SIAM J. Control Optim.
Vol. 42, No. 3, pp. 1002–1012

A NESTED COMPUTATIONAL APPROACH TO THE DISCRETE-TIME FINITE-HORIZON LQ CONTROL PROBLEM[*]

GIOVANNI MARRO[†], DOMENICO PRATTICHIZZO[‡], AND ELENA ZATTONI[†]

Abstract. A new algorithmic setting is proposed for the discrete-time finite-horizon linear quadratic (LQ) optimal control problem with constrained or unconstrained final state, no matter whether the problem is cheap, singular, or regular. The proposed solution, based on matrix pseudoinversion, is completed and made practically implementable by a nesting procedure for welding optimal subarcs that enables arbitrary enlargement of the control time interval.

Key words. linear quadratic problems, algorithms, discrete-time systems, pseudoinversion

AMS subject classifications. 49N05, 49N10, 93C05, 93C62

DOI. 10.1137/S0363012901384429

1. Introduction.

The linear quadratic (LQ) and linear quadratic regulator (LQR) optimal control problems have been extensively studied and are now well settled in the literature. In particular, the regular LQ and LQR problems are exhaustively treated in many well-known specialist books and textbooks, such as, for instance, those by Bryson and Ho [3], Kwakernaak and Sivan [12], Anderson and Moore [1], Chen and Francis [5], and Lewis and Syrmos [13]. Due to increasing interest in H_2 and H_∞ control problems, more recently much effort has been made to extend the solutions of LQ and LQR problems also to cheap and singular cases. In general, these new contributions concern the infinite-horizon problems, where optimality jointly with internal stability is required. The basic properties and algorithms referring to cheap and singular optimal control have been investigated by Arnold and Laub in [2], Van Dooren in [23], Ionescu, Oară, and Weiss in [11, 10, 9], Geerts in [6], Saberi, Sannuti, Chen, and Stoorvogel in [17, 18, 4, 22], and Soethoudt, Trentelman and Ran in [21, 16]. Extension to cheap and singular problems is obtained by using matrix pencils or linear matrix inequalities.

The above contributions mainly refer to continuous- or discrete-time *infinite-horizon* LQR problems. Conversely, this work focuses on the *finite-horizon* case for discrete-time systems. It applies to cheap, singular, or regular problems. Considering a *sharp* constraint on the final state, this paper extends the results given in [16], where the infinite-horizon case with *asymptotic* endpoint constraints is investigated. However, in the present paper, the requirement that the cost functions are positive semidefinite is still present, while in [16] it is not.

From the algorithmic viewpoint, the present work differs from those of the abovementioned papers and is inspired by the system structure algorithm introduced by Silverman in his well-known contribution [20]; see also [19, 8]. In the present work, the solution is achieved by the straightforward technique of pseudoinverting a particular system matrix as in [20], but the time interval considered can be arbitrarily

[*] Received by the editors February 6, 2001; accepted for publication (in revised form) November 28, 2002; published electronically June 25, 2003. This work was partially supported by the Italian Ministry of University and Research.

http://www.siam.org/journals/sicon/42-3/38442.html

[†] Dipartimento di Elettronica, Informatica e Sistemistica, Università di Bologna, viale Risorgimento, 2, 40136 Bologna, Italy (gmarro@deis.unibo.it, ezattoni@deis.unibo.it).

[‡] Dipartimento di Ingegneria dell'Informazione, Università di Siena, via Roma, 77, 53100 Siena, Italy (prattichizzo@dii.unisi.it).

enlarged through a suitable multilevel computational scheme; this enables overcoming any dimensionality constraint thus achieving a certain interest with respect to computational practice.

The following notation will be used: $\mathbb{R}^n$ stands for the set of all n-tuples of real numbers, A' and $A^{\#}$ are used for the transpose and pseudoinverse of matrix A, respectively, and I stands for the identity matrix with a suitable dimension.

2. Statement of the problem. Consider the linear discrete-time-invariant dynamical system

$$\begin{aligned} x(k+1) &= A\,x(k) + B\,u(k), \quad x(0)=x_0, \\ e(k) &= C\,x(k) + D\,u(k), \end{aligned} \tag{1}$$

with $x \in \mathbb{R}^n$, $u \in \mathbb{R}^p$, $e \in \mathbb{R}^q$, $k \in [0, N-1]$, the cost function

$$J := \sum_{k=0}^{N-1} e(k)'e(k) + x(N)'Z'Z\,x(N), \tag{2}$$

and the constraint on the final state

$$G\,x(N) = y_f, \quad y_f \in \mathbb{R}^r. \tag{3}$$

Constraint (3) is assumed to be *feasible*; i.e., the initial state x_0, the constraint on the final state (G, y_f), and the number of steps N of the control time interval are assumed to be such that at least one N-step state trajectory from x_0 to an $x(N)$ satisfying (3) exists. It is worth noting that the final state can be completely assigned by assuming $G=I$ in (3); in this case, the terminal cost becomes irrelevant to the problem solution, and in (2) matrix Z can be assumed to be zero. On the other hand, assumptions $G=0$ and $y_f=0$ correspond to consider the final state as completely free. The discrete-time finite-horizon LQ optimal control problem with constrained final state can be stated as follows.

PROBLEM 1. *Consider the dynamic system* (1)*, and find a control sequence*[1] $u(k)$ $(k=0,1,\dots,N-1)$ *such that the cost function* (2) *is minimized under the constraint* (3)*.*

The solution of the above problem is given with no distinctions among cheap $(D'D=0)$, singular $(\det(D'D)=0)$, and regular cases $(\det(D'D) \neq 0)$. To this aim, it is convenient to introduce the following notation for the sequences of controls and of extended outputs (i.e., of the outputs completed with the square root of the terminal cost):

$$u_N := \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-1) \end{bmatrix}, \quad e_N := \begin{bmatrix} e(0) \\ e(1) \\ \vdots \\ e(N-1) \\ Z\,x(N) \end{bmatrix}. \tag{4}$$

Note that, owing to the above definition of e_N, the cost function (2) can also be given as

$$J = \|e_N\|_2^2, \tag{5}$$

[1] The control sequence corresponding to the minimum value of cost may be nonunique, particularly if the system (1) is not left-invertible.

where $\|e_N\|_2$ is the 2-norm of vector e_N. Furthermore, let us express the final state $x(N)$ as a function of the initial state x_0 and the control sequence u_N,

$$x(N) = A^N x_0 + A^{N-1} B\, u(0) + \cdots + B\, u(N-1) = A^N x_0 + L_N\, u_N,$$

with

$$L_N := \begin{bmatrix} A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix}, \tag{6}$$

so that the constraint (3) can be written as

$$y_f - GA^N x_0 = GL_N u_N. \tag{7}$$

According to (4), (5), and (7), Problem 1 is restated as follows.

Problem 1a. *Referring to the dynamic system* (1)*, find a vector u_N such that $\|e_N\|_2$ is minimized under the constraint* (7)*.*

An algorithmic solution of Problem 1 is derived through a constructive proof of the following theorem.

Theorem 1. *Consider the dynamic system* (1)*. Assume that the constraint* (3) *is feasible. Let K be a matrix, the columns of which form a basis for* $\ker(GL_N)$*. Let*

$$T_N := -(I - K\,(B_N K)^{\#}\, B_N)\,(GL_N)^{\#}\, GA^N - K\,(B_N K)^{\#}\, A_N, \tag{8}$$

$$V_N := (I - K\,(B_N K)^{\#}\, B_N)\,(GL_N)^{\#}, \tag{9}$$

$$C_N := (I - B_N K\,(B_N K)^{\#})\,(A_N - B_N\,(GL_N)^{\#}\, GA^N), \tag{10}$$

$$D_N := (I - B_N K\,(B_N K)^{\#})\, B_N\,(GL_N)^{\#}, \tag{11}$$

with

$$A_N := \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{N-1} \\ ZA^N \end{bmatrix}, \quad B_N := \begin{bmatrix} D & 0 & \cdots & 0 \\ CB & D & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{N-2}B & CA^{N-3}B & \cdots & D \\ ZA^{N-1}B & ZA^{N-2}B & \cdots & ZB \end{bmatrix}. \tag{12}$$

Then

$$u_N^o := T_N\, x_0 + V_N\, y_f \tag{13}$$

is the optimal solution of Problem 1 *with the feasible constraint* (3)*, and vector*

$$e_N^o := C_N\, x_0 + D_N\, y_f \tag{14}$$

is the optimal extended output sequence providing the optimal cost $J^o = \|e_N^o\|_2^2$.

Proof. First, let us consider the plainer problem defined only by (1) and (2), i.e., with no constraints on the final state. The sequence of outputs $e(0), e(1), \ldots, e(N-1)$ and the square root $Z\,x(N)$ of the terminal cost are related to the input sequence $u(0), u(1), \ldots, u(N-1)$ by the set of equations

$$\begin{aligned} e(0) &= C\,x_0 + D\,u(0), \\ e(1) &= C\,(A\,x_0 + B\,u(0)) + D\,u(1), \\ &\ \vdots \\ e(N-1) &= C\,(A^{N-1}x_0 + A^{N-2}B\,u(0) + \cdots + B\,u(N-2)) + D\,u(N-1), \\ Z\,x(N) &= Z\,(A^N x_0 + A^{N-1}B\,u(0) + \cdots + B\,u(N-1)), \end{aligned}$$

which can be written in the compact form

$$e_N = A_N\, x_0 + B_N\, u_N. \tag{15}$$

The sequence of controls u_N minimizing $\|e_N\|_2$ is obtained by applying the pseudoinverse property to (15). Hence

$$u_N^o = -B_N^{\#} A_N\, x_0. \tag{16}$$

It is worth noting that, in the case of an unconstrained final state, the optimal control sequence depends solely on the initial state x_0. From (15) and (16), it follows that

$$e_N^o = (I - B_N B_N^{\#})\, A_N\, x_0.$$

Hence, in the absence of constraints on the terminal state, the theorem is proven with

$$\begin{aligned} T_N &:= -B_N^{\#} A_N, \\ C_N &:= (I - B_N B_N^{\#})\, A_N. \end{aligned}$$

In this case, of course, V_N and D_N are not defined.

The complete optimal control problem, including the constraint on the final state, can easily be solved by using Lemma 1 in the appendix. In fact, (15) and (7) are taken from (23) and (24), respectively, under the correspondences

$$\zeta = e_N, \quad \varPhi = B_N, \quad \lambda = u_N, \quad \bar\rho = A_N\, x_0, \quad \varGamma = GL_N, \quad \bar\sigma = y_f - GA^N x_0. \tag{17}$$

The optimal control sequence u_N^o, as expressed in (13), can then be derived from (25) by taking into account the correspondences in (17). Similarly, the optimal extended output sequence e_N^o, as expressed in (14), can be derived from (26). □

Remark 1. *Note that the optimal cost J^o can be expressed in terms of the matrices C_N and D_N defined in* (10) *and* (11) *as the quadratic form*

$$J^o = \begin{bmatrix} x_0 \\ y_f \end{bmatrix}' \begin{bmatrix} C_N' C_N & C_N' D_N \\ D_N' C_N & D_N' D_N \end{bmatrix} \begin{bmatrix} x_0 \\ y_f \end{bmatrix}. \tag{18}$$

The results of Theorem 1 apply whether the system (A, B) is stable or not, or stabilizable or not. However, stability of system (1) is usually required to avoid numerical problems concerning powers of dynamic matrix A arising in (12). A deep discussion on the ill-conditioning of powers of matrices with eigenvalues greater than one is provided in [7]. In the case of stabilizable systems, the following remark applies to overcome the ill-conditioning of matrices involved in Theorem 1.

Remark 2. *If the system* (1) *is stabilizable, a preliminary pole placement is performed through feedback $u(k) = \bar u(k) + Hx(k)$ such that $\bar A := A + BH$ is stable and powers of $\bar A$ can be robustly computed. Now, let $\bar u^o(k)$ be the input in* (13) *which solves the optimal control problem for $(\bar A, B, C + DH, D)$. Then the solution for the original system (A, B, C, D) is simply computed as $u^o(k) = \bar u^o(k) + H\, x^o(k)$, where $x^o(k)$ is the optimal state time history, which is the same for both systems.*

Remark 3. *Let P be a basis matrix of $\ker B_N \cap \ker GL_N$; see* (12) *and* (6). *From* (25) *of Lemma* 1 *in the appendix, it follows that all the solutions of Problem* 1 *lie on the linear variety*

$$u_N^o(\gamma) = T_N\, x_o + V_N\, y_f + P\, \gamma.$$

According to Remark 5 *in the appendix, the minimum* 2*-norm vector solving Problem* 1 *is*

$$\hat{u}_N^o = \hat{T}_N\, x_o + \hat{V}_N\, y_f,$$

with

$$\begin{aligned} \hat{T}_N &= (I - PP^{\#})\, T_N, \\ \hat{V}_N &= (I - PP^{\#})\, V_N. \end{aligned}$$

It is worth noting that, if the final state is unconstrained, i.e., $G = 0$ and $y_f = 0$ in (3), the technique herein devised proves to be an effective alternative to the solution of the difference (possibly generalized) Riccati equation. The main difference with respect to the Riccati equation is that the pseudoinversion procedure is not recursive in nature and, as already mentioned, is more general since it also works for constrained final state problems.

LQ optimal control problems with hard constraints on the final states are usually approached by means of dynamic programming techniques or gradient methods, which, although formalized in a wide nonlinear setting, always provide solutions through recursive procedures.

The results of Theorem 1 and Remarks 2 and 3 are summarized in the following algorithm solving Problem 1.

Algorithm 1.

Step 1. *Check the stability of matrix A, and stabilize it by a state feedback according to Remark* 2 *if necessary.*

Step 2. *Evaluate the matrices A_N, B_N, and L_N as defined in* (12) *and* (6). *Note that if the final state $x(N)$ is completely assigned, the computation of the last row of A_N and B_N is skipped.*

Step 3. *Evaluate a basis matrix K for* $\ker(GL_N)$ *and the matrices T_N, V_N, C_N, and D_N as defined by* (8), (9), (10), *and* (11). *Compute the basis matrix P and the matrices $\hat{T}_N$ and $\hat{V}_N$ defined in Remark* 3 *if u_N^o is required to be minimum* 2*-norm.*

Step 4. *Evaluate the optimal control sequence u_N^o as defined by* (13) *and the optimal cost J^o as defined by* (18).

3. The problem nesting procedure. In the proposed solution, the optimal control sequence is computed as a function of x_0 and y_f by pseudoinverting suitably defined matrices. Since the dimensions of the matrices to be pseudoinverted are proportional to the number of steps N of the control time interval, this technique is subject to a dimensionality constraint: N cannot be greater than a maximum which depends on the available computational capability. However, this drawback can be overcome by resorting to the contrivance of nesting problems of the same type at different levels, as described below for a two-level nesting procedure.

In what follows, extension of (4), (6), (12), (10), (11) is used to denote the variables defined on a generic N_1-step (or N_2-step) control time interval; i.e., the variable subscript is intended for denoting the control interval length.

Refer to Problem 1. For the sake of simplicity, assume that $N_1, N_2 \in \mathbb{Z}^+$ exist, satisfying the pseudoinversion dimensionality constraint and such that $N_1 N_2 = N$. Then the following corollary of Theorem 1 can be stated.

Corollary 1. *The solution of Problem* 1 *can be obtained through N_2 optimal control problems with N_1 steps and the final state sharply assigned* (first level problems) *and one optimal control problem with N_2 steps and the final state weighted* *and/or constrained as in the original problem* (second level problem)*. Let R_{N_1} be a basis matrix of the subspace* $\operatorname{im} L_{N_1}$*. Denote the initial and final states of the jth $(j = 1, \ldots, N_2 - 1)$ first level problem, respectively, by $\tilde{x}(j)$ and $\tilde{x}(j+1)$, and let them belong to the optimal state trajectory of the second level optimal control problem defined by the system*

$$\begin{aligned} \tilde{x}(j+1) &= \tilde{A}\tilde{x}(j) + \tilde{B}\alpha(j), \quad \tilde{x}(0) = x_0, \\ \tilde{e}(j) &= \tilde{C}\tilde{x}(j) + \tilde{D}\alpha(j), \end{aligned} \tag{19}$$

with

$$\begin{aligned} \tilde{A} &:= A^{N_1}, \\ \tilde{B} &:= R_{N_1}, \\ \tilde{C} &:= C_{N_1} + D_{N_1}A^{N_1}, \\ \tilde{D} &:= D_{N_1}R_{N_1}, \end{aligned} \tag{20}$$

and $\tilde{x}$, α, $\tilde{e}$ vectors of suitable dimension, by the cost function

$$\tilde{J} = \sum_{j=0}^{N_2-1} \tilde{e}(j)'\,\tilde{e}(j) + \tilde{x}(N_2)'\,Z'\,Z\,\tilde{x}(N_2) \tag{21}$$

and by the constraint

$$G\,\tilde{x}(N_2) = y_f. \tag{22}$$

Proof. Note that the optimal cost J_j of the jth first level N_1 steps problem is equal to the jth contribution to the optimal cost of the second level problem (this latter also equal to the original problem optimal cost). In fact,

$$\begin{aligned} J_j &= \begin{bmatrix} \tilde{x}(j) \\ \tilde{x}(j+1) \end{bmatrix}' \begin{bmatrix} C'_{N_1}C_{N_1} & C'_{N_1}D_{N_1} \\ D'_{N_1}C_{N_1} & D'_{N_1}D_{N_1} \end{bmatrix} \begin{bmatrix} \tilde{x}(j) \\ \tilde{x}(j+1) \end{bmatrix} \\ &= (C_{N_1}\tilde{x}(j) + D_{N_1}\tilde{x}(j+1))'\,(C_{N_1}\tilde{x}(j) + D_{N_1}\tilde{x}(j+1)), \end{aligned}$$

and

$$C_{N_1}\tilde{x}(j) + D_{N_1}\tilde{x}(j+1) = C_{N_1}\tilde{x}(j) + D_{N_1}\left(A^{N_1}\tilde{x}(j) + R_{N_1}\,\alpha(j)\right) = \tilde{C}\tilde{x}(j) + \tilde{D}\alpha(j) = \tilde{e}(j).$$

Hence the thesis is an immediate consequence of Theorem 1. □

For the nesting procedure to be convenient from the computational viewpoint, the row dimension of matrices $\tilde{C}$ and $\tilde{D}$ appearing in (19), equal to the row dimension of matrices C_{N_1} and D_{N_1}, should be reduced according to the following property.

PROPERTY 1. *Matrices $\bar{C}_{N_1} \in \mathbb{R}^{(r\times n)}$ and $\bar{D}_{N_1} \in \mathbb{R}^{(r\times n)}$ exist such that*

$$\begin{bmatrix} \bar{C}'_{N_1} \\ \bar{D}'_{N_1} \end{bmatrix} \begin{bmatrix} \bar{C}_{N_1} & \bar{D}_{N_1} \end{bmatrix} = \begin{bmatrix} C'_{N_1} \\ D'_{N_1} \end{bmatrix} \begin{bmatrix} C_{N_1} & D_{N_1} \end{bmatrix},$$

and whose row dimension is $r = \operatorname{rank}\,[C_{N_1}\ D_{N_1}] \le 2n$*.*

Matrices $\bar{C}_{N_1}$ and $\bar{D}_{N_1}$ can easily be evaluated through a standard singular value decomposition procedure. Note that their row dimension is not directly related to the value of N_1 and is $2n$ at most, while that of C_{N_1} and D_{N_1} is $q\,(N_1+1)$. On the other hand, substitution does not affect the optimal control inputs and the optimal cost for the finite-horizon N_1N_2-step LQ problem. This is due to the way matrices C_N and D_N enter the optimal cost (18).

REMARK 4. *The assumption of $N = N_1N_2$ does not affect the generality of the proposed approach because, if this is not the case, one can divide the interval N into two parts of lengths N_a and $N_b = N_1N_2$ not violating the computational capability limits. Find the cost and the optimal control matrices on the two intervals, and then simply weld the N_a and N_b intervals by optimizing the overall cost with respect to the unknown state variable $x(N_a)$. Note that $x(N_a)$ must belong to the linear variety reachable in N_a steps from x_0.*

The following algorithm, based on Corollary 1, solves Problem 1 in a nested computational framework.

ALGORITHM 2.

Step 1. *Divide the control time interval into N_2 parts of N_1 steps each, where N_1 and N_2 are such that the maximum computational capability condition is met.*

Step 2. *Consider any of the N_2 subintervals and the corresponding N_1-step optimal control problem with the final state completely assigned and zero weighting matrix Z, referred to as the first level LQ problem. Note that, at this stage, both the initial and final states are assumed to be fixed, but they are, in general, unknown. Compute the resolvent matrices T_{N_1}, V_{N_1}, C_{N_1}, and D_{N_1} for the first level problem as described in steps* 1 *and* 2 *of Algorithm* 1 *and, if convenient, reduce the row dimension of C_{N_1} and D_{N_1} according to Property* 1. *These matrices are common to all the N_1-step optimal control problems.*

Step 3. *Solve the second level problem, defined by* (19), (21), *and* (22), *with Algorithm* 1, *thus obtaining the sequence of the intermediate states (i.e., the initial/terminal state of each pair of subsequent first level subintervals) as $\tilde{x}(j)$, $j \in [0, N_2]$.*

Step 4. *Use the jth pair of initial/terminal states $\tilde{x}(j), \tilde{x}(j+1)$ to solve the jth first level N_1-step problem as described in step* 3 *of Algorithm* 1. *Note that the matrices in* (13) *and* (15) *are equal for all the N_1-step problems and have been computed at step* 1.

Step 5. *The optimal control sequence and the optimal cost for the original control problem are simply obtained by grouping in a unique vector the optimal control sequences and summing up the optimal costs of all the N_1-step optimal control subproblems.*

The nesting idea enables extending the pseudoinversion procedure to finite-horizon LQ optimal control problems with a large value of N. However, the nesting procedure is subject to a dimensionality constraint, as well, which depends on the available computational capability and limits the admissible value of N_1 and N_2.

3.1. The multilevel nesting algorithm. This section shows how to overcome the drawback on the size of N_1 and N_2 for the two-level nesting procedure by means of a multi-level nesting algorithm whose innermost procedure is represented by the simple nesting procedure of Corollary 1.

The multilevel nesting algorithm reduces the computational effort needed to solve a finite-horizon LQ problem. The multilevel procedure basically corresponds to iteratively solving the second level N_2-step optimal control problem, stated in Step 2, by means of a further nesting procedure.

It is an easy matter to verify that the simple implementation of the iteration idea leads to increasing at each level the dimension of the matrices needed to compute the optimal cost $\|e_N^o\|_2^2$. Property 1 enables overcoming the dimensionality drawback while iterating Algorithm 2 in the multilevel case. The multilevel nesting algorithm solves the N-step LQ optimal control problem as follows.

ALGORITHM 3.

Step 1. *Set* $k = 1$ *and* $N_k = N$.

Step 2. *Divide the control time interval into* $N_k = N_{k,1}\,N_{k,2}$ *steps each, where* $N_{k,1}$ *is such that the computational complexity needed to evaluate the resolvent matrices* $T_{N_{k,1}}$, $V_{N_{k,1}}$, $C_{N_{k,1}}$, *and* $D_{N_{k,1}}$ *is met. Compute these matrices.*

Step 3. *Compute the reduced-size cost matrices* $\bar{C}_{N_{k,1}} \in \mathbb{R}^{(r_k \times n)}$ *and* $\bar{D}_{N_{k,1}} \in \mathbb{R}^{(r_k \times n)}$ *as described in Property* 1.

Step 4. *State the second-level* $N_{k,2}$*-step problem as defined in* (19), (21), *and* (22) *with these reduced-size cost matrices.*

Step 5. *Solve the* $N_{k,2}$ *step problem if this is computationally possible. If not, set* $k = k + 1$ *and* $N_k = N_{k,2}$, *and go back to step* 1.

Step 6. *Move back over all the nested levels, and assess the optimal control input and the cost for the original* N*-step problem.*

Table 1 illustrates a four-level nesting algorithm. At the second and third levels, $k = 2$ and $k = 3$, only basic matrices for the innermost problems are evaluated. The two-level nested problem is completely solved only at the last nesting level, i.e., $k = 4$. It is worth noting that if matrix factorization of Property 1 were not applied, the dimensions of the matrices $C_{N_{k,1}}$ and $D_{N_{k,1}}$ would significantly increase with k. In fact, at the nesting level $k + 1$, both would be $q \prod_{j=1}^{k} N_{j,1} \times n$, while factorization implies the maximum value $r_{k+1} \times n$ with $r_{k+1} \le 2n$ at every step.

TABLE 1

A four-level nested procedure. Matrices are evaluated only for problems with $N_{1,1}$, $N_{2,1}$, $N_{3,1}$, *and* $N_{3,2}$ *steps.*

$k = 1$	$N = N_{1,1}N_{1,2}$			
$k = 2$	$\mathbf{N_1 = N_{1,1}}$	$N_{1,2} = N_{2,1}N_{2,2}$		
$k = 3$		$\mathbf{N_1 = N_{2,1}}$	$N_{2,2} = N_{3,1}N_{3,2}$	
$k = 4$			$\mathbf{N_1 = N_{3,1}}$	$\mathbf{N_2 = N_{3,2}}$

4. An example. As an illustrative example, let us consider the following constrained LQ optimal control problem for the system:

$$A = \begin{bmatrix} 0.5 & 1 & -0.4 & 0 \\ 0.1 & 0.7 & 0 & -0.5 \\ 0 & 0 & 0.4 & 0 \\ 0 & 0 & 0 & 0.6 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 1 & 0 \\ 1 & 0.5 \end{bmatrix},$$

with initial condition $x_0 = [1,\ 2,\ 3,\ 4]'$ and constrained final state and weighting matrix as

$$y_f = \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} x(N), \quad Z = \begin{bmatrix} 1 & 0 & 2 & 1 \\ 0 & 0 & 3 & 1 \end{bmatrix}.$$

Problem 1 was first solved according to Algorithm 1 implemented in Matlab© 5.3 running on a 350 MhZ Pentium© II. The corresponding CPU time was 77.72 s. Instead, the CPU time corresponding to the implementation of Algorithm 2, with $N_1 = 25$ and $N_2 = 8$, was 0.676 s. Note that the three-level nesting procedure, with $N_{1,1} = 8$, $N_{2,1} = 5$, and $N_{2,2} = 5$, yields a dramatic reduction of the CPU time: 0.16 s. The results for CPU times are summarized in Table 2.

Table 2
CPU time is compared for different nesting level algorithms.

Simple pseudoinversion	$N = 200$	CPU time $= 77.72\,s$
Two-level nesting	$N_1 = 8;\ N_2 = 25$	CPU time $= 0.676\,s$
Three-level nesting	$N_{1,1} = 8;\ N_{2,1} = 5;\ N_{2,2} = 5$	CPU time $= 0.16\,s$

It can be seen that the multilevel nesting algorithm greatly reduces the computational burden for the finite-horizon LQ optimal control problem.

The optimal cost is

$$\min_{u_{200}} J = 0.687,$$

and the final state is

$$x(200) = \begin{bmatrix} -0.4821 \\ 1.4821 \\ -0.5109 \\ 1.5109 \end{bmatrix}.$$

5. Concluding remarks. The problem considered in this paper is out of the standard scheme of LQ optimal control problems considered in the literature, not only because of the completely general approach which makes it possible to deal with cheap, singular, and regular problems with no distinction, but also because the state of the system can be sharply assigned at both the initial and final time instants. Furthermore, the proposed nesting procedure is very easily implementable thus making feasible the pseudoinversion solution of optimal control problems on a time interval of finite, but arbitrary, length.

A practical application of the algorithms described herein are the computations of convolution profiles for H_2-optimal tracking and H_2-optimal rejection of an N-step previewed signal in the nonminimum phase case. These problems were described by the authors in [14, 15], where geometric-type conditions for perfect or almost perfect tracking or rejection with stability were derived.

The possibility of using the algorithmic setting described in this paper for the solution of the standard discrete-time infinite-horizon LQR problem is under investigation.

Appendix. This appendix briefly presents the manipulations used in the proof of Theorem 1.

Lemma 1. *The problem of finding a vector λ which minimizes the 2-norm of the vector*

$$\zeta := \varPhi\,\lambda + \bar\rho \tag{23}$$

and satisfies the constraint

$$\varGamma\,\lambda = \bar\sigma, \tag{24}$$

where the matrices $\varPhi$, $\varGamma$ and the column vectors $\bar\rho$, $\bar\sigma$ are given and the constraint (24) *is assumed to be feasible (i.e., satisfying $\bar\sigma \in \operatorname{im}\varGamma$), admits as the set of all solutions the linear variety*

$$\lambda^o(\gamma) := \left(I - K\,(\varPhi K)^{\#}\,\varPhi\right)\varGamma^{\#}\bar\sigma - K\,(\varPhi K)^{\#}\,\bar\rho + P\,\gamma, \tag{25}$$

where K is a basis matrix for $\ker \mathit{\Gamma}$*, P is a basis matrix for* $\operatorname{im} K \cap \ker \mathit{\Phi}$*, and γ is a free vector parameterizing the solution λ in the column space of P.*

The corresponding ζ vector with minimum 2*-norm is given by*

$$\zeta^o := \left(I - \mathit{\Phi} K \left(\mathit{\Phi} K\right)^{\#}\right)\left(\mathit{\Phi}\mathit{\Gamma}^{\#}\bar{\sigma} + \bar{\rho}\right). \tag{26}$$

Proof. The constraint (24) can be solved with respect to λ as

$$\lambda = \mathit{\Gamma}^{\#}\,\bar{\sigma} + K\,\nu, \tag{27}$$

where ν parameterizes the solution in $\ker \mathit{\Gamma}$ whose basis matrix is K. Then the following expression for vector ζ can be obtained by substituting (27) into (23):

$$\zeta = \mathit{\Omega}\,\nu + \bar{\eta}, \tag{28}$$

where

$$\mathit{\Omega} := \mathit{\Phi}\,K, \qquad \bar{\eta} := \mathit{\Phi}\mathit{\Gamma}^{\#}\,\bar{\sigma} + \bar{\rho}\,. \tag{29}$$

The expression for ν minimizing the 2-norm of ζ is simply obtained as

$$\nu = -\mathit{\Omega}^{\#}\,\bar{\eta} + H\,\gamma, \tag{30}$$

where vector γ parameterizes the solution in $\ker \mathit{\Omega}$ whose basis matrix is H. The expression

$$\zeta^o = \left(I - \mathit{\Omega}\,\mathit{\Omega}^{\#}\right)\bar{\eta} \tag{31}$$

of the minimum norm vector ζ can then be obtained by substituting (30) into (28). Expression (26) for ζ^o is derived from (31) and (29). By substituting (30) into (27), it follows that

$$\lambda = \mathit{\Gamma}^{\#}\,\bar{\sigma} - K\mathit{\Omega}^{\#}\,\bar{\eta} + KH\,\gamma. \tag{32}$$

Moreover, since

$$\operatorname{im} H := \ker\left(\mathit{\Phi} K\right) = \ker K + \operatorname{im} Y,$$

with Y such that

$$\operatorname{im}\left(KY\right) = \operatorname{im} K \cap \ker \mathit{\Phi},$$

(32) can also be written as

$$\lambda = \mathit{\Gamma}^{\#}\,\bar{\sigma} - K\mathit{\Omega}^{\#}\,\bar{\eta} + P\,\gamma, \tag{33}$$

where P has been defined in the statement of the lemma. Finally, by substituting (29) into (33), linear variety (25) is obtained. □

REMARK 5. *The minimum* 2*-norm vector λ minimizing the cost function* (23) *is*

$$\lambda^o = \left(I - PP^{\#}\right)\left(\left(I - K\left(\mathit{\Phi} K\right)^{\#}\mathit{\Phi}\right)\mathit{\Gamma}^{\#}\bar{\sigma} - K\left(\mathit{\Phi} K\right)^{\#}\bar{\rho}\right),$$

obtained by choosing

$$\gamma = -P^{\#}\left(\left(I - K\left(\mathit{\Phi} K\right)^{\#}\mathit{\Phi}\right)\mathit{\Gamma}^{\#}\bar{\sigma} - K\left(\mathit{\Phi} K\right)^{\#}\bar{\rho}\right);$$

see (25)*. If the minimum norm is not required, a plainer expression for λ follows by assuming that $\gamma = 0$.*

REFERENCES

[1] B. Anderson and J. Moore, *Optimal Control: Linear Quadratic Methods*, Prentice–Hall International, London, 1989.

[2] W. Arnold and A. Laub, *Generalized eigenproblem algorithms and software for algebraic Riccati equations*, Proc. IEEE, 72 (1984), pp. 1746–1754.

[3] A. Bryson and Y. Ho, *Applied Optimal Control*, Blaisdell Publishing Company, Waltham, MA, 1969.

[4] B. Chen, *H_∞ Control and Its Applications*, Lecture Notes in Control and Inform. Sci. 235, Springer-Verlag, New York, 1999.

[5] T. Chen and B. Francis, *Optimal Sampled-Data Control Systems*, Springer-Verlag, London, 1995.

[6] T. Geerts, *The algebraic Riccati equation and singular optimal control: The discrete-time case*, in Proceedings of the International Symposium on Systems and Networks: Mathematical Theory and Applications (MTNS '93), Vol. 2, Regensburg, Germany, 1993, pp. 129–134.

[7] G. Golub and C. Van Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.

[8] M. Hautus and L. Silverman, *System structure and singular control*, Linear Algebra Appl., 50 (1983), pp. 369–402.

[9] C. Ionescu, V. Oară, and M. Weiss, *General matrix pencil techniques for the solution of algebraic Riccati equation: A unified approach*, IEEE Trans. Automat. Control, 42 (1997), pp. 1085–1097.

[10] V. Ionescu and C. Oară, *Generalized continuous-time Riccati theory*, Linear Algebra Appl., 232 (1996), pp. 111–130.

[11] V. Ionescu and M. Weiss, *On computing the stabilizing solution of the discrete-time Riccati equation*, Linear Algebra Appl., 174 (1992), pp. 229–238.

[12] H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems*, John Wiley & Sons, New York, 1972.

[13] F. Lewis and V. Syrmos, *Optimal Control*, John Wiley & Sons, New York, 1995.

[14] G. Marro, D. Prattichizzo, and E. Zattoni, *Convolution profiles for right-inversion of multivariable non-minimum phase discrete-time systems*, Automatica J. IFAC, 38 (2002), pp. 1695–1703.

[15] G. Marro, D. Prattichizzo, and E. Zattoni, *A unified algorithmic setting for signal-decoupling compensators and unknown input observers*, in Proceedings of the 39th IEEE Conference on Decision and Control, Sydney, Australia, 2000.

[16] A. C. M. Ran and H. L. Trentelman, *Linear quadratic problems with indefinite cost for discrete time systems*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 776–797.

[17] A. Saberi and P. Sannuti, *Cheap and singular controls for linear quadratic regulators*, IEEE Trans. Automat. Control, 32 (1987), pp. 208–219.

[18] A. Saberi, P. Sannuti, and B. Chen, *H_2 Optimal Control*, System and Control Engineering, Prentice–Hall International, London, 1995.

[19] L. Silverman, *Inversion of multivariable linear systems*, IEEE Trans. Automat. Control, 14 (1969), pp. 270–276.

[20] L. Silverman, *Discrete Riccati equations: Alternative algorithms, asymptotic properties, and system theory interpretations*, in Control and Dynamic Systems, C. Leondes, ed., Academic Press, New York, 1976, pp. 313–386.

[21] J. Soethoudt and H. Trentelman, *The regular indefinite linear-quadratic problem with linear endpoint constraints*, Systems Control Lett., 12 (1989), pp. 23–31.

[22] A. Stoorvogel and A. Saberi, *The discrete-time algebraic Riccati equation and linear matrix inequality*, Linear Algebra Appl., 274 (1998), pp. 317–365.

[23] P. Van Dooren, *A generalized eigenvalue approach for solving Riccati equations*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 121–135.