Received: 18 July 2019 | Accepted: 13 July 2020

DOI: 10.1002/oca.2669

RESEARCH ARTICLE

WILEY

Distributed controller design and performance optimization for discrete-time linear systems

Daniel Viegas[1,2] | **Pedro Batista**[2] | **Paulo Oliveira**[2,3] | **Carlos Silvestre**[1]

[1]Department of Electrical and Computer Engineering, Faculty of Science and Technology, University of Macau, Taipa, China

[2]Institute for Systems and Robotics (ISR), Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal

[3]IDMEC, Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal

Correspondence
Daniel Viegas, Department of Electrical and Computer Engineering, Faculty of Science and Technology, University of Macau, Macao, China.
Email: dviegas@isr.ist.utl.pt

Funding information
Fundação para a Ciência e a Tecnologia, Grant/Award Numbers: LISBOA-01-0145-FEDER-029605, UIDB/50009/2020, UIDB/50022/2020; Macao Science and Technology Development Fund, Grant/Award Number: FDCT 0146/2019/A3; University of Macau, Grant/Award Number: MYRG2018-00198-FST

Summary

This article addresses the problem of distributed controller design for linear discrete-time systems. The problem is posed using the classical framework of state feedback gain optimization over an infinite-horizon quadratic cost, with an additional sparsity constraint on the gain matrix to model the distributed nature of the controller. An equivalent formulation is derived that consists in the optimization of the steady-state solution of a matrix difference equation, and two algorithms for distributed gain computation are proposed based on it. The first method consists in a step-by-step optimization of said difference matrix equation, and allows for fast computation of stabilizing state feedback gains. The second algorithm optimizes the same matrix equation over a finite time window to approximate asymptotic behavior and thus minimize the infinite-horizon quadratic cost. To assess the performance of the proposed solutions, simulation results are presented for the problem of distributed control of a quadruple-tank process, as well as a version of that problem scaled up to 40 interconnected tanks.

KEYWORDS

discrete-time systems, distributed control, optimal control

1 | INTRODUCTION

Distributed control can be summarily stated as the use of multiple subcontrollers instead of a single centralized one to control a complex dynamic system. Examples of such systems where distributed control can be advantageous include large-scale industrial processes,[1,2] power systems,[3,4] and formations of autonomous vehicles.[5,6] In terms of design, this can be modeled as a classical control problem subjected to sparsity constraints on both the dynamics and the controller to reflect the division into subsystems and subcontrollers.[7,8] However, these additional constraints add significant complexity to the problem.[9]

In general, there is a tradeoff between performance and scope of applicability in the solutions presented in the literature, which can be broadly divided into two categories: suboptimal solutions for general classes of problems, and optimal solutions for specific classes of sparsity patterns. For the first case, the classical formulations for optimal control yield nonconvex optimization problems, for which suboptimal solutions can be achieved through varied optimization

Carlos Silvestre is on leave from Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal.

methods.[10-12] One notable method, called sparsity-promoting control,[13,14] investigates which sparsity patterns yield the best performance on top of computing suitable controller gains. Another interesting approach focuses not only on controller design but also on the actual modeling of large-scale systems.[15,16] Finally, there is a significant body of work that relies on decomposition structures such as overlapping decomposition and border block diagonal decomposition for controller synthesis.[17,18] While the results mentioned so far are suboptimal, researchers have shown that, for certain specific structures, it is possible to obtain optimality results. For example, for systems which verify a condition called quadratic invariance, it is possible to achieve a convex, and thus tractable, formulation.[19,20] A similar result can be derived for robust control over partially ordered sets, or posets, see, for example, Reference 21. The case of spatially invariant systems is treated in References 22,23, while References 24,25 address systems with bounded nonlinear interconnections. Finally, powerful results can be found for even more specific classes of systems, such as positive systems[26] and symmetric Hurwitz state matrices.[27]

The approach proposed in this article diverges from existing work in the literature by not working directly with a finite-horizon formulation to address the optimal distributed control problem. Instead, two methods are proposed that seek to obtain well-performing steady-state controller gains through optimization over a single time step or a finite time window. It should be noted that the idea behind these methods is different from model predictive control solutions: instead of computing a new controller gain at each time step during operation, the goal here is offline computation of a single set of constant controller gains with guaranteed performance. To do so, the problem is first posed using the classical formulation of infinite-horizon quadratic cost minimization through state feedback for a discrete-time linear time-invariant (LTI) dynamic system, with an additional sparsity constraint on the controller gain to model the division into several independent subcontrollers. Then, an equivalent problem is derived that consists in the minimization of the steady-state solution of a given difference matrix equation, and two different algorithms are proposed that take advantage of this framework. The first one, called the one-step algorithm, optimizes the solution of said matrix equation at each time step, to converge to a well-performing distributed state feedback gain. The second, called the finite-horizon algorithm, optimizes the solution of the difference matrix equation over a finite time window to approximate asymptotic behavior, and thus minimize the original infinite-horizon quadratic control cost. Closed-form solutions are derived for the optimization problems used in both algorithms, allowing for a more computationally efficient and scalable implementation. To assess the performance of the proposed methods, simulation results are presented for the quadruple-tank process introduced in Reference 28, as well as a larger scale version of the same problem with 40 interconnected tanks.

The rest of the article is organized as follows. Section 2 formulates the problem of optimal distributed state feedback, from which the one-step and finite-horizon algorithms are derived in Sections 3 and 4, respectively. Simulation results are then presented in Section 5 for the quadruple-tank model, while Section 6 details further simulation results for a larger scale version of that problem. Finally, Section 7summarizes the main conclusions of the article.

1.1 | Notation

Throughout the article, $\mathbf{I}$ and $\mathbf{0}$ denote the identity matrix and a vector or matrix of zeros, respectively, both of appropriate dimensions. A block diagonal matrix $\mathbf{A}$ with n blocks $\mathbf{A}_1,\mathbf{A}_2, \dots, \mathbf{A}_n$ is denoted by $\mathbf{A} = \text{diag}(\mathbf{A}_1,\mathbf{A}_2, \dots, \mathbf{A}_n)$. Similarly, a diagonal matrix $\mathbf{B}$ whose entries are given by the entries of a vector $\mathbf{b}$ is denoted by $\mathbf{B} = \text{diag}(\mathbf{b})$. The notation $\text{vec}(\mathbf{A})$ denotes the vectorization operator, which returns a vector composed of the columns of the matrix $\mathbf{A}$. For a symmetric matrix $\mathbf{P}$, $\mathbf{P} \succ 0$, and $\mathbf{P} \succcurlyeq 0$ indicate that $\mathbf{P}$ is positive definite or semidefinite, respectively. The Kronecker product of two matrices $\mathbf{A}$ and $\mathbf{B}$ is denoted by $\mathbf{A} \otimes \mathbf{B}$.

2 | PROBLEM FORMULATION

Consider the discrete-time LTI system

$$\boldsymbol{x}(k+1) = \mathbf{A}\boldsymbol{x}(k) + \mathbf{B}\boldsymbol{u}(k), \tag{1}$$

where $\mathbf{x}(k) \in \mathbb{R}^n$ and $\mathbf{u}(k) \in \mathbb{R}^m$ are the state and input of the system, respectively, and $\mathbf{A}$ and $\mathbf{B}$ are constant real matrices of appropriate dimensions.

For a given $\mathbf{E} \in \mathbb{R}^{m\times n}$, the set of matrices that follow the sparsity pattern of $\mathbf{E}$ is defined as

$$\text{Sparse}(\mathbf{E}) = \{\mathbf{F} \in \mathbb{R}^{m\times n} : [\mathbf{E}]_{ij} = 0 \Rightarrow [\mathbf{K}]_{ij} = 0,\ i = 1, \dots, m,\ j = 1, \dots, n\}.$$

The input of (1) is defined as

$$\boldsymbol{u}(k) = -\mathbf{K}\boldsymbol{x}(k), \text{ with } \mathbf{K} \in \text{Sparse}(\mathbf{E}), \tag{2}$$

where $\mathbf{K} \in \mathbb{R}^{m\times n}$ is a constant state feedback gain matrix, which is subject to the desired sparsity pattern $\mathbf{E}$. This sparsity constraint on the controller gain can be used to model distributed control solutions, as it is shown in the simulations detailed in later sections of this article. For a quick example, suppose that $\mathbf{B} = \mathbf{I}$. Then, a diagonal gain matrix $\mathbf{K}$ represents a fully decentralized controller, in which each individual state variable of the system (1) is controlled by an independent local controller, which has access to only that respective state variable. In the same fashion, a block diagonal structure for $\mathbf{K}$ can be used to represent different local controllers actuating on distinct subsystems.

The main problem addressed in this article is the computation of controller gains for the system (1) subject to a given sparsity constraint (2) that minimize the infinite-horizon quadratic cost function

$$J = \sum_{k=0}^{+\infty} (\boldsymbol{x}^{\mathsf{T}}(k)\mathbf{Q}\boldsymbol{x}(k) + \boldsymbol{u}^{\mathsf{T}}(k)\mathbf{R}\boldsymbol{u}(k)), \tag{3}$$

where $\mathbf{Q} \succcurlyeq 0$ and $\mathbf{R} \succ 0$. In the sequel, the following assumption is made.

Assumption 1. There exists a stabilizing distributed gain for the system (1), that is, there exists a $\mathbf{K} \in \text{Sparse}(\mathbf{E})$ such that the eigenvalues of the closed-loop matrix $(\mathbf{A} - \mathbf{BK})$ lie inside the unit circle.

2.1 | Cost function reformulation

For the centralized case, that is, when $\mathbf{K}$ is not subject to a sparsity constraint, the design of optimal controllers is well understood and extensively documented in the literature.[29] However, those results are not applicable to the distributed case, and one must develop alternative approaches when the gain is subject to a sparsity constraint.

The rest of this section is dedicated to the reformulation of the cost function (3) to achieve a form that allows for the derivation of the algorithms detailed in the next two sections. First, substituting (2) in (1) yields the closed-loop difference equation

$$\boldsymbol{x}(k+1) = (\mathbf{A} - \mathbf{BK})\boldsymbol{x}(k). \tag{4}$$

Now, assume that all the eigenvalues of $(\mathbf{A} - \mathbf{BK})$ lie inside the unit circle. Then, for the given $\mathbf{Q}$ and $\mathbf{R}$, there exists an unique positive-definite matrix $\mathbf{P}_\infty \in \mathbb{R}^{n\times n}$ such that

$$(\mathbf{A} - \mathbf{BK})^{\mathsf{T}}\mathbf{P}_\infty(\mathbf{A} - \mathbf{BK}) - \mathbf{P}_\infty = -\mathbf{Q} - \mathbf{K}^{\mathsf{T}}\mathbf{RK}, \tag{5}$$

see, for example [30, theorem 23.7]. Let

$$V(\boldsymbol{x}(k)) := \boldsymbol{x}^{\mathsf{T}}(k)\mathbf{P}_\infty\boldsymbol{x}(k).$$

Using (4) and (5), it follows that

$$\begin{aligned} V(\boldsymbol{x}(k+1)) - V(\boldsymbol{x}(k)) &= \boldsymbol{x}^{\mathsf{T}}(k+1)\mathbf{P}_\infty \boldsymbol{x}(k+1) - \boldsymbol{x}^{\mathsf{T}}(k)\mathbf{P}_\infty \boldsymbol{x}(k) \\ &= \boldsymbol{x}^{\mathsf{T}}(k)(\mathbf{A}-\mathbf{B}\mathbf{K})^{\mathsf{T}}\mathbf{P}_\infty(\mathbf{A}-\mathbf{B}\mathbf{K})\boldsymbol{x}(k) - \boldsymbol{x}^{\mathsf{T}}(k)\mathbf{P}_\infty \boldsymbol{x}(k) \\ &= -\boldsymbol{x}^{\mathsf{T}}(k)(\mathbf{Q}+\mathbf{K}^{\mathsf{T}}\mathbf{R}\mathbf{K})\boldsymbol{x}(k). \end{aligned} \tag{6}$$

Substituting (2) in (3) and using (6) yields

$$\begin{aligned} J &= \sum_{k=0}^{+\infty} (\boldsymbol{x}^{\mathsf{T}}(k)\mathbf{Q}\boldsymbol{x}(k) + \boldsymbol{u}^{\mathsf{T}}(k)\mathbf{R}\boldsymbol{u}(k)) \\ &= \sum_{k=0}^{+\infty} \boldsymbol{x}^{\mathsf{T}}(k)(\mathbf{Q}+\mathbf{K}^{\mathsf{T}}\mathbf{R}\mathbf{K})\boldsymbol{x}(k) \\ &= \sum_{k=0}^{+\infty} V(\boldsymbol{x}(k)) - V(\boldsymbol{x}(k+1)). \end{aligned} \tag{7}$$

As the closed-loop difference Equation (4) is assumed to be stable, it follows that $V(\boldsymbol{x}(k))$ goes to zero as k goes to infinity, and (7) is reduced to

$$J = V(\boldsymbol{x}(0)) = \boldsymbol{x}^{\mathsf{T}}(0)\mathbf{P}_\infty \boldsymbol{x}(0). \tag{8}$$

Thus, the optimal distributed gain $\mathbf{K} \in \text{Sparse}(\mathbf{E})$ is the one that minimizes (8), in which $\mathbf{P}_\infty$ is the unique positive-definite solution of (5).

Note that the optimal cost (8) depends explicitly on the initial state of the system. To circumvent this issue, suppose that $\boldsymbol{x}(0)$ is sampled from a normal distribution with zero mean and covariance $\alpha\mathbf{I}$, where $\alpha > 0$ is an arbitrary positive scalar constant. Then, the expectation of the cost function (8) follows

$$\begin{aligned} E\{J\} &= E\{\boldsymbol{x}^{\mathsf{T}}(0)\mathbf{P}_\infty \boldsymbol{x}(0)\} = E\{\text{tr}(\boldsymbol{x}^{\mathsf{T}}(0)\mathbf{P}_\infty \boldsymbol{x}(0))\} \\ &= E\{\text{tr}(\mathbf{P}_\infty \boldsymbol{x}(0)\boldsymbol{x}^{\mathsf{T}}(0))\} = \text{tr}(E\{\mathbf{P}_\infty \boldsymbol{x}(0)\boldsymbol{x}^{\mathsf{T}}(0)\}) \\ &= \text{tr}(\mathbf{P}_\infty \alpha\mathbf{I}) = \alpha\ \text{tr}(\mathbf{P}_\infty). \end{aligned} \tag{9}$$

The minimization of (9) can thus be seen as optimizing the controller for a wide range of possible initial states, possibly even the whole field of $\mathbb{R}^n$ as α can be made arbitrarily large. The optimal distributed controller design problem considered in this article can then be restated as finding the gain $\mathbf{K} \in \text{Sparse}(\mathbf{E})$ that minimizes the trace of $\mathbf{P}_\infty$, where $\mathbf{P}_\infty$ is the unique positive-definite solution of the algebraic matrix Equation (5), that is, solve the optimization problem

$$\begin{aligned} &\underset{\substack{\mathbf{K} \in \mathbb{R}^{m\times n} \\ \mathbf{P}_\infty \in \mathbb{R}^{n\times n}}}{\text{minimize}} \quad \text{tr}(\mathbf{P}_\infty) \\ &\text{subject to} \quad \mathbf{P}_\infty = (\mathbf{A}-\mathbf{B}\mathbf{K})^{\mathsf{T}}\mathbf{P}_\infty(\mathbf{A}-\mathbf{B}\mathbf{K}) + \mathbf{Q} + \mathbf{K}^{\mathsf{T}}\mathbf{R}\mathbf{K} \\ &\qquad\qquad \mathbf{P}_\infty \succ 0 \\ &\qquad\qquad \mathbf{K} \in \text{Sparse}(\mathbf{E}) \end{aligned} \tag{10}$$

Finally, consider the difference matrix equation

$$\mathbf{P}(k+1) = (\mathbf{A}-\mathbf{B}\mathbf{K})^{\mathsf{T}}\mathbf{P}(k)(\mathbf{A}-\mathbf{B}\mathbf{K}) + \mathbf{Q} + \mathbf{K}^{\mathsf{T}}\mathbf{R}\mathbf{K}. \tag{11}$$

For a given stabilizing gain $\mathbf{K}$, it can be shown that the solution $\mathbf{P}(k)$ of (11) converges to the unique positive-definite solution $\mathbf{P}_\infty$ of (5) as $k \to \infty$ for any positive-definite initial condition $\mathbf{P}(0) \succ 0$, see, for example, Reference 30. Then, the

optimal distributed controller design problem (10) can be reformulated as

$$\begin{aligned} \underset{\mathbf{K}\in\mathbb{R}^{m\times n}}{\text{minimize}} \; & \lim_{k\to\infty} \operatorname{tr}(\mathbf{P}(k)) \\ \text{subject to} \; & \mathbf{P}(k+1) = (\mathbf{A}-\mathbf{B}\mathbf{K})^{\mathsf{T}}\mathbf{P}(k)(\mathbf{A}-\mathbf{B}\mathbf{K}) + \mathbf{Q} + \mathbf{K}^{\mathsf{T}}\mathbf{R}\mathbf{K} \\ & \mathbf{P}(0) = \mathbf{P}_0 \succ 0 \\ & \mathbf{K} \in \text{Sparse}(\mathbf{E}) \end{aligned}$$

This formulation is used in the rest of the article to derive algorithms to compute well-performing distributed controller gains for the system (1).

3 | ONE-STEP ALGORITHM FOR DISTRIBUTED CONTROLLER DESIGN

This section details the first of two algorithms introduced in this article to compute distributed state feedback gains for the discrete-time LTI system (1).

Consider the difference matrix Equation (11) with a time-varying gain:

$$\mathbf{P}(k+1) = (\mathbf{A}-\mathbf{B}\mathbf{K}(k+1))^{\mathsf{T}}\mathbf{P}(k)(\mathbf{A}-\mathbf{B}\mathbf{K}(k+1)) + \mathbf{Q} + \mathbf{K}^{\mathsf{T}}(k+1)\mathbf{R}\mathbf{K}(k+1). \tag{12}$$

The proposed controller gain computation algorithm is the following: starting with an arbitrary $\mathbf{P}(0)\succ 0$, compute the gain $\mathbf{K}(1)$ subjected to the desired sparsity pattern that minimizes $\operatorname{tr}(\mathbf{P}(1))$. Then, given the newly obtained $\mathbf{P}(1)$, compute the gain $\mathbf{K}(2)$ that minimizes $\operatorname{tr}(\mathbf{P}(2))$, and so on. That is, at each step k the next $\mathbf{K}(k+1)$ is obtained by solving the optimization problem

$$\begin{aligned} \underset{\mathbf{K}(k+1)\in\mathbb{R}^{m\times n}}{\text{minimize}} \; & \operatorname{tr}(\mathbf{P}(k+1)) \\ \text{subject to} \; & \mathbf{K}(k+1) \in \text{Sparse}(\mathbf{E}) \end{aligned} \tag{13}$$

in which $\mathbf{P}(k+1)$ is computed from $\mathbf{P}(k)$ using (12). Note that, as (13) is a quadratic optimization problem subject to a linear constraint, its optimal solution can be computed efficiently. The difference matrix Equation (12) is propagated using the gains obtained by solving (13) until steady-state is achieved, yielding a constant gain $\mathbf{K}$ and associated performance metric $\operatorname{tr}(\mathbf{P}_\infty)$.

In the unconstrained case, this algorithm yields the optimal steady-state LQR gain. However, when operating with a sparsity constraint on the gain, the obtained steady-solution is suboptimal, as it is shown in the simulation results presented ahead in the article. Nevertheless, the one-step algorithm remains a versatile and computationally efficient method to computed distributed state feedback gains. To speed up the algorithm, the solution of the optimization problem (13) can be obtained in closed-form, as detailed in the following result.

Theorem 1. *Let $\boldsymbol{l}_i \in \mathbb{R}^m$ denote a column vector whose entries are all set to zero except for the ith one, which is set to 1, and define $\mathscr{L}_i := \operatorname{diag}(\boldsymbol{l}_i)$. Define a vector $\boldsymbol{m}_j \in \mathbb{R}^n$ to encode the nonzero entries in the jth column of $\mathbf{K}(k+1)$, following*

$$\begin{cases} \boldsymbol{m}_j(i) = 0 & \text{if} \quad [\mathbf{E}]_{ij} = 0 \\ \boldsymbol{m}_j(i) = 1 & \text{if} \quad [\mathbf{E}]_{ij} \neq 0 \end{cases}, \; i = 1, 2, \ldots, n,$$

and let $\mathscr{M}_j = \operatorname{diag}(\boldsymbol{m}_j)$. Then, the optimal solution of the optimization problem (13) is given by

$$\mathbf{K}(k+1) = \sum_{j=1}^{n} (\mathbf{I} - \mathscr{M}_j + \mathscr{M}_j\mathbf{S}(k+1)\mathscr{M}_j)^{-1}\mathscr{M}_j\mathbf{B}^{\mathsf{T}}\mathbf{P}(k)\mathbf{A}\mathscr{L}_j, \tag{14}$$

where

$$\mathbf{S}(k+1) = \mathbf{B}^{\mathsf{T}}\mathbf{P}(k)\mathbf{B} + \mathbf{R}. \tag{15}$$

Proof. See A1. ∎

4 | FINITE-HORIZON ALGORITHM FOR DISTRIBUTED CONTROLLER DESIGN

This section details another algorithm to compute distributed state feedback gains for the system (1).

To improve on the performance offered by the algorithm detailed in the previous section, instead of optimizing the controller gain one time step at a time, this method considers the following finite-horizon problem:

$$\begin{aligned} &\underset{\mathbf{K}(i)\in\mathbb{R}^{m\times n}\ i=1,2,\ \dots\ ,W}{\text{minimize}} \quad \sum_{k=1}^{W}\text{trace}(\mathbf{P}(k)) \\ &\text{subject to} \ \ \mathbf{K}(i)\in\text{Sparse}(\mathbf{E}),\ \ i=1,2,\ \dots\ ,W \end{aligned} \tag{16}$$

The idea behind this formulation is to approximate steady-state behavior by making W large enough, and thus obtain a constant state feedback gain that outperforms the one computed with the one-step method. However, the optimization problem (16) is nonconvex, as the objective function is polynomial in the optimization variables $\mathbf{K}(k)$, $k=1,2,\ \dots\ ,W$. This can be verified by expressing $\mathbf{P}(k)$, $k=1,2,\ \dots\ ,W$ in terms of the initial condition $\mathbf{P}(0)$ and the gains $\mathbf{K}(k)$ using (11).

On the other hand, suppose that only one of the gains $\mathbf{K}(k)$ is kept as an optimization variable while the rest are fixed to constant given values. Then, the optimization problem (16) takes the form

$$\begin{aligned} &\underset{\mathbf{K}(k)\in\mathbb{R}^{m\times n}}{\text{minimize}} \sum_{i=1}^{W}\text{trace}(\mathbf{P}(i)) \\ &\text{subject to} \ \ \mathbf{K}(k)\in\text{Sparse}(\mathbf{E}) \end{aligned} \tag{17}$$

which is quadratic, and can thus be solved using conventional methods. Taking advantage of this fact, the algorithm detailed in Table 1 is proposed to improve the solution found using the one-step algorithm. This method aims to improve the performance over the finite time window iteratively by computing a new $\mathbf{K}(k)$ for a different $k\in\{1,\ 2,\ \dots\ ,W\}$ at each step by solving (17).

The solution of the partial problem (17) can be computed, in closed-form, as detailed in the following result.

Theorem 2. *Define a matrix* $\mathbf{Z}$ *such that the vector* $\mathbf{Z}\text{vec}(\mathbf{K}(k))$ *contains the nonzero elements of* $\mathbf{K}(k)$ *according to the desired sparsity pattern. The closed-form solution of (17) is given by*

$$\text{vec}(\mathbf{K}(k)) = \mathbf{Z}^{\mathsf{T}}(\mathbf{Z}(\mathbf{\Lambda}(k)\otimes\mathbf{S}(k))\mathbf{Z}^{\mathsf{T}})^{-1}\mathbf{Z}\ \text{vec}(\mathbf{B}^{\mathsf{T}}\mathbf{P}(k-1)\mathbf{A}\mathbf{\Lambda}(k)), \tag{18}$$

where $\mathbf{S}(k)$ *is computed as in (15), and*

$$\mathbf{\Lambda}(k) = \sum_{i=k}^{W}\mathbf{\Gamma}(k+1,i)\mathbf{\Gamma}^{\mathsf{T}}(k+1,i),$$

TABLE 1 Finite-horizon algorithm

(1) **Initialization:** Select a window size, W, an initial $\mathbf{P}(0)\succ 0$, and compute a set of initial $\mathbf{P}(k)$, $k=1,\ \dots\ ,W$ and controller gains $\mathbf{K}(k)$, $k=1,\ \dots\ ,W$ using, for example, the one-step method.

Select a stopping criterion, for example, a minimum improvement on the objective function of (16) or a fixed number of iterations.

(2) Set i = W.

(3) **Inner loop:**

(a) Fix all gains except for $\mathbf{K}(i)$, and solve (17) for $\mathbf{K}(i)$, replacing the previous value for that gain.

(b) Set $i=i-1$. If $i=0$, exit the inner loop to Step (4). Otherwise, go to (a).

(4) Recompute $\mathbf{P}(1)$ through $\mathbf{P}(W)$ using the new gains.

(5) If the stopping criterion is met, go to (6). Otherwise, go to (2).

(6) Compute the steady-state performance obtained with each distinct $\mathbf{K}(k)$ using (11), and select the best performing gain to apply to the distributed controller.

with

$$\mathbf{\Gamma}(k_i, k_f) = \prod_{j=k_i}^{k_f} (\mathbf{A} - \mathbf{B}\mathbf{K}(j)). \tag{19}$$

When $k_i > k_f$, we follow the convention $\mathbf{\Gamma}(k_i, k_f) = \mathbf{I}$.

Proof. See B1. ∎

Remark 1. As an example of how to compute $\mathbf{Z}$, consider $\mathbf{K}(k)$ of the form

$$\mathbf{K}(k) = \begin{bmatrix} k_1 & 0 \\ k_2 & k_3 \\ 0 & k_4 \end{bmatrix}. \text{ Then, } \quad \text{vec}(\mathbf{K}(k)) = \begin{bmatrix} k_1 \\ k_2 \\ 0 \\ 0 \\ k_3 \\ k_4 \end{bmatrix} \text{ and } \quad \mathbf{Z} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

In this case, the chosen $\mathbf{Z}$ preserves the order of the nonzero elements with respect to vec($\mathbf{K}(k)$). However, this is not mandatory as any permutation of the rows of $\mathbf{Z}$ will still yield the optimal gain using (18).

5 | APPLICATION TO THE QUADRUPLE-TANK PROCESS

This section details the results of simulations that were carried out to assess the performance of the proposed distributed control solutions. The model chosen to test the controller design methods detailed in the previous sections is the quadruple-tank process proposed in Reference 28, as depicted in Figure 1.

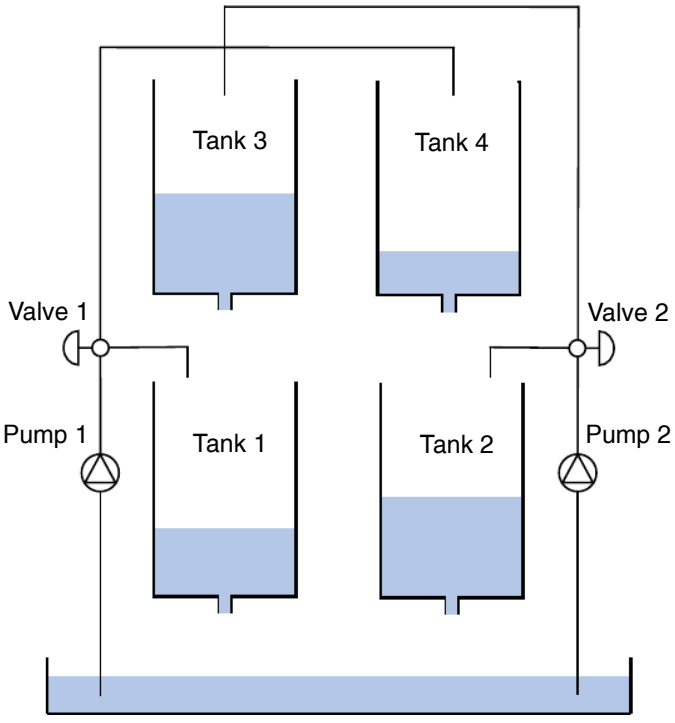

FIGURE 1 Schematic diagram of the quadruple-tank process[28] [Colour figure can be viewed at wileyonlinelibrary.com]

The dynamics of the quadruple-tank process are given by the nonlinear system

$$\begin{cases} \dot{h}_1(t) = -\frac{a_1}{A_1}\sqrt{2gh_1(t)} + \frac{a_3}{A_1}\sqrt{2gh_3(t)} + \frac{\gamma_1 k_1}{A_1}v_1(t) \\ \dot{h}_2(t) = -\frac{a_2}{A_2}\sqrt{2gh_2(t)} + \frac{a_4}{A_2}\sqrt{2gh_4(t)} + \frac{\gamma_2 k_2}{A_2}v_2(t) \\ \dot{h}_2(t) = -\frac{a_3}{A_3}\sqrt{2gh_3(t)} + \frac{(1-\gamma_2)k_2}{A_3}v_2(t) \\ \dot{h}_2(t) = -\frac{a_4}{A_4}\sqrt{2gh_4(t)} + \frac{(1-\gamma_1)k_1}{A_4}v_1(t) \end{cases}, \tag{20}$$

in which $h_i(t)$, A_i, and a_i are the water level, cross-section, and outlet hole cross-section of Tank i, respectively, and g is the acceleration of gravity. A voltage $v_i(t)$ is applied to each pump, resulting in the corresponding flow $k_i v_i$, and the parameters $\gamma_i \in [0, 1]$ model the effect of the corresponding valves. The measured output is the water level in Tanks 1 and 2. The parameters of the model were set to values similar to those detailed in Reference 28, see Table 2.

The model was then linearized around the equilibrium point

$$\begin{cases} h_1^0 = 12.26 \text{ (cm)} \\ h_2^0 = 12.78 \text{ (cm)} \\ h_3^0 = 1.63 \text{ (cm)} \\ h_4^0 = 1.41 \text{ (cm)} \\ v_1^0 = 3 \text{ (V)} \\ v_2^0 = 3 \text{ (V)} \end{cases},$$

yielding the LTI system

$$\begin{cases} \dot{\boldsymbol{x}}(t) = \mathbf{A}_C \boldsymbol{x}(t) + \mathbf{B}_C \boldsymbol{u}(t) \\ \boldsymbol{y}(t) = \mathbf{C}_C \boldsymbol{x}(t) \end{cases},$$

in which

$$\mathbf{A}_C = \begin{bmatrix} -\frac{1}{T_1} & 0 & \frac{A_3}{A_1 T_3} & 0 \\ 0 & -\frac{1}{T_2} & 0 & \frac{A_4}{A_2 T_4} \\ 0 & 0 & -\frac{1}{T_3} & 0 \\ 0 & 0 & 0 & -\frac{1}{T_4} \end{bmatrix}, \quad \mathbf{B}_C = \begin{bmatrix} \frac{\gamma_1 k_1}{A_1} & 0 \\ 0 & \frac{\gamma_2 k_2}{A_2} \\ 0 & \frac{(1-\gamma_2)k_2}{A_3} \\ \frac{(1-\gamma_1)k_1}{A_4} & 0 \end{bmatrix},$$

and $\mathbf{C}_C = \begin{bmatrix} \mathbf{I}_2 & \mathbf{0} \end{bmatrix}$. The time constants in $\mathbf{A}_C$ are given by

$$T_i = \frac{A_i}{a_i}\sqrt{\frac{2h_i^0}{g}}, \quad i = 1, \ldots, 4.$$

TABLE 2 Parameters of the quadruple-tank process

	1	2	3	4
A_i (cm^2)	28	32	28	32
A_i (cm^2)	0.071	0.057	0.071	0.057
k_i (cm^3/Vs)	3.33	3.35	–	–
γ_i	0.7	0.6	–	–

Following this, the linearized model was discretized with sampling time T_s, yielding the discrete-time LTI system

$$\begin{cases} \boldsymbol{x}(k+1) = \mathbf{A}_D\boldsymbol{x}(k) + \mathbf{B}_D\boldsymbol{u}(k) \\ \boldsymbol{y}(k) = \mathbf{C}_D\boldsymbol{x}(k) \end{cases},$$

where

$$\begin{cases} \mathbf{A}_D = e^{\mathbf{A}_C T_s} \\ \mathbf{B}_D = A_C^{-1}(e^{\mathbf{A}_C T_s} - \mathbf{I})\mathbf{B}_C \\ \mathbf{C}_D = \mathbf{C}_C \end{cases}.$$

Finally, to follow a reference signal $\boldsymbol{r}(k) \in \mathbb{R}^2$, two additional state variables $\boldsymbol{q}(k) := \begin{bmatrix} q_1(k) & q_2(k) \end{bmatrix}^\mathsf{T} \in \mathbb{R}^2$ are introduced that accumulate the output tracking error, following

$$\boldsymbol{q}(k+1) = \boldsymbol{q}(k) + (\boldsymbol{y}(k) - \boldsymbol{r}(k)),$$

yielding the augmented LTI system

$$\begin{cases} \begin{bmatrix} \boldsymbol{x}(k+1) \\ \boldsymbol{q}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_D & \mathbf{0} \\ \mathbf{C}_D & \mathbf{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(k) \\ \boldsymbol{q}(k) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_D & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}(k) \\ \boldsymbol{r}(k) \end{bmatrix} \\ \boldsymbol{y}(k) = \mathbf{C}_D\boldsymbol{x}(k) \end{cases}. \tag{21}$$

It is then straightforward to apply the methods detailed in the previous sections to (21) to design and optimize distributed controllers for the quadruple-tank process, as modeled by the nonlinear system (20). In the simulations that follow, one local controller is implemented at each pump. For Pump 1, the controller uses only the measured output $h_1(t)$ and its corresponding integral state $q_1(k)$, while the controller for Pump 2 has access to $h_2(t)$ and its corresponding integral state $q_2(k)$. Thus, the controller gain matrix $\mathbf{K}$ follows the sparsity pattern

$$\mathbf{K} = \begin{bmatrix} k_1^h & 0 & 0 & 0 & k_1^I & 0 \\ 0 & k_2^h & 0 & 0 & 0 & k_2^I \end{bmatrix}. \tag{22}$$

Figure 2 depicts the evolution of the trace of $\mathbf{P}(k)$ for both proposed methods applied to the augmented system (21) with the sparsity pattern (22). In this case, the weight matrices were set to $\mathbf{Q} = \mathbf{I}$ and $\mathbf{R} = \mathbf{I}$ and the sampling time was set to $T_s = 10$ (s). As it can be seen, in both cases a stable steady-state was achieved, and the finite-horizon algorithm improved the performance over the one-step algorithm. However, as the improvement was relatively small (around 2.5%) both controllers show very similar behavior, and thus the simulation results that follow depict only the controller computed using the finite-horizon algorithm.

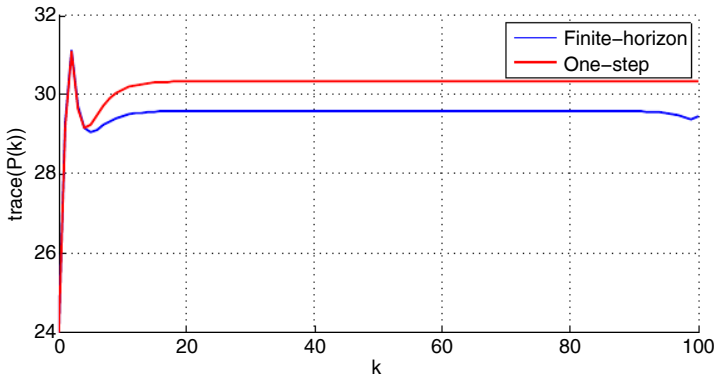

FIGURE 2 Evolution of trace($\mathbf{P}(k)$) over a 100-step window for both methods [Colour figure can be viewed at wileyonlinelibrary.com]

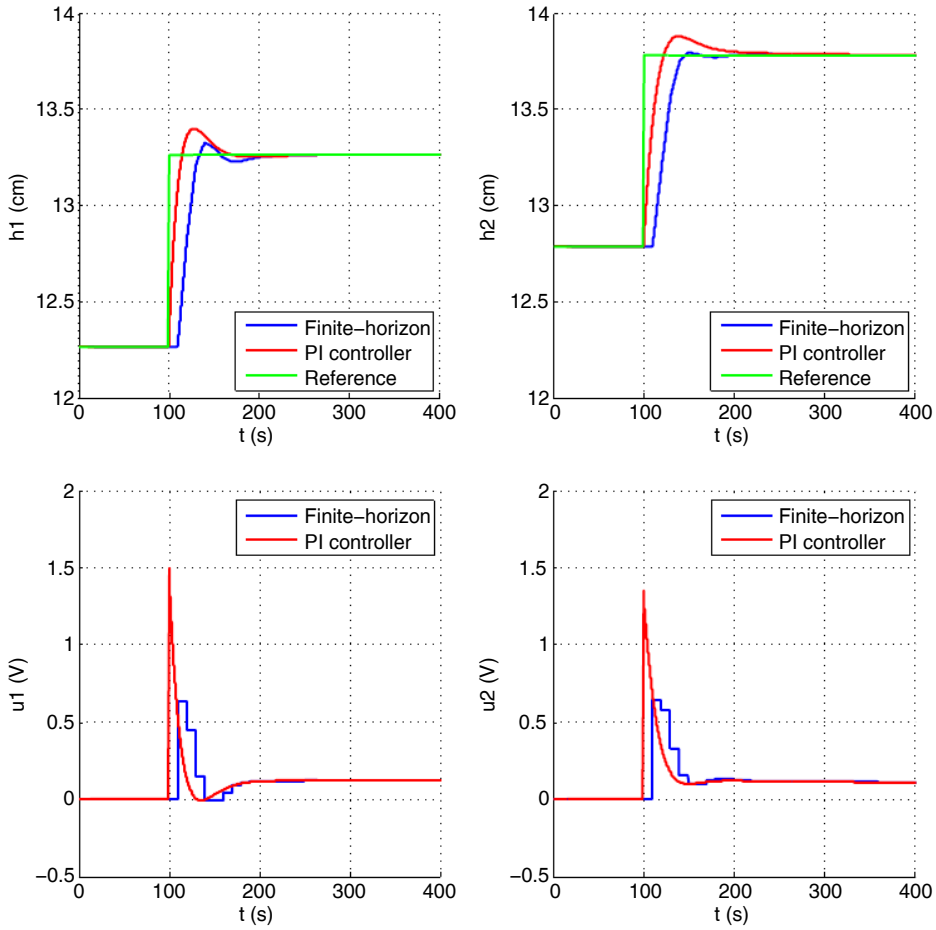

FIGURE 3 Response of the closed-loop system to a step in both tracked variables [Colour figure can be viewed at wileyonlinelibrary.com]

The resulting discrete-time distributed controller was first compared with the continuous-time distributed PI controller proposed in Reference 28. Figure 3 depicts the response of both controllers to a step in both tracking variables. As it can be seen, while the finite-horizon controller is slightly slower, mainly due to the delay introduced by the discrete-time implementation, it overshoots less while yielding a smaller control input. This is expected, as the continuous-time PI controller in Reference 28 was tuned manually while the discrete-time controller was optimized algorithmically.

Following this, further simulations were carried out to verify if the proposed controller design methods are able to follow design specifications, namely, through tuning the state and input weight matrices $\mathbf{Q}$ and $\mathbf{R}$. To do so, distributed controller gains were obtained using the finite-horizon method for three different values of $\mathbf{R}$: $\mathbf{I}$, $10\,\mathbf{I}$, and $100\,\mathbf{I}$. The results are depicted in Figure 4. As it can be seen, as the weight on the input increases, the controller generates an input of smaller magnitude, with the expected trade-off of a slower response to the step in the reference signal. This shows that the proposed methods can indeed be used to tune the performance of distributed controllers using the design parameters $\mathbf{Q}$ and $\mathbf{R}$.

6 | SIMULATIONS FOR A LARGER SCALE MODEL

This section details simulation results for a larger scale version of the model used in the previous section. In this case, instead of a quadruple-tank process we consider a system with 40 interconnected tanks, see Figure 5 for a simplified scheme of the process. The dynamics of this 40-tank process can be modeled by the nonlinear system

$$\dot{\boldsymbol{h}}(t) = (\mathbf{A}_{40} + \mathbf{D}_{40})\boldsymbol{s}(t) + \mathbf{B}_{40}\boldsymbol{v}(t), \tag{23}$$

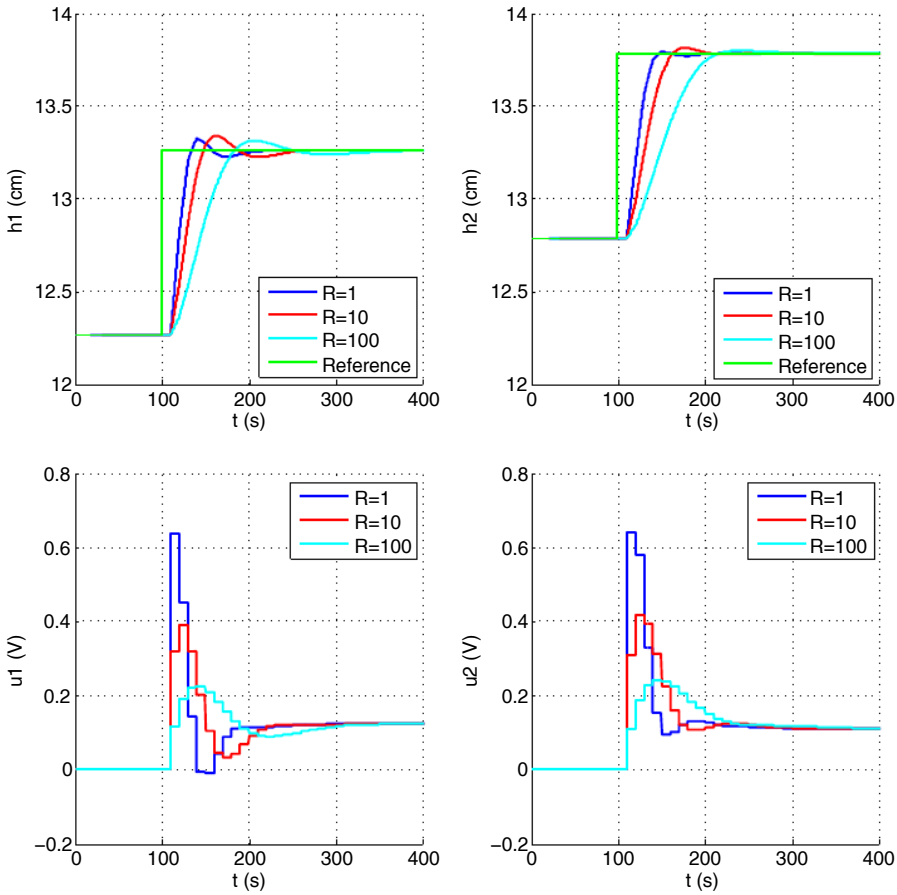

FIGURE 4 Response of the closed-loop system for different design parameters [Colour figure can be viewed at wileyonlinelibrary.com]

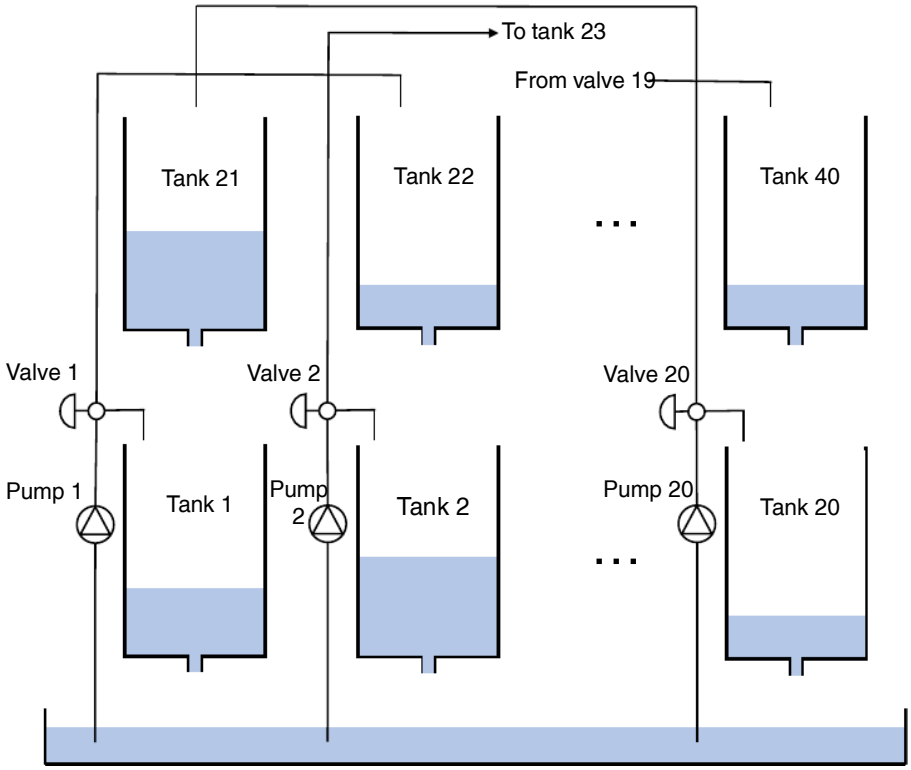

FIGURE 5 Schematic diagram of the 40-tank process[28] [Colour figure can be viewed at wileyonlinelibrary.com]

in which

$$\boldsymbol{h}(t) = \begin{bmatrix} h_1(t) \\ h_2(t) \\ \vdots \\ h_{40}(t) \end{bmatrix} \in \mathbb{R}^{40}$$

is the vector containing the water levels of each tank,

$$\boldsymbol{s}(t) = \begin{bmatrix} \sqrt{2gh_1(t)} \\ \sqrt{2gh_2(t)} \\ \vdots \\ \sqrt{2gh_{40}(t)} \end{bmatrix} \in \mathbb{R}^{40},$$

where g is the acceleration of gravity, and

$$\boldsymbol{v}(t) = \begin{bmatrix} v_1(t) \\ v_2(t) \\ \vdots \\ v_{20}(t) \end{bmatrix} \in \mathbb{R}^{20}$$

is the vector containing the input voltages of each pump. The parameters of the system are the cross-sections of the tanks A_i and the cross-sections of the outlet holes a_i, as well as the constants k_i and γ_i associated with the pumps and the valves, respectively. The matrices of the system are computed using those parameters, following

$$\mathbf{A}_{40} = \operatorname{diag}\left(-\frac{a_1}{A_1}, -\frac{a_2}{A_2}, \dots, -\frac{a_{40}}{A_{40}} \right),$$
$$\mathbf{D}_{40} = \begin{bmatrix} \mathbf{0} & \operatorname{diag}\left(\frac{a_{21}}{A_1}, \frac{a_{22}}{A_2}, \dots, \frac{a_{40}}{A_{20}} \right) \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

and

$$\mathbf{B}_{40} = \begin{bmatrix} \operatorname{diag}\left(\frac{\gamma_1 k_1}{A_1}, \frac{\gamma_2 k_2}{A_2}, \dots, \frac{\gamma_{20} k_{20}}{A_{20}} \right) \\ \mathbf{B}_{40}^a \end{bmatrix},$$

in which

$$\mathbf{B}_{40}^a = \begin{bmatrix} 0 & 0 & \dots & 0 & \frac{(1-\gamma_{20})k_{20}}{A_{21}} \\ \frac{(1-\gamma_1)k_1}{A_{22}} & 0 & \dots & 0 & 0 \\ 0 & \frac{(1-\gamma_2)k_2}{A_{23}} & \dots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \frac{(1-\gamma_{19})k_{19}}{A_{40}} & 0 \end{bmatrix}.$$

For the simulation results presented in this section, the parameters of the system were set to the values detailed in Table 3. To obtain a model suitable for application of the methods introduced in Sections 3 and 4, the nonlinear system (23) was first linearized around the operating point

TABLE 3 Parameters of the 40-tank process

	$i = 1, \ldots, 20$	$i = 21, \ldots, 40$
A_i (cm^2)	28	32
A_i (cm^2)	0.05	0.075
k_i (cm^3/Vs)	3.33	–
γ_i	0.55	–

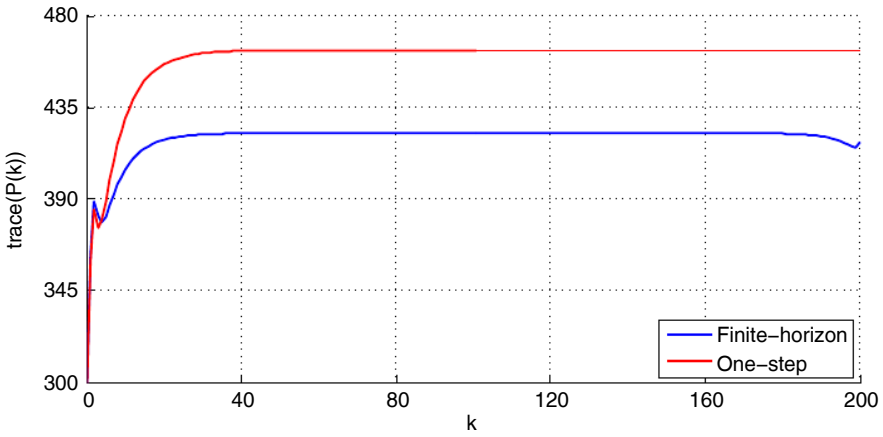

FIGURE 6 Evolution of trace($\mathbf{P}(k)$) over a 200-step window for both methods, with $\mathbf{R} = \mathbf{I}$ [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 4 Projected performance (trace($\mathbf{P}_\infty$)) for both distributed controller design methods

	One-step	Finite-horizon
$\mathbf{R} = \mathbf{I}$	4.63×10^2	4.22×10^2
$\mathbf{R} = 10\mathbf{I}$	1.57×10^3	1.35×10^3
$\mathbf{R} = 100\mathbf{I}$	7.65×10^3	6.39×10^3

$$\begin{cases} h_i^0 = 12.26 \text{ (cm)}, & i = 1, \ldots, 20 \\ h_i^0 = 12.78 \text{ (cm)}, & i = 21, \ldots, 40 \\ v_i^0 = 3 \text{ (V)}, & i = 1, \ldots, 20 \end{cases},$$

then discretized, and finally augmented with integral states similarly to what was done in the previous section. The distributed controller is composed of one local controller for each pump i, which has access to only the water level $h_i(t)$ and respective integral state. Gains for the distributed controller were then computed with both the one-step and finite-horizon algorithms, with $T_s = 10\,(s)$, $\mathbf{Q} = \mathbf{I}$, and various values for $\mathbf{R}$. The results are depicted in Figure 6 and Table 4. As it can be seen, in all cases the finite-horizon algorithm outperformed the one-step algorithm.

The gains obtained with both methods were then tested in simulation, and the results are depicted in Figures 7 and 8. Figure 7 depicts the evolution of $h_1(t)$, $h_2(t)$, and the inputs of their respective local controllers in response to a step in the reference for $h_1(t)$. As it can be seen, the reference is tracked correctly by both distributed controllers, and the finite-horizon controller has a slightly faster response at the cost of a larger input signal. To better assess the performance of both methods, Figure 8 depicts the evolution of $h_1(t)$ and respective input for different values of the input weight matrix $\mathbf{R}$. In both cases, larger values of $\mathbf{R}$ induce a smaller input signal, as well as the expected slower response to the step in the reference. However, note that using different values of $\mathbf{R}$ has more impact in shaping the behavior of the closed-loop system with the finite-horizon controller, as evidenced by the larger differences in the evolution of the water level as well as the input. This suggests that the finite-horizon algorithm is better suited to fine-tuning the controller gains using the state and input weight matrices $\mathbf{Q}$ and $\mathbf{R}$.

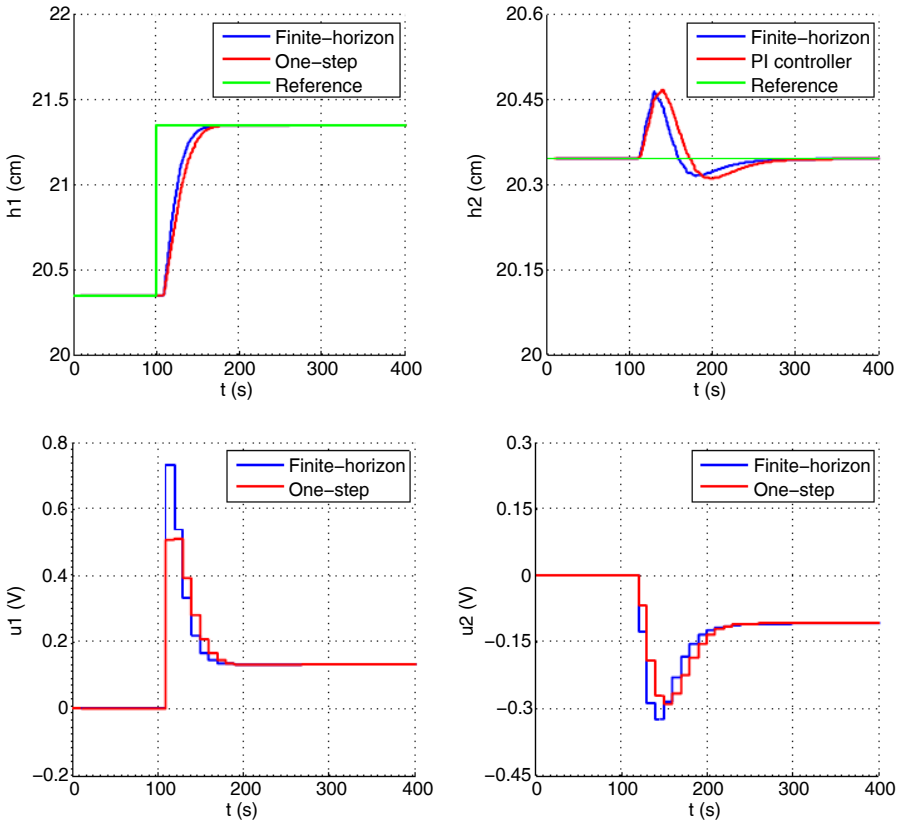

FIGURE 7 Response of the closed-loop system to a step in h_1 [Colour figure can be viewed at wileyonlinelibrary.com]

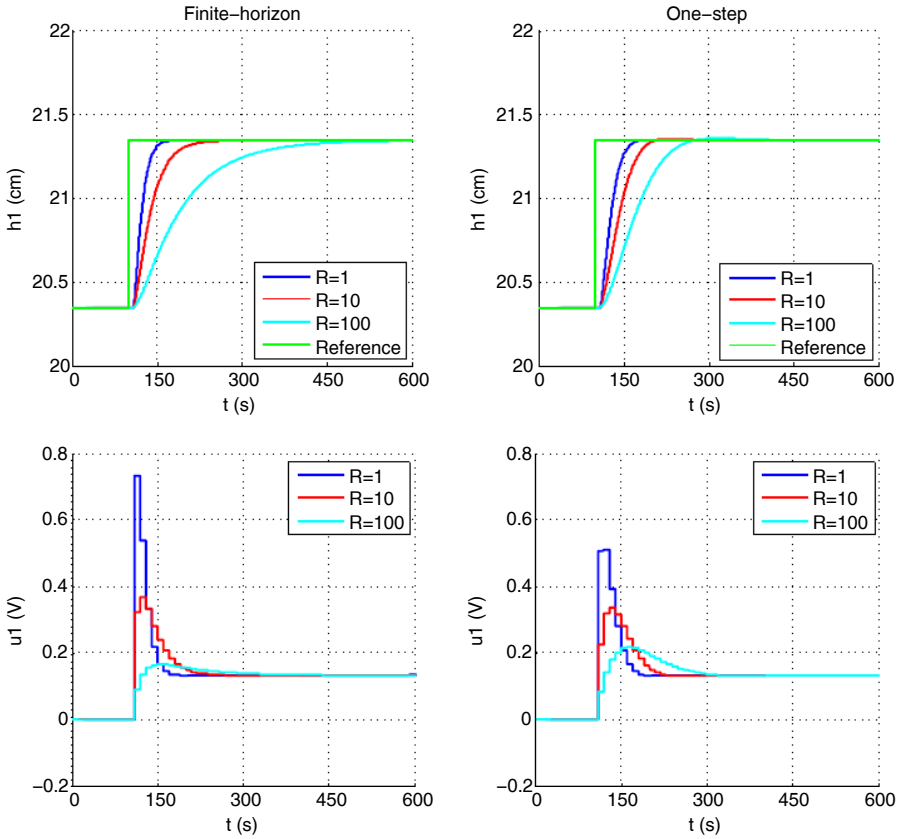

FIGURE 8 Comparison between the two methods for different design parameters [Colour figure can be viewed at wileyonlinelibrary.com]

7 | CONCLUSIONS

This article addressed the problem of distributed controller design for linear discrete-time systems. The problem was posed using the classical framework of state feedback gain optimization over an infinite-horizon quadratic cost, with an additional sparsity constraint on the gain matrix to model the distributed nature of the controller. An equivalent formulation was derived that consists in the optimization of the steady-state solution of a matrix difference equation, and two methods for distributed gain computation were proposed based on it.

The first one, called the one-step algorithm, optimizes the solution of said difference matrix equation at each time step, to converge to a well-performing distributed state feedback gain. A closed-form solution for the computations required at each step is detailed, allowing for fast computation of stabilizing gains.

The second method, called the finite-horizon algorithm, optimizes the solution of the difference matrix equation over a finite time window to approximate asymptotic behavior, and thus minimize the original infinite-horizon quadratic control cost as the size of the window increases. A closed-form solution for the optimization problems solved during each iteration is also presented for this method.

To assess the performance of the proposed solutions, simulation results were presented for the problem of distributed control of a quadruple-tank process, as well as a version of that problem scaled up to 40 interconnected tanks.

ACKNOWLEDGEMENTS

The work presented in this article was supported by the Macao Science and Technology Development Fund under Grant FDCT 0146/2019/A3, by the University of Macau, Macao, China, under Project MYRG2018-00198-FST, by the Fundação para a Ciência e a Tecnologia (FCT) through LARSyS - FCT Project UIDB/50009/2020, by the FCT project DECENTER [LISBOA-01-0145-FEDER-029605], funded by the Lisboa 2020 and PIDDAC programs, and through IDMEC, under contract LAETA FCT [UIDB/50022/2020].

ORCID

Daniel Viegas https://orcid.org/0000-0001-5847-3095
Pedro Batista https://orcid.org/0000-0001-6079-0436

REFERENCES

1. Tetiker M, Derya AA, Teymour F, Cinar A. Control of grade transitions in distributed chemical reactor networks–an agent-based approach. *Comput Chem Eng*. 2008;32(9):1984-1994.
2. Leitão P. Agent-based distributed manufacturing control: a state-of-the-art survey. *Eng Appl Artif Intell*. 2009;22(7):979-991.
3. Blaabjerg F, Teodorescu R, Liserre M, Timbus AV. Overview of control and grid synchronization for distributed power generation systems. *IEEE Trans Ind Electron*. 2006;53(5):1398-1409.
4. Prodanovic M, Green TC. High-quality power generation through distributed control of a power park microgrid. *IEEE Trans Ind Electron*. 2006;53(5):1471-1482.
5. Yan Z, Jouandeau N, Cherif AA. A survey and analysis of multi-robot coordination. *Int J Adv Robot Syst*. 2013;10:399.
6. Davison EJ, Aghdam AG. *Decentralized Control of Large-Scale Systems*. New York, NY: Springer Publishing Company; 2014.
7. Bakule L. Decentralized control: an overview. *Annu Rev Control*. 2008;32(1):87-98.
8. Antonelli G. Interconnected dynamic systems: an overview on distributed control. *IEEE Control Syst*. 2013;33(1):76-88.
9. Blondel VD, Tsitsiklis JN. A survey of computational complexity results in systems and control. *Automatica*. 2000;36(9):1249-1274.
10. Zhai G, Ikeda M, FY. Decentralized H_∞ controller design: a matrix inequality approach using a homotopy method. *Automatica*. 2001;37(4):565-572.
11. Bompart V, Noll D, Apkarian P. Second-order nonsmooth optimization for H_∞ synthesis. *Numer Math*. 2007;107(3):433-454.
12. Sojoudi S, Lavaei J. Exactness of semidefinite relaxations for nonlinear optimization problems with underlying graph structure. *SIAM J Optim*. 2014;24(4):1746-1778.
13. Lin F, Fardad M, Jovanović MR. Design of optimal sparse feedback gains via the alternating direction method of multipliers. *IEEE Trans Autom Control*. 2013;58(9):2426-2431.
14. Dörfler F, Jovanović MR, Chertkov M, Bullo F. Sparsity-promoting optimal wide-area control of power networks. *IEEE Trans Power Syst*. 2014;29(5):2281-2291.
15. van Schuppen JH, Boutin O, Kempker PL, et al. Control of distributed systems: tutorial and overview. *Eur J Control*. 2011;17(5):579-602.
16. Komenda J, Masopust T, Schuppen JH. Supervisory control synthesis of discrete-event systems using a coordination scheme. *Automatica*. 2012;48(2):247-254.
17. Bakule L. Decentralized control: status and outlook. *Annu Rev Control*. 2014;38(1):71-80.
18. Siljak DD. *Decentralized Control of Complex Systems*. North Chelmsford, Chelmsford, MA, USA.: Courier Corporation; 2011.
19. Lessard L, Lall S. Convexity of decentralized controller synthesis. *IEEE Trans Autom Control*. 2016;61(10):3122-3127.

20. Lamperski A, Lessard L. Optimal decentralized state-feedback control with sparsity and delays. *Automatica*. 2015;58:143-151.
21. Shah P, Parrilo PA. H_2 optimal decentralized control over posets: a state-space solution for state-feedback. *IEEE Trans Automat Control*. 2013;58(12):3084-3096.
22. Bamieh B, Paganini F, Dahleh MA. Distributed control of spatially invariant systems. *IEEE Trans Autom Control*. 2002;47(7):1091-1107.
23. Bamieh B, Voulgaris PG. A convex characterization of distributed control problems in spatially invariant systems with communication constraints. *Syst Control Lett*. 2005;54(6):575-583.
24. Mahmoud MS. *Decentralized Control and Filtering in Interconnected Dynamical Systems*. Boca Raton, FL: CRC Press; 2010.
25. Mahmoud MS. Decentralized stabilization of interconnected systems with time-varying delays. *IEEE Trans Autom Control*. 2009;54(11):2663-2668.
26. Tanaka T, Langbort C. The bounded real lemma for internally positive systems and H_∞ structured static state feedback. *IEEE Trans Automat Control*. 2011;56(9):2218-2223.
27. Rantzer A. Optimal H_∞ state feedback for systems with symmetric and Hurwitz state matrix. *Am Automat Control Council*. 2016;3366-3371.
28. Johansson KH. The quadruple-tank process: a multivariable laboratory process with an adjustable zero. *IEEE Trans Control Syst Technol*. 2000;8(3):456-465.
29. Middleton Richard H, Goodwin GC. *Digital Control and Estimation: A Unified Approach*. Upper Saddle River, NJ: Prentice Hall Professional Technical Reference; 1990.
30. Rugh WJ. *Linear System Theory*. Upper Saddle River, NJ: Prentice Hall; 1996.

How to cite this article: Viegas D, Batista P, Oliveira P, Silvestre C. Distributed controller design and performance optimization for discrete-time linear systems. *Optim Control Appl Meth*. 2021;42:126–143. https://doi.org/10.1002/oca.2669

APPENDIX A. PROOF OF THEOREM 1

The optimization problem (13) can be seen, instead of as a quadratic problem with a matrix variable $\mathbf{K}(k+1)$ subject to a sparsity constraint, as an unconstrained quadratic problem whose variables are the nonzero entries of $\mathbf{K}(k+1)$. Thus, the optimal solution can be found by differentiating the objective function with respect to these nonzero entries and equaling to zero.

Taking the trace of both sides of (12) and differentiating with respect to $\mathbf{K}(k+1)$, it can be shown that

$$\frac{\partial \mathrm{tr}(\mathbf{P}(k+1))}{\partial \mathbf{K}(k+1)} = -2\mathbf{B}^{\mathsf{T}}\mathbf{P}(k)\mathbf{A} + 2\mathbf{S}(k+1)\mathbf{K}(k+1). \tag{A1}$$

Define a set $\mathcal{X}$ of integer pairs of the form (i,j) to index the nonzero entries in $\mathbf{K}(k+1)$, that is,

$$\begin{cases} (i,j) \in \mathcal{X} & \text{if} \quad [\mathbf{E}]_{ij} \neq 0 \\ (i,j) \notin \mathcal{X} & \text{otherwise} \end{cases}, \quad i = 1, \ldots, m, \; j = 1, \ldots, n. \tag{A2}$$

Using (A1) and taking into account the sparsity constraint yields a set of $m \times n$ linear equations,

$$\begin{cases} \boldsymbol{l}_i^{\mathsf{T}}(\mathbf{S}(k+1)\mathbf{K}(k+1) - \mathbf{B}^{\mathsf{T}}\mathbf{P}(k)\mathbf{A})\boldsymbol{l}_j = 0 & \text{for all } (i,j) \in \mathcal{X} \\ \boldsymbol{l}_i^{\mathsf{T}}\mathbf{K}(k+1)\boldsymbol{l}_j = 0 & \text{for all } (i,j) \notin \mathcal{X} \end{cases}. \tag{A3}$$

Now, consider the jth column of $\mathbf{K}(k+1)$. From the first part of (A3), it follows that

$$\sum_{i}^{(i,j)\in\mathcal{X}} \mathscr{L}_i(\mathbf{S}(k+1)\mathbf{K}(k+1) - \mathbf{B}^{\mathsf{T}}\mathbf{P}(k)\mathbf{A})\boldsymbol{l}_j = 0. \tag{A4}$$

Using the notation introduced above, (A4) can be rewritten as

$$\mathscr{M}_j\mathbf{S}(k+1)\mathbf{K}(k+1)\boldsymbol{l}_j = \mathscr{M}_j\mathbf{B}^{\mathsf{T}}\mathbf{P}(k)\mathbf{A}\boldsymbol{l}_j. \tag{A5}$$

The left-hand side of (A5) can be expanded to

$$\mathcal{M}_j\mathbf{S}(k+1)\mathbf{K}(k+1)\boldsymbol{l}_j = \mathcal{M}_j\mathbf{S}(k+1)(\mathbf{I}+\mathcal{M}_j-\mathcal{M}_j)\mathbf{K}(k+1)\boldsymbol{l}_j$$

and then, using the second part of (A3), simplified to

$$\mathcal{M}_j\mathbf{S}(k+1)\mathbf{K}(k+1)\boldsymbol{l}_j = \mathcal{M}_j\mathbf{S}(k+1)\mathcal{M}_j\mathbf{K}(k+1)\boldsymbol{l}_j. \tag{A6}$$

Substituting (A6) in (A4) yields

$$\mathcal{M}_j\mathbf{S}(k+1)\mathcal{M}_j\mathbf{K}(k+1)\boldsymbol{l}_j = \mathcal{M}_j\mathbf{B}^\top\mathbf{P}(k)\mathbf{A}\boldsymbol{l}_j\,. \tag{A7}$$

However, (A7) cannot be solved for $\mathbf{K}(k+1)$ yet as the matrix $\mathcal{M}_j\mathbf{S}(k+1)\mathcal{M}_j$ is singular. To circumvent this, note that the second part of (A3) implies

$$\mathcal{M}_j\mathbf{S}(k+1)\mathcal{M}_j\mathbf{K}(k+1)\boldsymbol{l}_j = (\mathbf{I}-\mathcal{M}_j+\mathcal{M}_j\mathbf{S}(k+1)\mathcal{M}_j)\mathbf{K}(k+1)\boldsymbol{l}_j.$$

Then, substituting into (A7) yields

$$(\mathbf{I}-\mathcal{M}_j+\mathcal{M}_j\mathbf{S}(k+1)\mathcal{M}_j)\mathbf{K}(k+1)\boldsymbol{l}_j = \mathcal{M}_j\mathbf{B}^\top\mathbf{P}(k)\mathbf{A}\boldsymbol{l}_j,$$

which implies that the jth column of the optimal one-step gain is given by

$$\mathbf{K}(k+1)\boldsymbol{l}_j = (\mathbf{I}-\mathcal{M}_j+\mathcal{M}_j\mathbf{S}(k+1)\mathcal{M}_j)^{-1}\mathcal{M}_j\mathbf{B}^\top\mathbf{P}(k)\mathbf{A}\boldsymbol{l}_j. \tag{A8}$$

Finally, summing (A8) for all columns of $\mathbf{K}(k+1)$ yields the closed-form expression (14).

APPENDIX B. PROOF OF THEOREM 2

To begin with, note that the optimization problem

$$\begin{aligned} &\underset{\mathbf{K}(k)\in\mathbb{R}^{m\times n}}{\text{minimize}} \sum_{i=k}^{W}\text{trace}(\mathbf{P}(i)) \\ &\text{subject to}\ \ \mathbf{K}(k)\in\text{Sparse}(\mathbf{E}) \end{aligned} \tag{B1}$$

has the same optimal solution as (17), as the previous steps are not affected by the value of $\mathbf{K}(k)$. As for the one-step case, (B1) can be solved by differentiating the objective function with respect to the nonzero entries in $\mathbf{K}(k)$ and equaling to zero.

Using (11), it can be shown that, for $j \geq k$, $\mathbf{P}(k)$ can be expressed as

$$\mathbf{P}(j) = \boldsymbol{\Gamma}^\top(k,j)\mathbf{P}(k-1)\boldsymbol{\Gamma}(k,j) + \sum_{i=k}^{j}\boldsymbol{\Gamma}^\top(i+1,j)\mathbf{K}^\top(i)\mathbf{R}\mathbf{K}(i)\boldsymbol{\Gamma}(i+1,j) + \sum_{i=k}^{j}\boldsymbol{\Gamma}^\top(i+1,j)\mathbf{Q}\boldsymbol{\Gamma}(i,j), \tag{B2}$$

where $\boldsymbol{\Gamma}(k_i,k_f)$ is defined as in (19). Taking the trace of both sides of (B2) and differentiating with respect to $\mathbf{K}(k)$ yields

$$\frac{\partial\ \text{trace}(\mathbf{P}(j))}{\partial\mathbf{K}(k)} = 2(\mathbf{S}(k)\mathbf{K}(k)-\mathbf{B}^\top\mathbf{P}(k-1)\mathbf{A})(\boldsymbol{\Gamma}(k+1,j)\boldsymbol{\Gamma}^\top(k+1,j)). \tag{B3}$$

Summing (B3) over $j=k,k+1,\ \ldots\ ,W$ yields

$$\frac{\partial\ \sum_{j=k}^{W}\text{trace}(\mathbf{P}(j))}{\partial\mathbf{K}(k)} = 2(\mathbf{S}(k)\mathbf{K}(k)-\mathbf{B}^\top\mathbf{P}(k-1)\mathbf{A})\boldsymbol{\Lambda}(k).$$

Following the same procedure as in the previous section yields the set of linear equations

$$\begin{cases} \boldsymbol{l}_i^{\mathsf{T}}(\mathbf{S}(k)\mathbf{K}(k) - \mathbf{B}^{\mathsf{T}}\mathbf{P}(k-1)\mathbf{A})\boldsymbol{\Lambda}(k)\boldsymbol{l}_j = 0 & \text{for all } (i,j) \in \mathcal{X} \\ \boldsymbol{l}_i^{\mathsf{T}}\mathbf{K}(k)\boldsymbol{l}_j = 0 & \text{for all } (i,j) \notin \mathcal{X} \end{cases}, \tag{B4}$$

where $\mathcal{X}$ is defined as in (A2). Using vectorizing operators and the selection matrix $\mathbf{Z}$, the first part of (B4) can be rewritten as

$$\mathbf{Z}\ \mathrm{vec}(\mathbf{S}(k)\mathbf{K}(k)\boldsymbol{\Lambda}(k)) = \mathbf{Z}\ \mathrm{vec}(\mathbf{B}^{\mathsf{T}}\mathbf{P}(k-1)\mathbf{A}\boldsymbol{\Lambda}(k)). \tag{B5}$$

Note also that the second part of (B4) implies that

$$\mathrm{vec}(\mathbf{K}(k)) = \mathbf{Z}^{\mathsf{T}}\mathbf{Z}\ \mathrm{vec}(\mathbf{K}(k)). \tag{B6}$$

The left-hand side of (B5) can be rewritten as

$$\mathbf{Z}\ \mathrm{vec}(\mathbf{S}(k)\mathbf{K}(k)\boldsymbol{\Lambda}(k)) = \mathbf{Z}(\boldsymbol{\Lambda}(k) \otimes \mathbf{S}(k))\ \mathrm{vec}(\mathbf{K}(k)) = \mathbf{Z}(\boldsymbol{\Lambda}(k) \otimes \mathbf{S}(k))\bar{\mathbf{Z}}\mathbf{Z}\ \mathrm{vec}(\mathbf{K}(k)). \tag{B7}$$

Using (B7) in (B5) yields

$$\mathbf{Z}(\boldsymbol{\Lambda}(k) \otimes \mathbf{S}(k))\mathbf{Z}^{\mathsf{T}}\mathbf{Z}\ \mathrm{vec}(\mathbf{K}(k)) = \mathbf{Z}\ \mathrm{vec}(\mathbf{B}^{\mathsf{T}}\mathbf{P}(k-1)\mathbf{A}\boldsymbol{\Lambda}(k)),$$

which implies that

$$\mathbf{Z}\ \mathrm{vec}(\mathbf{K}(k)) = (\mathbf{Z}(\boldsymbol{\Lambda}(k) \otimes \mathbf{S}(k))\mathbf{Z}^{\mathsf{T}})^{-1}\mathbf{Z}\ \mathrm{vec}(\mathbf{B}^{\mathsf{T}}\mathbf{P}(k-1)\mathbf{A}\boldsymbol{\Lambda}(k)).$$

Finally, multiplying by $\mathbf{Z}^{\mathsf{T}}$ on both sides and using (B6) yields the closed-form solution (18).