European Journal of Control (2004)10:588–601

European Journal of Control

UDU Factored Discrete-Time Lyapunov Recursions Solve Optimal Reduced-Order LQG Problems

L.G. Van Willigenburg[1,*] and W.L. De Koning[2,†]

[1]Systems and Control Group, Wageningen University, Technotron, P.O. Box 17, 6700 AA Wageningen, The Netherlands; [2]Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Mekelweg 4, 2628 CD Delft

A new algorithm is presented to solve both the finite-horizon time-varying and infinite-horizon time-invariant discrete-time optimal reduced-order linear quadratic Gaussian (LQG) problem. In both cases the first order necessary optimality conditions can be represented by two non-linearly coupled discrete-time Lyapunov equations, which run forward and backward in discrete time. The algorithm iterates these two equations forward and backward in discrete time, respectively, until they converge. In the finite-horizon time varying case the iterations start from boundary conditions and the forward and backward in time recursions are repeated until they converge. The discrete-time recursions are suitable for UDU factorisation. It is illustrated how UDU factorisation may increase both the numerical efficiency and accuracy of the recursions. By means of several numerical examples and the benchmark problem proposed by the European Journal of Control, the results obtained with the new algorithm are compared to results obtained with algorithms that iterate the strengthened discrete-time optimal projection equations forward and backward in time. The convergence properties are illustrated to be comparable. Especially when the reduced compensator dimensions are significantly smaller than those of the controlled system, the algorithm presented in this paper is more efficient.

Keywords: Non-linearly Coupled Discrete-Time Lyapunov Equations; Optimal Reduced-Order LQG Control; Restricted Complexity Controllers; UDU Factorisation

1. Introduction

Controller reduction is a vital practical issue. Two approaches to controller reduction may be distinguished, direct versus indirect design [1]. Indirect design is characterized by the fact that the design is performed in two steps, instead of one. The two steps concern either model-reduction followed by full-order controller design or full-order controller design followed by controller-reduction. A major disadvantage of these indirect approaches is that stability of the closed loop system, in general, cannot be guaranteed, and optimality, in general, is lost. Direct design on the other hand incorporates both stability of the closed loop system and optimality. Therefore if, given the design criteria, a direct design method is feasible, it should always be preferred. This paper deals with the direct design of optimal reduced-order linear quadratic Gaussian (LQG) controllers for time-invariant and time-varying discrete-time systems. This design often is a fundamental part of the design of a digital optimal control systems [2], sampled either conventionally [9], or unconventionally [6,16,17].

So far, two approaches to compute infinite-horizon optimal reduced-order LQG compensators may be

*E-mail: gerard.vanwilligenburg@wur.nl
†E-mail: w.l.dekoning@ewi.tudelft.nl

Received 26 March 2002. Accepted 24 June 2004.
Recommended by H. Hjalmarsson and I.D. Landau.

distinguished. The first one uses parameter optimisation to optimise all the parameters of the compensator. Given a compensator the associated costs can be computed by solving a Lyapunov equation [7,18]. Unless the dimensions of the compensator are very small, this method becomes infeasible, due to the large number of parameters, and the non-linear nature of the optimisation. Other approaches use the so-called optimal projection equations (OPEs), initially presented by Hyland and Bernstein [10] and Bernstein *et al*. [3]. In the discrete-time case the OPE were strengthened [8,18,19] resulting in so-called strengthened discrete-time optimal projection equations (SDOPEs). These equations represent first-order necessary optimality conditions. Initially the algorithms based on the OPE were homotopy algorithms [14,15]. These algorithms solve a series (family) of problems and are therefore usually not very efficient. In the discrete-time case, based on results obtained for full-order compensation of discrete-time systems with white parameters [7], algorithms were presented that *iterate the first-order necessary optimality conditions forward and backward in discrete time*. These algorithms constitute a generalisation of the algorithm which solves the discrete-time estimation and control Riccati equations of full-order LQG control through forward and backward in time recursion. As demonstrated in [18] these algorithms are generally much more efficient than homotopy algorithms. Apart from these algorithms, to the best knowledge of the authors, no other algorithms have been presented in the literature that use forward and backward in time recursion of discrete-time equations representing first order necessary optimality conditions.

In this paper, a third approach to compute discrete-time optimal reduced-order LQG compensators is presented. Instead of the SDOPE the formulation of the necessary optimality conditions in terms of two non-linearly coupled recursive discrete-time Lyapunov equations is used. Like the SDOPE these equations are iterated forward and backward in discrete time to solve both the finite horizon time-varying and infinite horizon time-invariant optimal reduced-order LQG problem. As demonstrated in this paper the Lyapunov equations allow for UDU factorisation of the recursions, as opposed to the SDOPE. This paper illustrates how UDU factorisation enhances both the numerical efficiency and accuracy of the recursions. Furthermore the forward and backward in time recursions of the two Lyapunov equations guarantee the desired nonnegative definiteness of the matrices, which determine the solution, during the recursion, as opposed to recursions of the SDOPE. Finally it will become clear that if the dimension of the LQG compensator is small compared to that of the system the forward and backward in time iteration of the two Lyapunov equations is more efficient than those of the SDOPE.

According to [5], some homotopy algorithms that may be used to solve the continuous-time infinite horizon optimal reduced-order LQG problem perform iterations during which *linearly* coupled continuous-time Lyapunov equations are solved. These iterations differ significantly from the forward and backward discrete-time recursions performed in our paper. In our case the forward and backward in time recursions apply to two *non-linearly* coupled discrete-time recursive Lyapunov equations which constitute necessary optimality conditions. In our case the algorithm consists *solemnly* of these forward and backward in time recursions. The mentioned homotopy algorithms, among other things, *solve a set of linearly coupled Lyapunov equations during each single iteration*. Finally our approach solves both the time-invariant infinite horizon reduced-order LQG problem as well as the finite horizon time-varying reduced-order LQG problem. Solving the latter problem enables optimal reduced-order LQG feedback design for non-linear systems [2].

Because they are generally much more efficient than homotopy algorithms we compare the results obtained with our new algorithm with results obtained by algorithms that iterate the SDOPE forward and backward in discrete-time [18,19]. In the finite-horizon time-varying case one example, considered in [19], is used to make the comparison. An important feature of the finite-horizon time-varying case is that both the system and compensator may have a state, input and output the dimension of which varies with time. Although the dimensions of the state, input and output could in principle be fixed to the highest one occurring, we deliberately choose not to do so. The reasons for this are threefold: (1) allowing for time-varying dimensions may result in significantly smaller minimal realizations of compensators [20]; (2) systems with time-varying dimensions arise naturally if the sampling within a digital control system is not performed synchronously [16,17]; (3) with fixed dimensions, the inverse of some matrices occurring in the equations which determine the optimal compensator, no longer exist [20].

In the infinite horizon time-invariant case the dimension of the state, input and output is fixed and constant and the algorithm is compared with the one in [18] by means of some characteristic examples taken from the 3040 which were considered in [18]. In addition we make a comparison by considering the European Journal of Control benchmark problem.

In Sections 2–5 the paper treats the finite-horizon time-varying case. In Section 6 the infinite-horizon time-invariant case is treated as a special case. Section 7 concludes the paper.

2. The Optimal Fixed-Order Discrete-Time LQG Problem

Consider the time-varying discrete-time system,

$$x_{i+1} = \Phi_i x_i + \Gamma_i u_i + v_i, \quad i = 0, 1, \dots, N-1, \tag{1.1}$$

$$y_i = C_i x_i + w_i, \quad i = 0, 1, \dots, N, \tag{1.2}$$

where $x_i \in R^{n_i}$ is the state, $u_i \in R^{m_i}$ is the control, $y_i \in R^{l_i}$ is the observation, $v_i \in R^{n_{i+1}}$ the system noise, $w_i \in R^{l_i}$ the observation noise and $\Phi_i \in R^{n_{i+1} \times n_i}$, $\Gamma_i \in R^{n_{i+1} \times m_i}$, and $C_i \in R^{l_i \times n_i}$ are real matrices. Note that the dimension n_i of the system state, the dimension m_i of the input and the dimension l_i of the output, which are all bounded, may vary over time. As a result Φ_i may not be square. The processes $\{v_i\}$ and $\{w_i\}$ are zero-mean white noise sequences with covariance $V_i \in R^{n_{i+1} \times n_{i+1}}$, $W_i \in R^{l_i \times l_i}$ respectively and cross covariance $Y_i \in R^{n_{i+1} \times l_i}$, which satisfy,

$$V_i \geq 0, \quad W_i > 0 \Rightarrow \begin{bmatrix} V_i & Y_i \\ Y_i^{\mathrm{T}} & W_i \end{bmatrix} \geq 0. \tag{1.3}$$

The initial condition $x_0 \in R^{n_0}$ is a stochastic variable with mean $\bar{x}_0 \in R^{n_0}$ and covariance $X \in R^{n_0 \times n_0}$ and is independent of $\{v_i\}$ and $\{w_i\}$. As controller the following time-varying dynamic compensator is chosen:

$$\hat{x}_{i+1} = F_i \hat{x}_i + K_i y_i, \quad i = 0, 1, \dots, N-1, \tag{2.1}$$

$$u_i = -L_i \hat{x}_i, \quad i = 0,1, \dots, N-1, \tag{2.2}$$

where $\hat{x}_i \in R^{n_i^c}$ is the compensator state. The dimension n_i^c, $i = 0, 1, \dots, N$ of the compensator state is bounded and may be time-varying. $F_i \in R^{n_{i+1}^c \times n_i^c}$, $K_i \in R^{n_{i+1}^c \times l_i}$ and $L_i \in R^{m_i \times n_i^c}$ are real matrices. Note that F_i may not be square. The initial condition $\hat{x}_0 \in R^{n_0^c}$ is deterministic. Compensator (2.1)–(2.2) is denoted by $(\hat{x}_0, F^N, K^N, L^N)$ where $F^N = \{F_i, i = 0, 1, \dots, N-1\}$, $K^N = \{K_i, i = 0, 1, \dots, N-1\}$, $L^N = \{L_i, i = 0, 1, \dots, N-1\}$.

2.1. Problem Formulation

Given the system (1.1) and (1.2) the optimal fixed-order dynamic compensation problem is to find a compensator (2.1) and (2.2), with prescribed dimensions $n_i^c, i = 0, 1, \dots, N$, which minimizes the criterion,

$$J_N(\hat{x}_0, F^N, K^N, L^N) = E\left\{ x_N^{\mathrm{T}} Z x_N + \sum_{i=0}^{N-1} \begin{bmatrix} x_i^{\mathrm{T}} & u_i^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} Q_i & M_i \\ M_i^{\mathrm{T}} & R_i \end{bmatrix} \begin{bmatrix} x_i \\ u_i \end{bmatrix} \right\} \tag{3.1}$$

and to find the minimum value of J_N where $Q_i \in R^{n_i \times n_i}$, $R_i \in R^{m_i \times m_i}$, $M_i \in R^{n_i \times m_i}$, $Z \in R^{n_N \times n_N}$ satisfy,

$$Q_i \geq 0, \quad R_i > 0 \Rightarrow \begin{bmatrix} Q_i & M_i \\ M_i^{\mathrm{T}} & R_i \end{bmatrix} \geq 0, \quad Z \geq 0. \tag{3.2}$$

3. Minimality and First-Order Necessary Optimality Conditions

The minimality of compensators plays a crucial role in optimal fixed-order LQG compensator design. One reason is that, given the desire to reduce the compensator dimensions, minimal compensators are the interesting ones. The results to be presented in this paper require a suitable combination of the results presented by Van Willigenburg and De Koning [18,19,20]. This suitable combination of results is presented in this section and therefore proofs are omitted.

The dimensions of a finite-horizon minimal compensator satisfy [20],

$$\begin{aligned} &n_i^c - l_i \leq n_{i+1}^c \leq n_i^c + m_i \quad i = 0, 1, \dots, N-1, \\ &n_0^c = 1, \quad n_N^c = 0. \end{aligned} \tag{4.1}$$

On the other hand if a finite-horizon compensator has dimensions satisfying (4.1) then it can always be chosen such that it is minimal. If in addition to (4.1) the prescribed compensator dimensions satisfy

$$n_i^c \leq n_i^m, \tag{4.2}$$

where n_i^m, $i = 0, 1, \dots, N$ are the dimensions of a minimal realisation of the optimal full-order LQG compensator, then, if the conjecture in [20] holds, the global optimal reduced-order compensator is minimal.

Definition 1. Prescribed compensator dimensions that satisfy the conditions (4.1) and (4.2) are called max–min compensator dimensions.

In [20] it is explained how to compute a minimal realisation of a finite-horizon compensator and how to select prescribed compensator dimensions which satisfy (4.1) and (4.2). Let $P'_i, S'_i,\ i = 0, 1, \ldots, N$ denote the second moment of the closed loop system and the dual closed loop system respectively which satisfy,

$$P'_{i+1} = \Phi'_i P'_i \Phi'^{\mathrm{T}}_i + V'_i \in R^{(n_{i+1}+n^c_{i+1})\times(n_{i+1}+n^c_{i+1})},\quad i = 0, 1, \ldots, N-1,\quad P'_0 = \begin{bmatrix} \overline{x_0 x_0^{\mathrm{T}}} & \bar{x}_0 \hat{x}_0^{\mathrm{T}} \\ \hat{x}_0 \bar{x}_0^{\mathrm{T}} & \hat{x}_0 \hat{x}_0^{\mathrm{T}} \end{bmatrix} \tag{5.1}$$

$$S'_i = \Phi'^{\mathrm{T}}_i S_{i+1} \Phi'_i + Q'_i \in R^{(n_i+n^c_i)\times(n_i+n^c_i)},\quad i = 0, 1, \ldots, N-1,\quad S'_N = \begin{bmatrix} Z & 0 \\ 0 & 0 \end{bmatrix}, \tag{5.2}$$

where

$$\Phi'_i = \begin{bmatrix} \Phi_i & -\Gamma_i L_i \\ K_i C_i & F_i \end{bmatrix} \in R^{(n_{i+1}+n^c_{i+1})\times(n_i+n^c_i)}, \tag{6.1}$$

$$Q'_i = \begin{bmatrix} Q_i & -M_i L_i \\ -L_i^{\mathrm{T}} M_i^{\mathrm{T}} & L_i^{\mathrm{T}} R_i L_i \end{bmatrix} \in R^{(n_i+n^c_i)\times(n_i+n^c_i)},\quad i = 0, 1, \ldots, N-1 \tag{6.2}$$

$$V'_i = \begin{bmatrix} V_i & Y_i K_i^{\mathrm{T}} \\ K_i Y_i^{\mathrm{T}} & K_i W_i K_i^{\mathrm{T}} \end{bmatrix} \in R^{(n_{i+1}+n^c_{i+1})\times(n_{i+1}+n^c_{i+1})},\quad i = 0, 1, \ldots, N-1 \tag{6.3}$$

To state the main theorem consider the partitionings,

$$P'_i = \begin{bmatrix} P^1_i & P^{12}_i \\ P^{12^{\mathrm{T}}}_i & P^2_i \end{bmatrix},\quad P^1_i \in R^{n_i\times n_i},\quad P^{12}_i \in R^{n_i\times n^c_i},\quad P^2_i \in R^{n^c_i\times n^c_i},\ i = 0, 1, \ldots, N \tag{7.1}$$

and

$$S'_i = \begin{bmatrix} S^1_i & S^{12}_i \\ S^{12^{\mathrm{T}}}_i & S^2_i \end{bmatrix},\quad S^1_i \in R^{n_i\times n_i},\quad S^{12}_i \in R^{n_i\times n^c_i},\quad S^2_i \in R^{n^c_i\times n^c_i},\ i = 0, 1, \ldots, N \tag{7.2}$$

and furthermore let P^+ denote the Moore–Penrose pseudo inverse of P.

Theorem 1. A compensator $(\hat{x}_0, F^N, K^N, L^N)$ satisfies the first-order necessary optimality conditions associated to the optimal fixed-order compensation problem and is minimal if and only if

$$F_i = H_{i+1}[\Phi_i - K^0_i C_i - \Gamma_i L^0_i] G_i^{\mathrm{T}} \in R^{n^c_{i+1}\times n^c_i},\quad i = 0, 1, \ldots, N-1, \tag{8.1}$$

$$K_i = H_{i+1} K^0_i \in R^{n^c_{i+1}\times l_i},\quad i = 0, 1, \ldots, N-1, \tag{8.2}$$

$$L_i = L^0_i G_i^{\mathrm{T}} \in R^{m_i\times n^c_i},\quad i = 0, 1, \ldots, N-1, \tag{8.3}$$

$$\hat{x}_0 = H_0 \bar{x}_0 \in R^{n^c_0}, \tag{8.4}$$

where

$$K^0_i = (\Phi_i P_i C_i^{\mathrm{T}} + Y_i)(C_i P_i C_i^{\mathrm{T}} + W_i)^{-1},\quad i = 0, 1, \ldots, N-1 \tag{9.1}$$

$$L^0_i = (\Gamma_i^{\mathrm{T}} S_{i+1} \Gamma_i + R_i)^{-1}(\Gamma_i^{\mathrm{T}} S_{i+1} \Phi_i + M_i^{\mathrm{T}}),\quad i = 0, 1, \ldots, N-1 \tag{9.2}$$

$$G_i = P^{2^+}_i P^{12^{\mathrm{T}}}_i \in R^{n^c_i\times n_i},\quad i = 0, 1, \ldots, N \tag{9.3}$$

$$H_i = -S^{2^+}_i S^{12^{\mathrm{T}}}_i \in R^{n^c_i\times n_i},\quad i = 0, 1, \ldots, N \tag{9.4}$$

with

$$P_i = P^1_i - \hat{P}_i \in R^{n_i\times n_i},\quad i = 0, 1, \ldots, N \tag{10.1}$$

$$S_i = S^1_i - \hat{S}_i \in R^{n_i\times n_i},\quad i = 0, 1, \ldots, N \tag{10.2}$$

$$\hat{P}_i = P^{12}_i P^{2^+}_i P^{12^{\mathrm{T}}}_i \in R^{n^c_i\times n^c_i},\quad i = 0, 1, \ldots, N \tag{10.3}$$

$$\hat{S}_i = S^{12}_i S^{2^+}_i S^{12^{\mathrm{T}}}_i \in R^{n^c_i\times n^c_i},\quad i = 0, 1, \ldots, N \tag{10.4}$$

$$\operatorname{rank}(\hat{P}_i) = \operatorname{rank}(\hat{S}_i) = n^c_i,\quad i = 0, 1, \ldots, N \tag{10.5}$$

Observe from (5.1) and (5.2) that the conditions in Theorem 1 represent a two point boundary value problem (TPBVP). Furthermore from (5) and (10) $P^1_i, P^2_i, S^1_i, S^2_i, P_i, S_i, \hat{P}_i, \hat{S}_i, i = 0, 1, \ldots, N$ are all symmetric nonnegative matrices since $P'_i, S'_i,\ i = 0, 1, \ldots, N$ are nonnegative symmetric because they are second moment matrices.

4. The UDU Factored Numerical Algorithm

4.1. Description of the Algorithm

Equations (5.1) and (5.2) are coupled recursive Lyapunov equations that run forward and backward in time respectively. They apply for arbitrary compensators. A minimal optimal compensator has to satisfy Eqs (8.1)–(8.4) which through Eqs (9.1)–(10.5) introduces additional *non-linear* couplings in Eqs (5.1) and (5.2).

The algorithm starts with a choice, described in Section 4.4, of nonnegative symmetric initial values for $P_i', i = 0, 1, \ldots, N$ and $S_i', i = 0, 1, \ldots, N-1$. Note from (5.2) that S_N' is fully determined by Z. Then the backward in time recursion of Eq. (5.2) and forward in time recursion of Eq. (5.1) can be represented as follows:

$$P_i', S_{i+1}' \overset{(5),(8),(9),(10)}{\longrightarrow} F_i, K_i, L_i \overset{(6.1),(6.2)}{\longrightarrow} \Phi_i', Q_i' \overset{(5.2)}{\longrightarrow} S_i', \quad i = 0, 1, \ldots, N-1, \tag{11.1}$$

$$S_0', \bar{x}_0 \overset{(8.4),(9.4)}{\longrightarrow} \hat{x}_0 \overset{(5.1)}{\longrightarrow} P_0', \tag{11.2}$$

$$P_i', S_{i+1}' \overset{(5),(8),(9),(10)}{\longrightarrow} F_i, K_i, L_i \overset{(6.1),(6.3)}{\longrightarrow} \Phi_i', V_i' \overset{(5.1)}{\longrightarrow} P_{i+1}', \quad i = 0, 1, \ldots, N-1. \tag{11.3}$$

From the matrices at the left of each arrow, using the equations mentioned above the arrow, the matrices on the right of the arrow are computed. The matrices on the right of an arrow replace previous values that have been computed. The algorithm consists of repeated application of (11.1)–(11.3) respectively, until convergence is reached, that is, when new values of $P_i', S_i', i = 0, 1, \ldots, N$ become identical to the previous ones. In that case if the rank conditions (10.5) are satisfied a solution of the TPBVP (5.1)–(10.5) is found. These rank conditions are satisfied if during the recursions of the coupled Lyapunov equations no loss of rank has occurred.

Remark 1. The recursions of the coupled Lyapunov equations guarantee the desired positive nonnegative definiteness of the matrices $P_i, S_i, \hat{P}_i, \hat{S}_i$ during recursion, as opposed to recursions of the SDOPE. On the other hand recursions of the SDOPE can recover the rank of $\hat{P}_i, \hat{S}_i$ which is desirable if, during the recursions, a loss of rank occurs. It follows from the first-order necessary optimality conditions that these two desired properties exclude one another. From the results obtained for a large number of examples in [18,19] and the results presented in Sections 5 and 6 of this paper, it seems that, in practice, the lack of one of these two properties is hardly ever a problem.

4.2. Advantages and Implementation of UDU Factorisation

Given $\Phi_i', Q_i', V_i', \; i = 0, 1, \ldots, N-1$ the computation of one recursion of the Lyapunov equations (5.1) and (5.2) is identical to the computation of a time-update of the Kalman filter. Therefore two sequential UDU factored computational schemes of Bierman [4], namely the one on p. 53 and the one on pp. 132–133, intended for Kalman filtering, can be used to perform these computations. Observe that to implement Eqs (8.1)–(10.5) ordinary and pseudo inverses of symmetric matrices have to be computed. From Eqs (16.1)–(17.6) it becomes clear that the computation of the pseudo inverses is completely circumvented by employing UDU factorisation. Furthermore the UDU factorisation enables a very efficient computation of the ordinary inverses of the symmetric matrices in (9.1) and (9.2). Finally from the extensive numerical analysis in [4] it follows that UDU factorisation enhances the numerical accuracy and efficiency of the iterations of the Lyapunov equations. The increased accuracy becomes apparent only when the iterations tend to be ill conditioned [4].

The computational schemes in [4] are only suitable for positive definite covariance matrices having constant dimensions. In our case the covariance matrices have time-varying dimensions and may be nonnegative. Fortunately the algorithms can be easily adapted to deal with both. The following three equations represent the computations performed by the two algorithms, where a tilde denotes a symmetric nonnegative matrix:

$$\tilde{P}_i = U_{\tilde{P}_i} \tilde{D}_{\tilde{P}_i} U_{\tilde{P}_i}^{\mathrm{T}}, \tag{12.1}$$

$$\tilde{V}_i = U_{\tilde{V}_i} \tilde{D}_{\tilde{V}_i} U_{\tilde{V}_i}^{\mathrm{T}}, \tag{12.2}$$

$$\tilde{P}_{i+1} = \Phi_i \tilde{P}_i \Phi_i^{\mathrm{T}} + \tilde{V}_i = U_{\tilde{P}_{i+1}} \tilde{D}_{\tilde{P}_{i+1}} U_{\tilde{P}_{i+1}}^{\mathrm{T}}. \tag{12.3}$$

Equation (12.1) represents an UDU factorisation of $\tilde{P}_i$ in which $U_{\tilde{P}_i}, \tilde{D}_{\tilde{P}_i}$ are a unit upper triangular and a nonnegative diagonal matrix respectively. Similarly Eq. (12.2) represents an UDU factorisation of $\tilde{V}_i$. Our algorithm computes $U_{\tilde{P}_i}, \tilde{D}_{\tilde{P}_i}$ from $\tilde{P}_i$ and $U_{\tilde{V}_i}, \tilde{D}_{\tilde{V}_i}$ from $\tilde{V}_i$. Equation (12.3) represents one recursion of a discrete Lyapunov equation. From the matrices $U_{\tilde{P}_i}, \tilde{D}_{\tilde{P}_i}, U_{\tilde{V}_i}, \tilde{D}_{\tilde{V}_i}$ and Φ_i the algorithm computes $U_{\tilde{P}_{i+1}}, \tilde{D}_{\tilde{P}_{i+1}}$ which represents an UDU factorisation of $\tilde{P}_{i+1}$.

The fact that the inverse $U_{\tilde{P}_i}^{-1}$ of a non-singular upper triangular matrix $U_{\tilde{P}_i}$ can be computed with high accuracy and efficiency using another algorithm from [4], that is, the one on p. 65, will also be exploited. The algorithm applies to non-singular upper triangular matrices. Exploiting the fact that our matrices are *unit* upper triangular enables us to further simplify this algorithm. In addition we will exploit the fact that the inverse of a positive definite diagonal matrix $\tilde{D}_{\tilde{P}_i}$ is a diagonal matrix $\tilde{D}_{\tilde{P}_i}^{-1}$ with diagonal elements that are one over the corresponding ones in $\tilde{D}_{\tilde{P}_i}$. In particular we may compute the inverse $\tilde{P}_i^{-1}$ of $\tilde{P}_i$ in (12.1) as,

$$\tilde{P}_i^{-1} = U_{\tilde{P}_i}^{-\mathrm{T}} \tilde{D}_{\tilde{P}_i}^{-1} U_{\tilde{P}_i}^{-1}. \tag{13}$$

The actual implementation of these algorithms inside Matlab [12] was performed by the authors through mex files, the source of which is written in C-code [13]. The C-code of these mex files is available upon request.

4.3. UDU Factorisation of the Algorithm

Having computed UDU factorisations $U_{P_i'}, D_{P_i'}$ of P_i', $i = 0, 1, \ldots, N$ these can be partitioned in the same way as P_i' in (7.1),

$$U_{P_i'} = \begin{bmatrix} U_{P_i'}^1 & U_{P_i'}^{12} \\ 0 & U_{P_i'}^2 \end{bmatrix}, \quad U_{P_i'}^1 \in R^{n_i \times n_i}, \quad U_{P_i'}^{12} \in R^{n_i \times n_i^c},$$
$$U_{P_i'}^2 \in R^{n_i^c \times n_i^c}, \ i = 0, 1, \ldots, N, \tag{14.1}$$

$$D_{P_i'} = \begin{bmatrix} D_{P_i'}^1 & 0 \\ 0 & D_{P_i'}^2 \end{bmatrix}, \quad D_{P_i'}^1 \in R^{n_i \times n_i}, \quad D_{P_i'}^2 \in R^{n_i^c \times n_i^c},$$
$$i = 0, 1, \ldots, N, \tag{14.2}$$

where $U_{P_i'}^1, U_{P_i'}^2$, $i = 0, 1, \ldots, N$ are also unit upper triangular and $D_{P_i'}^1, D_{P_i'}^2$, $i = 0, 1, \ldots, N$ are also diagonal and nonnegative. From [11] note that,

$$\mathrm{rank}(\hat{P}_i) = n_i^c \quad \Leftrightarrow \quad P_i^2 > 0 \quad \Leftrightarrow \quad D_{P_i'}^2 > 0 \tag{15.1}$$

$$\mathrm{rank}(\hat{S}_i) = n_i^c \quad \Leftrightarrow \quad S_i^2 > 0 \quad \Leftrightarrow \quad D_{S_i'}^2 > 0. \tag{15.2}$$

Using (15.1), (15.2), (10.1)–(10.5) after multiplying out the partitioned UDU factorisation (14.1) and (14.2) we obtain

$$P_i^1 = U_{P_i'}^1 D_{P_i'}^1 U_{P_i'}^{1^\mathrm{T}} + U_{P_i'}^{12} D_{P_i'}^2 U_{P_i'}^{12^\mathrm{T}}, \tag{16.1}$$

$$P_i^{12} = U_{P_i'}^{12} D_{P_i'}^2 U_{P_i'}^{2^\mathrm{T}}, \tag{16.2}$$

$$P_i^2 = U_{P_i'}^2 D_{P_i'}^2 U_{P_i'}^{2^\mathrm{T}}, \tag{16.3}$$

$$P_i = U_{P_i'}^1 D_{P_i'}^1 U_{P_i'}^{1^\mathrm{T}}, \tag{16.4}$$

$$\hat{P}_i = U_{P_i'}^{12} D_{P_i'}^2 U_{P_i'}^{12^\mathrm{T}}, \tag{16.5}$$

$$G_i = U_{P_i'}^{2^{-\mathrm{T}}} U_{P_i'}^{12^\mathrm{T}}. \tag{16.6}$$

Dually after partitioning and multiplying out the UDU factorisations $U_{S_i'}, D_{S_i'}$ of S_i', we obtain

$$S_i^1 = U_{S_i'}^1 D_{S_i'}^1 U_{S_i'}^{1^\mathrm{T}} + U_{S_i'}^{12} D_{S_i'}^2 U_{S_i'}^{12^\mathrm{T}}, \tag{17.1}$$

$$S_i^{12} = U_{S_i'}^{12} D_{S_i'}^2 U_{S_i'}^{2^\mathrm{T}}, \tag{17.2}$$

$$S_i^2 = U_{S_i'}^2 D_{S_i'}^2 U_{S_i'}^{2^\mathrm{T}}, \tag{17.3}$$

$$S_i = U_{S_i'}^1 D_{S_i'}^1 U_{S_i'}^{1^\mathrm{T}}, \tag{17.4}$$

$$\hat{S}_i = U_{S_i'}^{12} D_{S_i'}^2 U_{S_i'}^{12^\mathrm{T}}, \tag{17.5}$$

$$H_i = -U_{S_i'}^{2^{-\mathrm{T}}} U_{S_i'}^{12^\mathrm{T}}. \tag{17.6}$$

Given (16.1)–(17.6) the repeated computation of (11.1) is performed as follows. To compute F_i, K_i, L_i in (11.1) first K_i^0, L_i^0 are computed according to Eqs (9.1) and (9.2). In (9.1) an UDU factorisation of $(C_i P_i C_i^\mathrm{T} + W_i)$ is computed first. As an input this computation uses the matrix C_i, the UDU factorisation (16.4) of P_i and an UDU factorisation of W_i. Next an UDU factorisation of the inverse of this matrix, that is, $(C_i P_i C_i^\mathrm{T} + W_i)^{-1}$, is computed according to Eq. (13). Similar arguments hold for the computation of $(\Gamma_i^\mathrm{T} S_i \Gamma_i + R_i)^{-1}$ in Eq. (9.2). From the UDU factorisations of these matrices the matrices themselves are recovered and next K_i^0, L_i^0 in (9.1) and (9.2) are computed through standard matrix calculations. The same holds for the calculation of G_i, H_i according to (16.6) and (17.6) and finally for the calculation of F_i, K_i, L_i according to Eqs (8.1)–(8.3). Having computed F_i, K_i, L_i in (11.1), via standard matrix calculations we calculate Φ_i', Q_i' according to Eqs (6.1) and (6.2). Next an UDU factorisation of Q_i' is computed. This UDU factorisation is used together with Φ_i' and the UDU factorisation of S_{i+1}' to compute an UDU factorisation of S_i' according to equation (5.2). This completes the computation of (11.1).

The computation of (11.2) is completely performed using standard matrix calculations. Finally the computation of (11.3) is dual to the computation of (11.1) and so is performed in the same manner.

4.4. Initialisation of the Algorithm

The nonnegative values of $P_i',\ i = 1, 2, \ldots, N$ and $S_i',\ i = 0, 1, \ldots, N-1$, with which the algorithm is initialised, are selected based on the following considerations. Equations (9.1)–(10.5) describe the link between the partitioned matrices which determine the Lyapunov equations and the SDOPE. The latter are linked to the standard estimation and control Riccati equations of full-order LQG control and so is the initialisation of the algorithms to solve these equations [6,19]. Therefore the initialisation of the Lyapunov equations is performed similar to the initialisation of the SDOPE, that is, $\hat{P}_i,\ i = 1, 2, \ldots, N-1$ are chosen to be random nonnegative while $P_i,\ i = 1, 2, \ldots, N$ are chosen to be zero matrices. According Eqs (16.4) and (16.5) this comes down to setting D_i^1 to zero, which makes the choice of U_i^1 irrelevant, and picking the free elements of U_i^{12}, U_i^2, D_i^2 random and positive in the case of D_i^2. Through Eqs (16.1)–(16.3) this fully determines P_i'. The matrices $S_i',\ i = 0, 1, \ldots, N-1$ are initialised the same way.

5. Numerical Issues, Examples and an Algorithm Comparison

5.1. Numerical Damping and Convergence Detection

The convergence of algorithms to compute optimal reduced-order LQG compensators, based on the SDOPE, in critical situations, is enhanced significantly by the introduction of a numerical damping [18,19]. The same holds for the algorithms presented here. In the finite horizon time-varying case the numerical damping is realised by computing:

$$P_i'^{j+1} := (1-\alpha)P_{i,(11)}'^{j+1} + \alpha P_i'^{j}, \quad i = 0, 1, \ldots, N, \quad 0 \le \alpha < 1, \tag{18.1}$$

$$S_i'^{j+1} := (1-\alpha)S_{i,(11)}'^{j+1} + \alpha S_i'^{j}, \quad i = 0, 1, \ldots, N, \quad 0 \le \alpha < 1, \tag{18.2}$$

where $P_i'^j, S_i'^j$ represents P_i', S_i' after the jth recursion of the algorithm and where $P_{i,(11)}'^j, S_{i,(11)}'^j$ represents P_i', S_i' obtained after execution of (11.1)–(11.3) during the jth recursion of the algorithm. The parameter $0 \le \alpha < 1$ is the numerical damping factor. Based on the fact that the scaling (scalar multiply) of an UDU factored matrix comes down to the scaling of the diagonal elements of the diagonal matrix we can again exploit the adapted algorithms from Bierman [4] to perform this computation.

Like in [19] convergence is assumed when the relative difference of consecutive values of trace$(P_N + S_0)$ falls repeatedly below a certain tolerance.

5.2. Numerical Examples and an Algorithm Comparison

Example 1. Consider the finite-horizon discrete-time LQG compensation problem (1.1)–(2.2), (4.1) and (4.2) with

$$\Phi_i = (1+\sin(i))\begin{bmatrix} -0.9653 & 0.7942 \\ -0.7942 & -0.9653 \end{bmatrix},$$
$$\Gamma_i = \begin{bmatrix} 0.4492 \\ 0.1784 \end{bmatrix}, \quad C_i = [0.6171\ 0.3187],$$

$$V_i = \mathrm{diag}(0.7327\ 0.8612), \quad Y_i = [-0.0677\ \ -0.05360]^{\mathrm{T}}, \quad W_i = 0.9334,$$

$$Q_i = \mathrm{diag}(0.0437\ 0.1108), \quad M_i = [-0.0859\ \ -0.0107]^{\mathrm{T}}, \quad R_i = 0.3311,$$

$$x_0 = [1\ 1]^{\mathrm{T}}, \quad X = \mathrm{diag}(0.1\ 0.1), \quad Z = \mathrm{diag}(0.1\ 0.1)$$

The spectral radius of Φ_i equals $(1 + 0.2\sin(i)).1.25$ so the system is unstable. The following prescribed compensator dimensions are max–min dimensions,

$$n_i^c = 1,\ \ i = 0, 1, 2, 3, 6, 7, 8, \quad n_i^c = 2,\ \ i = 4, 5, \quad n_i^c = 0,\ \ i = 9, \quad N = 9 \tag{19}$$

Table 1. Solutions of Example 1 for different prescribed compensator dimensions.

Equation	Costs associated to each solution					
(25)	33.5487	33.7895	34.2802			
(26)	36.7150	37.0523	37.7720	37.9787	38.0817	38.1795
(27)	35.8248	36.9570	37.1230			

The first column specifies the equation which determines the prescribed compensator dimensions $n_i^c, i = 0, 1, \ldots, N$. The remaining elements in each row represent the costs associated to each solution.

Table 2. Speed of convergence of the Lyapunov (L) equations and the SDOPE (P) for Example 1 with prescribed compensator dimensions (23) and a convergence tolerance of 10^{-8}.

Algorithm	Costs	Number of recursions before convergence/experiment index						
L	33.5487	40/4	44/10	44/11	46/9	47/2	47/13	48/6
		53/12	71/7	90/15				
P	33.5487	85/7	104/8	107/3	107/12	146/2	157/11	162/6
		175/13	197/15	207/14	218/1	224/10	274/4	
L	33.7895	191/8	196/5	219/1	228/14			
P	33.7895	439/9						
L	34.2802	85/3						
P	34.2802	152/5						

With these prescribed max–min compensators dimensions the problem is identical to the one in [19] when $\lambda = 0$. In [19] two different solutions were found using the algorithm based on the SDOPE. In our case, besides these two solutions, a third solution was found (see Table 1). The rank conditions (10.5) in each case were verified to hold.

The prescribed compensator dimensions

$$n_i^c = 1, \quad i = 0, 1, \ldots, 8, \qquad n_i^c = 0, \quad i = 9, \quad N = 9 \tag{20}$$

are also max–min dimensions. In this case our algorithm finds six solutions (see Table 1). Although the differences in costs are rather small, inspection of the matrices $P_i', S_i', i = 0, 1, \ldots, N-1$ reveals that they are truly different solutions. Again the rank conditions (10.5) in each case were verified to hold.

Finally, on purpose, the following prescribed compensator dimensions which are not max–min dimensions have been selected:

$$n_i^c = 1, \ i = 0, 1, \ldots, 6, \quad n_i^c = 2, \ i = 7, 8, \qquad n_i^c = 0, \ i = 9, \ N = 9. \tag{21}$$

In this case the global optimal compensator is not minimal because n_i^c, $i = 8, 9$ violate the conditions in Theorem 1. Then our algorithm should preferably produce non-minimal compensators, the minimal realisations of which have dimensions $n_i^{c_{\min}}, i = 0, 1, \ldots, N$ given by

$$n_i^{c_{\min}} = 1, \ i = 0, 1, 2, 3, 4, 5, 6, 8, \quad n_i^{c_{\min}} = 2, \qquad i = 7, \quad n_i^c = 0, \ i = 9, \ N = 9, \tag{22}$$

since these are the *maximal* values which satisfy the conditions in Theorem 1 and $n_i^{c_{\min}} \le n_i^c, i = 0, 1, \ldots, N$. The rank of both $\hat{P}_i, \hat{S}_i, i = 1, 2, \ldots, N-1$ determine the dimensions of a minimal realisation of a compensator satisfying the first-order necessary optimality conditions [20]. For all three solutions (see Table 1) the rank of both $\hat{P}_i, \hat{S}_i$ equals $n_i^{c_{\min}}, i = 1, 2, \ldots, N-1$.

Table 2 compares the speed of convergence of the algorithm based on the Lyapunov equations and the SDOPE. Both algorithms were executed 15 times. Each time the initial values were taken randomly, as described in Section 4.4, and were used for both algorithms. For each solution and both algorithms the number of recursions necessary to reach convergence is tabulated in ascending order followed by the experiment number which runs from 1 to 15. Observe that despite the equal initialisation of both algorithms they may generate different (locally) optimal solutions. For this example the number of recursions required by the Lyapunov equations, on average, equals 96.6 against 183.6 for the SDOPE. On the other hand one single recursion (one backward and one forward recursion) takes approximately 0.0563 seconds for the Lyapunov equations and 0.0403 seconds for the SDOPE using a pentium 866 MHz PC, and Matlab version 6.0 under Windows 2000. Although for this example one recursion of the SDOPE is computationally faster, for higher orders of the system and small orders of the compensator, the situation is the opposite, as demonstrated in the next section.

6. The Infinite-Horizon Time-Invariant Case

6.1. Description of the Algorithm

Assume that the state, output and input of the system and the compensator have a constant fixed dimension, that is, $n_i = n, m_i = m, l_i = l, n_i^c = n_c, i = 0, 1, \ldots$. Also assume that all the matrices in the problem formulation are time-invariant, that is, $\Phi_i = \Phi, \Gamma_i = \Gamma$, $C_i = C$, $V_i = V$, $W_i = W$, $Y_i = Y$, $Q_i = Q$, $R_i = R$, $M_i = M$, $F_i = F$, $K_i = K$, $L_i = L$, $i = 0, 1, \ldots, N-1$. Consider the criterion (3.1) and (3.2) divided by N, that is, divided by the final time. Then if the final time $N \to \infty$, the time-invariant infinite-horizon fixed-order LQG compensation problem is obtained. Within the time-invariant infinite-horizon fixed-order LQG compensation problem, the matrix Z and the boundary conditions, play no role anymore, and steady state solutions of the Lyapunov equations (5.1) and (5.2) determine optimal fixed-order compensators. As in the time-varying finite-horizon case, these solutions can be obtained through forward and backward in time recursion of Eqs (5.1) and (5.2). In this case however only a *single* recursion of each Lyapunov equation is performed simultaneously. Using the same notation as in (11.1)–(11.3), and disregarding the time indices and boundary conditions in the equations, these computations are represented by:

$$P', S' \overset{(5),(8),(9),(10)}{\longrightarrow} F, K, L \overset{(6.1),(6.2),(6.3)}{\longrightarrow} \Phi', Q', V' \overset{(5.1),(5.2)}{\longrightarrow} P', S'. \tag{23}$$

Starting from initial values P', S', computed as described in Section 4.4, the algorithm consists of repeated application of the computations (23), until convergence is reached, that is, when new values of P', S' become identical to the previous ones. If $P^2 > 0, S^2 > 0$ then a minimal compensator is obtained which satisfies the first-order necessary optimality conditions. The numerical damping and the convergence detection are implemented according to the description in Section 5.1 in which the time indices should now be removed.

In the infinite horizon case stability of the closed loop systems becomes an issue. Conditions which guarantee closed loop stability have been stated in [18].

6.2. Numerical Examples and an Algorithm Comparison

Algorithms for infinite-horizon time-invariant optimal reduced-order LQG compensation based on the SDOPE have been presented in [18]. In that paper the results of two computer experiments containing a total of 3040 examples were presented. In [18] the numerical damping applied in the algorithms was fixed. With this fixed numerical damping the algorithm failed to converge for a number of problems. If we select the numerical damping automatically, by increasing it if divergence or oscillations occur during the iterations, it turns out that the algorithms in [18] can be made to converge for all the 3040 examples. Using the same automatic selection of the numerical damping we verified that, apart from one example, this also applies to the algorithm presented in this paper.

In Table 3 we compare the speed of convergence of the Lyapunov and SDOPE algorithms on four characteristic examples out of the 3040 examples considered in [18]. Like in [18] the algorithms were executed 10 times. Each time the initial values were randomly selected, as described in Section 4.4, and used for both algorithms. The number of recursions necessary to reach convergence are tabulated in ascending order. For $n = 50$ one out of the ten Lyapunov algorithm runs did not converge.

The Lyapunov equations constitute two equations of dimension $n + n_c$ while the SDOPE constitute four equations of dimension n. As a result, when n_c is significantly smaller than n, Lyapunov recursions are faster. When on the other hand, n_c is close to n, then recursions of the SDOPE are faster. Furthermore the computation time of the SDOPE is more or less independent of the value of n_c. In Fig. 1 the computation time of a single recursion of each algorithm is plotted, as a function of n and n_c. Again these computation times are measured on a pentium 866 MHz PC, using Matlab version 6.0 under Windows 2000.

Table 3. Speed of convergence of the Lyapunov equations (L) and the SDOPE (P) with a convergence tolerance of 10^{-6}.

Algorithm	n	Number of recursions before convergence									
L	10	97	99	107	115	119	120	127	134	151	168
P	10	90	96	97	98	98	100	102	102	110	127
L	29	133	135	140	141	142	148	163	181	204	386
P	29	110	111	117	126	131	133	137	144	154	209
L	50	275	791	824	864	890	956	986	1104	1706	–
P	50	690	728	1013	1044	1842	2177	2208	2260	3986	4199

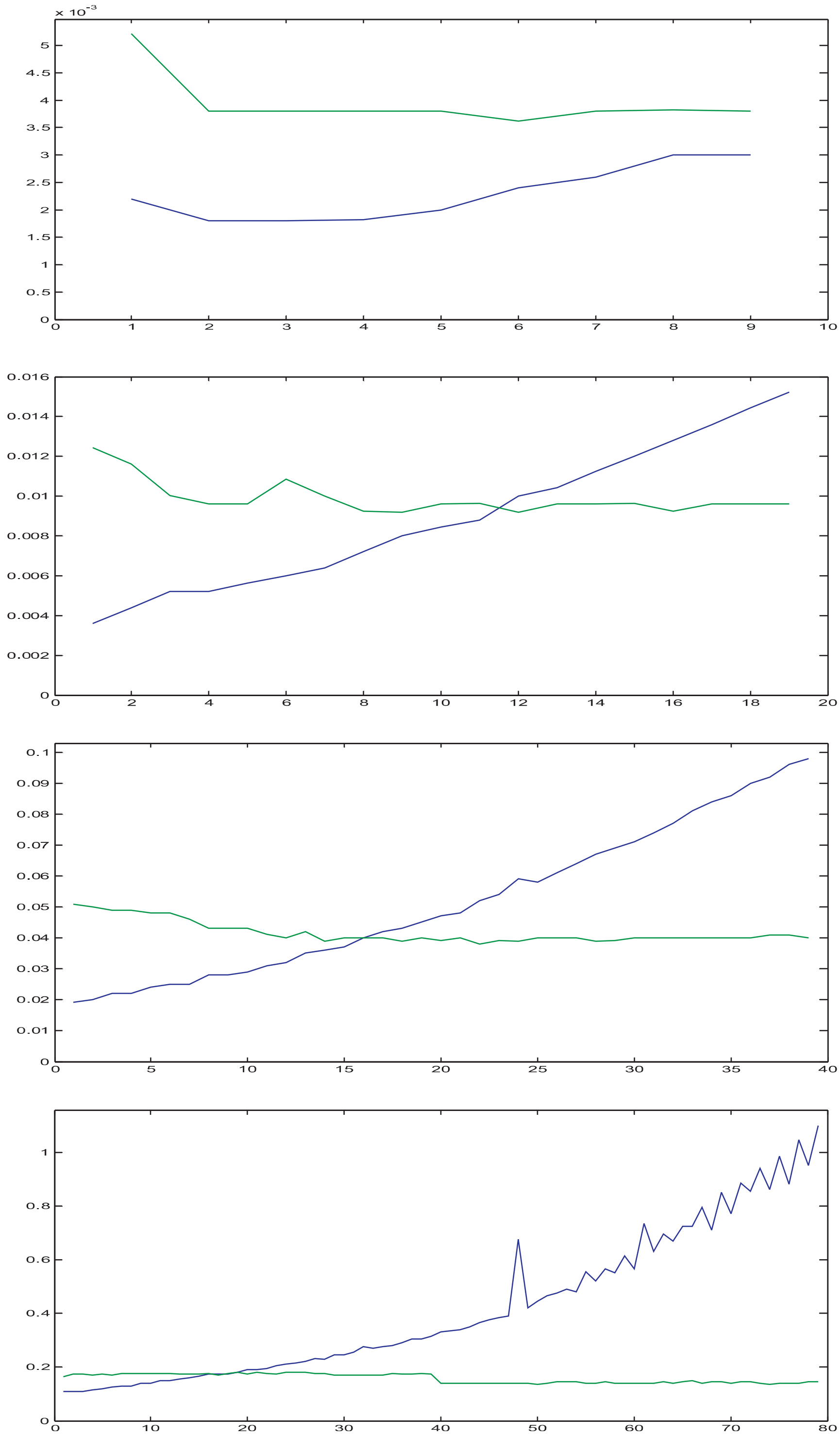

Fig. 1. Computation time [s] of one recursion of the Lyapunov equations and the SDOPE for $n = 10, 20, 40, 80$ versus $n_c = 1, 2, \ldots, n-1$.

6.3. The European Journal of Control Benchmark Problem

The benchmark problem for reduced-order controller design concerns an active suspension system used for disturbance attenuation in a large frequency band in the presence of load variations. Two linear discrete-time black box models in polynomial form have been identified separately, namely one describing the transfer function C/D from the single disturbance

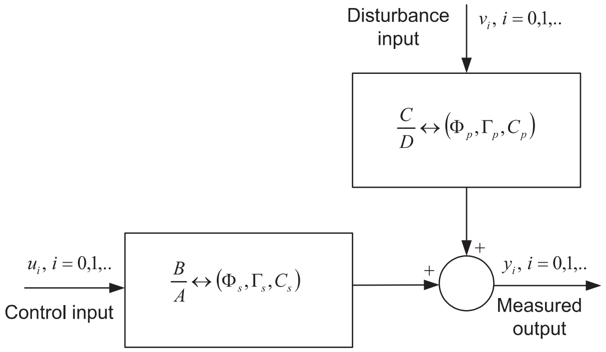

Fig. 2. Block diagram of the European Journal of Control benchmark.

input to the single measured output, and one describing the transfer function B/A from the single control input to the single measured output, see Fig. 2. Due to the separate and black box nature of the models both their states do not have a physical meaning and do not interact.

If an attempt would be made to model the system in state-space form and in continuous-time, using first principles from mechanics, then the full system state would be influenced by both the control input and the disturbance. Also then the disturbance attenuation could be formulated in terms of the full state of the system by means of a quadratic integral criterion rather then in terms of input and output sensitivity functions, as is done in the benchmark problem formulation. After computing the equivalent discrete-time optimal control problem [6] our discrete-time optimal reduced-order LQG controller design method would be very well suited to solve this problem. Unfortunately, the proposed state-space model is not available. We can however also apply our discrete-time optimal reduced-order LQG controller design when the disturbance attenuation is formulated in terms of a quadratic measure $y^{\mathrm{T}}Q_y y$ penalizing the magnitude of the residual force which is the single measured output y and when we convert the primary and secondary discrete-time transfer functions into two state space realizations (Φ_p, Γ_p, C_p) and (Φ_s, Γ_s, C_s) respectively. The states of these two state-space realizations are denoted by x_p and x_s respectively, and do not interact. This is the approach that we will adopt here. Note that our design penalizes the residual force equally over all frequencies. Therefore our design will not necessarily satisfy the design specifications of the benchmark which are imposed through bounds on the input and output sensitivity functions. Our problem formulation takes the following form. The system is described by

$$x_{i+1} = \Phi x_i + \Gamma u_i + v_i, \quad i = 0, 1, \ldots, \tag{24.1}$$

$$y_i = Cx_i + w_i, \tag{24.2}$$

where $x \in R^n$ is the state, $u \in R^m$ is the control $y \in R^l$ is the output, Φ, Γ, C are the system matrices. v_i, $i = 0, 1, \ldots$ is a discrete-time white noise process with a covariance matrix $V \in R^{n \times n}$ representing the disturbances acting on the system and w_i, $i = 0, 1, \ldots$ is a discrete-time white noise process with a covariance matrix $W \in R^{l \times l}$ representing the measurement errors. The quadratic criterion is given by

$$J = E\left\{ \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} x_i^{\mathrm{T}} Q x_i + u_i^{\mathrm{T}} R u_i \right\}. \tag{25}$$

The system state x and the matrices $\Phi, \Gamma, C, Q, R, V, W$ which make up the LQG problem equal

$$\begin{gathered} x = \begin{bmatrix} x_p \\ x_s \end{bmatrix}, \quad \Phi = \begin{bmatrix} \Phi_p & 0 \\ 0 & \Phi_s \end{bmatrix}, \quad \Gamma = \begin{bmatrix} 0 \\ \Gamma_s \end{bmatrix}, \\ C = [\,C_p \quad C_s\,], \quad V = \begin{bmatrix} \Gamma_p^{\mathrm{T}} V_p \Gamma_p & 0 \\ 0 & 0 \end{bmatrix}, \\ W = W_y, \quad Q = C^{\mathrm{T}} Q_y C, \quad R = R_u \end{gathered} \tag{26}$$

where $V_p \in R^{1\times 1}$ is the scalar covariance matrix of the scalar white noise process $v_i^p, i = 0, 1, \ldots$, the disturbance input to the primary model, $W_y \in R^{1\times 1}$ is the scalar covariance matrix of the scalar white measurement noise affecting the single output y. Furthermore $Q_y \in R^{1\times 1}$ determines the scalar quadratic penalty $y^{\mathrm{T}} Q_y y$ on the single system output reflecting the disturbance attenuation and R_u determines the scalar quadratic penalty $u^{\mathrm{T}} R_u u$ on the single control variable u. In Eq. (26) the scalars V_p, W_y, Q_y, R_u are design parameters related to the magnitude and attenuation of disturbances.

Assuming the measurement errors to be small compared to the disturbances and allowing for a reasonably large control to attenuate disturbances the following choice of the design parameters was made:

$$V_p = 1, \quad W_y = 10^{-4}, \quad Q_y = 1, \quad R_u = 10^{-4}. \tag{27}$$

The optimal reduced-order LQG problem obtained in this manner, especially for intermediate and small compensator dimensions, suffers from several local minima. We suspect that these local minima relate to the large number of complex conjugate eigenvalues of the stable system matrix Φ, several of which have absolute values close to 1. Also the spectral radius of the closed loop system, in each case, is very close to 1. Roughly speaking the closed loop system is only just stable and highly oscillatory. These properties slow down the convergence of the iterative algorithms and also make them sensitive to numerical errors. Despite these properties all solutions were obtained from recursions of both the SDOPE and the Lyapunov equations, except for $n_c = 1, 2$. For the compensator dimensions $n_c = 1, 2$ constrained non-linear parameter optimization outperforms our algorithms [18]. Otherwise the drawbacks associated to high dimensional constrained non-linear parameter optimization prevail and soon renders this approach practically impossible. For $n_c = 1, 2$ the optimal compensator was computed using constrained non-linear parameter optimisation and a controllability canonical representation of the compensator. Then for $n_c = 1$ two parameters need to be optimized and for $n_c = 2$ four parameters.

Figure 3 shows the results of four computations of the compensator order n_c versus the costs.

Two of them are computed as follows. For each value $n_c = 27, 26, \ldots, 4, 3$ the Lyapunov and SDOPE algorithm were executed five times with different random initializations and the best solution was selected. A monotone decrease of the cost of the global optimal compensator should occur if the conjecture in [20] holds. Due to the existence of several local minima the costs are not identical everywhere and also not monotonically decreasing everywhere. Two other attempts were made to find the global solutions and to obtain a monotonic decrease. In this case for each value of n_c the first of the five algorithm runs of both the SDOPE and the Lyapunov equations was initialized with the best solution obtained from the previous five algorithm runs for compensator order $n_c + 1$. In case of the SDOPE this initialization is straightforward since the dimensions of $P, S, \hat{P}, \hat{S} \in R^{n \times n}$ do not depend on n_c while $\operatorname{rank}(\hat{P}) = \operatorname{rank}(\hat{S}t) = n_c$ is automatically reduced by the algorithm. In the case of the Lyapunov equations however the dimensions of $P', S' \in R^{(n+n_c) \times (n+n_c)}$ do depend on n_c. From [18] we adopt the following approach. First $P', S' \in R^{(n+n_c+1) \times (n+n_c+1)}$ are translated into $P, S, \hat{P}, \hat{S} \in R^{n \times n}$ using Eqs (10.1)–(10.4). Then a

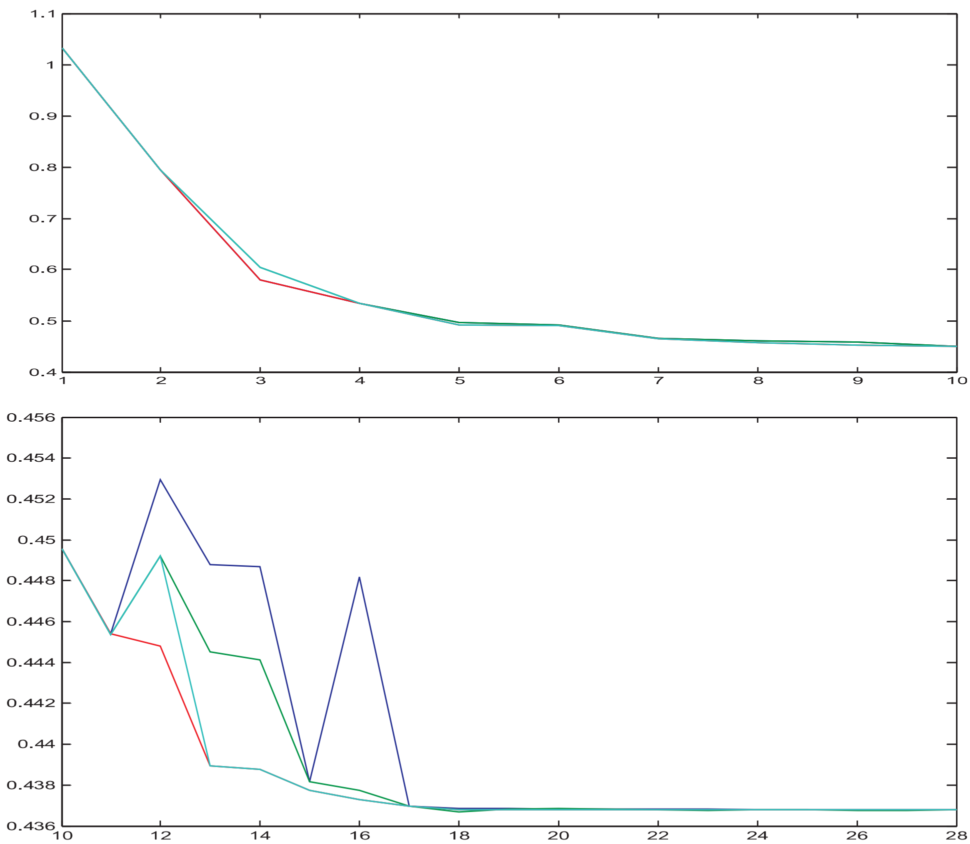

Fig. 3. Four computations of the costs versus the compensator dimension $n_c = 1, 2, \ldots, 28$ for the European Journal of Control benchmark problem.

(G, M, H) factorization of $\hat{P}\hat{S}$ is computed based on the largest n_c positive eigenvalues of $\hat{P}\hat{S}$ [18]. Then $G, H \in R^{n_c \times n}$ and from [18, Eq. (A22)]

$$P' = \begin{bmatrix} P + \hat{P} & \hat{P}H^{\mathrm{T}} \\ H\hat{P} & H\hat{P}H^{\mathrm{T}} \end{bmatrix} \in R^{(n+n_c)\times(n+n_c)},$$
$$S' = \begin{bmatrix} S + \hat{S} & \hat{S}G^{\mathrm{T}} \\ G\hat{S} & G\hat{S}G^{\mathrm{T}} \end{bmatrix} \in R^{(n+n_c)\times(n+n_c)}, \quad (28)$$

When the first of the five algorithm runs is initialised in this way a monotone decrease and the lowest costs in Fig. 3 for each value of n_c are obtained with the Lyapunov algorithm. The results obtained with the SDOPE are identical except for the costs for $n_c = 3, 12$. Note that the results obtained with the Lyapunov algorithm still do not guarantee that we have found the global minimum for each value of n_c although it is more likely.

7. Conclusions

A new algorithm has been presented to compute finite horizon time-varying and infinite horizon time-invariant discrete-time optimal reduced-order LQG compensators. The algorithm iterates two nonlinearly coupled recursive discrete-time Lyapunov equations forward and backward in time until convergence. Compared to the computation of optimal reduced-order compensators via forward and backward in time recursion of the SDOPE, this method has the following advantages. Since the Lyapunov equations constitute two matrix equations of dimension $n_i + n_i^c, i = 0, 1, \ldots, N$ $(n + n_c$ in the time-invariant infinite horizon case) while the SDOPE constitute four equations of dimension $n_i, i = 0, 1, \ldots, N, (n)$ if the dimensions of the compensator state $n_i^c(n_c)$ are small compared to those of the system, that is, $n_i(n)$, the Lyapunov equations are more efficient, assuming that the convergence properties are comparable. The results of this paper indicate that the latter is the case. Also the Lyapunov equations guarantee nonnegative definiteness of the matrices during the recursions, as opposed to the SDOPE. Furthermore the Lyapunov equations do not require the computation of the oblique projection, which is fundamental to the SDOPE, and which requires computationally expensive eigenvalue decompositions and inversions of non symmetric square matrices. As demonstrated in this paper the Lyapunov equations are suitable for UDU factorization, as opposed to the SDOPE, which cannot be guaranteed to be nonnegative during the recursions. It has been illustrated how UDU factorisation can improve the reliability and efficiency of the computations.

Recursions of the SDOPE on the other hand, are able to increase the rank of $\hat{P}_i, \hat{S}_i(\hat{P}, \hat{S})$ during the recursions, as opposed to recursions of the Lyapunov equations. The rank of both $\hat{P}_i, \hat{S}_i(\hat{P}, \hat{S})$ should be as high as possible and is constrained to be less or equal to the prescribed dimensions of the compensator state $n_i^c(n_c)$. The increase of this rank is important if, during the recursions, for some reason, a temporary loss of rank occurs. The results in this paper obtained with the Lyapunov equations indicate that this situation hardly ever occurs. The capability of the SDOPE to increase the rank is precisely the one that prevents that nonnegative definiteness of the matrices during the recursions can be guaranteed. The results of this paper also indicate that the possible negative definiteness during the recursions hardly ever presents a problem.

The mathematical proof of the convergence properties of forward and backward in time recursions of both the SDOPE and the Lyapunov equations remains an open area of research, except for the SDOPE associated to full-order compensation of systems with deterministic and white parameters. Only the latter two problems do not suffer from the possible existence of local minima which hamper the mathematical proof [18,19]. Despite this lack of mathematical proof, the practical experience obtained by the authors, with a large number of examples some of which have been presented in this paper, indicates that the algorithms have very impressive and comparable convergence properties, if automatic selection of the numerical damping is applied. Likely increasing the numerical damping makes the problem more convex locally.

References

1. Anderson BDO, Liu Y. Controller reduction: concepts and approaches. IEEE Trans Autom Control 1989; 34(8): 802–812
2. Athans M. The role and use of the stochastic linear quadratic gaussian problem in control system design. IEEE Trans Autom Control 1971; 16(6): 529–552
3. Bernstein DS, Davis LD, Hyland DC. The optimal projection equations for reduced-order discrete-time modeling, estimation and control. J Guidance Control Dyn 1986; 9(3): 288–293
4. Bierman GJ. Factorization methods for discrete sequential estimation. Acadamic Press, New York; 1977
5. Collins EG, Hodel AS. Efficient solution of linearly coupled Lyapunov equations. SIAM J Matrix Anal Appl 1997; 18(2): 291–304

6. De Koning WL. Equivalent discrete optimal control problem for randomly sampled digital control systems. Int J Syst Sci 1980; 11: 841–850
7. De Koning WL. Compensatability and optimal compensation of systems with white parameters. IEEE Trans Autom Control 1992; 37(5): 579–588
8. De Koning WL, Van Willigenburg LG. Numerical algorithms and issues concerning the discrete-time optimal projection equations for systems with white parameters. In: Proceedings UKACC Control '98 Conference, Vol. 2, Swansea UK, 1–4 September 1998; pp. 1605–1610
9. Halyo N, Caglayan AK. A seperation theorem for the stochastic sampled-data LQG problem. Int J Control 1976; 23(2): 237–244
10. Hyland DC, Bernstein DS. The optimal projection equations for reduced-order dynamic compensation. IEEE Trans Autom Control 1984; 29: 1034–1037
11. Kreindler E, Jameson A. Conditions for nonnegativeness of partitioned matrices. IEEE Trans Autom Control 1972; 17: 147–148
12. Matlab 6 User's Guide. The MathWorks Inc. 2000
13. Matlab 6 External Interface Guide. The MathWorks Inc. 2000
14. Richter S. A homotopy algorithm for solving the optimal projection equations for reduced-order dynamic compensation: existence, convergence and global optimality. In: Proceedings American Control Conference, Minneapolis, MN, June 1987; pp. 1527–1531
15. Richter SL, Collins EG. A homotopy algorithm for reduced-order compensator design using the optimal projection equations. In: Proceedings 28th Conference on Decision and Control, Tampa, FL 1989
16. Van Willigenburg LG, De Koning WL. Derivation and computation of the digital LQG regulator and tracker in case of asynchronous and aperiodic sampling. Control Theory Adv Tech 1995; 10(4): 1–13
17. Van Willigenburg LG. Digital optimal control and LQG compensation of asynchronous and aperiodically sampled non-linear systems. In: Proceedings of the European Control Conference, Vol. 1, Rome, 5–8 September 1995; pp. 496–500
18. Van Willigenburg LG, De Koning WL. Numerical algorithms and issues concerning the discrete-time optimal projection equations. Eur J Control 2000; 6: 93–110
19. Van Willigenburg LG, De Koning WL. Optimal reduced-order compensators for time-varying discrete-time systems with deterministic and white parameters. Automatica 1999; 35(1): 129–138
20. Van Willigenburg LG, De Koning WL. Minimal and non-minimal reduced-order compensators for time-varying discrete-time systems. Automatica 2002; 38(1): 157–165