Decentralized moving horizon estimation for large-scale networks of interconnected unconstrained linear systems

Leonardo Pedroso[1] and Pedro Batista[1]

[1] *Institute for Systems and Robotics, Instituto Superior Técnico, Universidade de Lisboa, Portugal*

Accepted version[1]

Available online on 10 Fev 2023

Cite as

L. Pedroso and P. Batista, "Decentralized moving horizon estimation for large-scale networks of interconnected unconstrained linear systems," in *IEEE Transactions on Control of Network Systems*, vol. 10, no. 4, pp. 1855-1866, doi: 10.1109/TCNS.2023.3244086.

[1] Level of access, as per info available on Sherpa Romeo: `https://v2.sherpa.ac.uk/id/publication/37938` (10 Fev 2022)

Decentralized moving horizon estimation for large-scale networks of interconnected unconstrained linear systems

Leonardo Pedroso and Pedro Batista, *Senior Member, IEEE*

***Abstract*— This paper addresses the problem of designing a decentralized state estimation solution for a large-scale network of interconnected unconstrained linear time invariant (LTI) systems. The problem is tackled in a novel moving horizon estimation (MHE) framework, while taking into account the limited communication capabilities and the restricted computational power and memory, which are distributed across the network. The proposed design is motivated by the fact that, in a decentralized setting, a Luenberguer-based framework is unable to leverage the full potential of the available local information. A method is derived to solve a relaxed version of the resulting optimization problem. It can be synthesized offline and its stability can be assessed prior to deployment. It is shown that the proposed approach allows for significant improvement on the performance of recent Luenberger-based filters. Furthermore, we show that a state-of-the-art distributed MHE solution with comparable requirements underperforms in comparison to the proposed solution.**

***Index Terms*— Decision/Estimation Theory, Moving Horizon Estimation, Distributed Algorithms/Control, Networked Control Systems, Networks of Autonomous Agents**

I. Introduction

Over the past decades, decentralized estimation and control have been highly researched topics since they provide a solution to the estimation and control problems of large-scale networks of interconnected systems. In fact, they emerge as an alternative to the use of well-known centralized solutions, which become unfeasible to implement as the dimension of the network increases. The popularity of decentralized solutions is also increasing with the widening of its applications to a broad range of engineering fields in which networks of interconnected systems arise. Examples of such applications are unmanned aircraft formation flight [1], unmanned underwater formations [2], satellite constellations [3], automated highway control [4], irrigation networks [5], industrial processes [6], and various physical models whose underlying principles are modeled by partial differential equations [7].

Although plenty of work has been carried out in decentralized estimation and control of linear time-invariant (LTI) systems, the problem of designing such controllers, which consists in solving an optimization problem subject to a constraint that arises from the distributed nature of the configuration, is intractable [8] and remains an open problem. In fact, the optimal solution for a linear system with Gaussian noise may be nonlinear [9]. Furthermore, it has been shown that the solution of a decentralized design control problem is the result of a convex optimization problem, and thus tractable, if and only if quadratic invariance of the controller set is ensured [10], [11].

This problem has already been addressed, in a classical Luenberger observer framework, making use of four distinct approaches: i) minimum variance of the steady-state global estimation error, modeling the process and observation noise as Gaussian distributions, which degenerates into a decentralized Kalman filter [12], [13]; ii) minimum $\mathcal{H}_2$ norm of the global system [14]–[16]; iii) particular results for systems which verify the aforementioned quadratic invariance condition [17]; and iv) for particular coupling topologies [18]. In this work, the estimation problem of a large-scale network of interconnected actuation capable LTI systems with non-overlapping states is considered. The case for which all states overlap degenerates into a large-scale network of sensors and the goal becomes to collaboratively estimate the global state. This is also known as the consensus estimation problem, which has been studied extensively for large-scale networks [19]–[21]. Furthermore, the case for which there is partial state overlapping has been addressed in [22], making use of consensus arguments.

One limitation of the classical Luenberger observer framework, for both centralized and decentralized estimation solutions, is that it does not take advantage of known bounds on noise and state variables. This degrades the estimation performance and can even lead to the instability of the filter [23]. The moving horizon estimation (MHE) framework was originally introduced to address this matter [24], [25], but has evolved to deal with nonlinear dynamics [26] and outputs [27] as well. It amounts to solving, at each time instant, an optimization problem that takes into account the measurement data of a window of past time instants subject to known bounds. Several distributed MHE approaches have been proposed to exploit known bounds on networks of interconnected linear systems [28]–[32] and validated in impactful applications, e.g., urban road networks [33] and cascade river reaches [34]. In particular, Farina *et al.* [28] proposed three promising distributed MHE methods for large-scale LTI systems with convergence guarantees. One of those, designated PMHE1

Leonardo Pedroso and Pedro Batista are with the Institute for Systems and Robotics, Instituto Superior Técnico, Universidade de Lisboa, Portugal. (e-mail: leonardo.pedroso@tecnico.ulisboa.pt; pbatista@isr.tecnico.ulisboa.pt)

method, only requires local communication between systems, at the expense of disregarding uncertainty associated with the dynamic couplings between systems. However, it does not support correlated process noise between systems with dynamic couplings.

On one hand, decentralized approaches based on the classical Luenberger framework can, generally, be synthesized offline and require communication of filtered state estimates between systems with dynamic couplings exclusively, once per iteration of the discrete-time filter implementation. This decentralized design has, for this reason, low memory, computational load, and communication requirements, allowing for an easy implementation to real large-scale systems. On the other hand, distributed approaches based on the MHE framework are able to exploit known bounds at the expense of demanding computations to solve local finite-horizon optimization problems and broadcasting estimation error covariance matrices in real-time for the whole MHE window. Thus, even though the methods based on a MHE framework are, in principle, able to exceed those based on a Luenberger observer formulation in terms of estimation performance, it comes with an increase in computational, memory, or communication requirements, which makes the application to the vast majority of real systems infeasible.

In this paper, we propose a decentralized state estimation solution for a large-scale network of interconnected unconstrained LTI systems with non-overlapping states subject to unbounded Gaussian process noise with access to local sensing outputs subject to unbounded Gaussian sensor noise. The novelty of the proposed method is that it is based on a MHE framework, even though no bounds are considered neither for the state variables nor for process or sensor noise. It is shown that it is possible to achieve a significant estimation performance improvement in relation to decentralized methods based on the classical Luenberger framework with manageable computational, memory, and communication requirements. Furthermore, we show that a state-of-the-art distributed MHE solution with comparable requirements underperforms in comparison with the solution presented herein.

The motivation for taking this approach is not clear at first sight. It is true that, for an unconstrained system, in a centralized scheme the Luenberger and MHE framework yield the same optimal estimation performance. This is due to the well-known fact that the Kalman filter is the optimal solution to the state estimation problem and that it can be written recursively in the form of a Luenberguer observer [35]. Thus, it is pointless to follow a MHE approach for unconstrained systems in a centralized setting. However, in a decentralized setting that is not the case. The formulation of the decentralized problem as a classical Luenberger observer, for which every newly computed estimate is computed making use of the previous filtered estimate and the newly obtained sensor measurement, comes with a significant decrease of performance. This performance loss is due to the fact that, for decentralized configurations, the optimization of the quality of the estimate of a given instant compromises the quality of future time instants. Thus, besides the obvious loss of performance associated with the decentralized design, there is also the loss associated with this effect, whose origin is detailed more thoroughly in Section II-B. As a means of mitigating the latter form of loss of performance, we consider a MHE framework.

The novel MHE framework for decentralized filter design for large-scale networks of interconnected LTI systems put forward in this paper is developed in discrete-time. It is built on multiple prediction-filtering steps employed in a Luenberger Kalman filter for each individual system, which ensures the consistency of the filter. The problem is formulated in a first instance for the global system, with a sparsity constraint on the filter gain. Such sparsity constraint imposes certain entries of the global gain matrix to be null, following a structure that reflects the communication restrictions in the network, necessary for the implementation of the decentralized state filters. The optimization problem that arises in the problem formulation is nonconvex. Hence, the algorithm that is proposed herein, designated moving finite-horizon (MFH) method, relies on conveniently defined convex relaxations of the original optimization problem to achieve a computationally efficient approximation to its solution. The MFH filter synthesis can (and should) be carried out offline in a computing server, which avoids intensive real-time computations. The local decentralized filters gains can then be extracted from the globally synthesized sparse gain matrix, allowing for its decentralized implementation, leveraging local communication exclusively. Herein, we designate this framework as decentralized, as opposed to distributed, given that it is formulated globally, synthesized offline, and deployed afterwards in each system resorting to local communication. It contrasts with general MHE schemes, which rely on the numerical solution of an optimization problem solved distributively across the network in real-time.

The computational, memory, and communication requirements of the MFH method are comparable with those of a decentralized Luenberger observer formulation:

1) The local measurement of each system is exclusively available for filter computations in that system;
2) Each system has one computational unit associated with it. The computational load of the estimation algorithm of the network is distributed across all computational units in such a way that each carries out computations concerning its own state estimate exclusively, which circumvents the curse of dimensionality;
3) Each computational unit only has to store a time window of parameters concerning its own state estimate;
4) The communication links between systems are restricted to pairs of systems which are dynamically coupled.

Furthermore, the stability of the error dynamics of the proposed MFH filter can be assessed after its offline synthesis and prior to deployment. Finally, the framework and proposed algorithm are validated resorting to numerical simulations. The performance of the MFH method is compared with several methods based on the classical Luenberger observer formulation and with a distributed MHE solution with comparable requirements. A MATLAB implementation of the MFH algorithm and of the simulations can be found in the DECENTER toolbox at https://decenter2021.github.io. The documentation of the MFH method implementation is available at https://decenter2021.github.io/documentation/MHEMovingFiniteHorizonLTI and extensive simulation results at https://decenter2021.github.io/examples/MHEMFH.

This paper is organized as follows. In Section II, the estimation problem is formulated, the assumptions that are considered are introduced, and the properties and requirements of the decentralized design are evaluated. In Section III, the MFH method is derived. In Section IV, the application of the MFH method to two large-scale networks of interconnected systems is simulated and its performance is compared with other state-of-the-art methods. Finally, some concluding remarks are provided in Section V.

A. Notation

Throughout this paper, the identity, null, and ones matrices, all of proper dimensions, are denoted by $\mathbf{I}$, $\mathbf{0}$, and $\mathbf{1}$, respectively. Alternatively, $\mathbf{I}_n$, $\mathbf{0}_{n\times m}$, and $\mathbf{1}_{n\times n}$ are also used to represent the $n\times n$ identity matrix, the $n\times m$ null matrix, and the $n\times m$ ones matrix, respectively. The i-th component of a vector $\mathbf{v}\in\mathbb{R}^n$ is denoted by $[\mathbf{v}]_i$. The entry (i,j) of a matrix $\mathbf{A}$ is denoted by $[\mathbf{A}]_{ij}$. The vectorization of a matrix $\mathbf{A}$, denoted herein by $\mathrm{vec}(\mathbf{A})$, returns a vector composed of the concatenated columns of $\mathbf{A}$. The column-wise concatenation of vectors $\mathbf{x}_1,\ldots,\mathbf{x}_N$ is denoted by $\mathrm{col}(\mathbf{x}_1,\ldots,\mathbf{x}_N)$ and $\mathrm{diag}(\mathbf{A}_1,...,\mathbf{A}_N)$ denotes the square block diagonal matrix whose diagonal blocks are given by matrices $\mathbf{A}_1,...,\mathbf{A}_N$. The product of matrices $\mathbf{A}_N\ldots\mathbf{A}_1$ is denoted by $\prod_{i=1}^{N}\mathbf{A}_i$. Given a symmetric matrix $\mathbf{P}$, $\mathbf{P}\succ\mathbf{0}$ and $\mathbf{P}\succeq\mathbf{0}$ are used to point out that $\mathbf{P}$ is positive definite and positive semidefinite, respectively. The Kronecker product of two matrices $\mathbf{A}$ and $\mathbf{B}$ is denoted by $\mathbf{A}\otimes\mathbf{B}$.

II. Problem statement

The statement of the MHE framework for decentralized filter design for large-scale networks of interconnected systems employed in this paper is carried out in three steps. First, in Section II-A, a global model of a generic network of interconnected systems is presented, built on the concatenation of the dynamics of each individual system. Second, in Section II-B, the decentralized MHE problem is formulated for the global model, i.e. for the network as a whole, and the novel formulation put forward in this paper is compared with the state-of-the-art formulations, mainly regarding the origin of the performance improvement. Moreover, the nonconvex filter design optimization problem is stated, which is posteriorly relaxed in Section III to devise the MFH method. Third, in Section II-C, the novel framework is analyzed as far as communication and computational requirements are concerned. A flowchart of the decentralized implementation is also included.

A. Model of a network of interconnected systems

Consider a network of N interconnected systems, $\mathcal{S}_i$, each associated with one computing unit $\mathcal{T}_i$, with $i=1,\ldots,N$. The topology of the network, which is assumed to be time invariant, is defined by the dynamic couplings between systems. Such dynamic coupling topology may be represented by a directed graph, or digraph, $\mathcal{G}:=(\mathcal{V},\mathcal{E})$, composed of a set $\mathcal{V}$ of vertices and a set $\mathcal{E}$ of directed edges. An edge e incident on vertices i and j, directed from j towards i, is denoted by $e=(j,i)$. For a vertex i, its in-degree, ν_i^-, is the number of edges directed towards it, and its in-neighborhood, $\mathcal{D}_i^-$, is the set of indices of the vertices from which such edges originate. Conversely, for a vertex i, its out-degree, ν_i^+, is the number of edges directed from it, and its out-neighborhood, $\mathcal{D}_i^+$, is the set of indices of the vertices towards which such edges are directed. For a more detailed overview of the elements of graph theory used to model this network, see [36]. In this framework, each system is represented by a vertex, i.e. system $\mathcal{S}_i$ is represented by node i, and if the dynamics of $\mathcal{S}_i$ depend on the dynamics of system $\mathcal{S}_j$, then this coupling is represented by an edge directed from vertex j towards vertex i, i.e. edge $e=(j,i)$. The case for which there is a measurement coupling, instead of (or in addition to) a dynamic coupling, is modeled similarly. In fact, the MHE framework for this case is analogous to the one put forward in this paper and, thus, it can be extended to accommodate it.

The dynamics of system $\mathcal{S}_i$ are modeled by the following discrete-time LTI system

$$\begin{cases}\mathbf{x}_i(k+1)=\mathbf{A}_{i,i}\mathbf{x}_i(k)+\mathbf{B}_{i,i}\mathbf{u}_i(k)+\\ \qquad\sum_{j\in\mathcal{D}_i^-}\left(\mathbf{A}_{i,j}\mathbf{x}_j(k)+\mathbf{B}_{i,j}\mathbf{u}_j(k)\right)+\mathbf{w}_{i,\mathcal{D}_i^-}(k)\\ \mathbf{y}_i(k)=\mathbf{C}_i\mathbf{x}_i(k)+\mathbf{v}_i(k)\end{cases}, \quad (1)$$

where $\mathbf{x}_i(k)\in\mathbb{R}^{n_i}$ is the state vector, $\mathbf{u}_i(t)\in\mathbb{R}^{m_i}$ is the input vector, which is assumed to be known, and $\mathbf{y}_i(t)\in\mathbb{R}^{o_i}$ is the output vector, all of system $\mathcal{S}_i$; matrices $\mathbf{A}_{i,j}\in\mathbb{R}^{n_i\times n_j}$, $\mathbf{B}_{i,j}\in\mathbb{R}^{n_i\times m_j}$, and $\mathbf{C}_i\in\mathbb{R}^{o_i\times n_i}$ are known constant matrices that model the dynamics of system $\mathcal{S}_i$ and its coupling with the other systems in its in-neighborhood; vectors $\mathbf{v}_i(k)\in\mathbb{R}^{o_i}$ and $\mathbf{w}_{i,\mathcal{D}_i^-}(k)\in\mathbb{R}^{n_i}$ are the observation and process noise, modeled as zero-mean white Gaussian processes. The global dynamics of the network are, then, modeled by the following discrete-time LTI system

$$\begin{cases}\mathbf{x}(k+1)=\mathbf{A}\mathbf{x}(k)+\mathbf{B}\mathbf{u}(k)+\mathbf{w}(k)\\ \mathbf{y}(k)=\mathbf{C}\mathbf{x}(k)+\mathbf{v}(k)\end{cases}, \quad (2)$$

where $\mathbf{x}(k):=\mathrm{col}(\mathbf{x}_1(k),\ldots,\mathbf{x}_N(k))\in\mathbb{R}^n$ is the global state vector, $\mathbf{u}(k):=\mathrm{col}(\mathbf{u}_1(k),\ldots,\mathbf{u}_N(k))\in\mathbb{R}^m$ is the global input vector, and $\mathbf{y}(k):=\mathrm{col}(\mathbf{y}_1(k),\ldots,\mathbf{y}_N(k))\in\mathbb{R}^o$ is the global output vector; vector $\mathbf{v}(k):=\mathrm{col}(\mathbf{v}_1(k),\ldots,\mathbf{v}_N(k))$ is the global observation noise; $\mathbf{w}(k):=\mathrm{col}(\mathbf{w}_{1,\mathcal{D}_1^-}(k),\ldots,\mathbf{w}_{N,\mathcal{D}_N^-}(k))$ is the global process noise. Matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are time invariant matrices of proper dimensions that model the global dynamics. Similarly, $\mathbf{Q}$ and $\mathbf{R}$ are covariance block matrices that model the zero-mean white Gaussian global process and sensor noise, respectively. For more details on the definition of these global matrices, see [37, Section 2.1]. Note that some of the block entries of matrices $\mathbf{Q},\mathbf{A}$, and $\mathbf{B}$ may be null due of the nonexistence of dynamic couplings between every pair of systems. However, it is important to stress that no assumption was made regarding either the sparsity or the structure of matrices $\mathbf{Q}, \mathbf{A}$, and $\mathbf{B}$, as it is imposed for some state-of-the-art MHE formulations. It is considered that the initial network state estimate, $\hat{\mathbf{x}}(0)$, is a random vector described by a Gaussian distribution with expected value $\mathbf{x}(0)$ and covariance $\mathbf{P_0} \succeq \mathbf{0} \in \mathbb{R}^{n\times n}$.

B. Decentralized MHE formulation

Exploiting the intricacies of the decentralized estimation problem, the framework proposed herein allows for significant performance improvement in comparison with the state-of-the-art formulation of decentralized recursive filters. To comprehend the conception and the advantages of using the proposed MHE formulation, it is important to stress a significant distinction between the centralized and decentralized formulations of the estimation problem. On one hand, the solution to the centralized estimation problem,

$$\underset{\substack{\mathbf{K}(i)\in\mathbb{R}^{n\times o}\\ i\in\mathbb{N}}}{\text{minimize}} \quad \lim_{T\to\infty} \frac{1}{T}\sum_{k=1}^{T} \operatorname{tr}(\mathbf{P}(k|k)) \tag{3}$$

is equivalent to the solution to an infinite number of independent optimization problems [35]

$$\underset{\mathbf{K}(k)\in\mathbb{R}^{n\times o}}{\text{minimize}} \quad \operatorname{tr}(\mathbf{P}(k|k))$$

for $k \in \mathbb{N}$, where $\mathbf{P}(k|k)$ is the estimation error covariance matrix and $\mathbf{K}(k)$ the gain of the Luenberger formulation of the Kalman filter. In other words, for a given time instant k, the corresponding estimate can be obtained using a gain designed to achieve the minimum $\operatorname{tr}(\mathbf{P}(k|k))$ possible without compromising the performance of the estimate at the following time instants. This result is, by no means, evident, but it is of great significance and convenience, since it originates a recursive linear filter for the centralized solution. On the other hand, this property is not satisfied in a decentralized setting. Take, as an example, the one-step and finite-horizon methods presented and compared in [12]. The one-step method, which minimizes $\operatorname{tr}(\mathbf{P}(k|k))$ independently for each time instant, underperforms in comparison with the finite-horizon method, which minimizes the sum of the trace of the estimation error covariance over a finite window. For this reason, it is the case that, for decentralized configurations, it is not possible, using a Luenberger formulation, to make use of all the potential of the available information for the computation of each state estimate, since it compromises the performance of future time steps.

The novel MHE framework proposed in this paper aims to compute a state estimate for each time instant resorting to a past window of measurements and local communication that satisfies the aforementioned guidelines. The scheme that is proposed is achieved by employing, at each time instant, the prediction-filtering steps of a Luenberger filter to that window of past instants, taking solely into consideration the estimation performance at the end of the window. This framework allows to leverage and extend state-of-the-art convex relaxation techniques [12]. Consider the following augmented notation to ease the characterization of the prediction-filtering recurrence that occurs at each time instant k. Let $\hat{\mathbf{x}}_i(\tau+1|\tau|k)$ denote the local predicted state estimate at time instant $\tau+1$ as computed at time instant k and $\hat{\mathbf{x}}_i(\tau|\tau|k)$ denote the local filtered state estimate at time instant τ as computed at time instant k, both for system $\mathcal{S}_i$. For each time instant k, consider the finite window $\{k-W+1,\ldots,k\}$, where $W \in \mathbb{N}$ is the finite window length. For the system $\mathcal{S}_i$, an iteration of the proposed filter for time instant k follows

$$\begin{cases} \hat{\mathbf{x}}_i(k-W|k-W|k) = \hat{\mathbf{x}}_i(k-W|k-W|k-W) \\ \mathbf{u}_{i,\mathcal{D}_i^-}(\tau-1) = \mathbf{B}_{i,i}\mathbf{u}_i(\tau-1) + \sum_{j\in\mathcal{D}_i^-}\mathbf{B}_{i,j}\mathbf{u}_j(\tau-1) \\ \hat{\mathbf{x}}_i(\tau|\tau-1|k) = \mathbf{A}_{i,i}\hat{\mathbf{x}}_i(\tau-1|\tau-1|k) + \\ \qquad \sum_{j\in\mathcal{D}_i^-}(\mathbf{A}_{i,j}\hat{\mathbf{x}}_j(\tau-1|\tau-1|k)) + \mathbf{u}_{i,\mathcal{D}_i^-}(\tau-1) \\ \hat{\mathbf{x}}_i(\tau|\tau|k) = \hat{\mathbf{x}}_i(\tau|\tau-1|k) + \\ \qquad \mathbf{K}_i(\tau|k)\left(\mathbf{y}_i(\tau) - \mathbf{C}\hat{\mathbf{x}}_i(\tau|\tau-1|k)\right), \end{cases} \tag{4}$$

where $\mathbf{K}_i(\tau|k) \in \mathbb{R}^{n_i\times o_i}$ denotes the filter gain for time instant τ computed at time instant k. The various predicted state estimates depend on the state estimates of the systems in the in-neighborhood of $\mathcal{S}_i$, $\hat{\mathbf{x}}_j(\tau-1|\tau-1|k)$ with $\tau = k-W+1,\ldots,k$, which have to be received via a communication link W times each iteration. Moreover, $\mathbf{u}_{i,\mathcal{D}_i^-}(k-1)$ is computed at the beginning of each iteration, meaning that $\mathcal{T}_i$ must receive $\mathbf{u}_j(k-1)$ with $j \in \mathcal{D}_i^-$ once each iteration. Thus, $\mathbf{u}_{i,\mathcal{D}_i^-}(\tau-1)$, $\tau = k-W+1,\ldots,k$, has to be stored, as well as $\hat{\mathbf{x}}_i(\tau-1|\tau-1|k)$, $\tau = k-W,\ldots,k-1$. The filtering steps depend exclusively on sensor readings of system $\mathcal{S}_i$, $\mathbf{y}_i(\tau)$ with $\tau = k-W+1,\ldots,k$, which have also to be stored. Note that, because the system is linear and the proposed MHE filter dynamics (4) are based on linear prediction-filtering steps, the conditional probability density functions of the state remain Gaussian and are fully characterized by their mean and covariance. Thus, the proposed filter is consistent, i.e, it is zero-mean and the computed estimation error covariance is not over-confident [38]. Fig. 1 shows the communication and storage requirements for an arbitrary node i of an illustrative network with string configuration for the recurrence carried

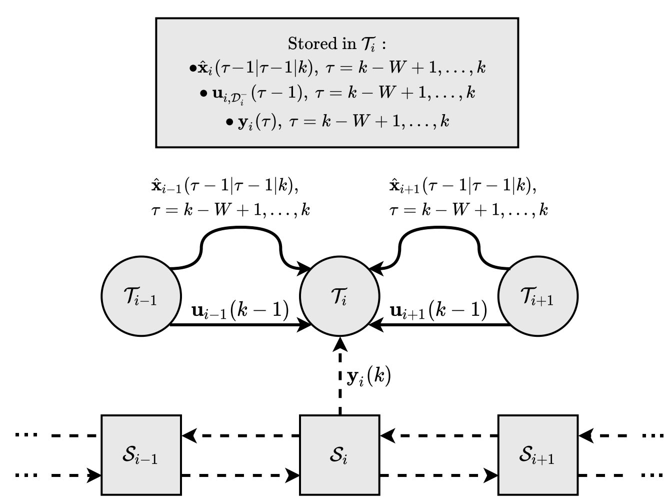

Fig. 1: Communication and storage requirements for an arbitrary computational unit $\mathcal{T}_i$ with string configuration. The solid arrows represent a communication link and the dashed arrows represent dynamic interaction.

out at time instant k.

The prediction-filtering steps of a single iteration of the Luenberger formulation of a Kalman filter can also be written for the global system, which yields

$$\begin{cases}\hat{\mathbf{x}}(k-W|k-W|k) = \hat{\mathbf{x}}(k-W|k-W|k-W)\\ \hat{\mathbf{x}}(\tau|\tau-1|k) = \mathbf{A}\hat{\mathbf{x}}(\tau-1|\tau-1|k) + \mathbf{B}\mathbf{u}(\tau-1)\\ \hat{\mathbf{x}}(\tau|\tau|k)=\hat{\mathbf{x}}(\tau|\tau-1|k)+\mathbf{K}(\tau|k)(\mathbf{y}(\tau)-\mathbf{C}\hat{\mathbf{x}}(\tau|\tau-1|k)),\end{cases} \tag{5}$$

where $\hat{\mathbf{x}}(\tau|\tau-1|k) := \mathrm{col}(\hat{\mathbf{x}}_1(\tau|\tau-1|k),\ldots,\hat{\mathbf{x}}_N(\tau|\tau-1|k))$, $\hat{\mathbf{x}}(\tau|\tau|k) := \mathrm{col}(\hat{\mathbf{x}}_1(\tau|\tau|k),\ldots,\hat{\mathbf{x}}_N(\tau|\tau|k))$, and $\mathbf{K}(\tau|k) \in \mathbb{R}^{n\times o}$ is the global filter gain for time instant τ computed at time instant k, with $\tau = k-W+1,\ldots,k$. Note that writing the multiple local state filters (4) together for the global system dynamics leads to a global gain of the form

$$\mathbf{K}(\tau|k) = \mathrm{diag}(\mathbf{K}_1(\tau|k),\ldots,\mathbf{K}_N(\tau|k)), \tag{6}$$

which means that the global gain is subject to a specific constraint. Such constraint falls in a broader category designated by sparsity constraints. Let matrix $\mathbf{E}\in\mathbb{R}^{n\times o}$ denote a sparsity pattern. The set of matrices which obey the sparsity constraint determined by $\mathbf{E}$ is defined as

$$\mathrm{Sparse}(\mathbf{E}) := \{[\mathbf{K}]_{ij}\in\mathbb{R}^{n\times o} : [\mathbf{E}_{ij}] = 0 \implies [\mathbf{K}]_{ij} = 0;\ i=1,...,n,\ j=1,...,o\}.$$

Thus, it follows from (6) that $\mathbf{K}(\tau|k)\in\mathrm{Sparse}(\mathbf{E})$, with $\mathbf{E} = \mathrm{diag}(\mathbf{1}_{n_1\times o_1},\ldots,\mathbf{1}_{n_N\times o_N})$. The MHE filter derived in this paper is designed to solve an optimization problem for the network as a whole, subject to an arbitrary time invariant sparsity constraint on the global filter gain and limited communication links between systems.

Let $\mathbf{P}(\tau+1|\tau|k)$ denote the global predicted estimation error covariance matrix at time instant $\tau+1$ as computed at time instant k and $\mathbf{P}(\tau|\tau|k)$ denote the global global filtered estimation error covariance matrix at time instant τ as computed at time instant k. The dynamics of the estimation error covariance matrix follow from (5), as given by

$$\begin{cases}\mathbf{P}(k-W|k-W|k) = \mathbf{P}(k-W|k-W|k-W)\\ \mathbf{P}(\tau|\tau-1|k) = \mathbf{A}\mathbf{P}(\tau-1|\tau-1|k)\mathbf{A}^T + \mathbf{Q}\\ \mathbf{P}(\tau|\tau|k) = \mathbf{K}(\tau|k)\mathbf{R}\mathbf{K}^T(\tau|k)+\\ \qquad(\mathbf{I}-\mathbf{K}(\tau|k)\mathbf{C})\mathbf{P}(\tau|\tau-1|k)(\mathbf{I}-\mathbf{K}(\tau|k)\mathbf{C})^T\end{cases}, \tag{7}$$

with $\tau = k-W+1,\ldots,k$, which is a recursive expression of prediction-filtering estimation error covariance steps of the Luenberger Kalman filter. The estimation problem in the MHE framework is analogous to (3). Furthermore, the aforementioned sparsity constrain (6) allows to formulate the decentralized problem globally explicitly considering the communication restrictions associated with the decentralized setting. For an infinite-horizon and a known and time invariant sparsity pattern $\mathbf{E}$, solve the optimization problem

$$\begin{array}{ll}\underset{\substack{\mathbf{K}(\tau|i)\in\mathbb{R}^{n\times o}\\ i\in\mathbb{N},\tau\in\{i-W+1,\ldots,i\}}}{\text{minimize}} & \lim_{T\to\infty}\dfrac{1}{T}\sum_{k=1}^{T}\mathrm{tr}(\mathbf{P}(k|k|k))\\ \text{subject to} & \mathbf{K}(\tau|i)\in\mathrm{Sparse}(\mathbf{E}),\ i\in\mathbb{N},\\ & \tau\in\{i-W+1,\ldots,i\}\quad\text{and}\quad(7).\end{array} \tag{8}$$

Remark 2.1: Note that, for each time step k, a sequence of gains is computed, instead of a single gain. This way, the current state estimate is dependent on the estimate of time instant $k-W$, instead of the previous state estimate. It is important to stress that, for each time instant k, we are exclusively concerned in minimizing the trace of the first computation of the estimation error covariance, i.e. $\mathbf{P}(k|k|k)$. It is the only relevant estimation error covariance, since it is the only one that is obtained in real-time and whose state estimate may be used in a feedback control law. In the computation of each state estimate $\hat{\mathbf{x}}(k|k|k)$ using (4), the state estimates that are computed in this recursion for the prior time instants, $\hat{\mathbf{x}}(\tau|\tau|k)$ with $\tau = k-W+1,\ldots,k-1$, may be as poor an estimate as necessary, as long as it allows for the best possible estimate $\hat{\mathbf{x}}(k|k|k)$. This is the reason for the performance improvement in comparison with the Luenberger formulation, since the previous state estimates do not compromise the quality of future estimates.

Remark 2.2: The MHE formulation arises from the necessity of making use of the full potential of the sensor information that has been received. In that sense, ideally, selecting a variable window size $W = k$ would yield the best performance possible, since all the sensor information up to time instant k would be taken into account. It is evident that it is unfeasible to make $W\to\infty$ as $k\to\infty$, due to the increasing computational load and communication requirements as W becomes large. Furthermore, the sensor information prior to time instant $k-W+1$ is used to compute $\hat{\mathbf{x}}(k-W|k-W|k-W)$ and $\mathbf{P}(k-W|k-W|k-W)$ and as the sensor information received gets older it becomes less relevant to the estimate of the current state. For this reason, a large enough steady-state constant window length W_{ss}, which offers a compromise between performance and computational load, is considered.

Notwithstanding, the goal of this paper is to seek a steady-state solution to (8), i.e. to find a constant sequence of global gains $\mathbf{K}_\infty(\tau)$, $\tau = 1,...,W_{ss}$, instead of a single constant gain. If such sequence stabilizes the error dynamics of the filter (4), then the estimation error covariance converges to a steady-state solution $\mathbf{P}_\infty\in\mathbb{R}^{n\times n}$. The steady-state equivalent of the optimization problem (8) is then written as

$$\begin{array}{ll}\underset{\substack{\mathbf{K}_\infty(\tau)\in\mathbb{R}^{n\times o},\\ \tau=1,...,W_{ss}}}{\text{minimize}} & \mathrm{tr}(\mathbf{P}_\infty)\\ \text{subject to} & \mathbf{K}_\infty(\tau)\in\mathrm{Sparse}(\mathbf{E}),\ \tau=1,...,W_{ss}\\ & \text{and}\quad(7)\end{array} \tag{9}$$

and the local filter dynamics as

$$\begin{cases}\hat{\mathbf{x}}_i(k-W_{ss}|k-W_{ss}|k)=\hat{\mathbf{x}}_i(k-W_{ss}|k-W_{ss}|k-W_{ss})\\ \mathbf{u}_{i,\mathcal{D}_i^-}(\tau-1) = \mathbf{B}_{i,i}\mathbf{u}_i(\tau-1) + \sum_{j\in\mathcal{D}_i^-}\mathbf{B}_{i,j}\mathbf{u}_j(\tau-1)\\ \hat{\mathbf{x}}_i(\tau|\tau-1|k) = \mathbf{A}_{i,i}\hat{\mathbf{x}}_i(\tau-1|\tau-1|k)+\\ \qquad\sum_{j\in\mathcal{D}_i^-}(\mathbf{A}_{i,j}\hat{\mathbf{x}}_j(\tau-1|\tau-1|k)) + \mathbf{u}_{i,\mathcal{D}_i^-}(\tau-1)\\ \hat{\mathbf{x}}_i(\tau|\tau|k) = \hat{\mathbf{x}}_i(\tau|\tau-1|k)+\\ \qquad\mathbf{K}_{\infty_i}(\tau-k+W_{ss})\left(\mathbf{y}_i(\tau)-\mathbf{C}\hat{\mathbf{x}}_i(\tau|\tau-1|k)\right),\end{cases} \tag{10}$$

where

$$\mathbf{K}_\infty(\tau) = \mathrm{diag}(\mathbf{K}_{\infty_1}(\tau),\ldots,\mathbf{K}_{\infty_N}(\tau)), \tag{11}$$

with $\tau = k-W_{ss}+1,\dots,k$. In the next section, the MFH method is put forward to design a steady-state sequence of global gains $\mathbf{K}_\infty(\tau), \tau = k-W_{ss}+1,\dots,k$. After the proposed offline synthesis, the local decentralized gains $\mathbf{K}_{\infty_i}(\tau), \tau = k-W_{ss}+1,\dots,k$ of the local filter dynamics (10) are extracted from the globally synthesized sparse gain matrix according to (11). Therefore, the proposed estimation solution can be deployed in the network, according to the local filter dynamics (10), leveraging local communication exclusively.

Thus, for a given constant sequence of global gains $\mathbf{K}_\infty(\tau)$, $\tau = 1,...,W_{ss}$, the decentralized filter (10) is stable if $\prod_{\tau=1}^{W_{ss}}((\mathbf{I}-\mathbf{K}_\infty(\tau)\mathbf{C})\,\mathbf{A})$ is Shur. As a result, a sequence of gains can be synthesized offline using, for instance, the MFH method proposed in this paper, and its stability can be easily verified using this criterion before it is deployed over the network.

C. Communication and computational requirements

The novel discrete-time MHE framework for decentralized filter design for large-scale networks of interconnected LTI systems is given as a solution to the optimization problem (9), whose local filter dynamics follow (10). Before setting out to solve the actual optimization problem, it is important to stress the properties of this novel formulation as far as computational power and communication requirements are concerned.

First, note that each computational unit $\mathcal{T}_i$ associated with system $\mathcal{S}_i$ has to compute (10) every new iteration of the estimation algorithm. It consists of W_{ss} Kalman filter prediction-filtering steps for the dynamics of $\mathcal{S}_i$, which is of reduced dimension no matter the size of the network. For this reason, the computational load on $\mathcal{T}_i$ is low, independently of the scale of the network, since the load of the global estimation algorithm is distributed across all computational units and carried out in a parallel manner. Second, note that the optimization problem (9) can be solved offline, requiring that the model of the network dynamics is known beforehand. Although the time required to find its solution increases with N, because it is formulated for the network as a whole, it can be run offline in a computing server. Once the constant sequence of global gains $\mathbf{K}_\infty(\tau)$, $\tau = 1,...,W_{ss}$, is found, the local sequence of gains, according to (11), are available. Hence, they can be uploaded to each of the computational units. Third, besides the system dynamics matrices and local sequence of gains, each computational unit is only required to store in memory $\mathbf{u}_{i,\mathcal{D}_i^-}(\tau-1)$, $\tau = k-W_{ss}+1,\dots,k$, $\hat{\mathbf{x}}_i(\tau-1|\tau-1|k)$, $\tau = k-W_{ss},\dots,k-1$, and $\mathbf{y}_i(\tau)$, $\tau = k-W_{ss}+1,\dots,k$, which are of low dimension. Note that there is no replication of computations or of data stored across computational units, which contributes to the efficiency of the algorithm as far as computational power and memory resources are concerned. Not only does this design allow for very fast computations of state estimates, but it is also suitable for the application to large-scale networks, requiring only cheap microcontrollers as computational units.

The local filter dynamics (10) shows that computational unit $\mathcal{T}_i$ has to receive data from every computational unit in the in-neighborhood of $\mathcal{S}_i$, i.e. $\mathcal{T}_j$, $j \in \mathcal{D}_i^-$. Thus, the communication network can be also represented by a directed graph, whose nodes are the computational units of each system and the communication links are the same directed edges as the dynamic coupling graph $\mathcal{G}$. It is important to stress that the only required communication links are between systems with dynamic interaction, which, for the vast majority of applications to large-scale networks, are in close proximity to one another. For this reason, it is generally easy to establish a wire connection for these communication links, which allows for fast data transfer. In fact, for every iteration k, each computational unit $\mathcal{T}_i$ has to receive, at the beginning of each iteration, the control input of the previous time instant of the systems in its in-neighborhood, i.e. $\mathbf{u}_j(k-1)$, $j \in \mathcal{D}_i^-$. Furthermore, (10) shows that each computational unit has to receive updated state estimates of the systems in its in-neighborhood computed at time instant k, i.e. $\hat{\mathbf{x}}_j(\tau-1|\tau-1|k)$, $\tau = k-W_{ss}+1,\dots,k$, $j \in \mathcal{D}_i^-$. It amounts to W_{ss} communications for each iteration, which have to be carried out synchronously, i.e. every computational unit in the network must undergo alternated prediction-filtering steps and updated state estimate exchanges synchronously. Although it may seem difficult to achieve in practice, it is important to stress that: i) the amount of information transmitted each step is very reduced, unlike the state-of-the-art MHE designs; ii) the communication links are restricted to pairs of systems with dynamic coupling, which eases the implementation of a fast and reliable data connection; and iii) the value of W_{ss} required to notice a significant improvement in the performance of the estimation over state-of-the-art designs is very low. Recall Fig. 1, which depicts the communication and storage requirements for an arbitrary node i of an illustrative large-scale network with string configuration. Fig. 2 shows the flowchart of the decentralized estimation algorithm implemented in $\mathcal{T}_i$ according to the proposed MHE framework.

III. Moving finite-horizon method

Because of the sparsity constraint, the optimization problems (8) and (9) are nonconvex and finding their optimal solution is still an open problem. To overcome this difficulty, the optimization problem may be relaxed so that it becomes convex, allowing for the use of well known optimization techniques. Albeit optimal for the modified problem, the relaxed solution is only an approximation to the solution of the original problem, thus careful relaxation is necessary to ensure that the separation between both solutions is minimal. This approach is designated convex relaxation [39] and it has been successfully employed in control theory [40], [41]. The goal is to find a sequence of steady-state gains that solves (9). Nevertheless, the MFH method is designed to solve the optimization problem (8) with time varying sequences of gains as $k\to\infty$. If the sequence of gains and estimation error covariance converges, then it approximates the behavior of the steady-state problem, consisting of an approximate solution to (9).

Note that the objective function of (8) is the sum, over an infinite window, of variances of the estimation error at time instant k, i.e., $\hat{\mathbf{x}}(k|k|k)-\mathbf{x}(k)$. Each of these state estimates

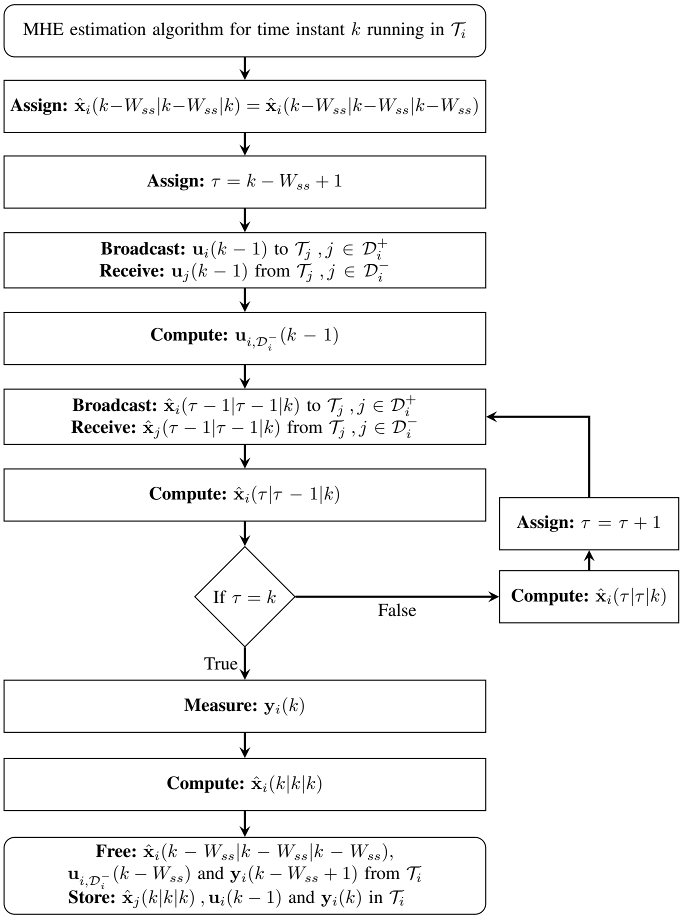

Fig. 2: Flowchart of the decentralized estimation algorithm implemented in $\mathcal{T}_i$ according to the proposed MHE framework.

is computed making use of W prediction-filtering steps in (4), which are initialized with the estimate $\hat{\mathbf{x}}(k-W|k-W|k-W)$. As the window length, W, increases, an increasing number of outputs is considered in the prediction-filtering steps, which results in a lower dependence of $\hat{\mathbf{x}}(k|k|k)$ on $\hat{\mathbf{x}}(k-W|k-W|k-W)$. This is the principle behind the first relaxation of (8), which consists in optimizing the accuracy of each state estimate $\hat{\mathbf{x}}(k|k|k)$ without taking into account its effect in the performance of the future estimates $\hat{\mathbf{x}}(k+nW|k+nW|k+nW)$, with $n \in \mathbb{N}$. According to this relaxation, (8) can be approximated by multiple independent optimization problems

$$\begin{array}{ll} \underset{\substack{\mathbf{K}(\tau|k)\in\mathbb{R}^{n\times o},\\ \tau\in\{k-W+1,\ldots,k\}}}{\text{minimize}} & \operatorname{tr}(\mathbf{P}(k|k|k)) \\ \text{subject to} & \mathbf{K}(\tau|k)\in\operatorname{Sparse}(\mathbf{E}), \\ & \tau \in \{k-W+1,\ldots,k\}, \quad \text{and} \quad (7) \end{array} \tag{12}$$

$k \in \mathbb{N}$. Notice that in the limit scenario of a varying maximum window length $W = k$, the relaxation above is exact. However, expanding the objective function of (12) as a function of the estimation error covariance boundary condition, i.e. $\mathbf{P}(k-W|k-W|k-W)$, yields a nonconvex polynomial expression on $\mathbf{K}(\tau|k)$, $\tau = k-W+1,\ldots,k$. For this reason, the optimization problem (12) is still nonconvex. Therefore, to make use of well-known optimization techniques, a second and last relaxation step is employed. Instead of minimizing the sequence of gains $\mathbf{K}(\tau|k)$, $\tau = k-W+1,\ldots,k$ as a whole as in (12), one can iteratively minimize each gain of the sequence individually, while taking into account its effect on the whole finite window. Following this relaxation principle, the optimization problem (12) can be relaxed by iterating on solutions to

$$\begin{array}{ll} \underset{\mathbf{K}(\tau|k)\in\mathbb{R}^{n\times o}}{\text{minimize}} & \operatorname{tr}(\mathbf{P}(k|k|k)) \\ \text{subject to} & \mathbf{K}(\tau|k) \in \operatorname{Sparse}(\mathbf{E}) \quad \text{and} \quad (7)\,, \end{array} \tag{13}$$

$k \in \mathbb{N}$ and $\tau \in \{k-W+1,\ldots,k\}$. Expanding the objective function of the optimization problem above, one readily concludes, after some algebraic manipulation using (7), that, for each k and each τ, it is quadratic in relation to $\mathbf{K}(\tau|k)$. Given that the sparsity constraint is also convex, not only is the relaxed optimization problem (13) convex for each k and each τ, but it also has a closed-form solution, as detailed in the following result.

Theorem 3.1: Define a matrix $\mathbf{Z}$ such that the vector $\mathbf{Z}\operatorname{vec}(\mathbf{K}(\tau|k))$ contains the non-zero entries of $\mathbf{K}(\tau|k)$ according to the desired sparsity pattern. The closed-form solution of (13) is given by

$$\begin{aligned} \operatorname{vec}(\mathbf{K}(\tau|k)) = \mathbf{Z}^T \left(\mathbf{Z}(\mathbf{S}(\tau|k)\otimes\mathbf{\Psi}(\tau|k))\mathbf{Z}^T\right)^{-1}\mathbf{Z} \\ \operatorname{vec}\left(\mathbf{\Psi}(\tau|k)\mathbf{P}(\tau|\tau-1|k)\mathbf{C}^T\right), \end{aligned} \tag{14}$$

where $\mathbf{S}(\tau|k)$ is the innovation covariance at time instant τ computed at time instant k, given by

$$\mathbf{S}(\tau|k) = \mathbf{C}\mathbf{P}(\tau|\tau-1|k)\mathbf{C}^T + \mathbf{R}$$

and

$$\mathbf{\Psi}(\tau|k) = \mathbf{\Gamma}^T(\tau+1,k)\mathbf{\Gamma}(\tau+1,k)\,, \tag{15}$$

with

$$\mathbf{\Gamma}(k_i,k_f) = \prod_{j=k_i}^{k_f}(\mathbf{I}_n - \mathbf{K}(j|k_f)\mathbf{C})\mathbf{A}\,, \tag{16}$$

for $k_i \le k_f$ and $\mathbf{\Gamma}(k_i,k_f) = \mathbf{I}_n$ for $k_i > k_f$.

Proof: See the Appendix. ■

For an example on how to compute matrix $\mathbf{Z}$ for a given sparsity pattern, see [12, Section 5]. For a given time instant k, each time a gain $\mathbf{K}(\tau|k)$ is modified, the sequence of error covariance matrices needs to be updated, which can be computationally expensive. Nevertheless, analysing the closed-form solution for the computation of $\mathbf{K}(\tau|k)$, given by (14), one readily notices it only makes use of the error covariance of instants up to τ. For this reason, the gains can be computed in reverse order, i.e. from the last time step of the window to the first, updating the covariances when all the gains of the window have already been computed. Repeating this process, that is, taking turns computing the sequence of gains backwards in time and recomputing the covariance matrices forward in time, the sequence of gains converges to a near-optimal solution of the optimization problem (12). This subroutine is presented in Table I.

Remark 3.1: As pointed out in Table I, the algorithm for the computation of the sequences of gains of the MFH method requires a sequence of gains for its initialization. There are

TABLE I: Subroutine for the computation of a sequence of gains for the MFH method for an arbitrary time instant k.

1) **Inputs:**
   a) k, current time instant;
   b) W, window length;
   c) ϵ, minimum relative improvement on the objective function of the optimization problem (12).
2) **Initialization:**
   a) Set an initial sequence of filter gains $\mathbf{K}(\tau|k)$, $\tau = k - W + 1, ..., k$, using the sequence computed in the previous time instant if $k > W$ or the one-step method otherwise. See Remark 3.1 for more details on the initialization.
   b) Compute the resulting initialization estimation error covariance matrices $\mathbf{P}(\tau|\tau|k)$, $\tau = k - W + 1, ..., k$.
3) **Do:**
   a) **For:** $\tau = k, ..., k - W + 1$
      i) Recompute $\mathbf{K}(\tau|k)$ using (14).
   b) Recompute the estimation error covariance matrices $\mathbf{P}(\tau|\tau|k)$, $\tau = k - W + 1, ..., k$, using (7).

   While: relative improvement on $\mathrm{tr}(\mathbf{P}(k|k|k))$ relative to the previous outer loop iteration is greater than ϵ.
4) **Return:** the sequence of gains $\mathbf{K}(\tau|k)$ and corresponding estimation error covariance matrices $\mathbf{P}(\tau|\tau|k)$, for $\tau = k - W + 1, ..., k$.

plenty of approaches that can be followed for this. Since it is not necessary that these initial gains stabilize the filter error dynamics, null gains may be used for the initialization. However, for unstable systems, its use may lead to numerical problems, especially for large window lengths, since the corresponding initial covariance grows unbounded with time. A second approach is to initialize the method with the sequence of centralized gains, which can be computed very rapidly. In fact, the use of (14) guarantees that, after the first iteration, all the gains follow the given sparsity pattern. A third alternative is to initialize the algorithm with a stabilizing sequence of sparse gains provided by a fast method such as the one-step method, put forward in [12, Section 4]. A fourth approach is to use the sequence of gains computed for the previous time step. In fact, if the sequence of gains, for each time instant k, converges to a constant sequence as $k \to \infty$, then using this initialization, the higher k is, the lesser is the number of required iterations for the convergence of each sequence of gains. For this reason, for time instants $k > W_{ss}$ (equivalently $k > W$), this is the initialization that should be used to allow for a significant reduction in computational load. For the initialization period, i.e. $k \leq W_{ss}$ (equivalently $k = W$), one of the first three approaches that were presented may be used, as the previously computed sequence of gains has a different length.

The algorithm proposed in this paper to compute a steady-state sequence of gains following a MHE framework is outlined in Table II. It returns an approximation, possibly suboptimal, to the solution of the optimization problem (9) as the limit, within a relative separation whose order of magnitude is set by ϵ_∞, of an approximate, possibly sub-optimal, solution to (8) as $k \to \infty$. Note that, similarly to the Luenberger formulation, the steady-state sequence of gains can, and should, be computed offline.

For the application of the MFH algorithm to a particular system, four parameters need to be selected: i) the initial estimation error covariance matrix $\mathbf{P}(0|0|0)$; ii) the steady-state window length, W_{ss}; iii) the minimum relative improvement on the objective function of the steady-state optimization problem (9), ϵ_∞; and iv) the minimum relative improvement on the objective function of the optimization problem (12), ϵ. The MFH algorithm may reach different local minima of the steady-state optimization problem (9) for different $\mathbf{P}(0|0|0)$. For that reason, for a particular steady-state window length W_{ss}, one aims to find one such initialization matrix $\mathbf{P}(0|0|0)$ which, ideally, converges to the global minimum. It is important to remark that the parameter $\mathbf{P}(0|0|0)$, used for the initialization of the MFH algorithm, is distinct from the estimation error covariance of the initial estimate, $\mathbf{P_0}$. In fact, if a given $\mathbf{P}(0|0|0)$ is selected for the gain synthesis and the MFH algorithm returns a sequence of steady-state gains that stabilizes the dynamics of the decentralized filter, then the estimation error covariance converges to a steady-state matrix $\mathbf{P}_\infty$, regardless of the initial estimation error covariance of the initial estimate, $\mathbf{P_0}$.

The selection of the parameters $\mathbf{P}(0|0|0)$ and W_{ss} can be conducted in one of three ways. First, if there is considerable computational power available, as the gain computation is carried out offline, it is possible to iterate through each window size value up to a maximum value, applying the algorithm to each of them for a large number of randomly generated initial estimation error covariance matrices. Then, the pair of parameters which yields a good compromise between performance and window length is selected. Second, one may select, in a first instance, the steady-state window size which allows for a good compromise between performance and window length for $\mathbf{P}(0|0|0) = \mathbf{0}_{n\times n}$. Then, for the previously selected window length, the MFH algorithm is run for a large number of randomly generated initial estimation error covariance matrices, the best of which is selected. This approach requires significantly lower computational power, which is particularly convenient for the application to large-scale systems. Third, although suboptimal, the performance obtained for $\mathbf{P}(0|0|0) = \mathbf{0}_{n\times n}$ and the respective steady-state

TABLE II: MFH algorithm for the computation of a steady-state sequence of gains.

1) **Initialization:**
   a) Select an initial covariance $\mathbf{P}(0|0|0) \succeq \mathbf{0}$.
   b) Select a steady-state window length W_{ss}.
   c) Select stopping criteria: i) ϵ_∞, the minimum relative improvement on the objective function of the steady-state optimization problem (9) and ii) $\epsilon \leq \epsilon_\infty/10$, the minimum relative improvement on the objective function of the optimization problem (12) for the computation of each sequence of gains.
   d) $k = 0$;
2) **Do:**
   a) $k = k+1$
   b) $W = \min(k, W_{ss})$
   c) **Call:** the algorithm outlined in Table I with input arguments $(k,\ W,\ \epsilon)$ to compute the sequence of gains $\mathbf{K}(\tau|k)$, and corresponding estimation error covariance matrices $\mathbf{P}(\tau|\tau|k)$, $\tau = k-W+1,\ldots,k$.

   While: $k \leq W_{ss}$ **or** relative improvement on $\mathrm{tr}(\mathbf{P}(k|k|k))$ relative to $\mathrm{tr}(\mathbf{P}(k-1|k-1|k-1))$ is greater than ϵ_∞.
3) **Return:** the steady-state sequence of gains $\mathbf{K}(\tau|k)$, for $\tau = k - W_{ss} + 1, ..., k$ and the corresponding steady-state estimation error covariance matrix $\mathbf{P}(k|k|k)$.

window size yields identical performance to the one obtained using the previous two approaches. Thus, a null initialization can be simply used instead to choose the window length and synthesize the sequence of gains. It is important to note that all approaches are easily implemented in a parallel framework across multiple cores. The parameter ϵ_∞ is adjusted according to the desired precision of the solution. Note that, depending on the rate of convergence of the solution, if a precision of $\epsilon^\star$ is required, it may be necessary to set ϵ_∞ some orders of magnitude below $\epsilon^\star$. Lastly, it is important to note that, since the MFH algorithm depends on the output of the subroutine for the computation of each sequence of gains, it is necessary that its precision is, at least, one order of magnitude higher than the precision selected for the steady-state problem solution approximation. For this reason, one should select ϵ such that $\epsilon \leq \epsilon_\infty/10$.

Remark 3.2: It is also worth noticing that a filter designed to use a steady-state sequence of gains, as the steady-state MHE filter presented in this paper, is not self-starting. Such design can only be used starting from time instant $k = W_{ss}$. The state estimates for time instant prior to $k = W_{ss}$ ought to be computed using a Luenberger filter whose steady-state gain is computed using one of the state-of-the-art algorithms.

Remark 3.3: The iterative gain computation procedure for the MFH method is based on the closed-form solution (14), which has a computational complexity of $\mathcal{O}(|\chi|(no)^2)$, where $|\chi|$ denotes the number of nonzero entries of $\mathbf{E}$. Instead of using it, the exact numeric algorithm proposed in [42] can be, alternatively, used to compute each iteration with a computational complexity of $\mathcal{O}(|\chi|^3)$. Usually, in decentralized control applications, $|\chi|$ is given by $|\chi| \approx cn$, where $c \in \mathbb{N}$ is a constant. It, thus, follows that a computational complexity of $\mathcal{O}(n^3)$ is achieved for each iteration, which is equal to the one of the centralized solution. An efficient MATLAB implementation of the MFH method can be carried out using the *DECENTER* toolbox. See https://decenter2021.github.io/documentation/MHEMovingFiniteHorizonLTI for more details.

IV. Simulation results

Extensive numerical simulations were carried out to assess the performance of the proposed MHE framework and of the MFH method. These are presented concisely in this section due to space constraints. The unabridged simulations for several systems are available at the aforementioned open-source repository, which can be easily reproduced and adapted. The performance of the solution proposed in this paper is compared with the centralized (C) solution; state-of-the-art decentralized Luenberger filters: i) one-step (OS) method [12, Section 4], and ii) finite-horizon (FH) algorithm [12, Section 5]; and the distributed MHE method PMHE1 put forward in [28], which is a state-of-the-art distributed MHE solution whose communication, computational, and memory requirements are of the same magnitude as those of the solution proposed in this work.

A large-scale network of $N = 500$ systems was randomly generated, whose interconnection configuration is represented by the digraph depicted in Fig. 3, with $n_i = 2$ and $o_i = 1$, $i = 1, \ldots, 500$, and considering a fully decentralized configuration with $\mathbf{E} = \mathrm{diag}(\mathbf{1}_{n_1 \times o_1}, \ldots, \mathbf{1}_{n_N \times o_N})$. The OS method was synthesized with $\epsilon = 10^{-5}$. The FH algorithm gain was synthesized with $\epsilon = 10^{-2}$ and $W = 100$. Often the FH algorithm does not converge to a steady-state gain, but this synthetic network was chosen so that it is not the case. An example of this behavior can be seen in the extensive simulations in the aforementioned open-source repository, for which the MFH method still converges. The MFH method gain sequence was synthesized with $\epsilon_\infty = 10^{-4}$, $\epsilon = \epsilon_\infty/10$, and using a null estimation error covariance matrix as initialization. Table III depicts the projected performance of the MFH method for several steady-state window sizes normalized by the centralized performance. The value $W_{ss} = 5$ was chosen. The PMHE1 method is not guaranteed to converge for a window size equal to the MFH method in this network, according to [28, Theorem 1]. In the unabridged simulations in the repository, it is shown that its error dynamics are unstable in this synthetic network. The PMHE1 method disregards the

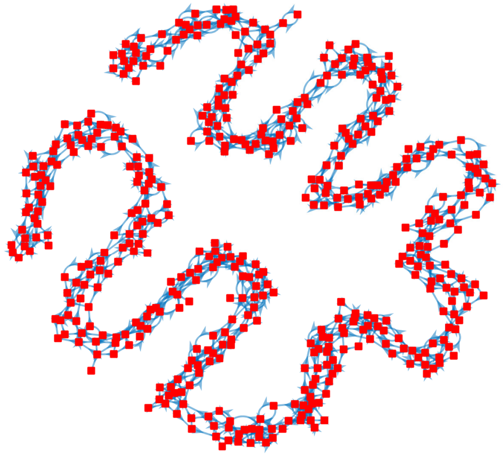

Fig. 3: Digraph of the interconnection configuration of the second large-scale network considered in the simulations.

TABLE III: Steady-state normalized projected performance comparison of the MFH method for different steady-state window sizes W_{ss}.

W_{ss}	2	3	4	5	6	7
$\mathrm{tr}(\mathbf{P}_\infty)/\mathrm{tr}(\mathbf{P}_\infty^{\mathrm{Cent}})$	1.911	1.764	1.647	1.590	1.555	1.520

uncertainty associated with the dynamic couplings between systems to allow for an implementation that resorts to local communication exclusively. Thus, unless couplings are very weak, it often leads to instability. Later, a second network with weaker couplings is considered.

Fig. 4 depicts the evolution of the trace of the estimation error covariance throughout the last iteration of the MFH algorithm, i.e. $\mathbf{P}(\tau|\tau|k)$, $\tau = k - W_{ss}, \ldots, k$, for the last time instant k for which a new sequence of gains is computed. First, given that the algorithm converges, the boundary estimation error covariance matrix $\mathbf{P}(k-W|k-W|k) = \mathbf{P}(k-W|k-W|k-W)$ and the one at the end of the finite window achieve the same performance. Second, recall Remark 2.1, in which the reason for the performance improvement of the proposed MFH algorithm over a Luenberger formulation is discussed. That reason is evident in this plot. In fact, given that we are exclusively concerned in minimizing $\mathbf{P}(k|k|k)$, the estimation error covariance matrices corresponding to the estimates that are computed in this recursion for the prior time instants, $\mathbf{P}(\tau|\tau|k)$ with $\tau = k-W+1, \ldots, k-1$, may achieve as poor a performance as necessary, as long as it allows for the best possible estimate $\hat{\mathbf{x}}(k|k|k)$. Analyzing Fig. 4, one notices that throughout the window, $\mathrm{tr}(\mathbf{P}(\tau|\tau|k))$ reaches four orders of magnitude above the steady-state value. This massive loss of performance throughout the window allows, nevertheless, for a very good state estimate at the end of the finite window, since the previous estimates do not compromise its performance as it is the case when the estimation problem is formulated in a Luenberger framework.

Fig. 5 depicts the evolution of the trace of the covariance of the estimation error for 1000 Monte Carlo simulations. The initial estimation error covariance of the simulations is set to $\mathbf{P_0} = \mathbf{0}_{5\times5}$, i.e. null initial estimation error, as a means of

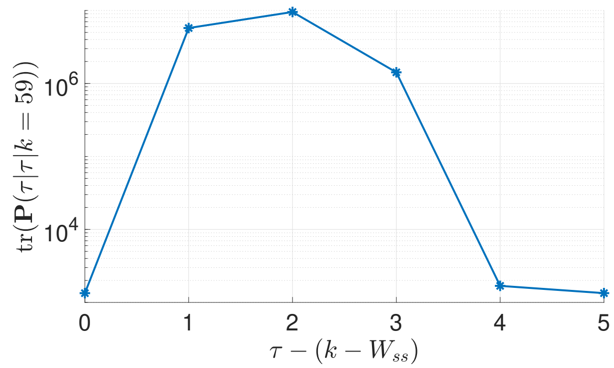

Fig. 4: Evolution of the trace of the estimation error covariance throughout the last iteration of the MFH algorithm.

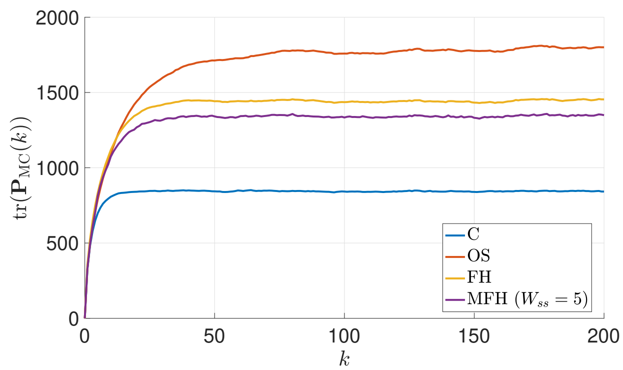

Fig. 5: Evolution of the trace of the estimation error covariance for 1000 Monte Carlo simulations.

contributing to the clarity of the plots. Table IV depicts the steady-state performance of the various decentralized methods obtained by averaging the trace of the covariance matrices obtained with the Monte Carlo simulations over the last 20 time instants, normalized by the projected centralized performance. Not only is it possible to conclude that the MFH method outperforms the remaining decentralized methods, but also that the use of the second best performing method, which is the FH algorithm, results in a penalty of 7.59% on the steady-state performance.

Another large-scale network of $N = 1000$ systems was randomly generated, with $n_i = 2$ and $o_i = 1$, $i = 1, \ldots, 1000$, and considering a fully decentralized configuration with $\mathbf{E} = \mathrm{diag}(\mathbf{1}_{n_1\times o_1}, \ldots, \mathbf{1}_{n_N\times o_N})$. The process noise of different systems is uncorrelated, so that the PMHE1 method can be employed. The OS method was synthesized with $\epsilon = 10^{-4}$. The FH synthesis could not be computed in a reasonable amount of time for this network. The MFH method gain was synthesized with $\epsilon_\infty = 10^{-4}$, $\epsilon = \epsilon_\infty/10$, $W_{ss} = 2$, and using

TABLE IV: Steady-state normalized performance comparison, obtained for 1000 Monte Carlo simulations.

	C	OS	FH	MFH ($W_{ss}=5$)
$\mathrm{tr}(\mathbf{P}_{\infty,\mathrm{MC}})/\mathrm{tr}(\mathbf{P}_\infty^{\mathrm{Cent}})$	1.001	2.129	1.719	1.596
Relative to MFH	−37.3%	+33.3%	+7.59%	–

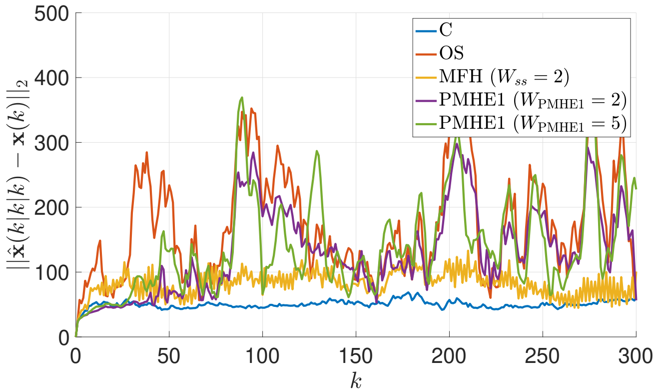

Fig. 6: Evolution of the norm of the estimation error.

TABLE V: Average estimation error norm for the different decentralized methods.

C	OS	MFH ($W_{ss} = 2$)	PMHE1 ($W_{\text{PMHE1}} = 2$)	PMHE1 ($W_{\text{PMHE1}} = 5$)
49.95	173.9	81.92	132.3	140.1

a null estimation error covariance matrix as initialization. The PMHE1 method is guaranteed to converge for a window size $W_{\text{PMHE1}} = 2$, according to [28, Theorem 1]. Alternative B for the PMHE1 methods is chosen according to [28, Section 1] because its requirements are more comparable to those of the MHE framework proposed in this work. Fig. 6 shows the evolution of the norm of the estimation error for these methods and Table V depicts the average error norm. The MFH method outperforms the remaining decentralized methods and the use of the second best performing method results in a penalty of 61.5% on the estimation performance. It is worth remarking that the PMHE1 method can handle constraints on the state variables and on the noise, which the proposed framework cannot. As a result, the PMHE1 method cannot be synthesized offline. Thus, to rely on local communication exclusively, it has to disregard uncertainty associated with the dynamic couplings between systems, which degrades performance. As a result, although it performs better than the OS method, it underperforms in comparison with the MHE solution proposed herein.

V. Conclusion

This paper addresses the problem of designing a decentralized state estimation solution for a large-scale network of interconnected unconstrained LTI systems. The problem is tackled in a novel MHE framework, motivated by an attempt to increase the estimation performance in relation to the state-of-the-art decentralized methods based on recursive Luenberger filters. First, the proposed MHE framework has low computational and memory requirements and requires local communication exclusively. Second, the proposed approach achieves a significant improvement in performance in comparison with recent Luenberger-based filtering methods. Third, we show that a state-of-the-art distributed MHE solution with comparable requirements underperforms in comparison with the solution proposed herein. Fourth, it can be concluded that other MHE-based decentralized methods could, in principle, achieve the same performance of the proposed MFH method for unconstrained systems. Nevertheless, by taking into account state and noise bounds, they cannot be synthesized offline, which imposes limiting constraints for their implementation feasibility that degrade performance.

Acknowledgements

This work was supported by the Fundação para a Ciência e a Tecnologia (FCT) through LARSyS - FCT Project UIDB/50009/2020 and through the FCT project DECENTER [LISBOA-01-0145-FEDER-029605], funded by the Programa Operacional Regional de Lisboa 2020 and PIDDAC programs.

Appendix

Assume, without loss of generality, that $\mathbf{A}$ is invertible. Using (7) to expand the objective function of (13) as a function of the estimation error covariance at time instant τ, i.e. $\mathbf{P}(\tau|\tau|k)$, yields a quadratic expression on $\mathbf{K}(\tau|k)$

$$\begin{aligned}\mathbf{P}(k|k|k) =& \mathbf{\Gamma}(\tau,k)\mathbf{P}(\tau-1|\tau-1|k)\mathbf{\Gamma}^T(\tau,k)+\\ &\sum_{i=\tau}^{k}\mathbf{\Gamma}(i+1,k)\mathbf{K}(i|k)\mathbf{R}\mathbf{K}^T(i|k)\mathbf{\Gamma}^T(i+1,k)+\\ &\sum_{i=\tau}^{k}\mathbf{\Gamma}(i,k)\mathbf{A}^{-1}\mathbf{Q}\mathbf{A}^{-T}\mathbf{\Gamma}^T(i,k)\,,\end{aligned}$$

where $\mathbf{\Gamma}(k_i,k_f)$ is defined as in (16). Taking the derivative of the trace of $\mathbf{P}(k|k|k)$ with respect to $\mathbf{K}(\tau|k)$ yields

$$\begin{aligned}\frac{\partial}{\partial\mathbf{K}(\tau|k)}\text{tr}(\mathbf{P}(k|k)) =& 2\mathbf{\Gamma}^T(\tau+1,k)\mathbf{\Gamma}(\tau+1,k)\\ &\left(\mathbf{K}(\tau|k)\mathbf{S}(\tau|k)-\mathbf{P}(\tau|\tau-1|k)\mathbf{C}^T\right).\end{aligned}$$

Let $\mathbf{l}_i$ denote a column vector whose entries are all set to zero except for the i-th one, which is set to 1. Then the optimal solutions is given by

$$\begin{cases}\mathbf{l}_i^T\mathbf{\Psi}(\tau|k)\mathbf{K}(\tau|k)\mathbf{S}(\tau|k)\mathbf{l}_j-\\ \quad\mathbf{l}_i^T\mathbf{\Psi}(\tau|k)\mathbf{P}(\tau|\tau-1|k)\mathbf{C}^T\mathbf{l}_j=0, & [\mathbf{E}]_{ij}\neq 0\\ \mathbf{l}_i^T\mathbf{K}(\tau|k)\mathbf{l}_j=0, & [\mathbf{E}]_{ij}=0,\end{cases} \quad (17)$$

where $\mathbf{\Psi}(\tau|k)$ is defined as in (15). For each k and each τ, (17) is identical to the equation that arises in the derivation of the finite-horizon method [12, Appendix B]. Therefore, the same techniques may be employed to solve (17). The solution of this optimization problem is, thus, given by (14).

References

[1] R. T. Thien and Y. Kim, “Decentralized formation flight via PID and integral sliding mode control,” *Aerospace Science and Technology*, vol. 81, pp. 322–332, 2018.

[2] D. Viegas, P. Batista, P. Oliveira, and C. Silvestre, “Decentralized observers for position and velocity estimation in vehicle formations with fixed topologies,” *Systems & Control Letters*, vol. 61, pp. 443–453, mar 2012.

[3] D. Ivanov, U. Monakhova, and M. Ovchinnikov, “Nanosatellites swarm deployment using decentralized differential drag-based control with communicational constraints,” *Acta Astronautica*, vol. 159, pp. 646–657, 2019.

[4] A. Alam, J. Mårtensson, and K. H. Johansson, "Experimental evaluation of decentralized cooperative cruise control for heavy-duty vehicle platooning," *Control Engineering Practice*, vol. 38, pp. 11–25, 2015.

[5] I. Prodan, L. Lefevre, D. Genon-Catalot, *et al.*, "Distributed model predictive control of irrigation systems using cooperative controllers," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 6564–6569, 2017.

[6] R. Vadigepalli and F. J. Doyle Iii, "Structural analysis of large-scale systems for distributed state estimation and control applications," *Control Engineering Practice*, vol. 11, no. 8, pp. 895–905, 2003.

[7] P. Benner, "Solving large-scale control problems," *IEEE Control Systems Magazine*, vol. 24, no. 1, pp. 44–59, 2004.

[8] V. D. Blondel and J. N. Tsitsiklis, "A survey of computational complexity results in systems and control," *Automatica*, vol. 36, no. 9, pp. 1249–1274, 2000.

[9] H. S. Witsenhausen, "A counterexample in stochastic optimum control," *SIAM Journal on Control*, vol. 6, no. 1, pp. 131–147, 1968.

[10] L. Lessard and S. Lall, "Internal quadratic invariance and decentralized control," in *Proceedings of the 2010 American Control Conference (ACC)*, pp. 5596–5601, IEEE, 2010.

[11] L. Lessard and S. Lall, "Convexity of decentralized controller synthesis," *IEEE Transactions on Automatic Control*, vol. 61, no. 10, pp. 3122–3127, 2015.

[12] D. Viegas, P. Batista, P. Oliveira, and C. Silvestre, "Discrete-time distributed Kalman filter design for formations of autonomous vehicles," *Control Engineering Practice*, vol. 75, pp. 55–68, jun 2018.

[13] D. Viegas, P. Batista, P. Oliveira, and C. Silvestre, "Discrete-time distributed Kalman filter design for multi-vehicle systems," in *2017 American Control Conference (ACC)*, pp. 5538–5543, IEEE, 2017.

[14] C. Fransson and B. Lennartson, "Low order multicriteria $\mathcal{H}_\infty$ design via bilinear matrix inequalities," in *42nd IEEE International Conference on Decision and Control (CDC)*, vol. 5, pp. 5161–5167, IEEE, 2003.

[15] M. Fardad and M. R. Jovanović, "On the design of optimal structured and sparse feedback gains via sequential convex programming," in *2014 American Control Conference (ACC)*, pp. 2426–2431, IEEE, 2014.

[16] D. Viegas, P. Batista, P. Oliveira, and C. Silvestre, "Gas decentralized navigation filters in a continuous-discrete fixed topology framework," in *21st Mediterranean Conference on Control and Automation*, pp. 1286–1291, IEEE, 2013.

[17] A. Lamperski and L. Lessard, "Optimal decentralized state-feedback control with sparsity and delays," *Automatica*, vol. 58, pp. 143–151, 2015.

[18] B. Chen, G. Hu, D. W. Ho, and L. Yu, "Distributed Kalman filtering for time-varying discrete sequential systems," *Automatica*, vol. 99, pp. 228–236, 2019.

[19] M. Farina, G. Ferrari-Trecate, and R. Scattolini, "A moving horizon scheme for distributed state estimation," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pp. 1818–1823, IEEE, 2009.

[20] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Transactions on automatic control*, vol. 55, no. 9, pp. 2069–2084, 2010.

[21] G. Battistelli, L. Chisci, G. Mugnai, A. Farina, and A. Graziano, "Consensus-based linear and nonlinear filtering," *IEEE Transactions on Automatic Control*, vol. 60, no. 5, pp. 1410–1415, 2014.

[22] U. A. Khan and J. M. Moura, "Distributing the Kalman filter for large-scale systems," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 4919–4935, 2008.

[23] C. V. Rao, *Moving horizon strategies for the constrained monitoring and control of nonlinear discrete-time systems*. The University of Wisconsin-Madison, 2000.

[24] C. V. Rao, J. B. Rawlings, and J. H. Lee, "Constrained linear state estimation—a moving horizon approach," *Automatica*, vol. 37, no. 10, pp. 1619–1628, 2001.

[25] 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, 2003.

[26] A. Alessandri, M. Baglietto, and G. Battistelli, "Moving-horizon state estimation for nonlinear discrete-time systems: New stability results and approximation schemes," *Automatica*, vol. 44, no. 7, pp. 1753–1765, 2008.

[27] G. Battistelli, L. Chisci, and S. Gherardini, "Moving horizon estimation for discrete-time linear systems with binary sensors: Algorithms and stability results," *Automatica*, vol. 85, pp. 374–385, 2017.

[28] 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, 2010.

[29] M. Farina, G. Ferrari-Trecate, and R. Scattolini, "Distributed moving horizon estimation for linear constrained systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 11, pp. 2462–2475, 2010.

[30] 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, 2013.

[31] A. Alessandri and M. Awawdeh, "Moving-horizon estimation with guaranteed robustness for discrete-time linear systems and measurements subject to outliers," *Automatica*, vol. 67, pp. 85–93, 2016.

[32] G. Battistelli, "Distributed moving-horizon estimation with arrival-cost consensus," *IEEE Transactions on Automatic Control*, vol. 64, no. 8, pp. 3316–3323, 2018.

[33] I. I. Sirmatel and N. Geroliminis, "Nonlinear moving horizon estimation for large-scale urban road networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 12, pp. 4983–4994, 2019.

[34] M. Farina, G. Ferrari-Trecate, C. Romani, and R. Scattolini, "Moving horizon estimation for distributed nonlinear systems with application to cascade river reaches," *Journal of Process Control*, vol. 21, no. 5, pp. 767–774, 2011.

[35] F. L. Lewis, L. Xie, and D. Popa, *Optimal and robust estimation: with an introduction to stochastic control theory*. CRC press, 2017.

[36] W. D. Wallis, *A beginner's guide to graph theory*. Springer Science & Business Media, 2010.

[37] L. Pedroso, P. Batista, P. Oliveira, and C. Silvestre, "Discrete-time distributed Kalman filter design for networks of interconnected systems with linear time-varying dynamics," *International Journal of Systems Science*, pp. 1–18, 2021.

[38] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[39] P. Jain, P. Kar, *et al.*, "Non-convex optimization for machine learning," *Foundations and Trends® in Machine Learning*, vol. 10, no. 3-4, pp. 142–363, 2017.

[40] S. H. Low, "Convex relaxation of optimal power flow—part i: Formulations and equivalence," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 1, pp. 15–27, 2014.

[41] S. H. Low, "Convex relaxation of optimal power flow—part ii: Exactness," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 2, pp. 177–189, 2014.

[42] L. Pedroso and P. Batista, "Efficient algorithm for the computation of the solution to a sparse matrix equation in distributed control theory," *Mathematics*, vol. 9, no. 13, 2021.

Leonardo Pedroso received the BSc and MSc in Aerospace Engineering in 2020 and 2022, from Instituto Superior Técnico (IST), University of Lisbon (ULisboa), Portugal. He has received four academic excellence diplomas, endowed by IST, and four outstanding academic performance awards, endowed by ULisboa. He was also awarded an honorable mention in the 48th International Physics Olympiad. Since 2019 he is a member of the research team of the DECENTER project, funded by the Portuguese National Science Foundation.

Pedro Batista (SM'18) received the Licenciatura degree in Electrical and Computer Engineering, in 2005, and the PhD degree, in 2010, both from the Instituto Superior Técnico (IST), Lisbon, Portugal. From 2004 to 2006, he was a Monitor with the Department of Mathematics, IST. Since 2012, he has been with the Department of Electrical and Computer Engineering of IST, where he is currently an Associate Professor. His research interests include sensor based navigation and control of autonomous vehicles. Dr. Batista has received the Diploma de Mérito twice during his graduation and his PhD thesis was awarded the Best Robotics PhD Thesis Award by the Portuguese Society of Robotics.