$-axis.

We will now show the existence of Z_{q+1} is guaranteed via the following two points:
1) Since result (II 4) holds by inductive assumptions, we have $\Pi \geq P_{q+1}$, and thus $(A + B_1B_1^TP_{q+1} - B_2B_2^T\Pi)$ is Hurwitz by Lemma 2 Part (ii). Hence $(A + B_1B_1^TP_{q+1}, B_2)$ is stabilizable and thus condition $(\alpha 1)$ and result (II 1) for $k = q + 1$ are satisfied;
2) Since $(A + B_1B_1^TP_q - B_2B_2^TP_{q+1})$ is Hurwitz by inductive assumption (II 3), condition $(\beta 1)$ is also satisfied.

Since conditions $(\alpha 1)$ and $(\beta 1)$ hold, there exists a unique positive semidefinite and stabilizing solution Z_{q+1} for (15) with $k = q+1$. Since $P_{q+2} = P_{q+1} + Z_{q+1}$, (II 2) is trivially satisfied for $k = q+1$ via Lemma 1. Since Z_{q+1} is the stabilizing solution to (15), it follows that $(A_{q+1} - B_2B_2^TZ_{q+1}) = (A + B_1B_1^TP_{q+1} - B_2B_2^TP_{q+2})$ is Hurwitz, hence (II 3) is satisfied when $k = q+1$. Since $P_{q+2} = P_{q+1} + Z_{q+1}$ and $Z_{q+1} \geq 0, P_{q+2} \geq P_{q+1}$. Also, since $\Pi \geq P_{q+1} = P_q + Z_q$ by inductive assumption (II 4) for $q \in \mathbb{Z}_{\geq 0}$, it follows that $(A + B_1B_1^TP_{q+1} - B_2B_2^T\Pi)$ is Hurwitz via Lemma 2 Part (ii) and this in turn gives $\Pi \geq P_{q+2}$ via Lemma 2 Part (i). Hence (II 4) is satisfied for $k = q + 1$.

• Inductive Conclusion

The case for $k = 0$ and the inductive step establish that (I) and (II) are true $\forall k \in \mathbb{Z}_{\geq 0}$, so the proof for (I) and (II) is completed.

(III) Since the sequence P_k is monotone (i.e., $P_{k+1} \geq P_k$) and bounded above by Π (i.e., $\Pi \geq P_k$), the sequence converges to a limit P_∞ (see pp. 33–34 in [37] for the details), and convergence of the sequence P_k to P_∞ implies convergence of Z_k to 0 since

$$Z_\infty := \lim_{k\to\infty} Z_k = \lim_{k\to\infty}(P_{k+1} - P_k) = 0.$$

Then it is clear from (II 4) that $P_\infty \geq 0$, from (II 2) that P_∞ must satisfy $F(P_\infty) = 0$, and from (II 3) that $(A + B_1B_1^TP_\infty - B_2B_2^TP_\infty)$ must be Hurwitz. Thus $P_\infty \geq 0$ is a stabilizing solution to $F(P_\infty) = 0$. Since $\Pi \geq 0$ is a stabilizing solution to $F(\Pi) = 0$ and the stabilizing solution of an ARE is always unique (see [10]), it is clear that $P_\infty = \Pi$. ■

The following corollary gives a condition under which there does *not* exist a stabilizing solution $\Pi \geq 0$ to $F(\Pi) = 0$. This is useful for terminating the recursion in finite iterations.

Corollary 4: Given real matrices A, B_1, B_2, C such that (C, A) has no unobservable modes on the $j\omega$-axis and (A, B_2) is stabilizable, and let $\{P_k\}$ and $F : \mathbb{R}^{n\times n} \rightarrow \mathbb{R}^{n\times n}$ be defined as in Theorem 3. If $\exists k \in \mathbb{Z}_{\geq 0}$ such that $(A + B_1B_1^TP_k, B_2)$ is not stabilizable, then there does not exist a stabilizing solution $\Pi \geq 0$ to $F(\Pi) = 0$.

Proof: Restatement of Theorem 3, implication (II 1). ■

IV. Algorithm

Given real matrices A, B_1, B_2, C with compatible dimensions, a specified tolerance $\Delta > 0$, and supposing (C, A) has no unobservable modes on the $j\omega$-axis and (A, B_2) is stabilizable, an iterative algorithm for finding the positive semidefinite stabilizing solution of (2), when it exists, is given as follows:

1) Let $P_0 = 0$ and $k = 0$.
2) Set $A_k = A + B_1B_1^TP_k - B_2B_2^TP_k$.
3) Construct (for example using the Kleinman algorithm in [2], though this is not necessary) the unique real symmetric stabilizing solution $Z_k \geq 0$ which satisfies

$$0 = Z_kA_k + A_k^TZ_k - Z_kB_2B_2^TZ_k + F(P_k) \tag{18}$$

where $F : \mathbb{R}^{n\times n} \rightarrow \mathbb{R}^{n\times n}$ is defined in (3).
4) Set $P_{k+1} = P_k + Z_k$.
5) If $\overline{\sigma}(B_1^TZ_k)^2 < \Delta$, then set $\Pi = P_{k+1}$ and exit. Otherwise, go to step 6.
6) If $(A + B_1B_1^TP_{k+1}, B_2)$ is stabilizable, then increment k by 1 and go back to step 2. Otherwise, exit as there does not exist a real symmetric stabilizing solution $\Pi \geq 0$ satisfying $F(\Pi) = 0$.

From Corollary 4 we see that if the stabilizability condition in step 6 fails at some $k \in \mathbb{Z}_{\geq 0}$, then there does not exist a stabilizing solution $\Pi \geq 0$ to $F(\Pi) = 0$ and the algorithm should terminate (as required by step 6). But when this stabilizability condition is satisfied $\forall k \in \mathbb{Z}_{\geq 0}$, construction of the series P_k and Z_k is always possible and either P_k converges to Π (which is captured by step 5) or P_k just diverges to infinity, which again means that there does not exist a stabilizing solution $\Pi \geq 0$ to $F(\Pi) = 0$.

Remark 1: It is worth pointing out that when the Kleinman algorithm is used to solve (18), how to stop the Kleinman iteration can be an important issue (see Example 2 in the Appendix for a demonstration of this). In fact, a simple criterion to stop the Kleinman iteration is to compute the residue of (18). When the residue of the right hand side terms of (18) is small enough, the Kleinman iteration should be stopped.

Remark 2: For a method to check the stabilizability of a matrix pair in step 6, one can refer to p. 50 in [4].

V. Rate of Convergence

The following theorem states that the local rate of convergence of the algorithm given in Section IV is quadratic.

Theorem 5: Given the suppositions of Theorem 3, and two series P_k, Z_k as defined in Theorem 3 Part I. Then there exists a $\theta > 0$ such that the rate of convergence of the series P_k is quadratic in the region $\|P_k - \Pi\| < \theta$.

Proof: We prove the rate of convergence of P_k by proving the rate of convergence of Z_k. Let $\acute{A}_k = A + B_1B_1^TP_k - B_2B_2^TP_{k+1}$ and $\tilde{A} = A + B_1B_1^T\Pi - B_2B_2^T\Pi$. Note that $\acute{A}_k$ is Hurwitz (see Theorem 3 Part II) and $\tilde{A}$ is Hurwitz since Π is the stabilizing solution of (2) (see [4]), and let $\tilde{Y}$ and $\acute{Y}_k$ be uniquely defined by:

$$0 = \tilde{Y}\tilde{A} + \tilde{A}^T\tilde{Y} + I \tag{19}$$
$$0 = \acute{Y}_k\acute{A}_k + \acute{A}_k^T\acute{Y}_k + I \tag{20}$$

where I is the identity matrix with appropriate dimensions. The matrices $\tilde{Y}$ and $\acute{Y}_k$ are positive definite because of the stability properties of $\acute{A}_k$ and $\tilde{A}$. Since $\tilde{Y}$ and $\acute{Y}_k$ are uniquely defined and $\lim_{k\rightarrow\infty} \acute{A}_k = \tilde{A}$, then $\lim_{k\rightarrow\infty} \acute{Y}_k = \tilde{Y}$, and thus for any small $\gamma > 0$, $\exists K_1 \in \mathbb{Z}_{\geq 0}$ such that

$$\overline{\sigma}(\acute{Y}_k - \tilde{Y}) \leq \gamma \quad \forall k \geq K_1.$$

This implies that

$$\overline{\sigma}(\acute{Y}_k) \leq \overline{\sigma}(\tilde{Y}) + \gamma \quad \forall k \geq K_1. \tag{21}$$

Now, define a monotonically non-increasing sequence ε_k by:

$$\varepsilon_k = \sup_{m\geq k} \overline{\sigma}(Z_m).$$

From (15) and Theorem 3, we have $\forall k \in \mathbb{Z}_{\geq 1}$:

$$0 = Z_kA_k + A_k^TZ_k - Z_kB_2B_2^TZ_k + Z_{k-1}B_1B_1^TZ_{k-1} \tag{22}$$

which can be equivalently rewritten as

$$0 = Z_k\acute{A}_k + \acute{A}_k^TZ_k + Z_kB_2B_2^TZ_k + Z_{k-1}B_1B_1^TZ_{k-1}. \tag{23}$$

Now, there exists $\eta > 0$ (e.g., $\eta = 4\max\{\overline{\sigma}(B_1)^2, \overline{\sigma}(B_2)^2\}$), independent of k, such that

$$Z_kB_2B_2^TZ_k + Z_{k-1}B_1B_1^TZ_{k-1} \leq \eta\varepsilon_{k-1}^2I \quad \forall k \in \mathbb{Z}_{\geq 1}. \tag{24}$$

Multiplying $(\eta\varepsilon_{k-1}^2)$ on each side of the (20), we obtain

$$0 = (\eta\varepsilon_{k-1}^2)\acute{Y}_k\acute{A}_k + (\eta\varepsilon_{k-1}^2)\acute{A}_k^T\acute{Y}_k + (\eta\varepsilon_{k-1}^2)I. \tag{25}$$

Then, subtracting (23) from the (25), we obtain

$$\begin{aligned}0 = {} & (\eta\varepsilon_{k-1}^2\acute{Y}_k - Z_k)\acute{A}_k + \acute{A}_k(\eta\varepsilon_{k-1}^2\acute{Y}_k - Z_k) + \eta\varepsilon_{k-1}^2I \\ & - (Z_kB_2B_2^TZ_k + Z_{k-1}B_1B_1^TZ_{k-1}).\end{aligned} \tag{26}$$

Now note that $\acute{A}_k$ is Hurwitz and the inequality (24) holds, then by (26) we have (see Lemma 3.18 in [4])

$$\eta\varepsilon_{k-1}^2\acute{Y}_k \geq Z_k. \tag{27}$$

Thus $\overline{\sigma}(Z_k) \leq \eta\varepsilon_{k-1}^2\overline{\sigma}(\acute{Y}_k)$ as $\eta\varepsilon_{k-1}^2\acute{Y}_k \geq Z_k$ (see [6]). Hence $\forall k \geq K_1 \geq 1$,

$$\begin{aligned}\varepsilon_k = \sup_{m\geq k}\overline{\sigma}(Z_m) &\leq \sup_{m\geq k}\left[\eta\varepsilon_{m-1}^2\overline{\sigma}(\acute{Y}_m)\right]\\ &\leq \eta(\overline{\sigma}(\tilde{Y})+\gamma)\sup_{m\geq k}\varepsilon_{m-1}^2 = \eta(\overline{\sigma}(\tilde{Y})+\gamma)\varepsilon_{k-1}^2.\end{aligned}$$

Thus, $\forall k \geq K_1 \geq 1$ such that $\varepsilon_{k-1} < 1/\eta(\overline{\sigma}(\tilde{Y})+\gamma)$, we obtain a quadratic rate of convergence, which concludes the proof. ■

VI. Game Theoretic Interpretation of the Algorithm

In this section, for the purpose of motivation, interpretation and further research, a game theoretic interpretation of the algorithm will be given. At the same time, we will also note that this interpretation is closely linked to an optimal control concept of approximation in policy space [11]–[13]. In this section, we will show that a game theory performance index can be approximated by a series of successive cost functions. At each iteration, the optimal policy is found to minimize the corresponding cost functions. With the increment of each iteration, the optimal policies approach the final optimum and the saddle point of the cost functions approaches the saddle point of the game theory performance index. In fact, it can be shown (see [14], [15]) that the technique of policy space iteration can be used to replace nonlinear Hamilton-Jacobi partial differential equations by a sequence of linear partial differential equations (even when the approximations and the optimal feedback law are nonlinear functions of the state). This is important because it is difficult [14] to solve Hamilton-Jacobi equations directly to obtain optimal feedback control in many cases.

Consider the dynamical system

$$\dot{x} = Ax + B_1w + B_2u \tag{28}$$

with the game theory performance index

$$J(x_0,u,w) = \int_0^\infty (u^Tu + x^TC^TCx - w^Tw)dt \tag{29}$$

where x_0 denotes the initial state of the system, x is the state vector, u denotes the control input, w denotes the disturbance input and A, B_1, B_2, C are given real matrices with compatible dimensions and appropriate stabilizability/detectability conditions. It is desired that u minimizes the cost function J and w maximizes it. It is well-known [9] that the optimal control law and the worst case disturbance (a saddle point of $J(x_0,u,w)$) are given by:

$$u_{optimal} = -B_2^T\Pi x, \tag{30}$$

$$w_{worst} = B_1^T\Pi x, \tag{31}$$

where $\Pi \geq 0$ is the unique stabilizing solution to (2). See [9] for more details on such game theory problems.

Let us now propose a heuristic induction which gives a game theoretic interpretation to our proposed algorithm. Suppose that at iteration k we have a trial control law $u_k = -B_2^TP_kx$ with P_k defined as in Theorem 3 Part I. Then we set $w_k = B_1^TP_kx$. Note that this is NOT the worst case w_k corresponding to $u_k = -B_2^TP_kx$ (unless $P_k = \Pi$), but it is a strategy we wish to impose since it will connect the heuristic ideas of this section to the earlier algorithm. The choice is also motivated by what happens at the optimum, as $P_k \rightarrow \Pi$ when $k \rightarrow \infty$. With this choice of w fixed, we now wish to find a new optimal control; i.e., u now has to minimize the following LQ cost function:

$$J_k(x_0,u) = \int_0^\infty (u^Tu + x^TC^TCx - x^TP_kB_1B_1^TP_kx)dt \tag{32}$$

subject to

$$\dot{x} = (A + B_1B_1^TP_k)x + B_2u \tag{33}$$

where $w_k = B_1^TP_kx$ has been substituted in (28) and (29) to yield the above problem. Under appropriate conditions (which will be analyzed in more details below via (36)), this LQ problem has an optimal solution for u given by:

$$u_{k+1} = -B_2^T\Lambda_{k+1}x$$

where Λ_{k+1} is the unique stabilizing solution to

$$\begin{aligned}0 = \Lambda_{k+1}(A + B_1B_1^TP_k) &+ (A + B_1B_1^TP_k)^T\Lambda_{k+1}\\ &-\Lambda_{k+1}B_2B_2^T\Lambda_{k+1} + (C^TC - P_kB_1B_1^TP_k).\end{aligned} \tag{34}$$

We will now show that Λ_{k+1} actually equals P_{k+1} as defined in Theorem 3 Part I. To do this, we will equivalently show that $\Lambda_{k+1} - P_k$ equals Z_k. Towards this end, let $W_k = \Lambda_{k+1} - P_k$. Firstly we will show that W_k satisfies the same equation as Z_k. Using $\Lambda_{k+1} = P_k + W_k$ in (34), we have

$$\begin{aligned}0 = P_k(A + B_1B_1^TP_k) &+ W_k(A + B_1B_1^TP_k)\\ &+ (A + B_1B_1^TP_k)^TP_k + (A + B_1B_1^TP_k)^TW_k\\ &- P_kB_2B_2^TP_k - W_kB_2B_2^TP_k - P_kB_2B_2^TW_k\\ &- W_kB_2B_2^TW_k + C^TC - P_kB_1B_1^TP_k.\end{aligned} \tag{35}$$

The terms above independent of W_k are

$$P_kA + A^TP_k + P_k(B_1B_1^T - B_2B_2^T)P_k + C^TC$$

which are equal to $F(P_k)$. Also, recall that $A_k = A + B_1B_1^TP_k - B_2B_2^TP_k$, hence (35) reduces to

$$0 = W_kA_k + A_k^TW_k - W_kB_2B_2^TW_k + F(P_k). \tag{36}$$

Since this is the same equation as (18), we conclude that W_k satisfies the same equation as Z_k. Also, existence of W_k is equivalent to existence of Λ_{k+1}. Necessary and sufficient conditions for the existence of W_k are: (A_k, B_2) is stabilizable and $(F(P_k), A_k)$ has no unobservable modes on the jw-axis. These conditions were analyzed in the proof of Theorem 3 and were shown to be fulfilled via the existence of the stabilizing solution $\Pi \geq 0$ to (2).

Now note that $\Lambda_{k+1} = P_k + W_k$ is a stabilizing solution to (34), meaning that $A + B_1B_1^TP_k - B_2B_2^T(W_k + P_k)$ is Hurwitz.

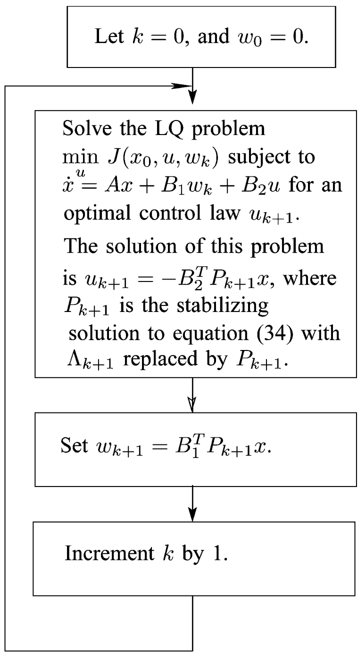

Let $k = 0$, and $w_0 = 0$.
Solve the LQ problem $\min_u J(x_0, u, w_k)$ subject to $\dot{x} = Ax + B_1 w_k + B_2 u$ for an optimal control law u_{k+1}. The solution of this problem is $u_{k+1} = -B_2^T P_{k+1} x$, where P_{k+1} is the stabilizing solution to equation (34) with Λ_{k+1} replaced by P_{k+1}.
Set $w_{k+1} = B_1^T P_{k+1} x$.
Increment k by 1.

Fig. 1. Heuristic game plan.

But $A + B_1 B_1^T P_k - B_2 B_2^T (W_k + P_k) = (A_k - B_2 B_2^T W_k)$. Hence W_k also makes $A_k - B_2 B_2^T W_k$ Hurwitz. Since Z_k is the unique stabilizing solution to (18) and since W_k satisfies the same equation and is also stabilizing, we conclude that $W_k = Z_k$ (see [10]), thereby in turn giving $\Lambda_{k+1} = P_{k+1}$.

Since the optimal control law that minimizes cost function (32) subject to constraint (33) is $u_{k+1} = -B_2^T P_{k+1} x$, we now set (according to our game plan) $w_{k+1} = B_1^T P_{k+1} x$ and proceed in this manner as outlined in the chart in Fig. 1.

This heuristic game plan converges to the optimal control (30) and the worst case disturbance (31), thereby also giving a game theoretic interpretation to the proposed algorithm.

VII. Numerical Examples

In this section, five examples are given. Example 1 provides a numerical comparison with other existing approaches, and shows that our algorithm works well when other approaches work well too. Example 2 provides a random test to compare our algorithm with the MATLAB command CARE and shows that our algorithm has good efficiency and accuracy when compared with CARE. Example 3 shows that our algorithm still works well when some other approaches (such as the MATLAB command CARE, the Schur method of [5], and the matrix sign function method of [31]) do not work. Example 4 demonstrates that there in fact does not exist a stabilizing solution $\Pi \geq 0$ to (2) when the stabilizability condition in step 6 of the algorithm is not satisfied. Example 5 shows that the computational cost of methods in [25], [35] depend greatly on the choice of initial condition. This is in contrast to our method that can always be initialized with the simple choice $P_0 = 0$.

Example 1: This simple example is used to illustrate that the algorithm proposed in this paper works well in situations where the traditional MATLAB command CARE (see [22] for the algorithm of CARE) also works well. Let

$$A = \begin{pmatrix} -3.4573 & -0.0313 & 0.1167 & 0.1295 \\ 0.6203 & -1.9884 & 1.9267 & 0.2827 \\ -1.8066 & 1.9929 & -3.4093 & -0.4120 \\ -0.3954 & 0.3908 & 0.4544 & -5.1381 \end{pmatrix}$$
$$B_1 = \begin{pmatrix} 1.6555 & 0.7164 & -1.5027 \\ -1.4300 & 0.5922 & 1.4075 \\ 2.8250 & 0.1516 & -0.4710 \\ -1.9743 & 1.5813 & -1.1708 \end{pmatrix}$$
$$B_2 = \begin{pmatrix} -1.6178 & -1.0622 \\ -1.0728 & 1.0278 \\ 0.8247 & 0.6979 \\ 0.7092 & 0.6806 \end{pmatrix}$$
$$C = \begin{pmatrix} -1.6758 & -0.4228 & 2.1930 & 0.8601 \\ 0.6654 & 0.9273 & -2.0392 & -1.3478 \\ -0.7585 & 0.1406 & 0.9184 & 0.7515 \\ 0.3357 & -0.0278 & 0.2078 & 0.7607 \end{pmatrix}.$$

The traditional MATLAB command CARE gives the solution:

$$\Pi = \begin{pmatrix} 0.4864 & -0.0021 & -0.5392 & -0.1774 \\ -0.0021 & 0.2259 & -0.0685 & -0.0907 \\ -0.5392 & -0.0685 & 0.7752 & 0.3601 \\ -0.1774 & -0.0907 & 0.3601 & 0.2532 \end{pmatrix}.$$

The algorithm proposed in this paper gives an almost identical solution after only 3 iterations with a specified tolerance $\Delta = 0.1$. The normalized error between the two solutions is 6.9717×10^{-5}.

Example 2: In this example, to show the efficiency and accuracy of our algorithm compared with the MATLAB command CARE, we have a random test including 200 samples (100 examples for the specified tolerance $\Delta = 0.1$ and 100 examples for the specified tolerance $\Delta = 0.01$). The test procedure is as follows:

1) Consider a state-space representation for a dynamic system
$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$
where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times(p+r)}$, $C \in \mathbb{R}^{m\times n}$, and u, x, y are input, state, output respectively;
2) Set the example number $i = 0$;
3) Choose n, m, p, r randomly and uniformly among the integers from 1 to 100;
4) Generate a random system by using MATLAB command sysrand with n, m, p, r obtained in step 3, and obtain A, B, C, D by MATLAB command unpck;
5) Partition the matrix $B = [B_1 \quad B_2]$, where $B_1 \in \mathbb{R}^{n\times p}$ and $B_2 \in \mathbb{R}^{n\times r}$;
6) Try MATLAB command CARE to solve the corresponding ARE with $E = I_n$, $S = 0$, $B = [B_1 \quad B_2]$, $Q = C^T C, R = \begin{pmatrix} -I_p & 0 \\ 0 & I_r \end{pmatrix}$, $Q = C^T C, R = \begin{pmatrix} -I_p & 0 \\ 0 & I_r \end{pmatrix}$, where I_n, I_p, and I_r are identity matrices with dimensions n, p, r, respectively. If there does not exist a stabilizing solution to the ARE, go back to step 3;
7) Use our algorithm to solve this ARE. For our algorithm, the iteration will be stopped when $\overline{\sigma}(B_1^T Z_k)^2 < \Delta = 0.1$, where Z_k is the matrix sequence defined in Theorem 3 Part (I);

TABLE I
ILLUSTRATION OF ITERATION COUNT OF OUR ALGORITHM AND ACCURACY COMPARISON WITH CARE FOR 100 RANDOM EXAMPLES $(\overline{\sigma}(B_1^T Z_k)^2 < \Delta = 0.1)$

Iterations / O(T$_i$)	2	3	4	5	6	7	8
10^{-4}	1	5	4		1	1	
10^{-5}		7	6	7	1	2	3
10^{-6}		13	10	3	2		
10^{-7}		5	3	3	1	1	
10^{-8}		1	4	7			
10^{-9}		4	3				
10^{-10}			2				
10^{-11}							
10^{-12}							
10^{-13}							
Number of Examples	1	35	32	20	5	4	3

TABLE II
ILLUSTRATION OF ITERATION COUNT OF OUR ALGORITHM AND ACCURACY COMPARISON WITH CARE FOR 100 RANDOM EXAMPLES $(\overline{\sigma}(B_1^T Z_k)^2 < \Delta = 0.01)$

Iterations / O(T$_i$)	2	3	4	5	6	7	8	9	10	11
10^{-4}										
10^{-5}										
10^{-6}		1	2	1	2	1				
10^{-7}	1	6	7	5	2	1	1		2	1
10^{-8}	1	6	2	4	1	1	2	1		
10^{-9}		2	11	1	1	1				
10^{-10}		1	5	2	1	2	1			
10^{-11}			1	4	1	2		1		1
10^{-12}			3		1					
10^{-13}			4	1	2					
Number of Examples	2	16	35	18	11	8	4	2	2	2

8) Let the solution of this ARE obtained by our algorithm be X_1 and the solution of this ARE obtained by the MATLAB command CARE be X_2;
9) Set $i = i + 1$, and let $T_i = \|X_1 - X_2\|/\|X_2\|$;
10) Repeat the steps 3–9 until $i = 100$;
11) Replace $\Delta = 0.1$ in step 7 by $\Delta = 0.01$, set $i = 0$ and repeat the steps 3–10.

For each random example in the following Table I and Table II, "Iterations" indicates the number of necessary to obtain the specified tolerance Δ in step 7; "$O(T_i)$" is the order of magnitude of T_i (defined in step 9), or the order of magnitude of the normalized comparison error between the stabilizing solutions obtained by our algorithm and the stabilizing solutions obtained by the MATLAB command CARE; "Number of examples" means the number of random examples that required "Iterations" number to converge. From Table I and Table II, we can conclude that our proposed algorithm works well in most cases with good efficiency (only 3–5 iterations in most examples) and accuracy when compared with the MATLAB command CARE.

A summary of the results in Table I and Table II is as follows:
1) In both Table I and Table II, our proposed algorithm converges in ONLY three to five iterations in most examples.
2) When the prescribed tolerance is $\Delta = 0.1$, we have $T_i < 10^{-3}$ for each random example in Table I; when the prescribed tolerance is $\Delta = 0.01$, we have $T_i < 10^{-5}$ for each random example in Table II.

Example 3: The following example illustrates that the proposed algorithm works well when other traditional methods fail. This example is a slight modification of example 6 in [5]. Choose the matrices $A \in \mathbb{R}^{21\times 21}$, $B_1 \in \mathbb{R}^{21\times 1}$, $B_2 \in \mathbb{R}^{21\times 1}$, $C \in \mathbb{R}^{21\times 21}$ in (2) as follows:

$$A = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ & \ddots & \ddots & \bigcirc & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ & \bigcirc & & \ddots & 1 \\ 0 & & \cdots & & 0 \end{pmatrix}, B_1 = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \delta \\ 0 \end{pmatrix}, B_2 = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ 1 \end{pmatrix}$$
$$C = \mathrm{diag}\{1, 0, \ldots, 0\}$$

where $\delta = 10^{-2}$ is the introduced modification. In this example, the Schur method in [5] does not produce an accurate result, similarly to the MATLAB command CARE. The algorithm proposed by Kleinman in [2] cannot be used because the term $(B_2B_2^T - B_1B_1^T)$ in the Riccati (2) is not positive semidefinite. However, the algorithm proposed in the paper easily gives the solution with the specified accuracy as will be shown next.

Firstly we attempt the Schur method in [5] with this example. Let F be defined as in (3), and S_1 be the solution to (2) by using this Schur method. We evaluate the accuracy of the solution S_1 by calculating $\rho[F(S_1)]$. The smaller $\rho[F(S_1)]$ is, the closer S_1 is to the correct solution. After calculation, we obtain $\rho[F(S_1)] = 1.9802 \times 10^3$ which is far too large. Thus, we can conclude that the Schur method in [5] fails to give a solution in this example. Similarly, let S_2 be the solution obtained by the MATLAB command CARE. For this solution, we can obtain $\rho[F(S_2)] = 1.9811 \times 10^3$ which again is too large. So we conclude that MATLAB command CARE also fails to give a solution in this example. If we were to try to refine the very coarse solution obtained by the Schur method in [5] using Kleinman's method in [2], this too fails as this algorithm diverges with each iteration (as expected). This can be shown as follows: let X_k with $k \in \mathbb{Z}_{\geq 1}$ denote the iterative series produced by the Kleinman algorithm, then we obtain $\rho[F(X_1)] = 5.7083 \times 10^2$, $\rho[F(X_2)] = 5.9959 \times 10^2, \cdots$, $\rho[F(X_{20})] = 8.2965 \times 10^7, \cdots, \rho[F(X_{100})] = 6.6206 \times 10^9$. If we use the matrix sign function method in [31] to solve this ARE and let Q be the corresponding solution, we obtain $\rho[F(Q)] = 1.235 \times 10^8$ which is again far too large.

However, when we use our proposed algorithm, we note that a unique stabilizing solution $P_4 > 0$ to (2) can be found with limiting accuracy after only 4 iterations with $\rho[F(P_4)] = \overline{\sigma}(B_1^T Z_3)^2 = 2.9205 \times 10^{-5}$.

Example 4: The following example shows that if $(A + B_1B_1^T P_{k+1}, B_2)$ is not stabilizable in step 5 of the algorithm, then there does not exist a stabilizing solution $\Pi \geq 0$ to (2). Choose

$$A = \begin{pmatrix} 1 & 100 \\ 1 & 0 \end{pmatrix}, B_1 = \begin{pmatrix} 10 \\ 0 \end{pmatrix}, B_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, C = \begin{pmatrix} 1 & 0 \end{pmatrix}.$$

We note that (C, A) has no unobservable modes on the $j\omega$-axis and (A, B_2) is stabilizable. When we run our algorithm, we find that $(A + B_1B_1^T P_1, B_2)$ is not stabilizable after one iteration since

$$P_1 = \begin{pmatrix} 0.3517 & 2.6442 \\ 2.6442 & 22.9964 \end{pmatrix}.$$

This is consistent with the fact that there does not exist a stabilizing solution $\Pi \geq 0$ to (2). In fact, we can find the unique stabilizing solution

$$\Pi = \begin{pmatrix} -0.5744 & -4.9147 \\ -4.9147 & -37.8481 \end{pmatrix}$$

which is clearly not positive semidefinite.

Example 5: The following example comes from example 1 in [25] and it gives a numerical comparison between our algorithm and the algorithms in [25], [35]. We set $\Delta = 0.01$ and choose

$$A = \begin{pmatrix} -4.0926 & -4.6586 \\ -4.6586 & -6.2726 \end{pmatrix}$$
$$B_1 = \begin{pmatrix} 3.0560 & 0 \\ 0 & 3.0560 \end{pmatrix}$$
$$B_2 = \begin{pmatrix} 3.1605 & 0.1545 \\ 0.1545 & 3.0617 \end{pmatrix}$$
$$C = \begin{pmatrix} 0.9028 & 1.0432 \\ 1.0432 & 1.3745 \end{pmatrix}.$$

We firstly try our algorithm on this example. After two iterations we have

$$P_2 = \begin{pmatrix} 0.0983 & 0.1146 \\ 0.1146 & 0.1486 \end{pmatrix}$$

with $\rho[F(P_2)] = \overline{\sigma}(B_1^T Z_1)^2 = 0.0046 < \Delta$. Now we try the algorithm in [25] with the initial guess

$$X_0 = \begin{pmatrix} 0.2304 & 0.2680 \\ 0.2680 & 0.3495 \end{pmatrix}.$$

We need five iterations to obtain

$$X_5 = \begin{pmatrix} 0.0984 & 0.1148 \\ 0.1148 & 0.1488 \end{pmatrix}$$

with $\rho[F(X_5)] = 0.0036 < \Delta$. If we try the algorithm in [25] with a different initial guess

$$\bar{X}_0 = \begin{pmatrix} 0.1008 & 0.1278 \\ 0.1278 & 0.1519 \end{pmatrix}$$

we need three iterations to obtain

$$X_3 = \begin{pmatrix} 0.0984 & 0.1148 \\ 0.1148 & 0.1488 \end{pmatrix}$$

with $\rho[F(X_3)] = 0.0036 < \Delta$. If we try the algorithm in [35] with the initial guess

$$X_0 = \begin{pmatrix} 0.2304 & 0.2680 \\ 0.2680 & 0.3495 \end{pmatrix}$$

we need two iterations to obtain

$$X_2 = \begin{pmatrix} 0.0984 & 0.1147 \\ 0.1147 & 0.1487 \end{pmatrix}$$

with $\rho[F(X_2)] = 4.65 \times 10^{-5} < \Delta$. If we try the algorithm in [35] with a different initial guess

$$\hat{X}_0 = \begin{pmatrix} 2 & 0.9 \\ 0.9 & 0.95 \end{pmatrix}$$

we need five iterations to obtain

$$X_5 = \begin{pmatrix} 0.0979 & 0.1152 \\ 0.1152 & 0.1487 \end{pmatrix}$$

with $\rho[F(X_5)] = 6.84 \times 10^{-4} < \Delta$. From this example, we can see that if the algorithms in [25], [35] are used to solve Riccati equations, the computational cost and the number of iterations will depend greatly on the choice of the initial condition. However, a good initial choice which leads to lower computational cost is not always straightforward to obtain since there appears to be no systematic procedure. For our algorithm, the initial condition can always be simply set to $P_0 = 0$.

VIII. Technical Remarks

Example 1 in the Appendix shows that $(A + B_1 B_1^T P_k - B_2 B_2^T P_k)$ is not guaranteed to always be Hurwitz for all $k \in \mathbb{Z}_{\geq 0}$. Consequently, if one were to use the Kleinman algorithm to solve (18), the Kleinman algorithm cannot always be initialized with the simple choice $G = 0$ (for $A_k + B_2 G$ Hurwitz). Also, we point out that while P_k is guaranteed to be monotonically increasing, Z_k is not guaranteed to be monotonically decreasing. The same Example 1 in the Appendix demonstrates this.

It is worth also pointing out that (18) cannot be simply replaced by one single Kleinman algorithm iteration (corresponding to a single Lyapunov equation). Example 2 in the Appendix shows that the algorithm would not converge in that situation.

IX. Conclusion

In this paper, we proposed an iterative algorithm to solve AREs arising from standard H_∞ control problems. Numerical examples have been provided to show that our algorithm has superior numerical reliability when compared with other existing methods. Furthermore, we have also proved that our algorithm has global convergence, can be initialized with a simple choice $P_0 = 0$, and has local quadratic rate of convergence. It can hence be anticipated that our algorithm can be used in situations where AREs need to be solved with high accuracy. We also presented a game theoretic interpretation of the algorithm, which underpins work currently being undertaken on tools for solving Hamilton–Jacobi–Bellman–Isaacs equations arising in game theory.

Appendix

Example 1: This is a counter-example to a conjecture that $(A + B_1 B_1^T P_k - B_2 B_2^T P_k)$ is Hurwitz and Z_k is monotonically decreasing. Choose equation shown at the bottom of following page.

Let $A_k = A + B_1 B_1^T P_k - B_2 B_2^T P_k$, where k is the iteration number. We compute the eigenvalues of A_k as follows:

$$\mathrm{spec}(A_1) = \begin{pmatrix} 0.4408 \\ -1.3058 + 2.0463i \\ -1.3058 - 2.0463i \\ -4.5065 \\ -3.2598 + 1.7983i \\ -3.2598 - 1.7983i \\ -1.9258 \end{pmatrix}$$

$$\mathrm{spec}(A_2) = \begin{pmatrix} 0.3298 \\ -1.3010 + 2.0211i \\ -1.3010 - 2.0211i \\ -4.4499 \\ -3.2633 + 1.8021i \\ -3.2633 - 1.8021i \\ -1.9645 \end{pmatrix}$$

$$\mathrm{spec}(A_3) = \begin{pmatrix} 0.1339 \\ -1.2885 + 2.0173i \\ -1.2885 - 2.0173i \\ -4.4274 \\ -3.2640 + 1.8035i \\ -3.2640 - 1.8035i \\ -1.9824 \end{pmatrix}$$

$$\mathrm{spec}(A_4) = \begin{pmatrix} -0.0840 \\ -1.2816 + 2.0202i \\ -1.2816 - 2.0202i \\ -4.4196 \\ -3.2645 + 1.8035i \\ -3.2645 - 1.8035i \\ -1.9932 \end{pmatrix}, \ldots.$$

It is clear that A_1, A_2 and A_3 are not Hurwitz because each of them has an eigenvalue with positive real part. For this example, we also note that the series Z_k is NOT monotonically decreasing as follows:

$$\mathrm{spec}(Z_1 - Z_0) = \begin{pmatrix} -20.2158 \\ -7.5453 \\ -3.0951 \\ -1.5818 \\ -1.1667 \\ -0.1587 \\ 188.6883 \end{pmatrix}$$

$$\mathrm{spec}(Z_2 - Z_1) = \begin{pmatrix} -0.1616 \\ -0.0449 \\ -0.0137 \\ -0.0120 \\ -0.0029 \\ -0.0008 \\ 108.1269 \end{pmatrix}$$

$$\mathrm{spec}(Z_3 - Z_2) = \begin{pmatrix} -0.0214 \\ -0.0036 \\ -0.0021 \\ 0.0021 \\ 0.0061 \\ 0.0138 \\ 18.8783 \end{pmatrix}$$

$$\mathrm{spec}(Z_4 - Z_3) = \begin{pmatrix} -60.2445 \\ -0.0195 \\ -0.0095 \\ -0.0004 \\ 0.0003 \\ 0.0059 \\ 0.0137 \end{pmatrix} \ldots$$

since all the above differences $Z_{k+1} - Z_k$ are not sign definite.

$$A = \begin{pmatrix} 1.0412 & -2.5067 & -1.1763 & 2.5711 & -1.7143 & 0.5608 & -0.9135 \\ -0.3583 & -0.2716 & -0.0375 & -1.3071 & 0.8196 & -1.3096 & 0.4449 \\ 0.4618 & -0.1297 & -0.1332 & 0.1534 & -1.0101 & 1.8364 & -0.1616 \\ -1.0737 & -0.4019 & -0.0681 & 0.4375 & -1.5821 & -0.6806 & 0.7751 \\ -3.4624 & -2.0130 & 1.2916 & -1.0136 & 2.4925 & -0.2745 & -0.2014 \\ -0.3231 & 2.8250 & 1.4170 & -0.8901 & 0.9891 & -0.5680 & 0.2871 \\ -0.3437 & 1.0811 & -1.5701 & -0.1852 & 0.8952 & -0.0836 & -0.4896 \end{pmatrix}$$

$$B_1 = \begin{pmatrix} 0.0200 & -0.0385 & -0.0652 \\ 0.0136 & -0.0936 & 0.0529 \\ 0.0107 & 0.1063 & 0.0514 \\ -0.0271 & -0.0025 & 0.0103 \\ 0.0203 & -0.0214 & 0.0119 \\ 0.0290 & -0.0272 & 0.0228 \\ 0.0190 & 0.0228 & -0.0248 \end{pmatrix}$$

$$B_2 = \begin{pmatrix} -0.9729 & 0.0985 \\ 0.0713 & 0.6554 \\ 0.2498 & 0.0731 \\ -0.6569 & -1.3706 \\ 0.0841 & -1.1481 \\ 0.2176 & -1.1295 \\ -0.0548 & -0.0935 \end{pmatrix}$$

$$C = \begin{pmatrix} 0.6127 & 2.0817 & 0.5971 & 1.4473 & 0.3839 & -0.3644 & 0.0087 \\ -1.4616 & 0.0258 & 0.9927 & 0.5751 & -1.2872 & 0.2052 & -1.1201 \\ 0.8081 & -0.6349 & 0.9720 & 1.0978 & 0.2316 & 1.4015 & 1.9333 \\ 0.1102 & -0.0355 & 0.3976 & -0.8304 & -0.1419 & 0.2122 & 0.3710 \end{pmatrix}.$$

Example 2: This example shows that the series P_k may not converge if we only use one Kleinman iteration (corresponding to a single Lyapunov equation) to solve (18) to obtain an approximation to the series Z_k at each iteration. Choose

$$A = \begin{pmatrix} -1.1201 & 1.3762 & 1.5297 & -0.1184 \\ 0.4080 & -2.2881 & -0.6889 & 0.7061 \\ 1.2388 & 0.7095 & -1.3935 & 2.5381 \\ 0.2662 & 1.3729 & -2.0505 & -3.2553 \end{pmatrix}$$
$$B_1 = \begin{pmatrix} -0.0224 & -0.0593 \\ -0.0438 & 0.0032 \\ 0.0176 & -0.0245 \\ -0.0386 & -0.0238 \end{pmatrix}$$
$$B_2 = \begin{pmatrix} 0.5783 & -0.5455 & 0.9183 \\ -0.5003 & -0.0943 & 0.1418 \\ 1.4989 & -1.5253 & -0.6415 \\ -0.8287 & -0.0760 & 1.0158 \end{pmatrix}$$
$$C = \begin{pmatrix} -1.9386 & 0.1690 & 0.1283 & -0.4507 \\ -0.6983 & 0.3399 & -0.4616 & -0.5974 \end{pmatrix}.$$

Let G_i denote the feedback matrices which we used to initialize the Kleinman algorithm at each iteration and i denotes the iteration number. We choose

$$G_1 = \begin{pmatrix} -1.7623 & -2.2315 & 1.9433 & -1.6744 \\ -1.9498 & -2.4289 & 1.2884 & -2.8354 \\ 3.6100 & 2.6947 & 1.6867 & -0.0434 \end{pmatrix}$$
$$G_2 = \begin{pmatrix} -23.3895 & -23.0941 & -7.5815 & -7.6735 \\ -13.2320 & -16.5666 & 3.0207 & -12.3761 \\ 37.6633 & 28.4162 & 10.7791 & 7.0096 \end{pmatrix}$$
$$G_3 = \begin{pmatrix} -675.4316 & -574.1795 & -185.9063 & -154.1866 \\ -297.0397 & -266.9176 & -55.3279 & -100.2321 \\ 860.0158 & 729.2770 & 231.0229 & 200.2686 \end{pmatrix}$$
$$G_4 = 10^6 \times \begin{pmatrix} -2.3632 & -2.0288 & -0.6084 & -0.5834 \\ -1.3482 & -1.1575 & -0.3469 & -0.3331 \\ 2.9464 & 2.5295 & 0.7585 & 0.7275 \end{pmatrix}.$$

Define F as in (3). In this example, the algorithm diverges if we use only one Kleinman iteration at each step to solve (18) as

$$\begin{aligned} \rho[F(P_1)] &= 82.2956 \\ \rho[F(P_2)] &= 1.1479 \times 10^3 \\ \rho[F(P_3)] &= 1.4475 \times 10^5 \\ \rho[F(P_4)] &= 1.9897 \times 10^9 \\ \rho[F(P_5)] &= 5.9053 \times 10^{15}. \end{aligned}$$

It is clear that the series P_k diverges if we only use one Kleinman iteration to solve (18) to obtain an approximation to the series Z_k at each iteration.

Acknowledgment

The authors wish to thank Dr. A. Varga for providing insight into some of the motivational arguments presented in the introduction.

References

[1] K. Martensson, "On the matrix Riccati equation," *Inform. Sci.*, vol. 3, pp. 17–49, 1971.

[2] D. L. Kleinman, "On an iterative technique for Riccati equation computation," *IEEE Trans. Automat. Control*, vol. AC-13, no. 1, pp. 114–115, Feb. 1968.

[3] W. M. Wonham, "On a matrix Riccati equation of stochastic control," *SIAM J. Control*, vol. 6, pp. 681–697, 1968.

[4] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. Upper Saddle River, NJ: Prentice Hall, 1996.

[5] A. J. Laub, "A Schur method for solving algebraic Riccati equation," *IEEE Trans. Automat. Control*, vol. AC-24, no. 6, pp. 913–921, Dec. 1979.

[6] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD: The John Hopkins Univ. Press, 1996.

[7] R. H. Bartels and W. Stewart, "Solution of the matrix $AX + XB = C$," *Commun. ACM*, vol. 15, pp. 820–826, 1972.

[8] S. Chen, "Necessary and sufficient conditions for the existence of positive solutions to algebraic Riccati equations with indefinite quadratic term," *Appl. Math. Optim.*, vol. 26, pp. 95–110, 1992.

[9] M. Green and D. J. N. Limebeer, *Linear Robust Control*. Englewood Cliffs, NJ: Prentice Hall, 1995.

[10] J. C. Willems, "Least squares stationary optimal control and the algebraic Riccati equation," *IEEE Trans. Automat. Control*, vol. AC-16, no. 6, pp. 621–634, Dec. 1971.

[11] R. E. Bellman, *Introduction to the Mathematical Theory of Control Process*. New York: Academic Press, 1971.

[12] R. E. Bellman, *Dymamic Programming*. Princeton, NJ: Princeton University Press, 1957.

[13] R. E. Bellman and R. E. Kalaba, *Quasilinearization and Nonlinear Boundary-Value Problems*. New York: American Elsevier, 1965.

[14] R. J. Leake and R. Liu, "Construction of suboptimal control sequences," *SIAM J. Control*, vol. 5, no. 1, pp. 54–63, 1967.

[15] J. B. Rosen, "Iterative solution of nonlinear optimal control problems," *SIAM J. Control*, vol. 4, no. 1, pp. 223–224, 1966.

[16] P. Benner, V. Herńndez, and A. Pastor, "On the Kleinman iteration for nonstabilizable systems," *J. Math. Control Signals Syst.*, vol. 16, no. 1, pp. 76–93, 2003.

[17] M. McAsey and L. Mou, "Generalized Riccati equations arising in stochastic games," *J. Linear Algebra Appl.*, vol. 416, no. 2-3, pp. 710–723, 2006.

[18] C. Luo and E. Feng, "Generalized differential Riccati equation and indefinite stochastic LQ control with cross term," *J. Appl. Math. Comput.*, vol. 13, no. 1-2, pp. 85–97, 2003.

[19] M. Ait-Rami, J. B. Moore, and X. Y. Zhou, "Indefinite stochastic linear quadratic control and generalized differential Riccati equation," *SIAM J. Control Optim.*, vol. 40, pp. 1296–1311, 2001.

[20] Y. Hu and X. Y. Zhou, "Indefinite stochastic Riccati equations," *SIAM J. Control Optim.*, vol. 42, no. 1, pp. 123–137, 2003.

[21] P. Benner, "Computational methods for linear-quadratic optimization," *Supplemento ai Rendiconti del Circolo Matematico di Palermo, Serie II*, no. 58, pp. 21–56, 1999.

[22] W. F. Arnold, III and A. J. Laub, "Generalized eigenproblem algorithms and software for algebraic Riccati equations," *Proc. IEEE*, vol. 72, pp. 1746–1754, 1984.

[23] V. Mehrmann and E. Tan, "Defect correction methods for the solution of algebraic Riccati equations," *IEEE Trans. Automat. Control*, vol. AC-33, no. 7, pp. 695–698, Jul. 1988.

[24] P. Lancaster and L. Rodman, *The Algebraic Riccati Equation*. Oxford, U.K.: Oxford University Press, 1995.

[25] L. Cherfi, H. Abou-Kandil, and H. Bourles, "Iterative method for general algebraic Riccati equation," in *Proc. ACSE'05 Conf.*, Cairo, Egypt, Dec. 19–21, 2005.

[26] C. H. Guo, "Iterative solution of a matrix Riccati equation arising in stochastic control, linear operators and matrices, the peter lancaster anniversary volume, Basel: Birkhuser," *Oper. Theory, Adv. Appl.*, vol. 130, pp. 209–221, 2002.

[27] C. H. Guo, "A note on the minimal nonnegative solution of a nonsymmetric algebraic Riccati equation," *Linear Algebra Appl.*, vol. 357, pp. 299–302, 2002.

[28] C. H. Guo, "Efficient methods for solving a nonsymmetric algebraic Riccati equation arising in stochastic fluid models," *J. Computat. Appl. Math.*, vol. 192, pp. 353–373, 2006.

[29] L. Dieci, "Some numerical considerations and Newton's method revisited for solving algebraic Riccati equations," *IEEE Trans. Automat. Control*, vol. 36, no. 5, pp. 608–616, May 1991.

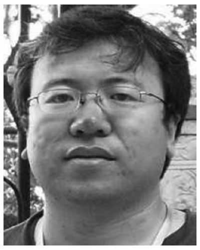
[30] T. Damm and D. Hinrichsen, "Newton's method for a rational matrix equation occurring in stochastic control," *Linear Algebra Appl.*, vol. 332–334, no. 1–3, pp. 81–109, 2001.

[31] V. Mehrmann, *The Autonomous Linear Quadratic Control Problem, Theory and Numerical Solution, Number 163 in Lecture Notes in Control and Information Sciences*. Heidelberg, Germany: Springer-Verlag, 1991.

[32] H. Abou-Kandil, G. Freiling, V. Ionescu, and G. Jank, *Matrix Riccati Equations in Control and Systems Theory*. Birkhäuser, Basel, Switzerland: , 2003.
[33] B. N. Datta, *Numerical Methods for Linear Control Systems*. New York: Elsevier Academic Press, 2004.
[34] C. Luo and E. Feng, "Indefinite stochastic LQ control with cross term via semidefinite programming," *J. Appl. Math. Comput.*, vol. 13, no. 1-2, pp. 85–97, 2003.
[35] T. Damm, *Rational Matrix Equations in Stochastic Control, Volume 297 of Lecture Notes in Control and Information Sciences*. Berlin, Germany: Springer-Verlag, 2004.
[36] V. Sima, *Algorithms for Linear-Quadratic Optimization, volume 200 in Pure and Applied Mathematics*. New York: Marcel Dekker, 1996.
[37] B. D. O. Anderson and J. B. Moore, *Linear Optimal Control*. Englewood Cliffs, N.J: Prentice-Hall, 1971.
[38] K. L. Hitz and B. D. O. Anderson, "An iterative method of computing the limiting solution of the matrix Riccati differential equation," *Proc. Inst. Elect. Eng.*, vol. 119, no. 9, pp. 1402–1406, Sep. 1972.
[39] B. D. O. Anderson, K. L. Hitz, and N. D. Diem, "Recursive algorithms for spectral factorization," *IEEE Trans. Circuits Syst.*, vol. CAS-21, no. 6, pp. 742–750, Nov. 1974.
[40] B. D. O. Anderson, "Second order convergence algorithms for the steady-state Riccati equation," *Int. J. Control*, vol. 28, no. 2, pp. 295–306, 1978.

Alexander Lanzon (SM'08) was born in Malta. He received the B.Eng. degree (with honors) in electrical engineering from the University of Malta, Msida, in 1995, and the M.S. and Ph.D. degrees in control engineering from the University of Cambridge, Cambridge, U.K., in 1997 and 2000, respectively.

Before joining the University of Manchester, Manchester, U.K., in 2006, he held academic positions at the Georgia Institute of Technology, Atlanta, and the Australian National University, Canberra. He also received earlier research training at Bauman Moscow State Technical University and industrial training at ST-Microelectronics, Ltd. and Yaskawa Denki Tokyo, Ltd. His research interests include fundamental theory of feedback systems and applications to aerospace and biological systems.



Yantao Feng (S'07) was born in China. He received the B.Eng. degree in industrial automation from Tianjin University, Tianjin, China, in 1999, the M.S. degree in navigation, guidance and control from the Shanghai Academy of Spaceflight Technology (SAST), Shanghai, China, in 2002, and is currently pursuing the Ph.D. degree at Australian National University, Canberra.

Since 2000, he has been an Aerospace Engineer with SAST. He is also associated with National ICT Australia, Ltd.. His research interests include robust control theory, game theory, periodic control systems, and complex networks.

Mr. Feng received the Australian Postgraduate Award (APA).

Brian D. O. Anderson (F'77–LF'03) was born in Sydney, Australia, on January 15, 1941. He received the B.Sc. degree in mathematics and the B.E. degree in electrical engineering from Sydney University, Sydney, Australia, in 1962 and 1964, respectively, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1966.

He is a Distinguished Professor with the Australian National University, Canberra, and a Distinguished Researcher with National ICT Australia.

Dr. Anderson received the IEEE Control Systems Award of 1997, the 2001 IEEE James H Mulligan, Jr. Education Medal, the Guillemin-Cauer Award, IEEE Circuits and Systems Society, in 1992 and 2001, the Bode Prize of the IEEE Control System Society in 1992, and the Senior Prize of the IEEE TRANSACTIONS ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING in 1986.

Michael Rotkowitz (M'00) received the B.S. degree in mathematical and computational science, the M.S. degree in statistics, and the M.S. and Ph.D. degrees in aeronautics and astronautics from Stanford University, Stanford, CA.

He also worked for J.P. Morgan Investment Management, New York, from 1996 to 1998. He was a Postdoctoral Fellow at the Royal Institute of Technology (KTH), Stockholm, Sweden, from 2005 to 2006, and at the Australian National University, Canberra, from 2006 to 2008. He is currently a Future Generation Fellow in the Department of Electrical and Electronic Engineering, University of Melbourne, Melbourne, Australia, as well as an Honorary Fellow in the Department of Mathematics and Statistics.

Dr. Rotkowitz received the George S. Axelby Outstanding Paper Award.