Structured computation of optimal controls for constrained cascade systems

Michael Cantoni, Farhad Farokhi, Eric Kerrigan, Iman Shames

ARTICLE HISTORY

Compiled June 23, 2017

ABSTRACT

Constrained finite-horizon linear-quadratic optimal control problems are studied within the context of discrete-time dynamics that arise from the series interconnection of subsystems. A structured algorithm is devised for computing the Newton-like steps of primal-dual interior-point methods for solving a particular re-formulation of the problem as a quadratic program. This algorithm has the following properties: (i) the computation cost scales linearly in the number of subsystems along the cascade; and (ii) the computations can be distributed across a linear processor network, with localized problem data dependencies between the processor nodes and low communication overhead. The computation cost of the approach, which is based on a fixed permutation of the primal and dual variables, scales cubically in the time horizon of the original optimal control problem. Limitations in these terms are explored as part of a numerical example. This example involves application of the main results to model data for the cascade dynamics of an automated irrigation channel in particular.

KEYWORDS

Interior-point methods; large-scale systems; model predictive control.

1. Introduction

The application of model predictive control involves computing the solution of constrained finite-horizon optimal control problems in a receding horizon fashion (Maciejowski, 2002; Rawlings & Mayne, 2009). Linear-quadratic (LQ) optimal control problems with discretely-indexed system dynamics are amenable to re-formulation as a quadratic program (QP). Particular QP formulations have favourable structure from the perspective of computation, in that it is possible to efficiently determine the modified Newton iterations of interior-point methods for solving the corresponding Karush-Kuhn-Tucker (KKT) optimality conditions. Specifically, each Newton step incurs a computation cost that scales linearly in the length of the horizon over which the state-space description of the dynamics is defined, and over which the separable penalty function and state/input constraints are specified (Diehl, Ferreau, & Haverbeke, 2009; Domahidi, Zraggen, Zeilinger, Morari, & Jones, 2012; Rao, Wright, & Rawlings, 1998; Wang & Boyd, 2010; Wills & Heath, 2003; Wright, 1993).

Cantoni, Farokhi and Shames are with the Department of Electrical and Electronic Engineering, The University of Melbourne, Australia (Email: {cantoni,farhad.farokhi,iman.shames}@unimelb.edu.au). Kerrigan is with the Departments of Electrical and Electronic Engineering and Aeronautics, Imperial College London, U.K. (Email: e.kerrigan@imperial.ac.uk). This work is supported by the Australian Research Council (LP130100605) and a McKenzie Fellowship.

This paper is about LQ optimal control problems with system dynamics arising from the cascade interconnection of N subsystems subject to constraints over a finite time horizon. Such systems can be found in the study of irrigation and drainage networks (Li, Cantoni, & Weyer, 2005; Puig et al., 2009; Soltanian & Cantoni, 2015), mutli-reservoir and hydro-power systems (Labadie, 2004; Şahin & Morari, 2010), vehicle platoons (Seiler, Pant, & Hedrick, 2004), and supply chain management (Disney & Towill, 2003). Attention is drawn to exploiting structure in the spatial dimension of the dynamics. The key observation relates to the block tridiagonal structure of the KKT conditions for a particular QP re-formulation of the optimal control problem. This can be exploited at each Newton step of a primal-dual interior-point method for solving the QP. Indeed, the computation cost of each step can be made to scale linearly in N . Moreover, the computations can be distributed across a linear network consisting of N processors, with localized problem data dependencies between nodes and low communication overhead.

The temporal structure of the system dynamics also contributes to sparsity in the KKT conditions. However, exploiting the fixed permutation of variables that underpins the structured algorithm developed below, and its aforementioned distributed implementation, leads to cubic scaling of the computation cost in the length of the time horizon of the original optimal control problem. The development illuminates the challenge of overcoming such a cubic dependence, while retaining linear scaling in N . Such limitations are also explored numerically, by comparison with the use of MATLAB’s ‘backslash’ operator to solve the system of linear equations associated with a Newton step. Note, a distributed implementation of ‘backslash’ would involve substantial information exchange.

The paper is organized as follows. The class of finite-horizon optimal control problems studied is defined in Section 2.1. The equivalent cascade-structured QP is formulated in Section 2.2. The correspondingly structured algorithm for computing Newton steps in a primal-dual interior-point method for solving this QP is then developed in Section 3. Finally, a numerical example based on model data for an automated irrigation channel is presented Section 4. Some concluding remarks are provided in Section 5. Proofs of all results are deferred to the Appendix.

2. Problem Formulation

A class of finite-horizon optimal control problems with cascade dynamics is defined below. The defining features relate to the cascade structure of subsystems involved and the assumed separability of the state and input inequality constraints, and the penalty function across the spatial and temporal dimensions of the system dynamics. An equivalent QP with computationally favorable structure, from the perspective of scalability in cascade length, is formulated in Section 2.2.

2.1. *Constrained finite-horizon optimal control of cascades*

Consider the cascade of $N \in \mathbb{N}$ linear discrete-time subsystems with dynamics modelled by

$$x_j(t+1) = A_j(t)x_j(t) + B_j(t)u_j(t) + E_j(t)x_{j-1}(t), \quad (1)$$

given initial conditions $x_j(0) = \xi_j \in \mathbb{R}^{n_j}$, and model data $A_j(t) \in \mathbb{R}^{n_j \times n_j}$, $B_j(t) \in \mathbb{R}^{n_j \times m_j}$, and $E_j(t) \in \mathbb{R}^{n_j \times n_{j-1}}$, for $j = 1, \dots, N$ and $t = 0, \dots, T-1$, with $E_1(t) = 0$ so that the spatial boundary value $x_0(t)$ can be taken to be zero. The parameter $T \in \mathbb{N}$ is a specified time (or prediction) horizon. The problem of interest is to select the control inputs $u_j(t)$, for each subsystem index $j = 1, \dots, N$ and sample time index $t = 0, \dots, T-1$, in order to minimize the separable penalty function

$$J = \frac{1}{2} \sum_{j=1}^N \left(\left(\sum_{t=0}^{T-1} \begin{bmatrix} x_j(t)^\top & u_j(t)^\top \end{bmatrix} \begin{bmatrix} Q_j(t) & S_j(t)^\top \\ S_j(t) & R_j(t) \end{bmatrix} \begin{bmatrix} x_j(t) \\ u_j(t) \end{bmatrix} \right) + x_j(T)^\top P_j x_j(T) \right), \quad (2)$$

subject to separable inequality constraints

$$M_j(t)x_j(t) + L_j(t)u_j(t) \leq c_j(t) \quad \text{for } j = 1, \dots, N, \text{ and } t = 0, \dots, T, \quad (3)$$

where

$$\begin{bmatrix} Q_j(t) & S_j(t)^\top \\ S_j(t) & R_j(t) \end{bmatrix} \succeq 0, \quad (4)$$

$M_j(t) \in \mathbb{R}^{\nu_j \times n_j}$, $L_j(t) \in \mathbb{R}^{\nu_j \times m_j}$ and $c_j(t) \in \mathbb{R}^{\nu_j}$, with $0 \prec R_j(t) = R_j(t)^\top \in \mathbb{R}^{m_j \times m_j}$, $M_j(0) = 0$, $L_j(T) = 0$, $S_j(T) = 0$, and $Q_j(T) = P_j = P_j^\top \succeq 0$, are given for $t = 0, \dots, T$ and $j = 1, \dots, N$. Note that (4) implies $Q_j(t) = Q_j(t)^\top \succeq 0$ and $Q_j(t) - S_j(t)^\top R_j(t)^{-1} S_j(t) \succeq 0$, since $R_j(t) \succ 0$.

The optimal control problem defined above can be reformulated as a standard QP in a number of ways. The style of QP reformulation described in (Wright, 1993) (within the context of exploiting temporal structure) leads to an interior-point method with Newton steps that involve the solution of a linear system of equations with favourable block tridiagonal structure. This is exploited in Section 3 to devise an algorithm with per-iteration computation cost that scales linearly with cascade length N .

2.2. A cascade-structured QP formulation

Note that the equality constraint corresponding to the dynamics (1) can be reformulated as follows. Define, for $j = 1, \dots, N$, the primal variables

$$\hat{u}_j = [u_j(0)^\top \quad \dots \quad u_j(T-1)^\top]^\top \in \mathbb{R}^{m_j T}$$

and

$$\hat{x}_j = [x_j(0)^\top \quad \dots \quad x_j(T)^\top]^\top \in \mathbb{R}^{n_j(T+1)}.$$

Then

$$-\hat{A}_j \hat{x}_j + \hat{E}_j \hat{x}_{j-1} + \hat{B}_j \hat{u}_j + \hat{H}_j \xi_j = 0, \quad (5)$$

where

$$\hat{A}_j = \begin{bmatrix} I & 0 & \cdots & \cdots & 0 \\ -A_j(0) & I & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -A_j(T-1) & I \end{bmatrix}, \quad \hat{B}_j = \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ B_j(0) & \ddots & & \vdots \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & B_j(T-1) \end{bmatrix},$$

$$\hat{E}_j = \begin{bmatrix} 0 & \cdots & \cdots & \cdots & 0 \\ E_j(0) & \ddots & & & \vdots \\ 0 & E_j(1) & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & E_j(T-1) & 0 \end{bmatrix}, \quad \text{and} \quad \hat{H}_j = \begin{bmatrix} I \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix},$$

for $j = 1, \dots, N$. The boundary condition for (5) can be taken as $\hat{x}_0 = 0$ since $\hat{E}_1 = 0$. Moreover, with

$$\begin{aligned} \hat{Q}_j &= \text{diag}(Q_j(0), \dots, Q_j(T)) \in \mathbb{R}^{n_j(T+1) \times n_j(T+1)}, \\ \hat{R}_j &= \text{diag}(R_j(0), \dots, R_j(T-1)) \in \mathbb{R}^{m_j T \times m_j T}, \\ \hat{S}_j &= [\text{diag}(S_j(0), \dots, S_j(T)) \quad 0] \in \mathbb{R}^{m_j T \times n_j(T+1)}, \\ \hat{M}_j &= \text{diag}(M_j(0), \dots, M_j(T)) \in \mathbb{R}^{\nu_j(T+1) \times n_j(T+1)}, \\ \hat{L}_j &= [\text{diag}(L_j(0), \dots, L_j(T-1))^\top \quad 0]^\top \in \mathbb{R}^{\nu_j(T+1) \times m_j T}, \text{ and} \\ \hat{c}_j &= [c_j^\top(0) \quad \cdots \quad c_j^\top(T)]^\top \in \mathbb{R}^{\nu_j(T+1)}, \end{aligned}$$

the performance index and constraints can be reformulated as

$$J = \frac{1}{2} \sum_{j=1}^N \left(\hat{x}_j^\top \hat{Q}_j \hat{x}_j + \hat{u}_j^\top \hat{R}_j \hat{u}_j + 2\hat{u}_j^\top \hat{S}_j \hat{x}_j \right) \quad (6)$$

and

$$\hat{M}_j \hat{x}_j + \hat{L}_j \hat{u}_j - \hat{c}_j \leq 0 \text{ for } j = 1, \dots, N, \quad (7)$$

respectively. To summarize, the problem of minimizing (2) subject to the equality constraints (1), and inequality constraints (3), takes the form of the following standard QP:

$$\min_{\substack{(\hat{u}_1, \dots, \hat{u}_N) \in \mathbb{R}^{m_1 T} \times \dots \times \mathbb{R}^{m_N T} \\ (\hat{x}_1, \dots, \hat{x}_N) \in \mathbb{R}^{n_1(T+1)} \times \dots \times \mathbb{R}^{n_N(T+1)}}} \quad (6) \quad \text{subject to} \quad (5) \text{ and } (7).$$

As mentioned in the introduction, interior point methods for solving QPs can be tailored to exploit the cascade structure of this particular formulation. This is explored below.

3. Exploiting the cascade structure in interior-point methods

Central-path (and other) primal-dual interior-point methods for solving a QP involve the application of modified Newton iterations to solve the corresponding KKT conditions for optimality (Wright, 1997, Ch. 16). For the QP formulated at the end of the preceding section, the KKT conditions are given by

$$\hat{Q}_j \hat{x}_j + \hat{S}_j^\top \hat{u}_j - \hat{A}_j^\top p_j + \hat{M}_j^\top \lambda_j + \hat{E}_{j+1}^\top p_{j+1} = 0, \quad (8a)$$

$$\hat{S}_j \hat{x}_j + \hat{R}_j \hat{u}_j + \hat{B}_j^\top p_j + \hat{L}_j^\top \lambda_j = 0, \quad (8b)$$

$$-\hat{A}_j \hat{x}_j + \hat{E}_j \hat{x}_{j-1} + \hat{B}_j \hat{u}_j + \hat{H}_j \xi_j = 0, \quad (8c)$$

$$\hat{M}_j \hat{x}_j + \hat{L}_j \hat{u}_j - \hat{c}_j + \theta_j = 0, \quad (8d)$$

$$\Lambda_j \Theta_j \mathbf{1} = 0, \quad (8e)$$

$$[\lambda_i^\top \quad \theta_j^\top]^\top \geq 0, \quad (8f)$$

for $j = 1, \dots, N$, where $\mathbf{1} = [1 \ \dots \ 1]^\top \in \mathbb{R}^{\nu_j}$, $\hat{x}_0 = 0$ (since $\hat{E}_1 = 0$), and $\hat{E}_{N+1} = 0$. With the exception of (8e), these optimality conditions are linear in the primal variables $\hat{x}_j$ and $\hat{u}_j$, the dual variables

$$p_j \in \mathbb{R}^{n_j \times (T+1)} \quad \text{and} \quad \lambda_j = [(\lambda_j)_1 \ \dots \ (\lambda_j)_{\nu_j}]^\top \in \mathbb{R}^{\nu_j},$$

with $\Lambda_j = \text{diag}((\lambda_j)_1, \dots, (\lambda_j)_{\nu_j}) \in \mathbb{R}^{\nu_j \times \nu_j}$, and the slack variables

$$\theta_j = [(\theta_j)_1 \ \dots \ (\theta_j)_{\nu_j}]^\top \in \mathbb{R}^{\nu_j},$$

with $\Theta_j = \text{diag}((\theta_j)_1, \dots, (\theta_j)_{\nu_j}) \in \mathbb{R}^{\nu_j \times \nu_j}$, for $j = 1, \dots, N$. At iteration $k \in \mathbb{N}$, the form of a modified Newton update is given by

$$s[k+1] = s[k] + \alpha[k] \cdot \delta[k], \quad (9)$$

where

$$s[k] = [s_1[k]^\top \ \dots \ s_N[k]^\top]^\top \quad (10)$$

is the outcome of the of the preceding step (or the initialization when $k = 1$) with $s_j[k] = [\hat{x}_j[k]^\top \ \hat{u}_j[k]^\top \ p_j[k]^\top \ \lambda_j[k]^\top \ \theta_j[k]^\top]^\top$ comprising the iterate of primal, dual and slack variables associated with subsystem index $j = 1, \dots, N$. The step direction

$$\delta[k] = [\delta_1[k]^\top \ \dots \ \delta_N[k]^\top]^\top$$

is computed to solve the linearized KKT conditions, expressed as

$$\begin{bmatrix} D_1[k] & -\Upsilon_2^\top & 0 & \dots & 0 \\ -\Upsilon_2 & D_2[k] & -\Upsilon_3^\top & \ddots & \vdots \\ 0 & -\Upsilon_3 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & D_{N-1}[k] & -\Upsilon_N^\top \\ 0 & \dots & 0 & -\Upsilon_N & D_N[k] \end{bmatrix} \begin{bmatrix} \delta_1[k] \\ \delta_2[k] \\ \vdots \\ \vdots \\ \delta_N[k] \end{bmatrix} = \begin{bmatrix} \rho_1[k] \\ \rho_2[k] \\ \vdots \\ \vdots \\ \rho_N[k] \end{bmatrix}. \quad (11)$$

The step-size parameter $\alpha[k] > 0$ is selected online to ensure $\lambda_j[k+1] > 0$ and $\theta_j[k+1] > 0$. In the system of linear equations (11),

$$\rho_j[k] = -\Upsilon_j s_{j-1}[k] + D_j[k] s_j[k] - \Upsilon_{j+1}^\top s_{j+1}[k] + \begin{bmatrix} 0^\top & 0^\top & (\hat{H}_j \xi_j)^\top & \hat{c}_j^\top & \eta_j[k]^\top \end{bmatrix}^\top, \quad (12a)$$

$$D_j[k] = \begin{bmatrix} \hat{Q}_j & \hat{S}_j^\top & -\hat{A}_j^\top & \hat{M}_j^\top & 0 \\ \hat{S}_j & \hat{R}_j & \hat{B}_j^\top & \hat{L}_j^\top & 0 \\ -\hat{A}_j & \hat{B}_j & 0 & 0 & 0 \\ \hat{M}_j & \hat{L}_j & 0 & 0 & I \\ 0 & 0 & 0 & \Theta_j[k] & \Lambda_j[k] \end{bmatrix}, \quad (12b)$$

$$\Upsilon_j = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -\hat{E}_j & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (12c)$$

$$\eta_j[k] = -\Theta_j[k] \lambda_j[k] - \Lambda_j[k] \theta_j[k] + \Lambda_j[k] \Theta_j[k] \mathbf{1} - \sigma[k] \mu[k] \mathbf{1}, \quad (12d)$$

$$\Lambda_j[k] = \text{diag}((\lambda_j[k])_1, \dots, (\lambda_j[k])_{\nu_j}), \quad (12e)$$

$$\Theta_j[k] = \text{diag}((\theta_j[k])_1, \dots, (\theta_j[k])_{\nu_j}), \quad (12f)$$

for $j = 1, \dots, N$ (with $\Upsilon_{N+1} = 0$, since $\hat{E}_{N+1} = 0$), $\sigma[k] \in (0, 1)$ is a centring parameter, and $\mu[k] = (\sum_{j=1}^N (\lambda_j[k])^\top \theta_j[k]) / (\sum_{j=1}^N \nu_j)$ is the so-called duality gap. As shown below, the linear equation (11) has a unique solution under the assumptions made on the system model and penalty function data when formulating the underlying LQ optimal control problem.

Computing the solution of (11) is facilitated by the apparent block tridiagonal structure. As such, the computation cost of solving for $\delta[k]$ can be made to scale linearly in N ; see e.g., (Meurant, 1992) and (Bevilacqua, Codenotti, & Romani, 1988). Ignoring the structure would incur computation cost of order N^3 . The temporal structure of the underlying optimal control problem also gives rise to sparse matrices $D_j[k]$. However, simultaneously exploiting the additional temporal structure appears to be difficult within the context of exploiting the cascade block tridiagonal structure of (11). Use of the following lemma, which is based on the fixed permutation of variables in (10) that yields the tridiagonal structure, leads to a solution algorithm with computation cost that scales as T^3 . The challenge of improving scalability in the time-horizon length is discussed subsequently. On the other hand, the computation involved can be dis-

tributed with low communication overhead and localized problem data dependencies between processor nodes. This aspect of the proposed method is described in more detail below. A proof of the lemma is deferred to the Appendix.

Lemma 1. *Dropping the explicit dependence on iteration index $k \in \mathbb{N}$, the unique solution of (11) is given by the backward and forward recursions*

$$\tilde{\rho}_j = \begin{cases} \rho_N & \text{for } j = N \\ \rho_j + \Upsilon_{j+1}^\top \Sigma_{j+1}^{-1} \tilde{\rho}_{j+1} & \text{for } j = N-1, \dots, 1 \end{cases},$$

and

$$\delta_j = \begin{cases} \Sigma_1^{-1} \tilde{\rho}_1 & \text{for } j = 1 \\ \Sigma_j^{-1} (\tilde{\rho}_j + \Upsilon_j \delta_{j-1}) & \text{for } j = 2, \dots, N \end{cases},$$

respectively, where

$$\Sigma_j = \begin{cases} D_N & \text{for } j = N \\ D_j - \Upsilon_{j+1}^\top (\Sigma_{j+1})^{-1} \Upsilon_{j+1} & \text{for } j = N-1, \dots, 1 \end{cases}. \quad (13)$$

In particular, each $\Sigma_j \in \mathbb{R}^{p_j \times p_j}$, where $p_j = ((2n_j + 2\nu_j)(T+1) + m_j T)$, is non-singular as required.

Remark 1. *To compute expressions involving the inverse of a matrix, one should solve the corresponding system of linear equations, rather than inverting and then multiplying. For example, to compute $X = \Sigma^{-1}Y$ given Σ and Y , one can solve $\Sigma X = Y$ for X . Structure in the matrix Σ can be exploited to efficiently construct factorizations (e.g., PLDU) that facilitate computation of the solution. This happens when ‘backslash’ is employed in MATLAB with sparse arguments; i.e., $X = \Sigma \backslash Y$. Indeed, exploiting structure in this way underlies Lemma 1. Another example can be seen in Lemma 2 below. Compositions of inverses and matrices appearing therein should also be computed as just described.*

Remark 2. *Using Lemma 1 to solve (11) incurs a computation cost that scales linearly in the number of subsystems N . However, the 11-block of Σ_j (cf. $\tilde{Q}_j \succeq 0$ in the proof of Lemma 1) becomes a full matrix for $j = N-1, \dots, 1$, despite the sparsity of $\hat{Q}_j$, $\hat{R}_j$, $\hat{S}_j$, $\hat{A}_j$, $\hat{B}_j$, $\hat{M}_j$, $\hat{L}_j$, Θ_j , and Λ_j . Therefore, the complexity of computing the above recursions ultimately scales cubically in T . This yields an overall computation cost of order NT^3 in the worst case.*

Remark 3. *Note that the backward and forward recursions in Lemma 1 are amenable to parallelization, which leads to a computation time that scales as $\log N$, given the computation and communication resources required; see e.g., (Nielsen & Axehill, 2014).*

The following result encapsulates a structured method for computing products such as $\Sigma_j^{-1}Y$ given Y . This method involves the inverse of block diagonal and block bidiagonal matrices of size $T \times T$, and the inverse of one unstructured positive-definite

$(m_j T) \times (m_j T)$ matrix. Note, m_j is typically smaller than n_j and ν_j , and thus, $p_j = ((2n_j + 2\nu_j)(T + 1) + m_j T)$. Nevertheless, the overall computation cost is order T^3 . A proof is deferred to the Appendix.

Lemma 2. *Given $Y_1 \in \mathbb{R}^{n_j(T+1) \times q}$, $Y_2 \in \mathbb{R}^{m_j T \times q}$, $Y_3 \in \mathbb{R}^{\nu_j(T+1) \times q}$, $Y_4 \in \mathbb{R}^{\nu_j(T+1) \times q}$, $Y_5 \in \mathbb{R}^{\nu_j(T+1) \times q}$, and Σ_j as defined in (13) for $j \in \{1, \dots, N\}$, the unique solution of*

$$\Sigma_j \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{bmatrix} = \begin{bmatrix} \hat{Q}_j - \hat{E}_{j+1}^\top (\Sigma_{j+1}^{-1})_{33} \hat{E}_{j+1} & \hat{S}_j^\top & -\hat{A}_j^\top & \hat{M}_j^\top & 0 \\ \hat{S}_j & \hat{R}_j & \hat{B}_j^\top & \hat{L}_j^\top & 0 \\ -\hat{A}_j & \hat{B}_j & 0 & 0 & 0 \\ \hat{M}_j & \hat{L}_j & 0 & 0 & I \\ 0 & 0 & 0 & \Theta_j & \Lambda_j \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{bmatrix} = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ Y_5 \end{bmatrix}, \quad (14)$$

has the following form:

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \Psi_j \Omega_j \Psi_j^\top \left(\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} + \begin{bmatrix} \hat{M}_j^\top \Theta_j^{-1} \Lambda_j & -\hat{M}_j^\top \Theta_j^{-1} \\ \hat{L}_j^\top \Theta_j^{-1} \Lambda_j & -\hat{L}_j^\top \Theta_j^{-1} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Y_4 \\ Y_5 \end{bmatrix} \right) \quad \text{and} \\ \begin{bmatrix} X_4 \\ X_5 \end{bmatrix} = \begin{bmatrix} -\Theta_j^{-1} \Lambda_j & \Theta_j^{-1} \\ I & 0 \end{bmatrix} \left(- \begin{bmatrix} \hat{M}_j & \hat{L}_j \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} + \begin{bmatrix} Y_4 \\ Y_5 \end{bmatrix} \right),$$

where

$$\Psi_j = \begin{bmatrix} I & 0 & 0 \\ -\tilde{R}_j^{-1} \tilde{S}_j & I & -\tilde{R}_j^{-1} \tilde{B}_j^\top \tilde{A}_j^{-\top} \\ 0 & 0 & \tilde{A}_j^{-\top} \end{bmatrix}, \quad (15)$$

$$\Omega_j = \begin{bmatrix} \tilde{Z}_j & 0 & -\tilde{W}_j^\top \\ 0 & \tilde{R}_j^{-1} & 0 \\ -\tilde{W}_j & 0 & -\tilde{W}_j \tilde{Q}_j \end{bmatrix}, \quad (16)$$

$$\tilde{Z}_j = \tilde{A}_j^{-1} \tilde{B}_j (\tilde{R}_j + \tilde{B}_j^\top \tilde{A}_j^{-\top} \tilde{Q}_j \tilde{A}_j^{-1} \tilde{B}_j)^{-1} \tilde{B}_j^\top \tilde{A}_j^{-\top}, \quad (17)$$

$$\tilde{W}_j = I - \tilde{Q}_j \tilde{Z}_j, \quad (18)$$

$$\tilde{H}_j = \begin{bmatrix} \tilde{Q}_j & \tilde{S}_j^\top \\ \tilde{S}_j & \tilde{R}_j \end{bmatrix} = \begin{bmatrix} \hat{Q}_j - \hat{E}_{j+1}^\top (\Sigma_{j+1}^{-1})_{33} \hat{E}_{j+1} & \hat{S}_j^\top \\ \hat{S}_j & \hat{R}_j \end{bmatrix} + \begin{bmatrix} \hat{M}_j^\top \\ \hat{L}_j^\top \end{bmatrix} \Theta_j^{-1} \Lambda_j \begin{bmatrix} \hat{M}_j & \hat{L}_j \end{bmatrix} \succeq 0, \quad (19)$$

$\tilde{Q}_j = \bar{Q}_j - \tilde{S}_j^\top \tilde{R}_j^{-1} \tilde{S}_j \succeq 0$, $\tilde{A}_j = \hat{A}_j + \hat{B}_j \tilde{R}_j^{-1} \tilde{S}_j$ (which is non-singular), $\tilde{B}_j = \hat{B}_j$, and $\hat{E}_{N+1} = 0$.

Remark 4. Constructing Θ_j^{-1} and $\tilde{A}_j^{-1}$ involves computation cost that scales linearly in T because of the diagonal structure of Θ_j and (lower) block bidiagonal structure of $\tilde{A}_j$, respectively. On the other hand, the $(m_j T) \times (m_j T)$ symmetric positive definite matrix $\tilde{R}_j = (\tilde{R}_j + \tilde{B}_j^\top \tilde{A}_j^{-\top} \tilde{Q}_j \tilde{A}_j^{-1} \tilde{B}_j)$ is full. So the cost of computing $\tilde{Z}_j$ (e.g., by $\tilde{A}_j^{-1} * \tilde{B}_j * (\tilde{R}_j \setminus \tilde{B}_j^\top) * \tilde{A}_j^{-\top}$ in subsequent MATLAB experimentation) scales as T^3 .

An alternative approach to Lemma 2 is possible based on the inverse of $\tilde{H}_j$, when it is non-singular, as in the case

$$\begin{bmatrix} \hat{Q}_j & \hat{S}_j^\top \\ \hat{S}_j & \hat{R}_j \end{bmatrix} \succ 0,$$

for example. However, the approach would also involve computation cost that scales as T^3 , since the $(n_j(T+1)) \times (n_j(T+1))$ matrix $\bar{Q}_j = \hat{Q}_j - \hat{E}_{j+1}^\top (\Sigma_{j+1}^{-1})_{33} \hat{E}_{j+1}$ in the 11-block of $\tilde{H}_j$ is full for $j < N$. Indeed, the cost would typically exceed the cost of the approach based on the inverse of $\hat{R}_j$, since $m_j < n_j$. By contrast, as seen in (Domahidi et al., 2012; Wang & Boyd, 2010) for example, taking such an approach can be fruitful within the context of exploiting temporal structure in optimal control problems, since then the 11-block (i.e., the Q block) of the corresponding Σ matrix is sparse. Scope for achieving better than T^3 scalability, while retaining linear scaling in N in computation time, is explored numerically in the next section.

To conclude this section, on exploiting cascade structure in the application of interior-point methods to the QP formulated at the end of Section 2.2, one final observation is made about the calculations required to implement the solution of (11) according to Lemma 1. Specifically, the computations can be distributed across a linear network of N processors, one for each subsystem, that mirrors the underlying cascade of subsystems. The processor associated with subsystem $j \in \{2, \dots, N-1\}$ needs to send $\Upsilon_j \Sigma_j^{-1} \bar{\rho}_j$ to, and receive $\Upsilon_j \delta_{j-1}$ from, the agent for subsystem $j-1$. Moreover, the processor for subsystem $j \in \{1, \dots, N-1\}$ only needs access to $\hat{E}_{j+1}$ (i.e., the influence of j on $j+1$) and no further model, penalty function, or constraint data associated with other subsystems. The processor for $j=1$ need only communicate with the agent for $j=2$ and the processor for $j=N$ need only communicate with the agent for $j=N-1$. For each Newton iteration of the interior-point method, information is communicated in a sequence of steps, once up the cascade, and then once down the cascade, without further iteration. Calculation of the the duality gap and step size updates can be determined by a similar pattern of up the cascade, and then down the cascade, exchange of information. Processor pipelining, by the provision of buffering in the communication, for example, could be exploited to improve throughput, without improving latency which could degrade slightly. Of course, it is necessary to process a sufficient number of Newton updates in the interior-point method. An appealing features of interior-point methods is the typically small number of iterations (e.g., ten to fifteen) needed to reach a good solution to the QP. As such, the overall communication overhead of a distributed version of the method is low. The localization of model and other problem data can also be considered advantageous from security and privacy perspectives. As noted in Remark 3, the structure of the recursions in Lemma 1 are amenable to parallelization, as proposed in (Nielsen & Axehill, 2014), for example. The communication overhead in this case would exceed the overhead of the distributed computations described above.

4. Example

Numerical results are obtained by applying the preceding developments to model data for an automated irrigation channel, within the context of a water-level reference planning problem. A state-space model of the form (1) can be constructed for channels

that operate under decentralized distant-downstream control (Soltanian & Cantoni, 2015); each subsystem corresponds to a stretch of channel between adjacent flow regulators called a pool. An optimal control problem can be formulated to determine the water-level reference input for each pool, for a planning horizon over which a load forecast may be known, subject to hard constraints on the water-levels and flows. For this example, the model data for the pools, the distributed controllers, the discretization sample-period, and constraint levels are all taken from (Neshastehriz, Cantoni, & Shames., 2014); the many pool channels considered here are constructed by concatenating sections of the channel considered therein. The model for each subsystem (i.e., pool) involves one control input, the water-level reference, and four states, including two for the water-level dynamics and two for the PI-type feedback controller that sets the upstream inflow on the basis of the measured downstream water-level error. Box type constraints on the water-level and controller flow output states are to be satisfied in addition to the water-level reference input constraints. For each subsystem $j \in \{1, \dots, N\}$, the following hold: $n_j = 4$; $m_j = 1$; and $\nu_j = 6$. The other model parameters (e.g., entries of state-space matrices) are non-uniform along the channel.

Figure 1. Normalized computation time for 16 Newton iterations. Top plots show case (i): Solving (11) via Lemma 1 and Lemma 2. Bottom plots shows case (ii): Solving (11) with MATLAB2016b ‘backslash’ operator and sparse argument representations.

Figure 1 shows the (forced single thread) normalized computation time of exactly sixteen Newton updates computed by two different methods for solving (11), as the

cascade length, N , and time-horizon length, T , are both varied from 1 to 55. The two methods considered are: (i) solution of (11) by using Lemma 1 and Lemma 2; and (ii) solution of (11) by using the MATLAB2016b implementation of the ‘backslash’ operator on sparse representations of the tridiagonal matrix X on the left-hand side and ρ on the right; i.e., $\mathbf{X}=\text{sparse}(X)$; $\mathbf{rho}=\text{sparse}(\rho)$; $\mathbf{delta}=\mathbf{X}\backslash\mathbf{rho}$. Comparison of ‘backslash’ to the algorithm based on Lemmas 1 and 2 serves to gauge the scope for achieving better than cubic scaling of the computation in T , while retaining linear scaling in N . It is expected that ‘backslash’ can do better in these terms because of the freedom it has to permute the variables and equations in different ways for different values of T and N . While this freedom can be exploited to manage the bandwidth of the sparse matrix X , and thus computation time, it would impact the extent to which the computation can be distributed with low communication overhead and localized problem data dependencies between processor nodes. By contrast, in case (i) the permutation of primal and dual variables is fixed in the same way for all N and T , which underlies the aforementioned distributed algorithm implementation. The proposed method is run in MATLAB as a script, coded with Remark 1 in mind. As such, the relative shape of the plots is important, not the value of normalized computation time (which is comparable.)

It can be seen from Figure 1 that the computation time in both cases scales (close to) linearly in the length N of the cascade. Clearly case (ii) is better in terms of how the computation scales in the length T of the time-horizon, since case (i) exhibits cubic scaling in T . This is consistent with the cubic scaling of computation cost in the dimension of the system state for related methods tailored to exploit the temporal structure of optimal control problems; see e.g., (Domahidi et al., 2012; Rao et al., 1998; Wang & Boyd, 2010)

Figure 2. Normalized computation time for 16 Newton iterations. Case (i), solving (11) via Lemma 1 and Lemma 2 (star). Case (ii), solving (11) with MATLAB2016b ‘backslash’ operator and sparse input representations (circle). Slice $N = T$ (blue). Slice $N = 27$ (black). Slice $T = 27$ (red).

Figure 2 shows the computation time along three 1-D slices of the two mesh plots shown in Figure 1. Specifically, slice $N = T$ (blue), slice $N = 27$ (black), and slice $T = 27$ (red), with results for case (i) marked as *, and results for case (ii) marked as o. Take note of the log-log scale. It can be seen that case (i) exhibits linear scalability of computation cost in N for fixed T , cubic scaling in T for fixed N , and overall complexity NT^3 , as confirmed by the quartic in N characteristic of the slice $N = T$. For case (ii), it can be seen that the computation cost of ‘backslash’ actually scales super-linearly, and that for the slice $N = T$ the computation time scales as N^3 . While there appears to be some scope for exploiting additional temporal structure present in the optimal control problem, the preceding analysis and numerical experimentation suggest that it may be difficult to achieve without sacrificing linear scaling in N .

5. Conclusion

The main contribution is an algorithm for computing the Newton iterations in an interior-point method for solving a structured QP formulation of constrained discrete-time optimal control problems with cascade dynamics. The algorithm exhibits computation cost that scales linearly in the number of subsystems along the cascade, and cubically in the length of the time horizon. Moreover, the algorithm admits a distributed implementation over a linear computation network with low communication overhead and localized problem data dependencies between the processor nodes. The underlying analysis, and numerical experimentation with data for an automated irrigation channel, illuminate the challenges associated with improving the way computation time scales in the length of the time-horizon, without sacrificing linear scalability in the cascade length, and scope for distributed computation in the way described. Ongoing work is focused on the exploration of avenues for simultaneous exploitation of temporal and spatial structure within the context of cascade systems, and on extending the approach to accommodate undirected spatial propagation of information in tree networks of dynamical systems.

Appendix A. A technical lemma and proofs

Lemma 3. *Given $Q = Q^\top \in \mathbb{R}^{n \times n}$, $R = R^\top \in \mathbb{R}^{m \times m}$, $S \in \mathbb{R}^{m \times n}$, $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $M \in \mathbb{R}^{\nu \times n}$, $L \in \mathbb{R}^{\nu \times m}$, suppose that $R \succ 0$, $Q \succeq 0$, and $Q - S^\top R^{-1} S \succeq 0$. Moreover, let*

$$D = \begin{bmatrix} Q & S^\top & -A^\top & M^\top & 0 \\ S & R & B^\top & L^\top & 0 \\ -A & B & 0 & 0 & 0 \\ M & L & 0 & 0 & I \\ 0 & 0 & 0 & \Theta & \Lambda \end{bmatrix},$$

with $\Theta = \text{diag}(\theta_1, \dots, \theta_\nu) \succ 0$ and $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_\nu) \succeq 0$, and define

$$\begin{bmatrix} \bar{Q} & \tilde{S}^\top \\ \tilde{S} & \tilde{R} \end{bmatrix} = \begin{bmatrix} Q & S^\top \\ S & R \end{bmatrix} + \begin{bmatrix} M^\top \\ L^\top \end{bmatrix} \Theta^{-1} \Lambda \begin{bmatrix} M & L \end{bmatrix} \succeq 0, \quad (\text{A1})$$

$\tilde{Q} = \bar{Q} - \tilde{S}^\top \tilde{R}^{-1} \tilde{S} \succeq 0$, $\tilde{A} = A + B\tilde{R}^{-1}\tilde{S}$, and $\tilde{B} = B$. If $\tilde{A}$ is non-singular, then D is non-singular. Moreover, the 33-block of D^{-1} is given by

$$-\tilde{A}^{-\top} \tilde{Q}^{\frac{1}{2}} (I + \tilde{Q}^{\frac{1}{2}} \tilde{A}^{-1} \tilde{B} \tilde{R}^{-1} \tilde{B}^\top \tilde{A}^{-\top} \tilde{Q}^{\frac{1}{2}})^{-1} \tilde{Q}^{\frac{1}{2}} \tilde{A}^{-1} \preceq 0.$$

Proof. First note that $\Theta \succ 0$ is invertible and that

$$\begin{bmatrix} 0 & I \\ \Theta & \Lambda \end{bmatrix}^{-1} = \begin{bmatrix} -\Theta^{-1}\Lambda & \Theta^{-1} \\ I & 0 \end{bmatrix}.$$

As such, it follows that D is invertible if and only if the Schur complement

$$\begin{bmatrix} \bar{Q} & \tilde{S}^\top & -A^\top \\ \tilde{S} & \tilde{R} & B^\top \\ -A & B & 0 \end{bmatrix} = \begin{bmatrix} Q & S^\top & -A^\top \\ S & R & B^\top \\ -A & B & 0 \end{bmatrix} - \begin{bmatrix} M^\top & 0 \\ L^\top & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & I \\ \Theta & \Lambda \end{bmatrix}^{-1} \begin{bmatrix} M & L & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

is non-singular. Moreover, the inverse of this matrix, when it exists, is precisely the matrix comprising the first three block rows and columns of D^{-1} ; as such, the 33-blocks coincide.

Now note that $\tilde{R} = R + L^\top \Theta^{-1} \Lambda L \succ 0$, since the diagonal matrix $\Theta^{-1} \Lambda \succeq 0$ and $R \succ 0$, and

$$\begin{aligned} & \begin{bmatrix} \tilde{Q} & 0 & -I \\ 0 & \tilde{R} & 0 \\ -I & 0 & -\tilde{A}^{-1} \tilde{B} \tilde{R}^{-1} \tilde{B}^\top \tilde{A}^{-\top} \end{bmatrix} \\ &= \begin{bmatrix} I & -\tilde{S}^\top \tilde{R}^{-1} & 0 \\ 0 & I & 0 \\ 0 & -\tilde{A}^{-1} \tilde{B} \tilde{R}^{-1} & \tilde{A}^{-1} \end{bmatrix} \begin{bmatrix} \bar{Q} & \tilde{S}^\top & -A^\top \\ \tilde{S} & \tilde{R} & B^\top \\ -A & B & 0 \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ -\tilde{R}^{-1} \tilde{S} & I & -\tilde{R}^{-1} \tilde{B}^\top \tilde{A}^{-\top} \\ 0 & 0 & \tilde{A}^{-\top} \end{bmatrix}, \end{aligned} \quad (\text{A2})$$

where $\tilde{A} = A + B\tilde{R}^{-1}\tilde{S}$ and $\tilde{Q} = \bar{Q} - \tilde{S}^\top \tilde{R}^{-1} \tilde{S} \succeq 0$ in view of (A1). Therefore, with $\tilde{B} = B$, the matrix D is non-singular if and only if the matrix

$$\begin{bmatrix} \tilde{Q} & -I \\ -I & -\tilde{A}^{-1} \tilde{B} \tilde{R}^{-1} \tilde{B}^\top \tilde{A}^{-\top} \end{bmatrix} = \begin{bmatrix} I & -\tilde{Q} \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 & -(I + \tilde{Q} \tilde{A}^{-1} \tilde{B} \tilde{R}^{-1} \tilde{B}^\top \tilde{A}^{-\top}) \\ -I & -\tilde{A}^{-1} \tilde{B} \tilde{R}^{-1} \tilde{B}^\top \tilde{A}^{-\top} \end{bmatrix} \quad (\text{A3})$$

is non-singular, which is the case if and only if $(I + \tilde{Q} \tilde{A}^{-1} \tilde{B} \tilde{R}^{-1} \tilde{B}^\top \tilde{A}^{-\top})$ is non-singular, or equivalently, if and only if $-1 \notin \text{spec}(\tilde{Q} \tilde{A}^{-1} \tilde{B} \tilde{R}^{-1} \tilde{B}^\top \tilde{A}^{-\top})$. That later holds because

$$\text{spec}(\tilde{Q} \tilde{A}^{-1} \tilde{B} \tilde{R}^{-1} \tilde{B}^\top \tilde{A}^{-\top}) \cup \{0\} = \text{spec}(\tilde{Q}^{\frac{1}{2}} \tilde{A}^{-1} \tilde{B} \tilde{R}^{-1} \tilde{B}^\top \tilde{A}^{-\top} \tilde{Q}^{\frac{1}{2}}) \cup \{0\} \subset \mathbb{R}_{\geq 0},$$

whereby D is invertible. In particular, the 22-block of the inverse of the left-hand side of (A3) is given by $-(I + \tilde{Q} \tilde{A}^{-1} \tilde{B} \tilde{R}^{-1} \tilde{B}^\top \tilde{A}^{-\top})^{-1} \tilde{Q} = -\tilde{Q}^{\frac{1}{2}} (I + \tilde{Q}^{\frac{1}{2}} \tilde{A}^{-1} \tilde{B} \tilde{R}^{-1} \tilde{B}^\top \tilde{A}^{-\top} \tilde{Q}^{\frac{1}{2}})^{-1} \tilde{Q}^{\frac{1}{2}}$, which is congruent to the 33-block of D^{-1} , via the transformation $\tilde{A}^{-1}$ in view of (A2). $\square$

A.1. Proof of Lemma 1

Proof. Recall that $\hat{R}_j \succ 0$, $\hat{Q}_j \succeq 0$, $\hat{Q}_j - \hat{S}_j^\top \hat{R}^{-1} \hat{S}_j \succeq 0$ and $\tilde{A}_j = \hat{A}_j + \hat{B}_j \hat{R}_j^{-1} \hat{S}_j$ is non-singular; n.b., the structure of $\tilde{A}_j$ is the same as the block bidiagonal structure of $\hat{A}_j$ with identity matrices along the block diagonal. Now using Lemma 3, observe that D_j is invertible for $j = 1, \dots, N$. Given the structure of Υ_j , the matrix Σ_j is the same as D_j except for the 11-block, which is $\tilde{Q}_j$ in the case of the latter and $\tilde{Q}_j = \hat{Q}_j - \hat{E}_j^\top (\Sigma_{j+1}^{-1})_{33} \hat{E}_j$ in the former. By Lemma 3, $(D_N^{-1})_{33} \preceq 0$, so that $(\Sigma_{j+1}^{-1})_{33} \preceq 0$, whereby $\tilde{Q}_j \succeq 0$, and thus, Σ_j is non-singular for $j = N-1$ by Lemma 3 again. Continuing this argument inductively yields the invertibility of Σ_j for $j = N-2, \dots, 1$. As such, the specified recursions for Σ_j , $\tilde{\rho}_j$ and δ_j are all well defined.

With $\Sigma_N = D_N$, solving the last block of (11) gives $\delta_N = \Sigma_N^{-1} \rho_N + \Sigma_N^{-1} \Upsilon_N \delta_{N-1}$. In turn,

$$\begin{bmatrix} D_1 & -\Upsilon_2^\top & 0 & \dots & 0 \\ -\Upsilon_2 & D_2 & -\Upsilon_3^\top & \ddots & \vdots \\ 0 & -\Upsilon_3 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & D_{N-2} & -\Upsilon_{N-1}^\top \\ 0 & \dots & 0 & -\Upsilon_{N-1} & \Sigma_{N-1} \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_{N-1} \end{bmatrix} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \tilde{\rho}_{N-1} \end{bmatrix},$$

where Σ_{N-1} and $\tilde{\rho}_{N-1}$ are defined as in the statement of the lemma. Continuing in this fashion yields the remaining expressions for Σ_j , $\tilde{\rho}_j$ and δ_j . The first block equation eventually becomes $\Sigma_1 \delta_1^k = \tilde{\rho}_1$, and thus, $\delta_1 = \Sigma_1^{-1} \tilde{\rho}_1$. $\square$

A.2. Proof of Lemma 2

Proof. It is shown in the proof of Lemma 1 that $-\hat{E}_{j+1}^\top (\Sigma_{j+1}^{-1})_{33} \hat{E}_{j+1} \succeq 0$. As such,

$$\begin{bmatrix} \hat{Q}_j & \hat{S}_j^\top \\ \hat{S}_j & \hat{R}_j \end{bmatrix} + \begin{bmatrix} -\hat{E}_{j+1}^\top (\Sigma_{j+1}^{-1})_{33} \hat{E}_{j+1} & 0 \\ 0 & 0 \end{bmatrix} \succeq 0,$$

and thus, $\tilde{Q}_j = \bar{Q}_j - \tilde{S}_j^\top \tilde{R}^{-1} \tilde{S} \succeq 0$ as claimed. Observe that $\tilde{A}_j$ is non-singular because it has the same lower block bidiagonal structure as $\hat{A}_j$, with identity blocks along the diagonal. Now recall that $\Theta_j \succ 0$, whereby $\begin{bmatrix} 0 & I \\ \Theta_j & \Lambda_j \end{bmatrix}^{-1} = \begin{bmatrix} -\Theta_j^{-1} \Lambda_j & \Theta_j^{-1} \\ I & 0 \end{bmatrix}$. Using this identity, the structure of (14), and the Schur complement, yields

$$\begin{bmatrix} X_4 \\ X_5 \end{bmatrix} = \begin{bmatrix} 0 & I \\ \Theta_j & \Lambda_j \end{bmatrix}^{-1} \left(\begin{bmatrix} Y_4 \\ Y_5 \end{bmatrix} - \begin{bmatrix} \hat{M}_j & \hat{L}_j & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} \right)$$

and

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} \bar{Q}_j & \tilde{S}_j^\top & -\hat{A}_j^\top \\ \tilde{S}_j & \tilde{R}_j & \hat{B}_j^\top \\ -\hat{A}_j & \hat{B}_j & 0 \end{bmatrix}^{-1} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} -\begin{bmatrix} \hat{M}_j^\top & 0 \\ \hat{L}_j^\top & 0 \end{bmatrix} \begin{bmatrix} 0 & I \\ \Theta_j & \Lambda_j \end{bmatrix}^{-1} \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ Y_5 \end{bmatrix}. \quad (\text{A4})$$

In view of (A2),

$$\begin{bmatrix} \bar{Q}_j & \tilde{S}_j^\top & -\hat{A}_j^\top \\ \tilde{S}_j & \tilde{R}_j & \hat{B}_j^\top \\ -\hat{A}_j & \hat{B}_j & 0 \end{bmatrix}^{-1} = \begin{bmatrix} I & 0 & 0 \\ -\tilde{R}_j^{-1}\tilde{S}_j & I & -\tilde{R}_j^{-1}\tilde{B}_j^\top\tilde{A}_j^{-\top} \\ 0 & 0 & \tilde{A}_j^{-\top} \end{bmatrix} \begin{bmatrix} \tilde{Q}_j & 0 & -I \\ 0 & \tilde{R}_j & 0 \\ -I & 0 & -\tilde{A}_j^{-1}\tilde{B}_j\tilde{R}_j^{-1}\tilde{B}_j^\top\tilde{A}_j^{-\top} \end{bmatrix}^{-1} \begin{bmatrix} I & -\tilde{S}_j^\top\tilde{R}_j^{-1} & 0 \\ 0 & I & 0 \\ 0 & -\tilde{A}_j^{-1}\tilde{B}_j\tilde{R}_j^{-1} & \tilde{A}_j^{-1} \end{bmatrix}.$$

As such, the result follows by noting that

$$\begin{aligned} & \begin{bmatrix} \tilde{Q}_j & 0 & -I \\ 0 & \tilde{R}_j & 0 \\ -I & 0 & -\tilde{A}_j^{-1}\tilde{B}_j\tilde{R}_j^{-1}\tilde{B}_j^\top\tilde{A}_j^{-\top} \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \tilde{A}_j^{-1}\tilde{B}_j\tilde{R}_j^{-1}\tilde{B}_j^\top\tilde{A}_j^{-\top}\tilde{W}_j & 0 & \tilde{A}_j^{-1}\tilde{B}_j\tilde{R}_j^{-1}\tilde{B}_j^\top\tilde{A}_j^{-\top}\tilde{W}_j\tilde{Q}_j - I \\ 0 & \tilde{R}_j^{-1} & 0 \\ -\tilde{W}_j & 0 & -\tilde{W}_j\tilde{Q}_j \end{bmatrix}, \quad (\text{A5}) \end{aligned}$$

where $\tilde{W}_j = (I + \tilde{Q}_j\tilde{A}_j^{-1}\tilde{B}_j\tilde{R}_j^{-1}\tilde{B}_j^\top\tilde{A}_j^{-\top})^{-1}$. Indeed, the matrix in (A5) is precisely Ω_j in (16). Application of the Sherman-Morrison-Woodbury matrix inversion lemma gives the equivalent expression, including the expression (18) for $\tilde{W}_j$, and (17) for $\tilde{Z}_j = \tilde{A}_j^{-1}\tilde{B}_j\tilde{R}_j^{-1}\tilde{B}_j^\top\tilde{A}_j^{-\top}\tilde{W}_j$. $\square$

References

- Bevilacqua, R., Codenotti, B., & Romani, F. (1988). Parallel solution of block tridiagonal linear systems. *Linear Algebra and its Applications*, 104, 39–57.
- Diehl, M., Ferreau, H. J., & Haverbeke, N. (2009). Efficient numerical methods for nonlinear MPC and moving horizon estimation. In *Nonlinear model predictive control* (pp. 391–417). Springer.
- Disney, S. M., & Towill, D. R. (2003). The effect of vendor managed inventory (VMI) dynamics on the bullwhip effect in supply chains. *International Journal of Production Economics*, 85(2), 199–215.
- Domahidi, A., Zraggen, A. U., Zeilinger, M. N., Morari, M., & Jones, C. N. (2012). Efficient interior point methods for multistage problems arising in receding horizon control. In *Proceedings 51st IEEE Conference on Decision and Control* (pp. 668–674).

- Labadie, J. W. (2004). Optimal operation of multireservoir systems: State-of-the-art review. *Journal of Water Resources Planning and Management*, 130(2), 93–111.
- Li, Y., Cantoni, M., & Weyer, E. (2005). On water-level error propagation in controlled irrigation channels. In *Proceedings of the 44th IEEE Conference on Decision and Control* (pp. 2101–2106).
- Maciejowski, J. (2002). *Predictive Control: With Constraints*. Prentice-Hall.
- Meurant, G. (1992). A review on the inverse of symmetric tridiagonal and block tridiagonal matrices. *SIAM Journal on Matrix Analysis and Applications*, 13(3), 707–728.
- Neshastehriz, A. R., Cantoni, M., & Shames, I. (2014). Water-level reference planning for automated irrigation channels via robust MPC. In *Proc. european control conference* (pp. 1331–1336).
- Nielsen, I., & Axehill, D. (2014). An $O(\log N)$ parallel algorithm for Newton step computation in model predictive control. *IFAC Proceedings Volumes*, 47(3), 10505–10511.
- Puig, V., Cembrano, G., Romera, J., Quevedo, J., Aznar, B., Ramón, G., & Cabot, J. (2009). Predictive optimal control of sewer networks using CORAL tool: Application to Riera Blanca catchment in Barcelona. *Water Science and Technology*, 60(4), 869–878.
- Rao, C. V., Wright, S. J., & Rawlings, J. B. (1998). Application of interior-point methods to model predictive control. *Journal of Optimization Theory and Applications*, 99(3), 723–757.
- Rawlings, J. B., & Mayne, D. Q. (2009). *Model Predictive Control: Theory and Design*. Nob Hill Pub.
- Şahin, A., & Morari, M. (2010). Decentralized model predictive control for a cascade of river power plants. In *Intelligent infrastructures* (pp. 463–485). Springer.
- Seiler, P., Pant, A., & Hedrick, K. (2004). Disturbance propagation in vehicle strings. *IEEE Transactions on Automatic Control*, 49(10), 1835–1841.
- Soltanian, L., & Cantoni, M. (2015). Decentralized string-stability analysis for heterogeneous cascades subject to load-matching requirements. *Multidimensional Systems and Signal Processing*, 26(4), 985–999.
- Wang, Y., & Boyd, S. (2010). Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, 18(2), 267–278.
- Wills, A., & Heath, W. (2003). Interior-point methods for linear model predictive control. *Report EE03016, University of Newcastle, NSW*, 2308.
- Wright, S. J. (1993). Interior point methods for optimal control of discrete time systems. *Journal of Optimization Theory and Applications*, 77(1), 161–187.
- Wright, S. J. (1997). *Primal-dual interior-point methods*. SIAM.