University Library



A gateway to Melbourne's research publications

Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:
Zafar, A;Farokhi, F;Cantoni, M

Title:
Linear quadratic control computation for systems with a directed tree structure

Date:
2020-01-01

Citation:
Zafar, A., Farokhi, F. & Cantoni, M. (2020). Linear quadratic control computation for systems with a directed tree structure. IFAC Papersonline, 53 (2), pp.6536-6541. https://doi.org/10.1016/j.ifacol.2020.12.069.

Persistent Link:
https://hdl.handle.net/11343/272712

Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 53-2 (2020) 6536–6541

Linear quadratic control computation for systems with a directed tree structure

Armaghan Zafar * **Farhad Farokhi** * **Michael Cantoni** *

* *Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville VIC 3010, Australia (e-mail: armaghanz@student.unimelb.edu.au, {cantoni,ffarokhi}@unimelb.edu.au).*

Abstract: A computational method is proposed for solving a structured quadratic optimal control problem subject to linear discrete-time dynamics arising from a directed tree structured interconnection of heterogeneous sub-systems. The problem is first formulated as a quadratic program with structure along three dimensions of the decision space. A nested algorithm based on block Jacobi iterations is proposed for the linear system of equations obtained from the corresponding first-order optimality conditions. It is shown that the per iteration computational burden scales favorably with increasing problem size in each dimension. The computations at each iteration are amenable to distributed implementation on a network of parallel processors mirroring the tree graph structure of the problem. Numerical experiments, based on a model data for an automated irrigation network, are used to demonstrate aspects of the approach, including the impact of early termination of the inner iterations in agreement with corresponding analysis.

Copyright © 2020 The Authors. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0)

Keywords: Linear Quadratic (LQ) control, tree graph, structured computations

NOTATIONS

Throughout this document, $\mathbb{R} := (-\infty, +\infty)$ is used to denote the set of real numbers and $\mathbb{N} := \{1, 2, \ldots\}$ is the set of natural numbers. $\mathbb{R}^n$ denotes an n-dimensional real vector and $\mathbb{R}^{p\times q}$ denotes a real matrix with p rows and q columns. $\mathcal{A}\backslash\{b\}$ denotes all elements of the set $\mathcal{A}$ except element b. I_q is the identity matrix of size $q \times q$ and $0_{p\times q}$ denotes a matrix of zeros of size $p \times q$. A' denotes the transpose of a matrix. $A \succ 0$ means the symmetric matrix $A = A' \in \mathbb{R}^{n\times n}$ is positive definite (i.e., there exists $c > 0$ such that $x'Ax \geq cx'x$ for all $x \in \mathbb{R}^n$) and $A \succeq 0$ means A is a positive semi-definite (i.e., $x'Ax \geq 0$ for all $x \in \mathbb{R}^n$). "diag(.)"" represents construction of block diagonal matrix from input arguments and "col(.)" represents concatenation of input arguments as column vector. $\otimes$ represents the Kronecker product.

1. INTRODUCTION

In this paper, computational aspects of solving structured large-scale finite-horizon linear-quadratic (LQ) optimal control problems are investigated. Specifically problems subject to linear discrete-time dynamics arising from the interconnection of heterogeneous sub-systems in the form of a directed tree graph. The key features are that (i) each node has in degree no greater than 2, (ii) the out degree is 1 for all nodes, except one node with no children, and (iii) the sub-graph comprised of the branch nodes is a directed path, where a branch node means a node of in degree 2 or more. This kind of structure is relevant in various domains, including water irrigation networks (Li et al., 2005), supply chains (Perea-López et al., 2003), and radial power distribution networks (Peng and Low, 2018).

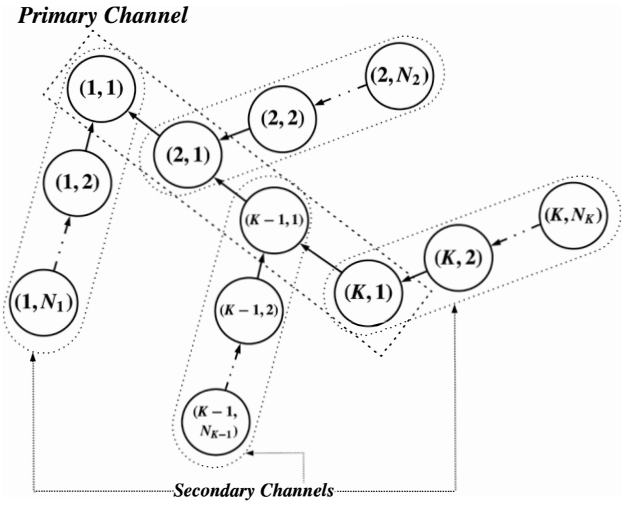

Fig. 1. Directed information flow in a tree graph network

To help clarify, consider a resource distribution network consisting of a primary channel and secondary distribution channels. Suppose that resource flow regulators are located at the source of and along channels immediately downstream of every supply point to a secondary channel or end-user as the case may be. Each of these regulators is controlled to achieve a specified level of the resource in the buffer at the supply points immediately downstream, based on observations of this level and the downstream flow load. Such a feedback control architecture results in automatic demand driven release of the resource from the source; e.g., see (Cantoni et al., 2007) for related work on irrigation networks. Indeed, the information flow corre-

2405-8963 Copyright © 2020 The Authors. This is an open access article under the CC BY-NC-ND license.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2020.12.069

sponds to the interaction of sub-systems on a directed tree graph with the features described previously; see Fig. 1. The type of LQ optimal control problem considered in this paper arises, for example, in determining suitable reference storage levels throughout the controlled network on the basis of a demand forecast, accounting for the underlying dynamics of the low-level feedback control system. The directed tree graph structured interconnection of dynamical sub-systems considered is shown in Fig. 1. Each node is a sub-system and is labeled with subscript (i,j) with $i \in \mathcal{K} := \{1,2,\ldots,K\} \subset \mathbb{N}$ and $j \in \mathcal{N}_i := \{1,2,\ldots,N_i\} \subset \mathbb{N}$, where K is the total number of sub-systems in the primary channel and N_i is the total number of sub-systems in the secondary channel originating from node i of the primary channel. It can be seen from Fig. 1 that each sub-system $(i,1)$ in the primary channel is only directly dependent on the sub-systems $(i+1,1)$ and $(i,2)$, while each sub-system (i,j) in the secondary channels only depends directly on the sub-system $(i,j+1)$. The state space dynamics of each sub-system are

$$\begin{aligned} x_{i,j}(t+1) = \; & A_{i,j}x_{i,j}(t) + B_{i,j}u_{i,j}(t) \\ & + E_{i,j}x_{i,j+1}(t) + F_{i,j}x_{i+1,j}(t), \end{aligned} \tag{1}$$

where $x_{i,j}(t) \in \mathbb{R}^{n_{i,j}}$ and $u_{i,j}(t) \in \mathbb{R}^{m_{i,j}}$ are the state and input of sub-system $(i,j) \in \mathcal{K} \times \mathcal{N}_i$ at time $t \in \mathcal{T} := \{0,1,\ldots,T\} \subset \mathbb{N} \cup \{0\}$, respectively. Initial conditions are given by $x_{i,j}(0) = \xi_{i,j} \in \mathbb{R}^{n_{i,j}}$. The model parameters are $A_{i,j} \in \mathbb{R}^{n_{i,j} \times n_{i,j}}$, $B_{i,j} \in \mathbb{R}^{n_{i,j} \times m_{i,j}}$, $E_{i,j} \in \mathbb{R}^{n_{i,j} \times n_{i,j+1}}$ and $F_{i,j} \in \mathbb{R}^{n_{i,j} \times n_{i+1,j}}$. Matrices $F_{i,j}$ and $E_{i,j}$ account for the coupling between the states of adjacent sub-systems in the primary and secondary channels respectively. $F_{i,j}$ are non zero for all $i \in \mathcal{K}\backslash\{K\}$, $j = 1$ and zero elsewhere while matrices $E_{i,j}$ are only zero for $j = N_i$ so that the boundary value of sub system $x_{i,N_{i+1}}(t)$ is effectively zero for all $t \in \mathcal{T}$.

The finite-horizon optimal control problem of interest is the following:

$$\begin{aligned} \min_{x,\,u} \; & \frac{1}{2} \sum_{i=1}^{K} \Bigg(\sum_{j=1}^{N_i} \Bigg(\Bigg(\sum_{t=0}^{T-1} \begin{bmatrix} x_{i,j}(t) \\ u_{i,j}(t) \end{bmatrix}' \begin{bmatrix} Q_{i,j} & 0 \\ 0 & R_{i,j} \end{bmatrix} \begin{bmatrix} x_{i,j}(t) \\ u_{i,j}(t) \end{bmatrix} \\ & + x_{i,j}(T) P_{i,j} x_{i,j}(T) \Bigg) \Bigg) \end{aligned} \tag{2a}$$

subject to

$$x_{i,j}(0) = \; \xi_{i,j} \text{ for } i \in \mathcal{K},\; j \in \mathcal{N}_i, \tag{2b}$$

$$\begin{aligned} x_{i,j}(t+1) = \; & A_{i,j}x_{i,j}(t) + B_{i,j}u_{i,j}(t) \\ & + E_{i,j}x_{i,j+1}(t) + F_{i,j}x_{i+1,j}(t) \\ & \text{for } i \in \mathcal{K},\; j \in \mathcal{N}_i,\; t \in \mathcal{T}\backslash\{T\}, \end{aligned} \tag{2c}$$

where $R_{i,j} \in \mathbb{R}^{m_{i,j} \times m_{i,j}}$, $Q_{i,j} \in \mathbb{R}^{n_{i,j} \times n_{i,j}}$ and $P_{i,j} \in \mathbb{R}^{n_{i,j} \times n_{i,j}}$ are symmetric positive definite matrices. The decision variables u and x consist of components $u_{i,j}(t)$ for $i \in \mathcal{K}$, $j \in \mathcal{N}_i$, $t \in \mathcal{T}\backslash\{T\}$ and $x_{i,j}(t)$ for $i \in \mathcal{K}$, $j \in \mathcal{N}_i$, $t \in \mathcal{T}$, respectively. It can be seen that the cost function (2a) is separable across the time horizon and along the two spatial dimensions indexed by i and j,, respectively. However, there is coupling between states of adjacent sub-systems in the constraint (2c), which encodes the temporal and spatial dynamics of the directed tree structured interconnection of sub-systems..

The finite horizon LQ optimal control just described is a structured quadratic program (QP). The solution of this QP can be computed by solving the linear system of equations arising from the first-order optimality conditions also known as Karush-Kuhn-Tucker (KKT) conditions (Nocedal and Wright, 2000). Due to the special structure of equality constraint which encodes the dynamics, this linear system of equations is also structured.

The main contribution of this paper relates to exploiting structure in the aforementioned KKT conditions, extending prior work of the authors pertaining to the special case of a path graph network (i.e., no secondary channels (Cantoni et al., 2017; Zafar et al., 2019)). Specifically, an algorithm based on nested block Jacobi iterations is devised for solving the corresponding linear system of equations, and thus, the optimal control problem. These iterations retain structure in all three dimensions of the problem and yield a favorable scaling of the computational burden with the number of sub-systems along the primary channel and secondary channels, and the length of the time horizon. In particular, for a fixed number of inner iterations, the per out-iteration computational cost scales linearly in all three dimensions. The computations of all iterations are amenable to distributed implementation on a network of parallel processors mirroring the tree graph structure of the problem.

The rest of the paper is organized as follows. In Section 2, the aforementioned structured QP is formulated and the corresponding KKT conditions are identified. In Section 3, the nested algorithm based on block Jacobi iteration is presented. The convergence of these nested Jacobi iterations including error analysis when the inner iterations are terminated early is considered in Section 4. The proposed method is applied to determine the set-point references for an automated irrigation network model. Computation time results for a single processor implementation are presented in Section 5. Finally, some concluding remarks are given in Section 6.

2. PROBLEM FORMULATION

For the sake of simplifying the arguments and notation, the following assumptions are made:

Assumption 1. The number of pools in each secondary channel is equal to $N_1 = N_2 = \ldots = N_K = N$ and $\mathcal{N}$ represents the common set of secondary channel indexes.

Remark 2. Assumption 1 only aids in the simplification of notations and clarity of presentation. All subsequent developments remain valid if Assumption 1 does not hold. In such case, the size of the block matrices presented in equation (4) will be different for each index j and the limit of summation goes to $N = \max_i(N_i)$.

Following (Cantoni et al., 2017) and (Zafar et al., 2019), where cascade (i.e., directed path) interconnections are studied within an LQ optimal control context, by defining, $\bar{u}_{i,j} := \text{col}(u_{i,j}(0),\ldots,u_{i,j}(T-1)) \in \mathbb{R}^{m_{i,j}T}$ and $\bar{x}_{i,j} := \text{col}(x_{i,j}(0),\ldots,x_{i,j}(T)) \in \mathbb{R}^{n_{i,j}(T+1)}$, the problem (2) can be reformulated as the following temporally stacked QP:

$$\min_{\substack{\bar{x}_{i,j},\bar{u}_{i,j}, \\ i\in\mathcal{K},\; j\in\mathcal{N}}} \; \frac{1}{2} \sum_{i=1}^{K} \Bigg(\sum_{j=1}^{N} \Bigg(\begin{bmatrix} \bar{x}_{i,j} \\ \bar{u}_{i,j} \end{bmatrix}' \begin{bmatrix} \bar{Q}_{i,j} & 0 \\ 0 & \bar{R}_{i,j} \end{bmatrix} \begin{bmatrix} \bar{x}_{i,j} \\ \bar{u}_{i,j} \end{bmatrix} \Bigg) \Bigg) \tag{3a}$$

subject to

$$0 = \bar{A}_{i,j}\bar{x}_{i,j} + \bar{B}_{i,j}\bar{u}_{i,j} + \bar{E}_{i,j}\bar{x}_{i,j+1} + \bar{F}_{i,j}\bar{x}_{i+1,j} + \bar{H}_{i,j}\xi_{i,j},\ i \in \mathcal{K},\ j \in \mathcal{N}, \tag{3b}$$

where

$$\bar{Q}_{i,j} = \begin{bmatrix} \mathrm{I}_T \otimes Q_{i,j} & 0_{n_{i,j}T \times n_{i,j}} \\ 0_{n_{i,j} \times n_{i,j}T} & P_{i,j} \end{bmatrix} \in \mathbb{R}^{n_{i,j}(T+1) \times n_{i,j}(T+1)},$$
$$\bar{R}_{i,j} = \mathrm{I}_T \otimes R_{i,j} \in \mathbb{R}^{m_{i,j}(T) \times m_{i,j}(T)},$$
$$\bar{H}_{i,j} = \mathrm{col}(\mathrm{I}_{n_{i,j}}\ 0_{n_{i,j} \times n_{i,j}} \cdots 0_{n_{i,j} \times n_{i,j}}) \in \mathbb{R}^{n_{i,j}(T+1) \times n_{i,j}},$$
$$\bar{A}_{i,j} = -\mathrm{I}_{n_{i,j}(T+1)} + \begin{bmatrix} 0_{n_{i,j}T \times n_{i,j}T} & 0_{n_{i,j}T \times n_{i,j}} \\ \mathrm{I}_T \otimes A_{i,j} & 0_{n_{i,j}T \times n_{i,j}} \end{bmatrix},$$
$$\bar{B}_{i,j} = \begin{bmatrix} 0_{n_{i,j} \times m_{i,j}(T)} \\ \mathrm{I}_T \otimes B_{i,j} \end{bmatrix}, \bar{E}_{i,j} = \begin{bmatrix} 0_{n_{i,j}T \times n_{i,j}T} & 0_{n_{i,j}T \times n_{i,j}} \\ \mathrm{I}_T \otimes E_{i,j} & 0_{n_{i,j}T \times n_{i,j}} \end{bmatrix},$$
$$\bar{F}_{i,j} = \begin{bmatrix} 0_{n_{i,j}T \times n_{i,j}T} & 0_{n_{i,j}T \times n_{i,j}} \\ \mathrm{I}_T \otimes F_{i,j} & 0_{n_{i,j}T \times n_{i,j}} \end{bmatrix},$$

for $i \in \mathcal{K}$, $j \in \mathcal{N}$. $\bar{A}_{i,j} \in \mathbb{R}^{n_{i,j}(T+1) \times n_{i,j}(T+1)}$ and $\bar{B}_{i,j} \in \mathbb{R}^{n_{i,j}(T+1) \times m_{i,j}(T)}$. $\bar{F}_{i,j} \in \mathbb{R}^{n_{i,j}(T+1) \times n_{i+1,j}(T+1)}$ and $\bar{F}_{i,j} = 0$ for all $\{(i,j) |\ i = K \text{ or } j \neq 1\}$. $\bar{E}_{i,j} \in \mathbb{R}^{n_{i,j}(T+1) \times n_{i,j+1}(T+1)}$ and $\bar{E}_{i,N_i} = 0$ (so that effectively $\bar{x}_{i,N_{i+1}} = 0$). Notice that all matrices in the temporally stacked QP (3) are block diagonal except matrices $\bar{A}_{i,j}$ which are block bi-diagonal.

Now defining $\hat{n}_j := \sum_{i=1}^{K} n_{i,j}(T+1)$, $\hat{m}_j := \sum_{i=1}^{K} m_{i,j}T$, $\hat{\nu}_j := \sum_{i=1}^{K} n_{i,j}$, $\hat{u}_j := \mathrm{col}(\bar{u}_{1,j}, \ldots, \bar{u}_{K,j}) \in \mathbb{R}^{\hat{m}_j}$ and $\hat{x}_j := \mathrm{col}(\bar{x}_{1,j}, \ldots, \bar{x}_{K,j}) \in \mathbb{R}^{\hat{n}_j}$, problem (3) can be reformulated by stacking across the dimension of the primary channel which results in the following sparse and structured quadratic program (QP).

$$\min_{\substack{\hat{x}_j, \hat{u}_j, \\ j \in \mathcal{N}}} \frac{1}{2} \sum_{j=1}^{N} \left(\begin{bmatrix} \hat{x}_j \\ \hat{u}_j \end{bmatrix}' \begin{bmatrix} \hat{Q}_j & 0 \\ 0 & \hat{R}_j \end{bmatrix} \begin{bmatrix} \hat{x}_j \\ \hat{u}_j \end{bmatrix} \right) \tag{4a}$$

subject to

$$0 = \hat{A}_j\hat{x}_j + \hat{B}_j\hat{u}_j + \hat{E}_j\hat{x}_{j+1} + \hat{H}_j\hat{\xi}_j,\ j \in \mathcal{N}, \tag{4b}$$

where

$$\hat{A}_j = \mathrm{diag}(\bar{A}_{1,j}, .., \bar{A}_{K,j}),\ j \in \mathcal{N} \backslash \{1\},$$
$$\hat{A}_1 = \begin{bmatrix} \bar{A}_{1,1} & \bar{F}_{1,1} & & \\ & \bar{A}_{2,1} & \ddots & \\ & & \ddots & \bar{F}_{K-1,1} \\ & & & \bar{A}_{K,1} \end{bmatrix} \in \mathbb{R}^{\hat{n}_j \times \hat{n}_j},$$

$\hat{Q}_j = \mathrm{diag}(\bar{Q}_{1,j}, .., \bar{Q}_{K,j}) \in \mathbb{R}^{\hat{n}_j \times \hat{n}_j}$, $\hat{R}_j = \mathrm{diag}(\bar{R}_{1,j}, .., \bar{R}_{K,j}) \in \mathbb{R}^{\hat{m}_j \times \hat{m}_j}$, $\hat{B}_j = \mathrm{diag}(\bar{B}_{1,j}, .., \bar{B}_{K,j}) \in \mathbb{R}^{\hat{n}_j \times \hat{m}_j}$, $\hat{E}_j = \mathrm{diag}(\bar{E}_{1,j}, .., \bar{E}_{K,j}) \in \mathbb{R}^{\hat{n}_j \times \hat{n}_j}$ and $\hat{H}_j = \mathrm{diag}(\bar{H}_{1,j}, .., \bar{H}_{K,j}) \in \mathbb{R}^{\hat{n}_j \times \hat{\nu}_j}$ are all block diagonal matrices with $\hat{E}_N = 0$.

2.1 First Order Optimality Conditions

The KKT conditions for problem (4) are given by

$$\hat{Q}_j\hat{x}_j + \hat{A}_j'\hat{p}_j + \hat{E}_{j-1}'\hat{p}_{j-1} = 0,$$
$$\hat{R}_j'\hat{u}_j + \hat{B}_j'\hat{p}_j = 0,$$
$$\hat{A}_j\hat{x}_j + \hat{B}_j\hat{u}_j + \hat{E}_j\hat{x}_{j+1} + \hat{H}_j\xi_j = 0,$$

for $j \in \mathcal{N}$. The variables $\hat{p}_j = \mathrm{col}(\bar{p}_{1,j}, \ldots, \bar{p}_{K,j}) \in \mathbb{R}^{\hat{n}_j}$ are Lagrange multipliers. Since Q and R are positive definite, the problem is strictly convex and the KKT conditions are necessary and sufficient for optimality (Nocedal and Wright, 2000).

Defining $\tilde{n} = \sum_{j=1}^{N}(\hat{n}_j)$, $\tilde{m} = \sum_{j=1}^{N}(\hat{m}_j)$, $\tilde{\nu} = \sum_{j=1}^{N}(\hat{\nu}_j)$, $\tilde{x} = \mathrm{col}(\hat{x}_1, \ldots, \hat{x}_N)$, $\tilde{u} = \mathrm{col}(\hat{u}_1, \ldots, \hat{u}_N)$, $\tilde{p} = \mathrm{col}(\hat{p}_1, \ldots, \hat{p}_N)$, the KKT conditions can be written as

$$\begin{bmatrix} \tilde{Q} & 0 & \tilde{A}' \\ 0 & \tilde{R} & \tilde{B}' \\ \tilde{A} & \tilde{B} & 0 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{u} \\ \tilde{p} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -r_{\tilde{p}} \end{bmatrix}, \tag{5}$$

where

$$\tilde{A} = \begin{bmatrix} \hat{A}_1 & \hat{E}_1 & & \\ & \hat{A}_2 & \ddots & \\ & & \ddots & \hat{E}_{N-1} \\ & & & \hat{A}_N \end{bmatrix} \in \mathbb{R}^{\tilde{n} \times \tilde{n}}.$$

and matrices $\tilde{Q} = \mathrm{diag}(\hat{Q}_1, \ldots, \hat{Q}_N) \in \mathbb{R}^{\tilde{n} \times \tilde{n}}$, $\tilde{R} = \mathrm{diag}(\hat{R}_1, \ldots, \hat{R}_N) \in \mathbb{R}^{\tilde{m} \times \tilde{m}}$, $\tilde{B} = \mathrm{diag}(\hat{B}_1, \ldots, \hat{B}_N) \in \mathbb{R}^{\tilde{n} \times \tilde{m}}$, $\tilde{H} = \mathrm{diag}(\hat{H}_1, \ldots, \hat{H}_N) \in \mathbb{R}^{\tilde{n} \times \tilde{\nu}}$, $r_{\tilde{p}} = -\tilde{H}\tilde{\xi} \in \mathbb{R}^{\tilde{n}}$ and $\tilde{\xi} = \mathrm{col}(\hat{\xi}_1, \ldots, \hat{\xi}_N) \in \mathbb{R}^{\tilde{\nu}}$. It is apparent that the linear system of equations (5) is very sparse and structured. In the next section, this structure is exploited to devise an iterative algorithm based on block Jacobi iterations to solve the system of equations (5).

3. BLOCK JACOBI ITERATIONS

Defining $\tilde{v} := \mathrm{col}(\tilde{x},\ \tilde{u})$, (5) can be written as

$$\begin{bmatrix} Z & Y' \\ Y & 0 \end{bmatrix} \begin{bmatrix} \tilde{v} \\ \tilde{p} \end{bmatrix} = \begin{bmatrix} 0 \\ -r_{\tilde{p}} \end{bmatrix}, \tag{6}$$

where $Z = \begin{bmatrix} \tilde{Q} & 0 \\ 0 & \tilde{R} \end{bmatrix}$ and $Y = \begin{bmatrix} \tilde{A} & \tilde{B} \end{bmatrix}$.

Note that matrix Z is positive definite and hence, that the inverse is positive definite. Equation (6) can be solved as

$$(YZ^{-1}Y')\tilde{p} = r_{\tilde{p}}, \tag{7}$$
$$\tilde{v} = Z^{-1}r_{\tilde{v}},$$

where $r_{\tilde{v}} = -Y'\tilde{p}$. Since $Z \succ 0$ and Y is full row rank, $YZ^{-1}Y'$ is positive definite. The matrix Z is block diagonal with block size independent of N, K and T. Therefore computation of Z^{-1} can be carried out efficiently. However, $YZ^{-1}Y'$ is a block tri-diagonal matrix and solving (7) is computationally more expensive.

The linear system (7) can be solved by block Jacobi iterations. First, note that $YZ^{-1}Y'$ has the block tri-diagonal structure

$$\Delta = YZ^{-1}Y' = \begin{bmatrix} \hat{\Phi}_1 & \hat{\Omega}_1 & & & \\ \hat{\Omega}_1' & \hat{\Phi}_2 & \hat{\Omega}_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \hat{\Omega}_{N-2}' & \hat{\Phi}_{N-1} & \hat{\Omega}_{N-1} \\ & & & \hat{\Omega}_{N-1}' & \hat{\Phi}_N \end{bmatrix}, \tag{8}$$

where

$$\hat{\Phi}_j = \hat{A}_j\hat{Q}_j^{-1}\hat{A}_j' + \hat{B}_j\hat{R}_j^{-1}\hat{B}_j' + \hat{E}_j\hat{Q}_{j+1}^{-1}\hat{E}_j' \in \mathbb{R}^{\hat{n}_j \times \hat{n}_j},$$
$$\hat{\Omega}_j = \hat{E}_j\hat{Q}_{j+1}^{-1}\hat{A}_{j+1}' \in \mathbb{R}^{\hat{n}_j \times \hat{n}_j},$$

for $j \in \mathcal{N}$ and $\hat{Q}_{N+1} = 0$. Let

$$\Psi = \mathrm{diag}(\hat{\Phi}_1, \ldots, \hat{\Phi}_N) \in \mathbb{R}^{\tilde{n}_j \times \tilde{n}_j} \tag{9}$$

and $\Xi := \Psi - \Delta$. The Jacobi iterations for solving (7) are given by the following:

$$\Psi \tilde{p}^{k+1} = r_{\tilde{p}} + \Xi \tilde{p}^k, \tag{10}$$

where k is the iteration index (Hackbusch, 2016). Since $\Delta \succ 0$, the block diagonal matrix $\Psi \succ 0$ (Golub and Van Loan, 2013, Cor 4.2.2). The convergence of these iterations to the solution of (7) is considered in Section 4.

3.1 Structure of Matrices

At each Jacobi iteration, the linear system of equations in (10) needs to be solved. The computational complexity of each Jacobi iteration depends on the structure of the matrix Ψ. It can be seen from (9) that $\Psi \succ 0$ is block diagonal. Each diagonal block $\hat{\Phi}_j$ is also very sparse and structured. For $j \in \mathcal{N}\backslash\{1\}$, each $\hat{A}_j$ is block diagonal, whereby both $\hat{\Phi}_j$ and $\hat{\Omega}_j$ are block diagonal. However, $\hat{\Phi}_1$ is block tri-diagonal because $\hat{A}_1$ is block bi-diagonal. In summary,

$$\hat{\Phi}_j = \begin{bmatrix} \bar{\Phi}_{1,j} & & \\ & \ddots & \\ & & \bar{\Phi}_{K,j} \end{bmatrix} \in \mathbb{R}^{\hat{n}_j \times \hat{n}_j} \text{ for } j \in \mathcal{N}\backslash\{1\}, \tag{11a}$$

$$\hat{\Phi}_1 = \begin{bmatrix} \bar{\Phi}_{1,1} & \phi_{1,1} & & & \\ \phi'_{1,1} & \bar{\Phi}_{2,1} & \phi_{2,1} & & \\ & \ddots & \ddots & \ddots & \\ & & \phi'_{K-2,1} & \bar{\Phi}_{K-1,1} & \phi_{K-1,1} \\ & & & \phi'_{K-1,1} & \bar{\Phi}_{K,1} \end{bmatrix} \in \mathbb{R}^{\hat{n}_j \times \hat{n}_j}, \tag{11b}$$

where

$$\bar{\Phi}_{i,j} = \bar{A}_{i,j}\bar{Q}^{-1}_{i,j}\bar{A}'_{i,j} + \bar{B}_{i,j}\bar{R}^{-1}_{i,j}\bar{B}'_{i,j} + \bar{E}_{i,j}\bar{Q}^{-1}_{i,j+1}\bar{E}'_{i,j} \in \mathbb{R}^{n_{i,j}(T+1)\times n_{i,j}(T+1)} \ \forall\ i \in \mathcal{K},\ j \in \mathcal{N}\backslash\{1\},$$

$$\bar{\Phi}_{i,1} = \bar{A}_{i,1}\bar{Q}^{-1}_{i,1}\bar{A}'_{i,1} + \bar{B}_{i,1}\bar{R}^{-1}_{i,1}\bar{B}'_{i,1} + \bar{E}_{i,1}\bar{Q}^{-1}_{i,2}\bar{E}'_{i,1} + \bar{F}_{i,1}\bar{Q}^{-1}_{i+1,1}\bar{F}'_{i,1} \in \mathbb{R}^{n_{i,1}(T+1)\times n_{i,1}(T+1)} \ \forall\ i \in \mathcal{K},$$

$$\phi_{i,1} = \bar{F}_{i,1}\bar{Q}^{-1}_{i+1,1}\bar{A}'_{i+1,1} \in \mathbb{R}^{n_{i,1}(T+1)\times n_{i,1}(T+1)} \ i \in \mathcal{K}\backslash\{K\},$$

and $\bar{Q}_{i,N+1} = \bar{Q}_{K+1,1} = 0$.

Remark 3. If the variable stacking were done across the dimension of the secondary channels first, instead of the primary channel as in (4), then the structure of every $\hat{\Phi}_j$ would be the same as $\hat{\Phi}_1$.

The size of each sub-block of $\hat{\Phi}_j$ scales linearly with T. Moreover, $\bar{\Phi}_{i,j}$ is block tri-diagonal with block-size that is independent of K, N and T. As such, in solving (10), for $i \in \mathcal{K}$, $j \in \mathcal{N}\backslash\{1\}$, the computation of each block component $\bar{p}^{k+1}_{i,j}$ of $\tilde{p}^{k+1}$ decomposes into a small block tri-diagonal equation $\bar{\Phi}_{i,j}\bar{p}^k_{i,j} = \bar{r}^k_{i,j}$ to solve. This can be achieved by backward and forward recursions (Meurant, 1992), with aggregate complexity that scales linearly in T, the number of blocks, and cubically with respect to $n_{i,j}$, the size of each block. On the other hand, for $j = 1$, a block tri-diagonal system of equations with block-size that depends linearly on T needs to be solved. With a view to retaining linear in T complexity, it is proposed to (approximately) solve this system of equations by inner Jacobi iterations.

Dropping the explicit dependence on the outer iteration index k, the first block equation of (10) is given by

Algorithm 1 Outer Block Jacobi Iterations

1: Compute $\hat{Q}_j^{-1},\ \hat{R}_j^{-1},\ \hat{\Phi}_j,\ \hat{\Omega}_j$ for $j \in \mathcal{N}$
2: Initialize $\hat{p}_j^0 = 0$ for $j \in \mathcal{N}$
3: $k \leftarrow 0$; $\zeta \leftarrow \epsilon + 1$
4: **while** $\zeta \geq \epsilon$ **do**
5: $\quad \hat{r}_1^k \leftarrow -\hat{H}_1\xi_1 - \hat{\Omega}_1\hat{p}_2^k$
6: $\quad$ **for** $j = 2, \ldots, N-1$ **do**
7: $\quad\quad \hat{r}_j^k \leftarrow -\hat{H}_j\xi_j - \hat{\Omega}_j\hat{p}_{j+1}^k - \hat{\Omega}'_{j-1}\hat{p}_{j-1}^k$
8: $\quad$ **end for**
9: $\quad \hat{r}_N^k \leftarrow -\hat{H}_N\xi_N - \hat{\Omega}'_{N-1}\hat{p}_{N-1}^k$
10: $\quad$ **Using Algorithm 2 solve** $\hat{\Phi}_1\hat{p}_1^{k+1} = \hat{r}_1^k$
11: $\quad$ **for** $j = 2, \ldots, N$ **do**
12: $\quad\quad$ **for** $i = 1, 2, \ldots, K$ **do**
13: $\quad\quad\quad$ **Noting that** $\hat{r}_j^k = \mathrm{col}(\bar{r}^k_{1,j}, \ldots, \bar{r}^k_{K,j})$ **use Backward-Forward recursions to solve** $\bar{\Phi}_{i,j}\bar{p}^{k+1}_{i,j} = \bar{r}^k_{i,j}$
14: $\quad\quad$ **end for**
15: $\quad$ **end for**
16: $\quad k \leftarrow k+1$; $\zeta \leftarrow \max_j \left\|\hat{p}_j^k - \hat{p}_j^{k-1}\right\|_\infty$
17: **end while**

Algorithm 2 Inner Block Jacobi Iterations

1: Initialize $\bar{p}^0_{i,1} = 0$ for $i \in \mathcal{K}$
2: $s \leftarrow 0$; $\hat{\zeta} \leftarrow \hat{\epsilon} + 1$
3: **while** $\hat{\zeta} \geq \hat{\epsilon}$ and $s < S_{max}$ **do**
4: $\quad r^s_{1,1} \leftarrow \bar{r}^k_{1,1} - \phi_{1,1}\bar{p}^s_{2,1}$
5: $\quad$ **for** $i = 2, \ldots, K-1$ **do**
6: $\quad\quad r^s_{i,1} \leftarrow \bar{r}^k_{i,1} - \phi_{i,1}\bar{p}^s_{i+1,1} - \phi'_{i-1,1}\bar{p}^s_{i-1,1}$
7: $\quad$ **end for**
8: $\quad r^s_{i,1} \leftarrow \bar{r}^k_{K,1} - \phi'_{K-1,1}\bar{p}^s_{K-1,1}$
9: $\quad$ **for** $i = 1, 2, \ldots, K$ **do**
10: $\quad\quad$ **Use Backward-Forward recursions to solve** $\bar{\Phi}_{i,1}\bar{p}^{s+1}_{i,1} = r^s_{i,1}$
11: $\quad$ **end for**
12: $\quad s \leftarrow s+1$; $\hat{\zeta} \leftarrow \max_i \left\|\bar{p}^s_{i,1} - \bar{p}^{s-1}_{i,1}\right\|_\infty$
13: **end while**

$$\hat{\Phi}_1\hat{p}_1 = \hat{r}_1, \tag{12}$$

where $\hat{\Phi}_1$ is the block tri-diagonal matrix (11b). Let

$$\hat{\Upsilon}_1 = \mathrm{diag}(\bar{\Phi}_{1,1}, \ldots, \bar{\Phi}_{K,1}) \in \mathbb{R}^{\hat{n}_j \times \hat{n}_j} \tag{13}$$

and $\hat{\Lambda}_1 := \hat{\Upsilon}_1 - \hat{\Phi}_1$. Then, at each iteration k of outer block Jacobi iterations, the inner block Jacobi iterations for solving (12) are given by the following:

$$\hat{\Upsilon}_1\hat{p}_1^{s+1} = \hat{r}_1 + \hat{\Lambda}_1\hat{p}_1^s. \tag{14}$$

This results in a nested algorithm. All steps of the outer and inner Jacobi iterations are summarized in Algorithm 1 and Algorithm 2, respectively.

Remark 4. The nested block Jacobi iterations can be parallelized on a network of $N \times K$ processing agents mirroring the structure of the given tree graph, but with undirected information flow as Ψ in (10) is block tri-diagonal. For example, the computations in Algorithm 1 can be distributed on a linear network of N parallel processing agents with each agent having K parallel computing threads. The information exchange required between outer iterations of the while-loop 4-17 is very localized. In lines 5 through 9, communication between adjacent processors is required. This amounts to $2(N-1)$ exchanges of information per

iteration. The update in line 16 can be determined via a forward-backward pass along the cascade to update the max of $\left\|\hat{p}_j^k - \hat{p}_j^{k-1}\right\|_\infty$, which incurs a further $2(N-1)$ exchanges. Therefore, the overall per-iteration communication overhead scales linearly with N for a distributed implementation of outer iterations.

Remark 5. For $j \in \mathcal{N}\backslash\{1\}$, K block tri-diagonal systems of equations are each solved only once per outer iteration k. However, for $j = 1$ these K block tri-diagonal systems of equations are solved up to $S_{\max}$ times. Therefore, the total computational time required per outer iteration is upper bounded by that of agent 1. With a fixed bound on the number of inner iterations, the per iteration computational complexity of every outer block Jacobi iteration scales linearly with N, K and T.

4. NESTED JACOBI ITERATIONS

Let $\Gamma := \Psi^{-1}\Xi$ denote the iteration matrix of the outer Jacobi iterations (10). Let $\tilde{p}^*$ denote the exact solution of (7). This is a fixed-point of the iterations, for which $\tilde{p}^k \to \tilde{p}^*$ as $k \to \infty$ from any initial value if and only if

$$\text{rad}(\Gamma) < 1, \tag{15}$$

where $\text{rad}(\cdot)$ denotes spectral radius. See (Hackbusch, 2016, Thm 2.16), for example. Since the positive definite matrix $\Delta = YZ^{-1}Y' = \Psi - \Xi$ is block tri-diagonal, it follows that (15) holds, as noted in (Zafar et al., 2019, Theorem 6). Similarly, the inner iterates $\hat{p}_1^s$ converge, as $s \to \infty$, to the solution $\hat{p}_1^*$ of (12) if and only if $\text{rad}(\hat{\Upsilon}_1^{-1}\hat{\Lambda}_1) < 1$. Again, this holds because $\hat{\Phi}_1 = \hat{\Upsilon}_1 - \hat{\Lambda}_1 \succ 0$ is also block tri-diagonal.

When the number of inner Jacobi iterations is limited to $S_{\max} > 0$, the update at outer iteration k corresponds to

$$\tilde{p}^{k+1} = \Psi^{-1}(r_{\tilde{p}} + \Xi\tilde{p}^k) + w^k \tag{16}$$

where

$$w^k = \text{diag}(\Theta_1, 0, \ldots, 0)(r_{\tilde{p}} + \Xi\tilde{p}^k)$$

and $\Theta_1 = (\sum_{l=0}^{S_{\max}}(\hat{\Upsilon}_1^{-1}\hat{\Lambda}_1)^l\hat{\Upsilon}_1^{-1}) - \hat{\Phi}_1^{-1}$. In particular,

$$\tilde{p}^{k+1} = (\Psi^{-1} + \text{diag}(\Theta_1, 0, \ldots, 0))(r_{\tilde{p}} + \Xi\tilde{p}^k).$$

This converges to a fixed point if and only if $\text{rad}(\Gamma + \text{diag}(\Theta_1, 0, \ldots, 0)\Xi) < 1$, which holds provided Θ_1 is sufficiently small because eigenvalues vary continuously (Horn and Johnson, 2013, Thm D1) and $\text{rad}(\Gamma) < 1$. It is possible to make Θ_1 arbitrarily small since $\text{rad}(\hat{\Upsilon}_1^{-1}\hat{\Lambda}_1) < 1$, whereby $\sum_{l=0}^{S_{\max}}(\hat{\Upsilon}_1^{-1}\hat{\Lambda}_1)^l\hat{\Upsilon}_1^{-1} \to \hat{\Phi}_1^{-1}$ as $S_{\max} \to \infty$. The proximity of the perturbed fixed-point to the solution $\tilde{p}^*$ of (10) can also be made arbitrarily close by increasing $S_{\max}$.

In (16), if $\|w^k\|_\infty \le \epsilon_w$ for all outer iterations k, then the ℓ_∞-to-ℓ_∞ gain from the disturbance w^k to the error defined relative to the unperturbed trajectory of the iterates $\tilde{p}^k$ is relevant to nested algorithm performance. This gain is finite since the corresponding dynamics is bounded-input bounded-output (BIBO) stable because $\text{rad}(\Psi^{-1}\Xi) < 1$ (Bacciotti, 2019). As such, smaller ϵ_w implies a smaller perturbation from the exact solution.

In the next section, numerical experiments are performed to illustrate the effect of early termination of inner iterations on the convergence of outer iterations to a preset tolerance.

5. NUMERICAL RESULTS

In this section, numerical experiments are performed by applying the Algorithm 1 to solve an optimal control problem involving model data for an automated irrigation network under so-called distant-downstream control (Cantoni et al., 2007). The irrigation network consists of K number of controlled pools in the primary channel and N number of controller pools in each secondary channel originating from the pools of primary channel. After temporal discretization, the dynamics of each controlled pool can be described by a sub-system in the form of (1). For each sub-system, the control input $u_{i,j}(t)$ is the adjustment of the water-level reference, while the state variable $x_{i,j}(t)$ is the deviation of state trajectory from equilibrium. For each sub-system $i \in \mathcal{K},\ j \in \mathcal{N}$, which takes the form (1) with $n_{i,j} = 4$, $m_{i,j} = 1$, the corresponding cost is set such that $Q_{i,j} = I$ and $R_{i,j} = 1$.

The implementation involves a single thread on one processor. That is, the possibility of exploiting parallelism in Algorithm 1 and 2 is not considered. The experiments are performed by setting $K = N$ and $T = K + N$ and varying the value of N from 10 to 100, whereby the largest problem considered has 10.04 million primal variables and 8.04 million dual variables.

Fig. 2 shows the processor time for single thread implementations of Algorithm 1 with stopping condition set to $\epsilon = 10^{-9}$ and Algorithm 2 with stopping condition set to $\epsilon_{in} = 10^{-9}$ and varying S_{max}. It shows that the increase in computational time has a quartic slope. Due to different number of inner iterations, the total computational time is different with $S_{max} = 1$ being fastest for each of the above implementations.

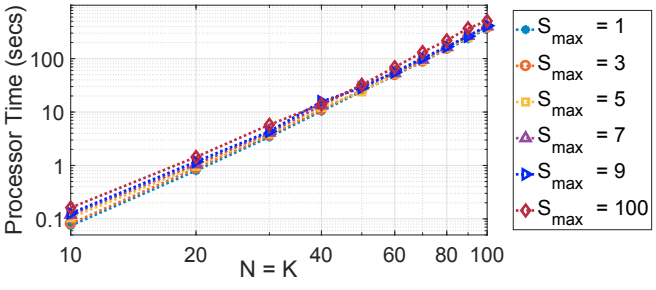

Fig. 2. Comparison of processor time for solving KKT Equations

Fig. 3 shows the total number of iterations performed by Algorithm 1 until convergence. The total number of iterations grow modestly with system size due to increase in $\text{rad}(\Gamma)$. Also, the total number of outer iterations are same for different S_{max}. This is by virtue of the stopping criteria $\left\|\hat{p}_j^{k+1} - \hat{p}_j^k\right\|_\infty < \epsilon,\ \forall\ j \in \mathcal{N}$. It shows that each component $\hat{p}_j^k$ of $\tilde{p}$ converges to a stationary point $\hat{p}_j^+$. Accuracy of the solution is shown in Fig. 4.

Fig. 4 shows the error $\|\tilde{A}\tilde{x} + \tilde{B}\tilde{u} + \tilde{H}\tilde{\xi}\|_\infty$ for the outcome $\tilde{x}$ and $\tilde{u}$ of nested Jacobi iterations. For $S_{max} = 1,\ 3$, this error is large but its modest for $S_{max} = 5,\ 7$. For $S_{max} = 9$, the error is of order 10^{-5}. However, for fixed number of inner iterations, the error increases with increase in system

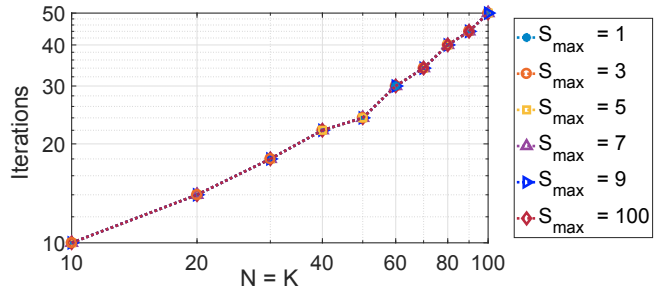

Fig. 3. Total number of iterations of Algorithm 1 for different inner iterations

size. This is due to the modest increase in the spectral radius of $(\hat{\Upsilon}_1^{-1}\hat{\Lambda}_1)$. For $S_{max} = 100$, Jacobi iterations converge to set tolerance ϵ_{in} and the error is of order 10^{-12}.

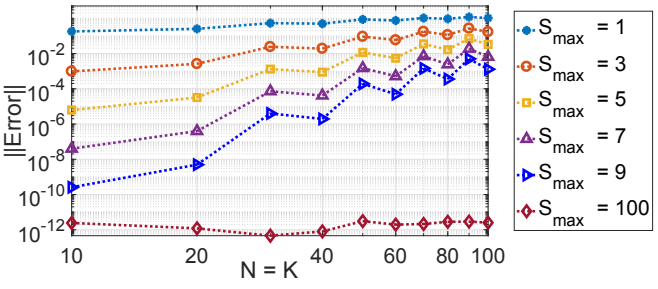

Fig. 4. Infinity norm of the error at solution

Fig. 5 shows the maximum of the total number of inner iterations performed by Algorithm 2 for each iteration of Algorithm 1. For $S_{max} = 100$, it can be seen that the Algorithm 2 converges to specified tolerance with in few tens of iterations. However, there is a modest increase in the total number of iterations performed as the system size increases.

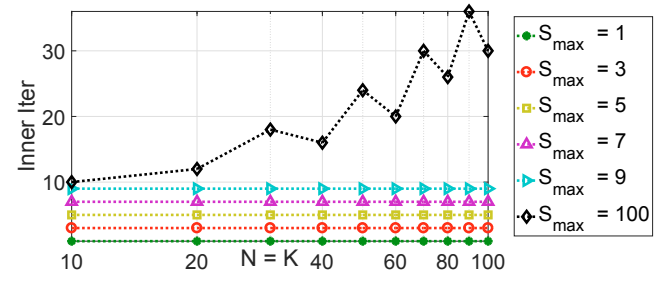

Fig. 5. Total number of iterations of Algorithm 2

6. CONCLUSIONS

A nested algorithm based on block Jacobi iterations is devised for solving a class of tree-structured finite horizon optimal control problems subject to the linear discrete-time dynamics. The proposed algorithm exhibits per-iteration computational cost which scales linearly with the number of sub-systems N, K and the length of time horizon T. The computations at each iteration can be distributed across $N \times K$ parallel processing agents in a network mirroring the tree structure of the graph. A preliminary convergence analysis of the proposed algorithm is provided. It is observed numerically that there is a modest growth in spectral radius of the iteration matrix Γ which results in increase of the number of iterations required to converge in the size $N = K$, $T = N + K$ of problems. Future work includes a detailed theoretical analysis for convergence of nested Jacobi iterations subject to early termination of inner iterations. It also includes the consideration of inequality constraints on the state and control input variables and extending the results for a more general class of tree graphs.

7. ACKNOWLEDGMENTS

This work was supported by the Australian Research Council (LP160100666). Aspects of it were done while F. Farokhi held a joint appointment with CSIRO's Data61, Docklands, VIC 3008, Australia.

REFERENCES

Bacciotti, A. (2019). *Stability and Control of Linear Systems*, volume 185 of *Studies in Systems, Decision and Control*. Springer International Publishing, Cham.

Cantoni, M., Farokhi, F., Kerrigan, E., and Shames, I. (2017). Structured computation of optimal controls for constrained cascade systems. *International Journal of Control*, 7179(October), 1–10.

Cantoni, M., Weyer, E., Li, Y., Ooi, S.K., Mareels, I., and Ryan, M. (2007). Control of large-scale irrigation networks. *Proceedings of the IEEE*, 95(1), 75–91.

Golub, G.H. and Van Loan, C.F. (2013). *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press.

Hackbusch, W. (2016). *Iterative Solution of Large Sparse Systems of Equations*, volume 95 of *Applied Mathematical Sciences*. Springer International Publishing, Cham.

Horn, R.A. and Johnson, C.R. (2013). *Matrix analysis*. Cambridge University Press.

Li, Y., Cantoni, M., and Weyer, E. (2005). On water-level error propagation in controlled irrigation channels. In *Proceedings of the 44th IEEE Conference on Decision and Control*, volume 2005, 2101–2106.

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.

Nocedal, J. and Wright, S.J. (2000). *Numerical optimization 2nd edition*. Springer.

Peng, Q. and Low, S.H. (2018). Distributed optimal power flow algorithm for radial networks, I: Balanced single phase case. *IEEE Transactions on Smart Grid*, 9(1), 111–121.

Perea-López, E., Ydstie, B.E., and Grossmann, I.E. (2003). A model predictive control strategy for supply chain optimization. *Computers and Chemical Engineering*, 27(8-9), 1201–1218.

Zafar, A., Cantoni, M., and Farokhi, F. (2019). Optimal control computation for cascade systems by structured Jacobi iterations. *IFAC-PapersOnLine*, 52(20), 291–296.