University Library

MINERVA ACCESS

A gateway to Melbourne's research publications

Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:
Guo, M;Lang, A;Cantoni, M

Title:
Structured moving horizon estimation for linear system chains

Date:
2019-06-01

Citation:
Guo, M., Lang, A. & Cantoni, M. (2019). Structured moving horizon estimation for linear system chains. 2019 18th European Control Conference (ECC), pp.1830-1835. IEEE. https://doi.org/10.23919/ECC.2019.8795717.

Persistent Link:
https://hdl.handle.net/11343/307664

Structured Moving Horizon Estimation for Linear System Chains*

Meichen Guo[1] Adair Lang[2] Michael Cantoni[2]

***Abstract*— Computational aspects of moving horizon state estimation are studied for a class of chain networks with bidirectional coupling in the linear state dynamics, and measured outputs. Moving horizon estimation involves solving a quadratic program to minimize the estimation error relative to a model over a fixed window of past input-output observations. By exploiting the spatial structure of a chain, two algorithms for solving this quadratic program are considered. Both algorithms can be distributed in the sense that the computations associated with each sub-system component of the state depend only on information associated with the immediate neighbours. The algorithms differ in the way that the linear Karush-Kuhn-Tucker conditions for optimality are solved. Computational and information dependency overheads are analyzed. Numerical results are presented for a 1-D mass-spring-damper chain.**

I. INTRODUCTION

Estimation techniques are used in control engineering when the system state is difficult to measure directly. Moving horizon estimation (MHE) is an optimization based method for state estimation. Specifically, observations of system inputs and outputs over a fixed time horizon into the past, and a prediction of the system state at the start of this window, are the inputs to an optimization problem to be solved at each time. This problem is formulated to minimize the estimation error over the horizon, relative to a given model of the dynamics [1], [2], [3], [4].

In recent years, work has emerged on decentralized and distributed MHE for networks of sub-systems. For instance, in [5] and [6] partition-based MHE algorithms for estimating each sub-system component of the state are developed. However, all-to-all information exchange is required between the computations for each partition, at each time step. A distributed MHE strategy is proposed in [7] for unconstrained large-scale systems having sparse banded or sparse multi-banded system matrices, based on a Chebyshev approximation of the centralized solution to the MHE problem. However, the hop proximity of information required to compute the estimate of a sub-system component of the state grows with the time horizon.

In this paper, computational aspects of the situation in which the linear system dynamics has chain structure are investigated. This structure corresponds to an interconnection of sub-systems over a path graph, with localized bi-directional coupling in the state dynamics, and the dependence of the observed sub-system output on the sub-system states. For this case, a tailored formulation of the optimization problem to be solved at each time step in an MHE scheme is amenable to structured computations. More specifically, by exploiting structure, the computations associated with each sub-system component of the state can be distributed across a (real or virtual) network of information processing agents, one for each sub-system. Moreover, the computational dependencies between these agents can be localized to the agents associated with the (at most two) neighbouring sub-systems in the chain only.

For the linear model considered here, without inequality constraints on the state estimates along the horizon, the tailored optimization problem to solve at each step is an equality constrained quadratic program. In particular, the Karush-Kuhn-Tucker (KKT) conditions that characterize optimality (*e.g.*, see [8]) amount to a block tri-diagonal system of linear equations to be solved. Although not explicitly considered below, with inequality constraints along the horizon, each Newton step of an interior point method (*e.g.*, see [8]) would also involve a block tri-diagonal system of linear equations of the form studied. Two structure exploiting algorithms are proposed for solving these equations. One is based on direct solution by backward-forward recursion [9]. A similar approach is studied in [10], [11], [12] for moving horizon control/estimation of linear cascade systems; *i.e.*, unidirectional coupling. The other algorithm is based on consensus iterations [13], [14], [15]. For both algorithms, the computational dependencies are localized in the sense described above. However, the computational and information exchange overheads are quite different.

The paper is organized as follows. In Section II, the MHE problem is formulated as a quadratic program. In Section III, two algorithms are developed for solving the corresponding KKT conditions. These are applied to a one dimensional mass-spring-damper chain in Section IV. Simulation results are illustrated and analyzed for the algorithms. Finally, the conclusions are drawn in Section V.

II. PROBLEM FORMULATION

Consider the standard discrete-time system model

$$x(t+1) = Ax(t) + Bu(t), \quad y(t) = Cx(t), \qquad (1)$$

with state $x(t) \in \mathbb{R}^n$, input $u(t) \in \mathbb{R}^m$, and output $y(t) \in \mathbb{R}^p$, for $t \in \mathbb{N} \cup \{0\}$. Moving horizon estimation (MHE) involves minimizing

$$J_t = \frac{1}{2}(\hat{x}(t-T) - \bar{x}(t-T))^{\intercal} P(\hat{x}(t-T) - \bar{x}(t-T)) + \frac{1}{2}\sum_{k=t-T}^{t}(\hat{y}(k) - y(k))^{\intercal} Q(\hat{y}(k) - y(k)),$$

*Funded in part by the Australian Research Council (LP160100666).

[1]Faculty of Science and Engineering, University of Groningen, 9700 AB Groningen, The Netherlands.

[2]Dept. Electrical and Electronic Engineering, The University of Melbourne, VIC 3010, Australia. Email:cantoni@unimelb.edu.au

by selecting the estimator output $\hat{x}(t-T|t)$ at each time step $t > T$. This selection determines $\hat{y}(k) = C\hat{x}(k)$ for $k = t-T, \ldots, t$, via $\hat{x}(k+1) = A\hat{x}(k) + Bu(k)$ and $\hat{x}(t-T) = \hat{x}(t-T|t)$ in accordance with (1), for the given observations $u(t-T), y(t-T), \ldots, u(t), y(t)$ and prediction $\bar{x}(t-T) = A\hat{x}(t-T-1|t-1) + Bu(t-T-1)$, based on the estimate $\hat{x}(t-T-1|t-1)$ from time step $t-1$. The integer $T > 0$ is the estimation horizon, $Q \in \mathbb{R}^{p\times p}$ is positive definite and $P \in \mathbb{R}^{n\times n}$ is positive semi-definite. If (A, C) is $T+1$ step observable, or P is positive definite, then the optimal estimate is unique [2]. When the observability hypothesis holds, the estimation error is stable for any positive semi-definite P, with exponential decay rate related to the size of P [2]; for $P = 0$ it is deadbeat. When this observability hypothesis does not hold, initial error may persist.

The special case of MHE for a chain of $N \in \mathbb{N}$ sub-systems with bidirectional coupling is the focus below. In particular, attention is restricted to dynamics of the form

$$
\begin{aligned}
x_i(t+1) &= A_i x_i(t) + B_i u_i(t) + E_{i-} x_{i-1}(t) + E_{i+} x_{i+1}(t),\\
y_i(t) &= C_i x_i(t) + H_{i-} x_{i-1}(t) + H_{i+} x_{i+1}(t),
\end{aligned} \tag{2}
$$

where $x_i(t) \in \mathbb{R}^{n_i}$, $u_i(t) \in \mathbb{R}^{m_i}$, and $y_i(t) \in \mathbb{R}^{p_i}$ are respectively the components of the overall state $x(t)$, input $u(t)$ and output $y(t)$ associated with sub-system $i = 1, \ldots, N$ at time $t \in \mathbb{N} \cup \{0\}$. The spatial boundary conditions are consistent with setting $E_{1-} = E_{N+} = 0$ and $H_{1-} = H_{N+} = 0$.

Assumption 2.1: The pair (A, C) is observable over time horizon $T + 1$, where

$$
A = \begin{bmatrix} A_1 & E_{1+} & \cdots & \cdots & 0 \\ E_{2-} & A_2 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & E_{(N-1)+} \\ 0 & \cdots & 0 & E_{N-} & A_N \end{bmatrix}, \text{ and}
$$

$$
C = \begin{bmatrix} C_1 & H_{1+} & \cdots & \cdots & 0 \\ H_{2-} & C_2 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & H_{(N-1)+} \\ 0 & \cdots & 0 & H_{N-} & C_N \end{bmatrix}.
$$

Subsequently, the special structure in the model (2) is exploited to compute the minimizer of a correspondingly separable estimation error $J_t = \sum_{i=1}^{N} J_{t,i}$, where

$$
\begin{aligned}
J_{t,i} = &\frac{1}{2}(\hat{x}_i(t-T) - \bar{x}_i(t-T))^{\mathsf{T}} P_i (\hat{x}_i(t-T) - \bar{x}_i(t-T)) \\
&+ \frac{1}{2}\sum_{k=t-T}^{t} (\hat{y}_i(k) - y_i(k))^{\mathsf{T}} Q_i (\hat{y}_i(k) - y_i(k)),
\end{aligned} \tag{3}
$$

$Q_i \in \mathbb{R}^{p_i\times p_i}$ is positive definite, $P_i \in \mathbb{R}^{n_i\times n_i}$ is positive semi-definite, and $\hat{y}_i(k)$ is constructed according to the model (2) with initial condition $\hat{x}_i(t-T) = \hat{x}_i(t-T|t)$, given input-output observations

$$
\begin{aligned}
u_{i|t} &= \left[u_i(t-T)^{\mathsf{T}}, \ldots, u_i(t-1)^{\mathsf{T}}\right]^{\mathsf{T}} \in \mathbb{R}^{m_i T},\\
y_{i|t} &= \left[y_i(t-T)^{\mathsf{T}}, \ldots, y_i(t)^{\mathsf{T}}\right]^{\mathsf{T}} \in \mathbb{R}^{p_i (T+1)},
\end{aligned}
$$

and the prediction $\bar{x}_i(t-T) \in \mathbb{R}^{n_i}$ from the previous step of the MHE scheme, for each $i = 1, \ldots, N$. Specifically, it is of interest to localize the spatial dependencies between the computations associated with construction of each sub-system component $\hat{x}_i(t-T|t)$ of the estimate for each $i = 1, \ldots, N$. In particular, it may be desirable to structure the computations in a way that distributes across a network (virtual or real) of processing agents with an information exchange graph that mirrors the chain structure of the system dynamics (*i.e.*, an undirected path). The approach taken is based on considering the collection of state-estimates across the whole horizon as the decision variables when characterizing the solution of the optimization problem, instead of resolving the structured equality constraints by taking $\hat{x}(t-T)$ as the only decision variable, along the lines suggested in the generic problem formulation above. Indeed, it is the resolution of the equality constraints in this way that results in all-to-all exchanges in the partition based distributed MHE schemes of [5], [6].

Lemma 2.1: With reference to (3), the problem $\min_{\hat{x}_i(t-T|t)\in\mathbb{R}^{n_i}} \sum_{i=1}^{N} J_{t,i}$, can be formulated as the quadratic program (QP):

$$
\begin{aligned}
\min_{\hat{x}_{i|t}\in\mathbb{R}^{n_i(T+1)},\ \delta_i\in\mathbb{R}^{n_i},\ \eta_i\in\mathbb{R}^{p_i(T+1)}} \ &\frac{1}{2}\Big(\sum_{i=1}^{N} \delta_i^{\mathsf{T}} P_i \delta_i \\
&+ (y_{i|t} - \bar{C}_i\hat{x}_{i|t} - \eta_i)^{\mathsf{T}} \bar{Q}_i (y_{i|t} - \bar{C}_i\hat{x}_{i|t} - \eta_i)\Big)
\end{aligned} \tag{4}
$$

subject to

$$
\begin{aligned}
\bar{A}_i\hat{x}_{i|t} - \bar{E}_{i+}\hat{x}_{(i+1)|t} - \bar{E}_{i-}\hat{x}_{(i-1)|t} - K_i\delta_i - d_i = 0,\\
\eta_i - \bar{H}_{i+}\hat{x}_{(i+1)|t} - \bar{H}_{i-}\hat{x}_{(i-1)|t} = 0,
\end{aligned} \tag{5}
$$

for $i = 1, \ldots, N$, where $d_i = -\bar{B}_i u_{i|t} + \tilde{x}_i$, $\tilde{x}_i = \begin{bmatrix}\bar{x}_i^{\mathsf{T}}(t-T) & 0_{1\times n_i T}\end{bmatrix}^{\mathsf{T}}$, $K_i = \begin{bmatrix} I_{n_i} & 0_{n_i\times n_i T}\end{bmatrix}^{\mathsf{T}}$,

$$
\bar{A}_i = I_{n_i(T+1)} - \begin{bmatrix} 0_{n_i\times n_i T} & 0_{n_i\times n_i} \\ I_T\otimes A_i & 0_{n_i T\times n_i}\end{bmatrix},\ \bar{B}_i = \begin{bmatrix} 0_{n_i\times m_i T} \\ I_T \otimes B_i\end{bmatrix},
$$

$$
\bar{E}_{i+} = \begin{bmatrix} 0_{n_i\times n_{i+1}T} & 0_{n_i\times n_{i+1}} \\ I_T\otimes E_{i+} & 0_{n_i T\times n_{i+1}}\end{bmatrix},
$$

$$
\bar{E}_{i-} = \begin{bmatrix} 0_{n_i\times n_{i-1}T} & 0_{n_i\times n_{i-1}} \\ I_T\otimes E_{i-} & 0_{n_i T\times n_{i-1}}\end{bmatrix},
$$

$\bar{C}_i = I_{(T+1)}\otimes C_i$, $\bar{Q}_i = I_{(T+1)}\otimes Q_i$, $\bar{H}_{i-} = I_{(T+1)}\otimes H_{i-}$, $\bar{H}_{i+} = I_{(T+1)}\otimes H_{i+}$, and $\otimes$ denotes Kronecker product, I_k a $k\times k$ identity matrix, and $0_{k\times l}$ a $k\times l$ zero matrix.

Proof: With the definitions above, and with $\delta_i = \hat{x}_i(t-T) - \bar{x}_i(t-T)$, $\hat{x}_{i|t} = [\hat{x}_i(t-T), \ldots, \hat{x}_i(t)]^{\mathsf{T}}$, and $\eta_i = \bar{H}_{i+}\hat{x}_{(i+1)|t} + \bar{H}_{i-}\hat{x}_{(i-1)|t}$ for $i = 1, \ldots, N$, it follows that

$$
J_{t,i} = \frac{1}{2}\Big(\delta_i^{\mathsf{T}} P_i\delta_i + (y_{i|t} - \bar{C}_i\hat{x}_{i|t} - \eta_i)^{\mathsf{T}}\bar{Q}_i(y_{i|t} - \bar{C}_i\hat{x}_{i|t} - \eta_i)\Big).
$$

The constraints (5) simply encode these additional definitions, which are consistent with (2). ∎

The solution of the QP (4–5) is unique under Assumption 2.1. The rest of the paper is about structured computation of this solution, via the KKT conditions. The aim is to achieve distributed computations in the aforementioned fashion with localized information dependencies.

It is important to note that the formulation of QP (4–5) is tailored to exploiting the chain structure of the MHE problem at hand. The temporal structure in (2) is reflected in the structure of $\bar{A}_i$, $\bar{B}_i$, $\bar{E}_{i+}$, and $\bar{E}_{i-}$. However, it is difficult to simultaneously exploit this structure and the spatial structure in the KKT conditions; see Remark 3.2.

III. Structured Computations

Two algorithms for solving the KKT conditions for the QP (4–5) are developed in this section. First the structure of the corresponding KKT conditions is identified. The two different approaches to solve the resulting block tri-diagonal linear system of equations are then devised. One involves direct solution of this system of equations. The other involves consensus iterations to compute the solution. In both cases, only localized dependencies arise in computations that can be associated with each sub-system in the chain. In this sense, the algorithms are amenable to distribution across a network of processing agents with undirected path graph for information exchange.

A. Structure of the KKT conditions

The KKT conditions for QP (4–5) are given by

$$\begin{aligned} \bar{C}_i^\intercal\bar{Q}_i\bar{C}_i\hat{x}_{i|t} + \bar{C}_i^\intercal\bar{Q}_i\eta_i - \bar{A}_i^\intercal q_i \\ +\bar{E}_{(i-1)+}^\intercal q_{(i-1)} + \bar{E}_{(i+1)-}^\intercal q_{(i+1)} \\ -\bar{H}_{(i-1)+}^\intercal l_{(i-1)} - \bar{H}_{(i+1)-}^\intercal l_{(i+1)} \\ -\bar{C}_i^\intercal\bar{Q}_i y_{i|t} = 0, \\ \bar{Q}_i\bar{C}_i\hat{x}_{i|t} + \bar{Q}\eta_i + l_i - \bar{Q}_i y_{i|t} = 0, \\ P_i\delta_i + K_i^\intercal q_i = 0, \\ \bar{A}_i\hat{x}_{i|t} - \bar{E}_{i+}\hat{x}_{(i+1)|t} - \bar{E}_{i-}\hat{x}_{(i-1)|t} - K_i\delta_i - d_i = 0, \\ \eta_i - \bar{H}_{i+}\hat{x}_{(i+1)|t} - \bar{H}_{i-}\hat{x}_{(i-1)|t} = 0, \end{aligned} \tag{6}$$

for $i = 1,\dots,N$, where $q_i \in \mathbb{R}^{n_i(T+1)}$ and $l_i \in \mathbb{R}^{p_i(T+1)}$ are the dual variables associated with the equality constraints in (5). With $\zeta_i = [(\hat{x}_{i|t})^\intercal,(\eta_i)^\intercal,(\delta_i)^\intercal,(q_i)^\intercal,(l_i)^\intercal]^\intercal$ for $i = 1,\dots,N$, the conditions (6) can be re-written as

$$\begin{bmatrix} D_1 & \Upsilon_2^\intercal & 0 & \cdots & 0 \\ \Upsilon_2 & D_2 & \Upsilon_3^\intercal & \ddots & \vdots \\ 0 & \Upsilon_3 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \Upsilon_N^\intercal \\ 0 & \cdots & 0 & \Upsilon_N & D_N \end{bmatrix} \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \vdots \\ \vdots \\ \zeta_N \end{bmatrix} = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \vdots \\ \psi_N \end{bmatrix} \tag{7}$$

where

$$D_i = \begin{bmatrix} \bar{C}_i^\intercal\bar{Q}_i\bar{C}_i & \bar{C}_i^\intercal\bar{Q}_i & 0 & -\bar{A}_i^\intercal & 0 \\ \bar{Q}_i\bar{C}_i & \bar{Q}_i & 0 & 0 & I_{p_i(T+1)} \\ 0 & 0 & P_i & K_i^\intercal & 0 \\ -\bar{A}_i & 0 & K_i & 0 & 0 \\ 0 & I_{p_i(T+1)} & 0 & 0 & 0 \end{bmatrix},$$

$$\Upsilon_i = \begin{bmatrix} 0 & \tilde{\Upsilon}_{(i-1)+}^\intercal \\ \tilde{\Upsilon}_{i-} & 0 \end{bmatrix}, \ \tilde{\Upsilon}_{i+} = \begin{bmatrix} 0 & 0 \\ \bar{E}_{i+} & 0 \\ -\bar{H}_{i+} & 0 \end{bmatrix},$$

$$\tilde{\Upsilon}_{i-} = \begin{bmatrix} 0 & 0 \\ \bar{E}_{i-} & 0 \\ -\bar{H}_{i-} & 0 \end{bmatrix}, \text{ and } \ \psi_i = \begin{bmatrix} \bar{C}_i^\intercal\bar{Q}_i y_{i|t} \\ \bar{Q}_i y_{i|t} \\ 0 \\ -d_i \\ 0 \end{bmatrix},$$

for $i = 1,\dots,N$.

Note that (7) admits a unique solution by Assumption 2.1. Two structured algorithms for computing the solution ζ_i, $i = 1,\dots,N$, are developed in the next two sub-sections. From this solution the state estimate $\hat{x}_i(t-T|t)$ can be extracted. By virtue of the block tri-diagonal structure of the matrix Ω on the left-hand side of (7), for each i the computation, in both cases, depends on information pertaining to sub-systems $i-1$, i and $i+1$ only. In the first algorithm, the dependence is sequential. In the second algorithm, the dependence is across iterations of the underlying consensus approach.

B. Computation by backward-forward recursion

The first approach, summarized in Algorithm 1, is based on the following lemma.

Lemma 3.1: If P_N is positive definite, then given $\psi = [\,\psi_1^\intercal \cdots \psi_N^\intercal\,]^\intercal$ the solution $\zeta = [\,\zeta_1^\intercal \cdots \zeta_N^\intercal\,]^\intercal$ of (7) can be determined by the following backward and forward recursions:

$$\widetilde{\psi}_i = \begin{cases} \psi_N & \text{for } i = N \\ \psi_i - \Upsilon_{i+1}^\intercal \Sigma_{(i+1)}^{-1}\widetilde{\psi}_{(i+1)} & \text{for } i = N-1,\dots,1 \end{cases}$$

$$\zeta_i = \begin{cases} \Sigma_1^{-1}\widetilde{\psi}_1 & \text{for } i = 1 \\ \Sigma_i^{-1}(\widetilde{\psi}_i - \Upsilon_i\zeta_{(i-1)}) & \text{for } i = 2,\dots,N \end{cases}$$

where

$$\Sigma_i = \begin{cases} D_N & \text{for } i = N \\ D_i - \Upsilon_{i+1}^\intercal \Sigma_{(i+1)}^{-1}\Upsilon_{i+1} & \text{for } i = N-1,\dots,1 \end{cases}.$$

Proof: The backward-forward recursions stated above arise as described in [9], for example. Note that D_N is non-singular, as $\bar{A}_N$ and P_N are non-singular for $i = 1,\dots,N$, whereby D_N clearly has full column rank. As such, Σ_N is non-singular. To see that Σ_i remains non-singular for $i = N-1,\dots,1$, suppose there exists $j < N$ such that this holds for $i = N,\dots,j+1$ but not $i = j$; *i.e.*, there exists $\omega_j \neq 0$ such that $\Sigma_j\omega_j = (D_j - \Upsilon_{(j+1)}^\intercal\Sigma_{(j+1)}^{-1}\Upsilon_{(j+1)})\omega_j = 0$. Then with $\omega_i = -\Sigma_i^{-1}\Upsilon_i\omega_i$ for $i = j+1,\dots,N$, and $\omega_i = 0$ for $i = 1,\dots,j-1$, it follows that $\Omega[\omega_1^\intercal,\dots,\omega_N^\intercal]^\intercal = 0$. This is a contradiction, since Ω is non-singular by Assumption 2.1. Therefore, no such j exists, and the recursions are well defined as claimed. ∎

Remark 3.1: It can be shown that D_N is non-singular if (A_N, C_N) is $T+1$ step observable. This observability condition can replace the positive definiteness condition on P_N in Lemma 3.1. Note that similar forward-backward recursions can be devised in the case that D_1 is non-singular, which hold if P_1 is positive definite, or (A_1, C_1) is $T+1$ step observable.

Remark 3.2: Although each D_i is structured in accordance with the temporal dynamics expressed in (2), this is difficult to exploit in line 6 of Algorithm 1. As the algorithm evolves in i from N downto 1, the matrices Σ_i

Algorithm 1: Backward-forward recursion

1 $\widehat{\psi}_N = 0$
2 $\Delta_N = 0$
3 **for** $i = N$ **downto** 1 **do**
4 $\quad \widetilde{\psi}_i \leftarrow \psi_i - \widehat{\psi}_i$
5 $\quad \Sigma_i \leftarrow D_i - \Delta_i$
6 $\quad$ **solve** $\Sigma_i \begin{bmatrix} X_i & \chi_i \end{bmatrix} = \begin{bmatrix} \Upsilon_i & \widetilde{\psi}_i \end{bmatrix}$
7 $\quad \widehat{\psi}_{i-1} \leftarrow \Upsilon_i^\intercal \chi_i$
8 $\quad \Delta_{i-1} \leftarrow \Upsilon_i^\intercal X_i$
9 **end**
10 $\zeta_0 \leftarrow 0$
11 **for** $i = 1$ **to** N **do**
12 $\quad \zeta_i \leftarrow \chi_i - X_i \zeta_{(i-1)}$
13 **end**

Algorithm 2: Consensus iterations for sub-system i

1 $s \leftarrow 0$
2 **solve** $D_i\, \zeta = \psi_i$
3 $\zeta_{[i]}^s = \mathbf{e}_i \otimes \zeta$
4 $r \leftarrow \varepsilon + 1$
5 **while** $r \geq \varepsilon$ **do**
6 $\quad \rho_{[i]}^s \leftarrow \sum_{j \in \mathcal{N}_i} (\zeta_{[i]}^s - \zeta_{[j]}^s)$
7 $\quad \zeta_{[i]}^{s+1} \leftarrow \zeta_{[i]}^s + \frac{1}{\mathrm{card}(\mathcal{N}_i)} \Pi_i \rho_{[i]}^s$
8 $\quad s \leftarrow s + 1$
9 $\quad r \leftarrow \max_j \|\rho_{[j]}\|_\infty$
10 **end**

lose structure. Therefore, the **solve** step in line 6 eventually incurs computational burden that scales as T^3. On the other hand, the number of such solves scales as N.

Remark 3.3: Note the sequential nature of Algorithm 1. It can be distributed (but not parallelized) across a network of N processing agents with computational dependencies (through the variables $\widehat{\psi}_i$, Δ_i, and ζ_i) that mirror the chain structure of the system dynamics. The tasks of agent $i \in \{1, \ldots, N\}$ in such an implementation would be as follows: (a) to process the body of the for loop in lines 3–9, after agent $i+1$ has completed the corresponding task and made the result available to agent i, starting with agent N; and then (b) to process the body of the for loop in lines 11–13, after agent $i-1$ has completed the corresponding task and made the result available to agent i, starting from agent 1. In this way, the overall number of localized information exchanges between agents is limited to $2(N-1)$.

Remark 3.4: In the absence of inequality constraints, as above, the part of step 6 in Algorithm 1 that yields X_i, need only be done for $t = T$. However, with inequality constraints over the time horizon, the corresponding Newton sub-steps required to solve the bi-linear KKT conditions in an interior point method (IPM) would involve a different Σ_i at each iteration, and as such, the computation of X_i would need to be done repeatedly.

C. Computation by consensus iterations

Alternatively, the block tri-diagonal system (7) could be solved by consensus based iterations as the underlying graph structure is connected. This can be done without sacrificing localized dependencies of the computations that can be associated with each sub-system $i = 1, \ldots, N$, in line with the preceding developments. Moreover, the consensus updates for each sub-system can be parallelized. However, many iterations may be required to converge to the solution.

For $i = 1, \ldots, N$, let $\mathbf{e}_i$ denote the i-th canonical basis vector in $\mathbb{R}^N$, and let Π_i denote the orthogonal projection onto the kernel of the i-th block row Ω_i of the block tri-diagonal matrix Ω on the left-hand side of (7); *i.e.*, $\Pi_i = I - \Omega_i^\intercal(\Omega_i\Omega_i^\intercal)^{-1}\Omega_i$. Moreover, let $\mathcal{N}_i$ denote the set of the indexes of sub-systems that neighbour sub-system i in the chain; *i.e.*, $\mathcal{N}_i = \{(i-1), (i+1)\}$ for $i = 2, \ldots, N-1$, and $\mathcal{N}_1 = \{2\}$, and $\mathcal{N}_N = \{(N-1)\}$. With this notation, and $\mathrm{card}(\mathcal{X})$ denoting the cardinality of set $\mathcal{X}$, Algorithm 2 is an implementation of the consensus iterations studied in [14], [15], for example.

Remark 3.5: Note that the computations in Algorithm 2 naturally distribute across a network of N processing agents in a parallel fashion, with inter-iteration dependencies for lines 6–7 that can be characterized by a path graph. These dependencies lead to $2N$ localized information exchanges per consensus iteration. The residual update in line 9 can be determined via a forward-backward pass along the chain to update the max of $\|\rho_{[i]}\|_\infty$ sequentially, which incurs a further $2(N-1)$ exchanges per consensus iteration. The per-iteration overhead in these terms scales comparably to Algorithm 1.

Remark 3.6: Many iterations may be required for consensus to be reached, leading to large computational and information exchange overhead overall. Indeed, while a path graph is connected, its algebraic connectivity is given by $1 - \cos(\pi/(N-1))$, which approaches 0 as the number of nodes N approaches infinity. This property of path graphs translates to poor scalability of the number of iterations required to reach consensus as the path graph grows in length, as confirmed in the numerical example below.

Remark 3.7: The per agent computational burden per consensus iteration, scales as T^2, due to the matrix vector product $\Pi_i \rho_{[i]}^s$.

IV. Numerical Example

In this section, the two algorithms from the previous section are applied to the problem of state estimation of mass-spring-damper chains of various lengths. To facilitate this initial comparison in terms of overall computational overhead, the implementation of the algorithm in both cases involves a single thread on one processor; *i.e.*, the possibility of exploiting parallelism in the implementation of Algorithm 2 is not considered, and in each case the computations for each component of the state are done sequentially.

Consider a mass-spring-damper chain consisting of $N > 0$ masses, $m_i > 0$ for $i = 1, \ldots, N$, with corresponding spring constants $k_i > 0$, and damper coefficient $b_i > 0$ for $i = 1, \dots, N+1$, as illustrated in Figure 1. The continuous time dynamics of the system can be expressed as

$$\begin{aligned} m_i \ddot{q}_i = &-b_{i+1}(\dot{q}_i - \dot{q}_{i+1}) + b_i(\dot{q}_{i-1} - \dot{q}_i) \\ &- k_{i+1}(q_i - q_{i+1}) + k_i(q_{i-1} - q_i) \end{aligned} \quad (8)$$

for $i = 1, \dots, N$, where $q_i = z_i - \sum_{k=1}^{i} \lambda_k$, z_i is the position of m_i from the left wall, λ_i is equilibrium position, and $q_0 = 0 = q_{N-1}$. Defining $x_i = [q_i \ \dot{q}_i]^\intercal$ and $u_i = F_i$, the dynamics (8) can be written in the form of

$$\dot{x}_i = \tilde{A}_i x_i + \tilde{E}_{i+} x_{i+1} + \tilde{E}_{i-} x_{i-1},$$

where

$$\tilde{A}_i = \begin{bmatrix} 0 & 1 \\ -\frac{k_i + k_{i+1}}{m_i} & -\frac{b_i + b_{i+1}}{m_i} \end{bmatrix},$$
$$\tilde{E}_{i+} = \begin{bmatrix} 0 & 0 \\ \frac{k_{i+1}}{m_i} & \frac{b_{i+1}}{m_i} \end{bmatrix}, \tilde{E}_{i-} = \begin{bmatrix} 0 & 0 \\ \frac{k_i}{m_i} & \frac{b_i}{m_i} \end{bmatrix},$$

for $i = 1, \dots, N$, and $\tilde{E}_{1-} = \tilde{E}_{N+} = 0_{2\times 2}$. Discretizing the continuous time model via forward Euler gives

$$x_i(t + \Delta t) = A_i x_i(t) + E_{i+} x_{i+1}(t) + E_{i-} x_{i-1}(t) \quad (9)$$

where $A_i = \Delta t \tilde{A}_i + I_{2\times 2}$, $E_{i+} = \Delta t \tilde{E}_{i+}$, and $E_{i-} = \Delta t \tilde{E}_{i-}$.

The following setting are made for the numerical example: $m_i = 1kg$; $k_i = 1N/m$; $b_i = 1Ns/m$, $\Delta t = 0.5s$. The observation matrices C_i, H_{i-} and H_{i+} are set such that $y_i = \left[\begin{smallmatrix} z_i - z_{i-1} \\ z_i - z_{i+1} \end{smallmatrix}\right] = \left[\begin{smallmatrix} q_i - q_{i-1} + \lambda_i \\ q_i - q_{i+1} - \lambda_{i+1} \end{smallmatrix}\right]$ for $i = 1, \dots, N$, where $z_0 = 0 = q_0$ and $z_{N+1} = \sum_{k=1}^{N+1} \lambda_k$ and $q_{N+1} = 0$; *i.e.*, relative positions between neighbouring masses are measured. The cost function J_i is set with $P_i = 10^{-3} I_2$ and $Q_i = I_2$ for $i = 1, \dots, N$. Note that the cost doesn't depend on λ_i as this is included in the estimates $\hat{y}_i$. The algorithms are applied varying chain length N. As expected, while Algorithm 1 is able to give a good solution, Algorithm 2 can take many iterations to converge. To help speed up the convergence of Algorithm 2, an equivalent state-space model of the mass-spring-damper chain is obtained by pre-scaling the model, before application of the algorithms.

Figure 2 shows the processor time for single thread implementations of the two algorithms for varying N, fixed $T = 5$, and stopping conditions $\varepsilon \in \{10^{-4}, 10^{-6}, 10^{-8}\}$ in the iterative Algorithm 2. For Algorithm 1, the processor time scales linearly as N grows from 2 to 200. Algorithm 2 takes much longer to determine a solution, and the processor time scales poorly. It is observed that decreasing the stopping condition ε increases the processor time for Algorithm 2. For $N > 10$ the consensus iterations exceed 10^4 in all cases!

Figure 3 shows the error $\|\psi - \Omega\zeta\|_\infty$ for the outcome ζ of each algorithm for fixed $T = 5$ and varying N and ε. For Algorithm 1, the errors remain in the order of 10^{-15} for increasing N. For Algorithm 2, the lower bound of the errors depends on the stopping condition. Smaller ε yields smaller errors, and thus more accurate solution.

Figure 4 shows the number of iterations required to reach consensus to the level of $\varepsilon \in \{10^{-4}, 10^{-6}\}$ for $N \in \{10, 20, 30\}$. Even with relatively small N and T, Algorithm 2 requires thousands of iterations to reach the stopping conditions. When N or T increases, the number of iterations needed grows substantially. In fact, Figure 4 explains why the processor time of Algorithm 2 is much longer than that of Algorithm 1, and why smaller ε requires longer processor time in Figure 2.

Finally, Figure 5 shows how the processor time varies with the estimation horizon T for $N = 10$ and $\varepsilon = 10^{-6}$. Similar to Figure 2, the processor time of Algorithm 2 is longer and scales in a less favourable way than for Algorithm 1. Compared with Figure 2, Figure 5 shows that the processor time of Algorithm 1 scales as T^3 as anticipated.

V. Conclusions

Computational aspects of moving horizon estimation are investigated for a class of linear system chains with bidirectional couplings in system dynamics, and measured outputs. By exploiting the spatial structure of such systems, two algorithms are developed that are amenable to distributed computations. Algorithm 1 involves strictly sequential subsystem related computations that are bounded in number in proportion to the length of the chain. Algorithm 2 involves consensus iterations, which can be parallelized, however overall computational burden is high because the number of iterations required to reach consensus scales poorly as the length of the chain grows. An alternative approach, that also leads to parallelizable iterations, could be obtained by application of the alternating direction method of multipliers (ADMM) to the QP (4–5). This and the accommodation of inequality constraints is documented in papers under preparation.

References

[1] W. H. Kwon, P. S. Kim, and P. Park, "A receding horizon kalman FIR filter for linear continuous-time systems," *IEEE Transactions on Automatic Control*, vol. 44, no. 11, pp. 2115–2120, Nov. 1999.

[2] A. Alessandri, M. Baglietto, and G. Battistelli, "Receding-horizon estimation for discrete-time linear systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 3, pp. 473–478, Mar. 2003.

[3] ——, "Moving-horizon state estimation for nonlinear discrete-time systems: New stability results and approximation schemes," *Automatica*, vol. 44, no. 7, pp. 1753–1765, Jul. 2008.

[4] M. A. Müller, "Nonlinear moving horizon estimation in the presence of bounded disturbances," *Automatica*, vol. 79, no. 5, pp. 306–314, May 2017.

[5] M. Farina, G. Ferrari-Trecate, and R. Scattolini, "Moving-horizon partition-based state estimation of large-scale systems," *Automatica*, vol. 46, no. 5, pp. 910–918, May 2010.

[6] R. Schneider and W. Marquardt, "Convergence and stability of a constrained partition-based moving horizon estimator," *IEEE Transactions on Automatic Control*, vol. 61, no. 5, pp. 1316–1321, May 2016.

[7] A. Haber and M. Verhaegen, "Moving horizon estimation for large-scale interconnected systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 11, pp. 2834–2847, Nov. 2013.

[8] S. J. Wright, *Primal-Dual Interior-Point Methods*. Philadelphia: SIAM, 1997.

[9] G. Meurant, "A review on the inverse of symmetric tridiagonal and block tridiagonal matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 13, no. 3, pp. 707–728, Jul. 1992.

[10] M. Cantoni, F. Farokhi, E. Kerrigan, and I. Shames, "Structured computation of optimal controls for constrained cascade systems," *International Journal of Control*, 2017, https://doi.org/10.1080/00207179.2017.1366668.

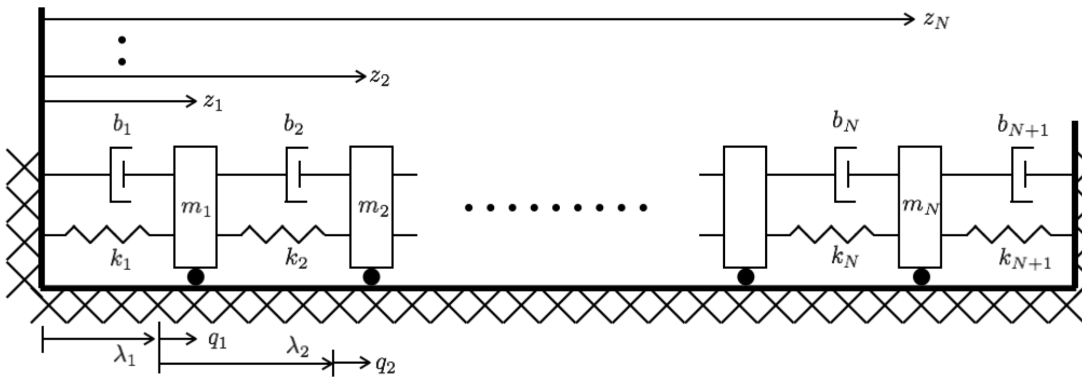

Fig. 1. Multiple mass-spring-damper system.

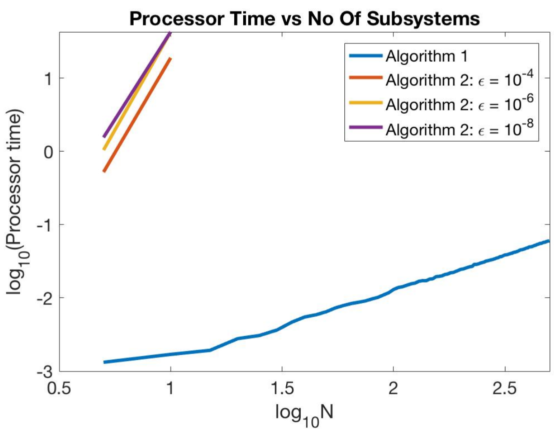

Fig. 2. Single thread processor time of Algorithms 1 and 2 for varying N and fixed $T = 5$.

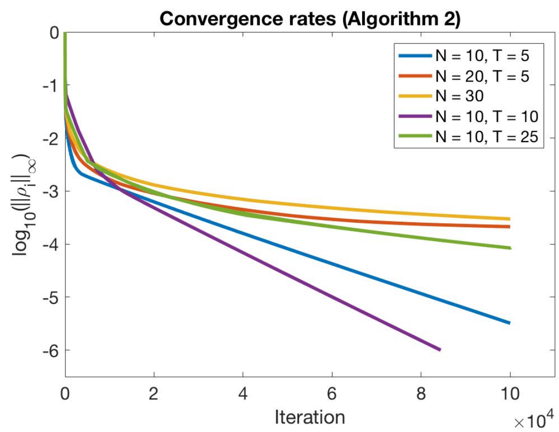

Fig. 4. Convergence rates of Algorithm 2 for varying N and T.

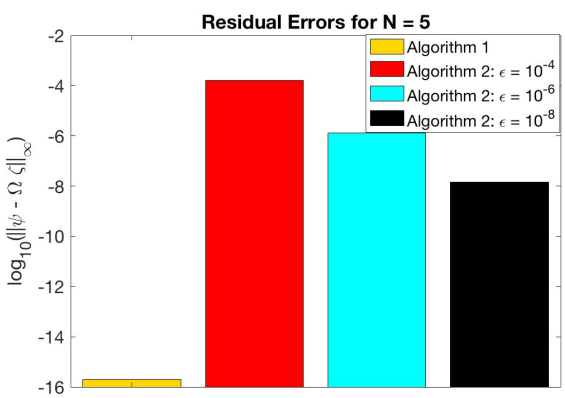

Fig. 3. Error of outcomes of Algorithms 1 and 2.

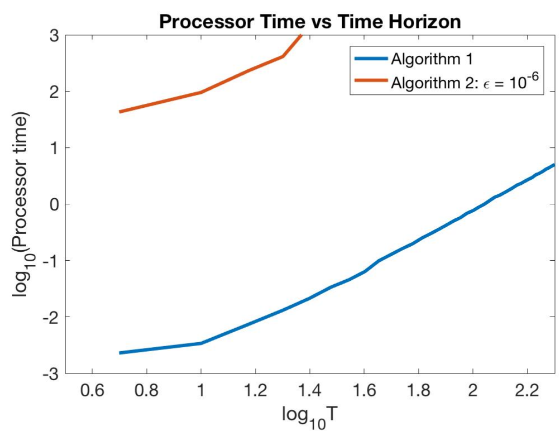

Fig. 5. Single thread processor time for varying T, fixed $N = 10$.

[11] M. Cantoni, A. Zafar, and F. Farokhi, "Scalable iterations for solving constrained LQ control problems with cascade dynamics," in *Proc. 23rd International Symposium on Mathematical Theory of Networks and Systems*, Hong Kong, 2018.

[12] M. Guo, A. Lang, and M. Cantoni, "Moving horizon estimation for linear cascade systems," in *The 15th International Conference on Control, Automation, Robotics and Vision*, Singapore, 2018.

[13] B. D. O. Anderson, S. Mou, A. S. Morse, and U. Helmke, "Decentralized gradient algorithm for solution of a linear equation," *Numerical Algebra, Control & Optimization*, vol. 6, no. 3, pp. 319–328, Sep. 2016.

[14] G. Shi, B. D. O. Anderson, and U. Helmke, "Network flows that solve linear equations," *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 2659–2674, Jun. 2017.

[15] J. Liu, A. S. Morse, A. Nedić, and T. Başar, "Exponential convergence of a distributed algorithm for solving linear algebraic equations," *Automatica*, vol. 83, pp. 37–46, Sep. 2017.