



**HAL**  
open science

## Distributed moving horizon estimation with pre-estimating observer

Antonello Venturino, Sylvain Bertrand, Cristina Stoica Maniu, Teodoro Alamo, Eduardo Fernandez Camacho

► **To cite this version:**

Antonello Venturino, Sylvain Bertrand, Cristina Stoica Maniu, Teodoro Alamo, Eduardo Fernandez Camacho. Distributed moving horizon estimation with pre-estimating observer. 24th International Conference on System Theory, Control and Computing (ICSTCC 2020), Oct 2020, Sinaia, Romania. 10.1109/icstcc50638.2020.9259766 . hal-02972554

**HAL Id: hal-02972554**

**<https://hal.science/hal-02972554>**

Submitted on 20 Oct 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Distributed moving horizon estimation with pre-estimating observer

Antonello Venturino<sup>\*,†</sup>, Sylvain Bertrand<sup>†</sup>, Cristina Stoica Maniu<sup>\*</sup>, Teodoro Alamo<sup>‡</sup> and Eduardo F. Camacho<sup>‡</sup>

**Abstract**—This paper proposes a new Distributed Moving Horizon Estimation (DMHE) algorithm for state estimation of a discrete-time linear system by a sensor network. The main contribution consists in using a pre-estimating Luenberger observer in the formulation of the local problem to be solved by each sensor. This results in a significant reduction of the computation time while preserving at the same time the accuracy of the estimates. The state estimate computed by each sensor is capable to converge even under weak observability conditions. Comparison with existing approaches is proposed through a simulation example, for both centralized and distributed cases, in terms of estimation accuracy and computation time.

**Index Terms**—distributed state estimation, moving horizon estimation, sensor networks, Luenberger observers, linear systems.

## I. INTRODUCTION

In recent decades, there has been an increasing interest in research on distributed state estimation due to its variety of applications over sensor networks such as target tracking [1]–[2], exploration [3], surveillance [4], etc. The estimation algorithms used in these applications are mainly formulated as centralized fusion architectures, where all the sensors transmit their local measurements to a central unit that processes the provided measurements to update the estimate. In general, the centralized algorithms are not scalable, since with the increasing number of sensors the complexity of the problem to solve increases, too. Furthermore, in case of large-scale networks, communication with a unique central unit can hardly be achieved e.g. due to communication bandwidth limitations. Unlike centralized schemes, in the distributed approaches [5]–[7] each sensor computes a local estimate using information acquired by other connected sensors within a local neighborhood. This can improve robustness to sensor failure exploiting redundancy and also lower the communication burden since data is transmitted amongst local nodes in the network.

The continuous decreasing costs of sensors is making these applications realizable, even though there are still open problems to face with. In fact, the distributed algorithms need to have particular properties in order to make them attractive for the industrial community. Thus, in the context of large-scale systems, the algorithms must be scalable to be able to deal

with large networks, must have low computation load since low-cost sensors are not powerful devices, must minimize the utilization of communication resources, etc. In [8], the authors reviewed several works on distributed state estimation over low-cost sensors networks, pointing out their characteristics, advantages, and challenging issues.

In recent years, Moving Horizon Estimation (MHE) techniques and the distributed counterpart (DMHE) have been used to successfully deal with large sensor networks [9]. The first idea of MHE proposed in [10] consists in estimating the current states by solving a least-square optimization problem penalizing on one hand the deviation between acquired and predicted measurements, and on the other hand the distance between the estimated state and some *a priori* information on it. MHE is a practical strategy for constrained state estimation and a lot of research has been devoted to develop stability guarantees on the estimation error, e.g., [11]–[13]. Although this approach is functional for control engineers offering the freedom to tune the parameters of the cost function, a strength and a weakness of this approach is the use of an optimization problem. This has to be solved within the sampling time; however, for large-scale systems this issue becomes critical. There have been several attempts in trying to reduce the computation demanding of MHE. One idea is to add a pre-estimating observer in the formulation. The authors of [12] proposed a MHE strategy with a Luenberger observer that leads to good performance especially for large estimation horizons. They also propose an optimization problem to tune the parameters minimizing the effects of measurement noise and model errors. In [14], the formulation is generalized using a weight matrix for the penalty function and adding states constraints. A MHE with pre-estimation has been proposed also for non-linear systems [13]. In [6] a distributed algorithm is proposed for linear systems with a stability proof under weak observability conditions (exploiting a *consensus-on-estimate* and a *consensus weight* term in the DMHE formulation). However, the computation time issue becomes crucial, since usually the network could be composed of low-cost sensors.

In this paper we propose a DMHE algorithm with pre-estimation, based on the ideas of [6], [12], [14]. A pre-estimating Luenberger observer is considered in the formulation of the local problem to be solved by each sensor, resulting in a significant reduction of the computation time. The main contribution of this paper covers: i) reducing the computation time of the optimization problems, ii) preserving the accuracy of the estimation errors. This allows the use of this type of algorithms for time-sensitive applications.

<sup>\*</sup>A. Venturino and C. Stoica Maniu are with Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes, 91190, Gif-sur-Yvette, France (e-mail: {antonello.venturino; cristina.stoica}@12s.centralesupelec.fr).

<sup>†</sup>A. Venturino and S. Bertrand are with Université Paris-Saclay, ONERA, Traitement de l'information et systèmes, 91123, Palaiseau, France (e-mail: {antonello.venturino; sylvain.bertrand}@onera.fr).

<sup>‡</sup>T. Alamo and E.F. Camacho are with Department of Ingeniería de Sistemas y Automática, Universidad de Sevilla, Camino de los Descubrimientos, 41092 Sevilla, Spain (e-mail: talamo@us.es, eduardo@esi.us.es).

The paper is organized as follows. Section II introduces the main theoretical statements and definitions. Sections III and IV present the proposed versions of centralized MHE and distributed MHE approaches, respectively. In Section V a simulation example is proposed, while Section VI provides some concluding remarks.

## II. NOTATIONS AND DEFINITIONS

Consider the following discrete-time linear time invariant (LTI) system:

$$x_{t+1} = Ax_t + w_t, \quad (1)$$

where  $x_t \in \mathcal{X} \subseteq \mathbb{R}^n$  is the state vector and  $w_t \in \mathcal{W} \subseteq \mathbb{R}^n$  is a white noise with covariance matrix  $Q$ . The sets  $\mathcal{X}$  and  $\mathcal{W}$  are assumed convex with  $0 \in \mathcal{X}$  and  $0 \in \mathcal{W}$ . The initial state  $x_0$  is unknown and assumed to be modelled by a random variable. Let us denote respectively by  $\mu$  and  $\Pi_0$  its mean and covariance matrix. The measurements are carried out by  $M$  sensors, that could be different from each other:

$$y_t^i = C^i x_t + v_t^i, \quad i = 1, \dots, M, \quad (2)$$

where  $v_t^i \in \mathbb{R}^{p_i}$  is a white noise with covariance  $R^i$ .

Consider a sensor network described by the directed graph  $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ , where nodes in  $\mathcal{N} = \{1, 2, \dots, M\}$  represents the sensors and the edge  $(j, i) \in \mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$  represents the communication link from sensor  $j$  to sensor  $i$ . We assume that all the links have a self-loop  $(i, i) \in \mathcal{E}$ ,  $\forall i \in \mathcal{N}$ . The transmission neighbourhood  $\mathcal{N}_\tau^i$  of the sensor  $i$  is  $\mathcal{N}_\tau^i = \{j \in \mathcal{N} : (i, j) \in \mathcal{E}\}$ , the reception neighbourhood  $\mathcal{N}_\rho^i$  of the sensor  $i$  is  $\mathcal{N}_\rho^i = \{j \in \mathcal{N} : (j, i) \in \mathcal{E}\}$  and the regional neighbourhood  $\mathcal{N}^i$  of the sensor  $i$  is  $\mathcal{N}^i = \mathcal{N}_\tau^i \cup \mathcal{N}_\rho^i$ . The number of nodes  $j \in \mathcal{N}_\rho^i$  is denoted by  $M^i$ .

A stochastic matrix  $K \in \mathbb{R}^{M \times M}$  is associated to the graph  $\mathcal{G}$  such that the elements:

$$k_{ij} > 0 \quad \text{if } (j, i) \in \mathcal{E}, \quad (3a)$$

$$k_{ij} = 0 \quad \text{otherwise}, \quad (3b)$$

$$\sum_{j=1}^M k_{ij} = 1, \quad \forall i = 1, \dots, M. \quad (3c)$$

The topology of the sensor network is assumed to be time-invariant thus the matrix  $K$  is constant. This matrix will be used to compute the consensus terms in the DMHE algorithm described in Section IV.

We use the notation introduced in [6] to recognize *local*, *regional* and *collective* information. Considering Sensor  $i$ , an information is considered to be *local* if it is related only to the node  $i$ , *regional* if it is related to the nodes in  $\mathcal{N}_\rho^i$ , and *collective* if it is related to the entire network. Corresponding notations from [6] are used to distinguish *local*, *regional* and *collective* variables. For a given variable  $z$ ,  $z^i$  denotes its local version,  $\bar{z}^i$  the regional one and  $\mathbf{z}$  the collective one. Using these notations, the regional measurements of a sensor  $i$  with reception neighborhood  $\mathcal{N}_\rho^i = \{j_1^i, \dots, j_{M^i}^i\}$  are

$$\bar{y}_t^i = \bar{C}^i x_t + \bar{v}_t^i, \quad (4)$$

where  $\bar{y}_t^i = [(y_t^{j_1^i})^T \dots (y_t^{j_{M^i}^i})^T]^T \in \mathbb{R}^{\bar{p}_i}$  with  $\bar{p}_i = \sum_{i \in \mathcal{N}_\rho^i} p_i$ ,  $\bar{C}^i = [(C^{j_1^i})^T \dots (C^{j_{M^i}^i})^T]^T$  and  $\bar{v}_t^i = [(v_t^{j_1^i})^T \dots (v_t^{j_{M^i}^i})^T]^T$ . The covariance matrix associated to the regional noise  $\bar{v}_t^i$  of sensor  $i$  is  $\bar{R}^i = \text{diag}(R^{j_1^i}, \dots, R^{j_{M^i}^i})$ .

Base on this nomenclature, three different observability notions are defined in [6] as follows.

**Definition 1.** The system is *locally observable* by sensor  $i$  (sensor  $i$  is locally observable) if the pair  $(A, C^i)$  is observable. The system is *regionally observable* by sensor  $i$  (sensor  $i$  is regionally observable) if the pair  $(A, \bar{C}^i)$  is observable. The system is *collective observable* if the pair  $(A, C)$  is observable.

## III. CENTRALIZED MHE WITH PRE-ESTIMATION

Moving Horizon Estimation with pre-estimation is based on the idea in [12] and [14], where the authors proposed a MHE formulation relying on a Luenberger observer. The centralized scheme makes use of the collective information solely.

In this strategy, for a given horizon length  $N \geq 1$ , an estimate  $\hat{x}_{t|t}$  of the state at time  $t$  is computed by solving the constrained minimization problem, hereafter denoted by  $\text{MHE}_{pre}$ :

$$\hat{x}_{t-N|t} = \arg \min_{\hat{x}_{t-N}} J(t-N, t, \hat{x}_{t-N}, \hat{\mathbf{v}}, \Gamma_{t-N}) \quad (5)$$

$$\text{s.t.} \quad \hat{x}_{k+1} = A\hat{x}_k + L\hat{\mathbf{v}}_k, \quad (6)$$

$$\hat{\mathbf{y}}_k = \mathbf{C}\hat{x}_k + \hat{\mathbf{v}}_k, \quad (7)$$

$$\hat{x}_k \in \mathcal{X}, \quad (8)$$

$$\forall k = t-N, \dots, t.$$

The gain matrix  $L \in \mathbb{R}^{n \times p}$  in (6), with  $p = \sum_{i=1}^M p_i$ , is chosen such that  $\Phi = A - LC$  is Schur stable. The cost function  $J$  given by:

$$J(t-N, t, \hat{x}_{t-N}, \hat{\mathbf{v}}, \Gamma_{t-N}) = \frac{1}{2} \sum_{k=t-N}^t \|\hat{\mathbf{v}}_k\|_{\mathbf{R}^{-1}}^2 + \Gamma_{t-N}(\hat{x}_{t-N}, \hat{x}_{t-N|t-1}), \quad (9)$$

where  $\mathbf{R} = \text{diag}(R^1, \dots, R^M)$ . The so called *initial penalty* function  $\Gamma_{t-N}(\hat{x}_{t-N}, \hat{x}_{t-N|t-1})$  in (9) is defined as follows:

$$\Gamma_{t-N}(\hat{x}_{t-N}, \hat{x}_{t-N|t-1}) = \frac{1}{2} \|\hat{x}_{t-N} - \hat{x}_{t-N|t-1}\|_{\Pi_{t-N|t-1}^{-1}}^2, \quad (10)$$

where  $\hat{x}_{t-N|t-1}$  is the second estimated state of the sequence computed at the previous time. The positive-definite symmetric weighting matrix  $\Pi_{t-N|t-1}$  in (10) is the unique solution of the following discrete-time algebraic Riccati equation [15]:

$$\Pi_{t-N|t-1} = Q + A\Pi_{t-N-1|t-2}A^T - A\Pi_{t-N-1|t-2}C^T(\mathbf{R} + C\Pi_{t-N-1|t-2}C^T)^{-1}C\Pi_{t-N-1|t-2}A^T, \quad (11)$$

subject to the initial condition  $\Pi_0$ . When  $t \leq N$ , one sets  $N = t$ . We denote by  $\{\hat{x}_{k|t}\}_{k=t-N}^t$  the sequence of the estimated states computed using (6) with the optimum solution  $\hat{x}_{t-N|t}$  of (5) as initial condition. The last term  $\hat{x}_{t|t}$  of this sequence corresponds to the estimate of the state  $x_t$  of the system at the current time instant  $t$ .

#### IV. DISTRIBUTED MHE WITH PRE-ESTIMATION

In [6] the authors proposed a distributed Moving Horizon Estimation method to estimate the state and the input of the model (1). In their strategy, the optimization problem to be solved online at time  $t$  involves the computation of the state trajectory over the past horizon. As in classical MHE schemes, this computation is done by propagating forward the state, from its initial condition at  $t - N$ , using the dynamic model of the system. This can accumulate the estimation error, especially when  $N$  is large and the system is unstable. In this section, we propose a new DMHE strategy with pre-estimation by introducing a Luenberger observer in its formulation. This will mitigate the effect of model uncertainty in the *a priori* estimate and thus will contribute to enhance the estimation accuracy. This improvement will also reduce the computation time required to solve the optimization problem, because the number of optimization variables involved is lower and does not depend on the horizon length  $N$ . Further details are given in the next sections.

##### A. Local minimization problem

Here we formulate the proposed DMHE scheme, from now on denoted as  $\text{DMHE}_{pre}$ , where each sensor  $i \in \mathcal{N}$  solves its own local moving horizon estimation problem based on regional measurements  $\hat{y}_t^i$  and some shared information amongst the neighbourhood  $\mathcal{N}^i$ . For a given horizon length  $N \geq 1$ , each sensor  $i \in \mathcal{N}$  computes an estimate  $\hat{x}_{t|t}^i$  of the state  $x_t$  at time  $t$  by solving the following constrained minimization problem with pre-estimation, hereafter denoted as  $\text{DMHE}_{i-pre}$ :

$$\hat{x}_{t-N|t}^i = \arg \min_{\hat{x}_{t-N}^i} J^i(t-N, t, \hat{x}_{t-N}^i, \hat{v}^i, \Gamma_{t-N}^i) \quad (12)$$

$$\text{s.t.} \quad \hat{x}_{k+1}^i = A\hat{x}_k^i + L^i\hat{v}_k^i, \quad (13)$$

$$\hat{y}_k^i = \bar{C}^i\hat{x}_k^i + \hat{v}_k^i, \quad (14)$$

$$\hat{x}_k^i \in \mathcal{X}, \quad (15)$$

$$\forall k = t-N, \dots, t.$$

Note that the equations remind the ones of the  $\text{MHE}_{pre}$  problem but using regional information instead. There are also some differences due to the *consensus* terms embodied in the initial penalty, which will be discussed in detail later on. First of all, the Luenberger gain  $L^i$  is computed such that  $\Phi^i = A - L^i\bar{C}^i$  is Schur stable when the sensor  $i$  is regionally observable, i.e. the pair  $(A, \bar{C}^i)$  is observable. Otherwise, as *extrema ratio*, one can compute  $L^i$  in order to minimize the propagation of the error along the prediction horizon by keeping the spectrum radius of  $\Phi^i$  as small as possible.

The cost function  $J^i$  is given by:

$$J^i(t-N, t, \hat{x}_{t-N}^i, \hat{v}^i, \Gamma_{t-N}^i) = \frac{1}{2} \sum_{k=t-N}^t \left\| \hat{v}_k^i \right\|_{(\bar{R}^i)^{-1}}^2 + \Gamma_{t-N}^i(\hat{x}_{t-N}^i, \hat{x}_{t-N|t-1}^i), \quad (16)$$

where the initial penalty function  $\Gamma_{t-N}^i(\hat{x}_{t-N}^i, \hat{x}_{t-N|t-1}^i)$  in (16) is defined as follows:

$$\Gamma_{t-N}^i(\cdot) = \frac{1}{2} \left\| \hat{x}_{t-N}^i - \hat{x}_{t-N|t-1}^i \right\|_{(\bar{\Pi}_{t-N|t-1}^i)^{-1}}^2. \quad (17)$$

Let  $\hat{x}_{t-N|t-1}^i$  denote the weighted average of the state estimates produced by sensors  $j \in \mathcal{N}^i$ , computed as follows:

$$\hat{x}_{t-N|t-1}^i = \sum_{j \in \mathcal{N}^i} k_{ij} \hat{x}_{t-N|t-1}^j, \quad (18)$$

where  $\hat{x}_{t-N|t-1}^j$  is the second estimated state in the sequence computed at the previous time by sensor  $j$ . A consensus-like term on the estimates is therefore introduced by (18) in the initial penalty (17). It helps to improve the accuracy of the local estimates and enables to guarantee convergence of the estimates to the state of the system even without regional observability [6].

The term  $\bar{\Pi}_{t-N|t-1}^i$  is computed as in [6]. For the sake of clarity, we recall here the procedure to compute it by:

$$\bar{\Pi}_{t-N|t-1}^i = \sum_{j \in \mathcal{N}^i} M^j k_{ij}^2 \bar{\Pi}_{t-N|t-1}^j, \quad (19)$$

where the update of  $\bar{\Pi}_{t-N|t-1}^i$  is performed by the sensor  $i$  based on regional information. More specifically, the matrix  $\bar{\Pi}_{t-N|t-1}^i$ , with  $i \in \mathcal{N}$ , is given by one iteration of the difference Riccati equation associated to a Kalman filter for the system:

$$\begin{cases} x_{t-N} = Ax_{t-N-1} + w_{t-N-1} \\ \bar{z}_{t-N}^i = \bar{O}_N^i x_{t-N} + \bar{V}_{t-N}^i \end{cases}$$

where  $\bar{V}_{t-N}^i$  represents the measurements noise and  $\bar{O}_N^i$  defines the  $i$ -th sensor regional observability matrix:

$$\bar{O}_N^i = [(\bar{C}^i)^T \quad (\bar{C}^i A)^T \quad \dots \quad (\bar{C}^i A^{N-1})^T]^T \quad (20)$$

If we define

$$C_N^i = \begin{bmatrix} 0 & 0 & \dots & 0 \\ \bar{C}^i & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \bar{C}^i A^{N-2} & \bar{C}^i A^{N-3} & \dots & \bar{C}^i \end{bmatrix} \in \mathbb{R}^{\bar{p}_i N \times n(N-1)}, \quad (21)$$

$$\bar{R}_N^i = \text{diag}(\bar{R}^i, \dots, \bar{R}^i) \in \mathbb{R}^{\bar{p}_i N \times \bar{p}_i N}, \quad (22)$$

$$Q_{N-1} = \text{diag}(Q, \dots, Q) \in \mathbb{R}^{n(N-1) \times n(N-1)}, \quad (23)$$

$$\text{Cov}[\bar{V}_t^i] = \bar{R}_N^{*i} = \bar{R}_N^i + C_N^i Q_{N-1} (C_N^i)^T, \quad (24)$$

and set the covariance of the estimate  $\hat{x}_{t-N-1}^i$  as

$$\bar{\Pi}_{t-N-1|t-2}^{*i} = \left( \left( \bar{\Pi}_{t-N-1|t-2}^i \right)^{-1} + (\bar{C}^i)^T (\bar{R}^i)^{-1} \bar{C}^i \right)^{-1}, \quad (25)$$

the resulting Riccati recursive equation is given by

$$\begin{aligned}\Pi_{t-N|t-1}^i &= \mathcal{R}^i(\Pi_{t-N-1|t-2}^{*i}, Q, \bar{R}_N^{*i}) \\ &= A\Pi_{t-N-1|t-2}^{*i}A^T + Q - A\Pi_{t-N-1|t-2}^{*i}(\bar{O}_N^i)^T \\ &\quad \times \left(\bar{O}_N^i\Pi_{t-N-1|t-2}^{*i}(\bar{O}_N^i)^T + \bar{R}_N^{*i}\right)^{-1} \\ &\quad \times \bar{O}_N^i\Pi_{t-N-1|t-2}^{*i}A^T.\end{aligned}\quad (26)$$

Since the communication network topology is assumed to be time-invariant, these equations can be computed off-line.

### B. Network information exchange

It is worth highlighting the way each node exchanges the information within its neighbourhood. For this purpose, we recall here some assumptions that play a major role:

- the sensors can be different from each other;
- the sensors characteristics (noise covariance and type of measurements) are not time-varying;
- the network topology is not time-varying;
- each sensor  $i$  knows its reception neighbourhood  $\mathcal{N}_\rho^i$ ;
- no time delay in the communication network.

This implies that the matrices  $C^i$  can be different for all  $i \in \mathcal{N}$ . Since the reception neighbourhood  $\mathcal{N}_\rho^i$  is known *a priori*, it is not necessary to exchange the information on the matrices  $C^i$  and  $R^i$  at each time. Moreover, this allows one to compute off-line the Luenberger gains  $L^i$ .

### C. DMHE<sub>pre</sub> procedure

Finally, we have all the elements to describe the procedure of the proposed distributed scheme.

#### 1) Off-line computation:

- a) Since the Luenberger gains  $L^i$  are constants, they can be computed off-line.
- b) All nodes store the estimate  $\hat{x}_{0|0}^i = \hat{x}_0 = \mu$  of  $x_0$ , where  $\mu$  is given, and the covariance matrix  $\Pi_0$  of  $x_0$ .

#### 2) Initialization:

- a) At the first step  $t = 0$ , each node  $i \in \mathcal{N}$  performs a first measurement  $y_0^i$  and sends the information to the nodes  $j \in \mathcal{N}_\tau^i$ .

#### 3) Online computation: each node $i \in \mathcal{N}$

- a) performs the measurement  $y_t^i$  and sends it to the nodes  $j \in \mathcal{N}_\tau^i$ ;
- b) **if**  $1 \leq t \leq N$ , then  $N$  is set to  $N = t$ ,  $\bar{\Pi}_{t-N|t-1}^i = \bar{\Pi}_{0|t-1}^i = \Pi_0$  and  $\hat{x}_{t-N|t-1}^i = \hat{x}_{0|t-1}^i$ ;
- c) **if**  $t > N$ :
  - i) computes  $\Pi_{t-N|t-1}^i$  according to (24), (25) and (26), and sends it to the nodes  $j \in \mathcal{N}_\tau^i$ ;
  - ii) computes  $\bar{\Pi}_{t-N|t-1}^i$  according to (19);
- d) solves the problem DMHE<sub>*i-pre*</sub> and store the solution  $\hat{x}_{t|t}^i$ ;
- e) sends  $\hat{x}_{t-N+1|t}^i$  to the nodes  $j \in \mathcal{N}_\tau^i$ .

Notice that the steps 3(c)i and 3(c)ii in the DMHE<sub>pre</sub> procedure could be computed off-line. However, for clarity

reasons with respect to the exchanging information timing, they have been described in the online step.

## V. EXAMPLE

A performance analysis is provided in this section for the proposed DMHE algorithm with pre-estimation through a simulation example. Comparison is performed to the centralized MHE of [11] and the DMHE of [6]. The system considered for illustration is the one used in [6] and given by

$$x_{t+1} = \begin{bmatrix} 0.9962 & 0.1949 & 0 & 0 \\ -0.1949 & 0.3819 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1.21 & 1.98 \end{bmatrix} x_t + w_t, \quad (27)$$

where  $x_t = [x_{1,t} \ x_{2,t} \ x_{3,t} \ x_{4,t}]^T$  is the state vector and  $w_t \in \mathbb{R}^4$  is a white noise with covariance  $Q = \text{diag}(0.0012, 0.038, 0.0012, 0.038)$ . Notice that the system is unstable since the eigenvalues of  $A$  are 0.9264, 0.4517,  $0.99 \pm 0.4795i$  and  $|0.99 \pm 0.4795i| > 1$ . For this example, one considers  $\mathcal{X} = \mathbb{R}^4$ .

The initial values of the algorithms have been set as  $\mu = [0 \ 0 \ 0 \ 0]^T$ ,  $\Pi_0 = I_4$  and  $N = 5$ .

To compare the results of all algorithms we use  $M = 4$  sensors both for the centralized and distributed cases. For the centralized scheme the following measurement equation is considered:

$$y_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x_t + v_t,$$

with  $\text{Var}(v_t) = R = I_4$ . For the distributed schemes, the following measurement equations are used:

$$\begin{aligned}y_t^i &= [1 \ 0 \ 0 \ 0] x_t + v_t^i \quad \text{if } i = 1, 2, \\ y_t^i &= [0 \ 0 \ 1 \ 0] x_t + v_t^i \quad \text{if } i = 3, 4,\end{aligned}$$

where  $\text{Var}(v_t^i) = R^i = 1$ ,  $i = 1, \dots, 4$ . The nodes are connected as reported by the graph in Fig. 1 and the associated matrix is defined as follows:

$$K = \begin{bmatrix} 0.5 & 0 & 0 & 0.5 \\ 0.5 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0.5 \end{bmatrix}.$$

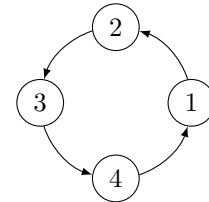


Fig. 1. Communication network.

As pointed out in [6], and recalled next, this example is challenging due to the non regional observability of some of

the sensors. In fact, analyzing the sensor 1, we notice that at each time instant it has available the information about  $x_{1,t}$ , directly measured, and  $x_{3,t}$ , transmitted by sensor 4. Likewise, the information available to sensor 2 consists of  $x_{1,t}$ , directly measured, and  $x_{1,t}$ , transmitted by sensor 1. The same applies to the sensors 3 and 4, which have information about  $(x_{1,t}, x_{3,t})$  and  $(x_{3,t}, x_{3,t})$ , respectively. Consequently, the sensors 1 and 3 are regionally observable, while the sensors 2 and 4 are not, since the pairs  $(A, \bar{C}^2)$  and  $(A, \bar{C}^4)$  are not observable.

The Luenberger gains, both for  $MHE_{pre}$  and  $DMHE_{pre}$ , have been chosen such that the eigenvalues of  $\Phi = A - LC$ , for the centralized scheme, and  $\Phi^i = A - L^i \bar{C}^i$  for the distributed one, are equal to the values shown in the Table I. These values have been chosen to ensure that matrices  $\Phi$ ,  $\Phi^1$  and  $\Phi^3$  are Schur stable, and to keep the spectrum radius of  $\Phi^2$  and  $\Phi^4$  "small" enough.

TABLE I  
EIGENVALUES OF  $\Phi = A - LC$  AND  $\Phi^i = A - L^i \bar{C}^i$ .

	$\lambda_1$	$\lambda_2$	$\lambda_3$	$\lambda_4$
$MHE_{pre}$	0.9	0.6	0.7	0.8
$DMHE_{1-pre}$	0.9	0.6	0.7	0.8
$DMHE_{2-pre}$	0.45	0.58	$0.99 + 0.48i$	$0.99 - 0.48i$
$DMHE_{3-pre}$	0.9	0.6	0.7	0.8
$DMHE_{4-pre}$	0.93	0.45	$1.17 + 0.98i$	$1.17 - 0.98i$

We consider two different performance metrics for the evaluation of the algorithms: computation time  $\tau_t$ , of which we examine sum, minimum and maximum time, and root mean square error (RMSE):

$$RMSE = \left( \sum_{t=t_c}^{t_f} \frac{\|e_t\|^2}{t_f - t_c} \right)^{\frac{1}{2}}, \quad (28)$$

where  $t_c$  is the convergence of the algorithms,  $t_f$  is the final time instant, with  $t_f > t_c$ , and  $e_t = x_t - \hat{x}_{t|t}$  is the estimation error. In this example, the sampling period is chosen to be 1s, while  $t_f = 20$ s corresponds to the simulation duration.

The simulation has been carried out by using a setup implemented within the MATLAB R2019b environment and the solver *LMI Lab* [16] over a Linux Ubuntu 20.04 PC equipped with an Intel Core i7-7700HQ, 2.80 GHz.

Fig. 2 shows the evolution of the system (27) and the estimates by  $MHE_{pre}$  and  $DMHE_{pre}$  algorithms. It is worth noting that the last two states have unstable dynamics and are larger in terms of magnitude with respect to the first two, nevertheless one can consider that the convergence time is  $t = 5$ s.

Estimation errors corresponding to  $MHE_{pre}$  and  $DMHE_{pre}$  are represented in Fig. 3. It can be notice that the  $MHE_{pre}$  compensates the initial estimation error very fast, within a few iterations, since the system is collective observable. As pointed out in [6] for the DMHE, when using the  $DMHE_{pre}$  the estimates produced by sensor 2 (respectively 4) relative to the states  $x_{3,t}$ ,  $x_{4,t}$  (respectively  $x_{1,t}$ ,  $x_{2,t}$ ) also exhibit large errors

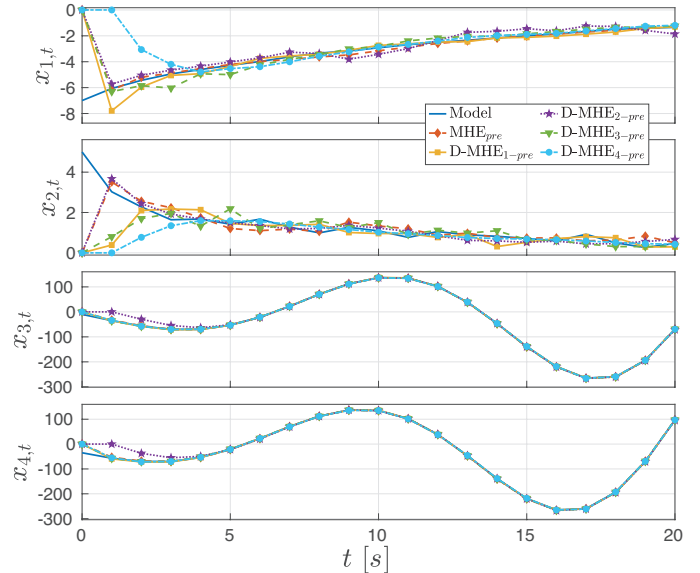


Fig. 2. Component of the states  $x_t = [x_{1,t} \ x_{2,t} \ x_{3,t} \ x_{4,t}]^T$  and the estimates  $\hat{x}_t = [\hat{x}_{1,t} \ \hat{x}_{2,t} \ \hat{x}_{3,t} \ \hat{x}_{4,t}]^T$  computed by  $MHE_{pre}$  and  $DMHE_{pre}$  algorithms.

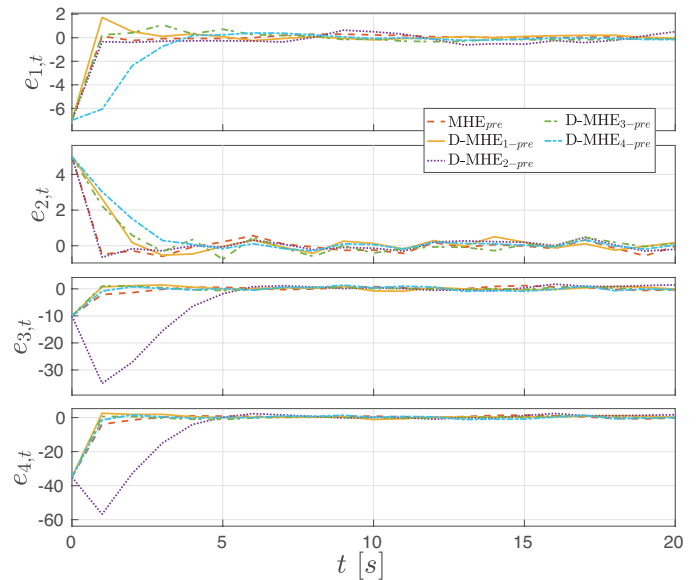


Fig. 3. Components of the estimation error  $e_t = [e_{1,t} \ e_{2,t} \ e_{3,t} \ e_{4,t}]^T = x_t - \hat{x}_{t|t}$  of the  $MHE_{pre}$  and  $DMHE_{pre}$  algorithms.

for  $t < 5$ . In fact, as previously mentioned, these states cannot be observed by these sensors using regional information. Effectiveness of the consensus terms also introduced in the  $DMHE_{pre}$  scheme is therefore illustrated here since all the estimation errors tend to converge to the same value despite the weak observability conditions of this example.

For the sake of clarity, in Fig. 2 and Fig. 3 the estimates and the estimation errors, respectively, from  $MHE$  and  $DMHE$  are omitted. It is also because  $e_t$  and  $x_t$  from  $MHE$  and  $DMHE$  are very similar with the ones from  $MHE_{pre}$  and  $DMHE_{pre}$ , respectively, as can be checked by comparing Fig. 3 with Fig. 4(a) in [6, p. 2468].

Table II shows the performance metrics that have been taken into account. In particular, the computation time is always lower for our proposed algorithms with respect to the ones without pre-estimation. This not come at a cost in terms of accuracy, because the RMSEs produced are similar amongst them. The maximum computation time for the algorithms without pre-estimation is greater than the sampling period. This means that there are some time steps where real time feasibility is not obtained. This can be checked on Fig. 4. Pre-estimation enables to reduce computation time by a factor close to 5 and would enable real time implementation without requiring fast optimization or *ad hoc* implementation.

TABLE II  
MINIMUM, MAXIMUM AND SUM OF THE COMPUTATION TIME  $\tau_t$  AND RMSE WITH  $t_c = 5s$  OF ALL ALGORITHMS COLLECTED IN THE SIMULATION.

	$\min\tau_t$	$\max\tau_t$	$\sum\tau_t$	RMSE
MHE	0.27	1.13	17.25	1.4
MHE <sub>pre</sub>	0.24	0.27	5.21	0.99
DMHE <sub>1</sub>	0.26	1.10	17.30	0.74
DMHE <sub>1-pre</sub>	0.24	0.29	5.20	0.84
DMHE <sub>2</sub>	0.26	1.02	16.30	1.53
DMHE <sub>2-pre</sub>	0.24	0.27	5.09	1.63
DMHE <sub>3</sub>	0.27	1.08	16.49	0.77
DMHE <sub>3-pre</sub>	0.25	0.29	5.18	0.76
DMHE <sub>4</sub>	0.26	1.10	17.24	0.90
DMHE <sub>4-pre</sub>	0.24	0.27	5.11	1.08

Figure 4 shows the computation time  $\tau_t$  of all algorithms. In this figure is more clear that our proposed schemes are less time demanding. In fact, the computation time mainly relies on the number of optimization parameters  $n_{op}$ , the evaluation of the cost and constraints. As it can be seen,  $n_{op}$  is the most important, since the algorithms without pre-estimation have  $n_{op} = n(N + 1)$ , while the ones with pre-estimation have only  $n_{op} = n$ . Thus, the latter ones do not depend on the horizon length  $N$ . Since  $N = t$  for  $t \leq 5$  and  $N = 5$  for  $t > 5$ , as expected, the computation time rises as  $N$  increases for MHE and DMHE.

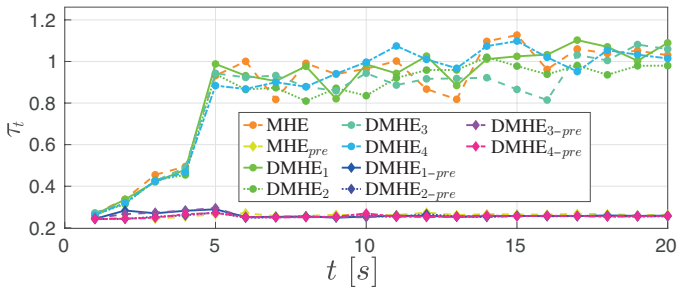


Fig. 4. Comparison amongst the computation times of all algorithms run in the simulation: MHE, MHE<sub>pre</sub>, DMHE, DMHE<sub>pre</sub>.

## VI. CONCLUSION

We presented a novel algorithm based on Moving Horizon Estimation (MHE) concept for distributed state estimation of discrete-time linear time-invariant systems. The use of a pre-estimation observer results in a significant reduction of the computation time. The proposed strategy has been validated via an illustrative numerical example. The estimation errors

produced by the DMHE<sub>pre</sub> are capable to converge even if some sensors in the network are not observable, due to the consensus terms embodied in the optimization problem. Moreover, the accuracy of the estimation errors is preserved (together with a reduced computation time), in the sense that it is comparable with the one of the original formulation [6].

Further improvements will include stability analysis of the estimation error provided by the proposed DMHE strategy with pre-estimation.

## ACKNOWLEDGMENT

The authors acknowledge support of MEyC Spain (contract PID2019-106212-RB-C41), the European Research Council and the UE ERDF (Advanced Grant OCONTSOLAR, Project ID: 789051).

## REFERENCES

- [1] A. Albert and L. Imsland, "Survey: mobile sensor networks for target searching and tracking," *Cyber-Physical Systems*, vol. 4, no. 2, pp. 57–98, 2018.
- [2] A. Pettiti, D. Di Paola, A. Rizzo, and G. Cicerelli, "Consensus-based distributed estimation for target tracking in heterogeneous sensor networks," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 6648–6653.
- [3] R. Vincent, D. Fox, J. Ko, K. Konolige, B. Limketkai, B. Morisset, C. Ortiz, D. Schulz, and B. Stewart, "Distributed multirobot exploration, mapping, and task allocation," *Annals of Mathematics and Artificial Intelligence*, vol. 52, pp. 229–255, 2008.
- [4] Q.-J. Kong, Y. Chen, and Y. Liu, "A fusion-based system for road-network traffic state surveillance: a case study of Shanghai," *IEEE Intelligent Transportation Systems Magazine*, vol. 1, no. 1, pp. 37–42, 2009.
- [5] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138–162, 2007.
- [6] 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.
- [7] M. Farina, G. Ferrari-Trecate, and R. Scattolini, "Distributed moving horizon estimation for nonlinear constrained systems," *International Journal of Robust and Nonlinear Control*, vol. 22, no. 2, pp. 123–143, 2012.
- [8] S. He, H.-S. Shin, S. Xu, and A. Tsourdos, "Distributed estimation over a low-cost sensor network: A review of state-of-the-art," *Information Fusion*, vol. 54, pp. 21–43, 2020.
- [9] G. Battistelli, "Distributed moving-horizon estimation with arrival-cost consensus," *IEEE Transactions on Automatic Control*, vol. 64, no. 8, pp. 3316–3323, 2018.
- [10] K. R. Muske, J. B. Rawlings, and J. H. Lee, "Receding horizon recursive state estimation," in *1993 American Control Conference*, 1993, pp. 900–904.
- [11] 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.
- [12] D. Sui, T. A. Johansen, and L. Feng, "Linear moving horizon estimation with pre-estimating observer," *IEEE Transactions on Automatic Control*, vol. 55, no. 10, pp. 2363–2368, 2010.
- [13] R. Suwantong, S. Bertrand, D. Dumur, and D. Beauvois, "Stability of a nonlinear moving horizon estimator with pre-estimation," in *2014 American Control Conference*, 2014, pp. 5688–5693.
- [14] D. Sui and T. A. Johansen, "Linear constrained moving horizon estimator with pre-estimating observer," *Systems & Control Letters*, vol. 67, pp. 40–45, 2014.
- [15] A. H. Jazwinski, *Stochastic processes and filtering theory*. Courier Corporation, 2007.
- [16] J. Löfberg, "Yalmip: A toolbox for modeling and optimization in MATLAB," in *2004 IEEE International Conference on Robotics and Automation*, 2004, pp. 284–289.