A New ℓ-step Neighbourhood Distributed Moving Horizon Estimator

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

Abstract— **This paper focuses on Distributed State Estimation over a peer-to-peer sensor network composed by possible low-computational sensors. We propose a new *ℓ-step Neighbourhood Distributed Moving Horizon Estimation* technique with fused arrival cost and pre-estimation, improving the accuracy of the estimation, while reducing the computation time compared to other approaches from the literature. Simultaneously, convergence of the estimation error is improved by means of spreading the information amongst neighbourhoods, which comes natural in the sliding window data present in the Moving Horizon Estimation paradigm.**

I. INTRODUCTION

Distributed algorithms have pervaded, in the last few years, many aspects of control engineering with applications for multi-robot systems, sensor networks, and others, covering topics such as control [1]–[4], state estimation [5]–[10], fault detection and mitigation [11], cyber-attack detection and mitigation on cyber-physical systems [12], [13], etc. Despite their different purposes, these topics share common characteristics as a consequence of the development in distributed schemes. Indeed, they face problems like scalability and communications between agents. If, on one hand, sharing more data could lead to have better performance in terms of accuracy, on the other hand the complexity could raise as well as the communication burden. In centralised frameworks, with the same purpose, usually a central unit manages all the resources involved in the network; nevertheless it can hardly deal with scalability issues due to physical and computational limitations. Moreover it is not robust with respect to the loss of the central unit.

For centralised state estimation problems, Moving Horizon Estimation (MHE) techniques have been investigated over the past years due to their capability to take into account constraints in the finite horizon "least-square" optimisation problem on which their formulation is based [14].

Within the context of distributed state estimation by sensor networks, each sensor is delegated to estimate (at least partially) the state of the system on the basis of local measurements and information received from its neighbours. Several works on Distributed MHE (DMHE) methods have been dedicated to guarantee stability of the estimation error dynamics, e.g. [6], [7], [10]. In [6] the authors have proposed a DMHE algorithm proving that it is stable even under weak observability conditions (due to *consensus on estimates* and a *consensus weight* term in the DMHE formulation). More recently, [10] introduced another consensus-based mechanism in a DMHE approach to fuse local arrival costs and guarantee stability of the estimation errors in a fully distributed way. The main drawback of the MHE paradigm is its computational load, since an optimisation problem must be solved online, at each instant, to compute the state estimate. This may be problematic when dealing with limited-computational resources, especially in the distributed case with low-computational sensors. Although adapted optimisation methods have been developed and can be used to reduce the computation time, an additional way of improvement concerns the structure of the optimisation problem itself. One idea, *inter alia*, is to introduce in the MHE formulation the use of an observer, that would enable to decrease the number of optimisation parameters. This has been first introduced in [15] for centralised linear MHE, considering a Luenberger observer, and then extended to the centralised non-linear case in [16], and to the distributed linear case in [8] as an extension of the DMHE formulated in [6].

This paper extends the approach proposed by the authors in [8] to the DMHE formulation of [10] which has proven to obtain more general stability results as well as enhanced performance compared to [6]. The current paper leads to a reduced computation time due to a pre-estimating observer. Another contribution concerns the improvement of the convergence of the estimation error by mitigating unobservability issues. This situation could arise in sensor networks when some nodes may have no sensing capacities (inactive sensors), or are able to only measure some parts of the state of the system that would make it non observable using only these sensors. For this purpose, the new proposed DMHE technique exploits the exchanges of information amongst local nodes based on an ℓ-step neighbourhood information spreading mechanism.

The paper is structured as follows. Section II introduces the problem formulation and proposed communication protocol. The proposed DMHE algorithm is presented in Section III. Before concluding remarks, simulations examples are presented and analysed in Section IV.

II. PROBLEM STATEMENT

This section describes the state estimation problem of a system over a sensor network. The dynamical system

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}@l2s.centralesupelec.fr).

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).

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).

is described as a discrete-time linear time-invariant (LTI) system

$$x_{t+1} = Ax_t + w_t, \tag{1}$$

with $x_t \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$ representing the state and $w_t \in \mathcal{W} \subseteq \mathbb{R}^{n_x}$ the disturbance or unknown input. The measurements are performed by heterogeneous sensors, and can be modelled by

$$y_t^i = C^i x_t + v_t^i, \qquad i = 1, \ldots, M \tag{2}$$

where $y_t^i \in \mathbb{R}^{n_y^i}$ is the measurement vector, $v_t^i \in \mathcal{V}^i \subseteq \mathbb{R}^{n_y^i}$ the measurement noise and $M > 1$ the number of the sensors composing the network. The superscript i refers to the sensor i. The sets $\mathcal{X}$, $\mathcal{W}$ and $\mathcal{V}^i$ are assumed to be convex sets.

In the distributed architecture, each sensor also shares data with its neighbours. The sensor network is described by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \{1, 2, \ldots, M\}$ is the set of all nodes (sensors) and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ is the set of all edges (communication links). Specifically, the pair $(i, j) \in \mathcal{E}$ is defined if and only if the sensor j can receive information from the sensor i. The *neighbourhood* $\mathcal{N}^i$ of the sensor i is denoted by $\mathcal{N}^i = \{j \in \mathcal{N} : (i, j) \in \mathcal{E}\}$ and its cardinality $M^i = \text{card}(\mathcal{N}^i)$.

In addition to the shared data coming from the neighbour sensors in $\mathcal{N}^i$, each sensor i could exploit past information from other sensors $j \notin \mathcal{N}^i$, if there exists a path that connects these sensors to sensor i.

For this reason, denoting $d(i, j)$ the distance, in terms of number of edges, between node i and j, we define the *ℓ-step neighbourhood* $\mathcal{N}_\ell^i = \{j \neq i \in \mathcal{N} : d(i, j) \leqslant \ell\}$, i.e. the set of sensors $j \in \mathcal{N}$ for which there is a path of length at most ℓ to node i. Notice that $\mathcal{N}_1^i = \mathcal{N}^i$.

A. Communication protocol

As mentioned above, the network is composed by possibly different types of nodes, some of them with no sensing capabilities, i.e. $C^i = 0$, or at least partially, meaning that a sensor may observe only some part of the state vector of the system, i.e. the pair (A, C^i) is possibly not detectable. Moreover, the network could be deployed such that some neighbourhoods are composed only by nodes resulting in weak local or regional observability properties [6], meaning that the pair $(A, \bar{C}^i)$ could be not detectable, where $\bar{C}^i$ is the regional output matrix, i.e. $\bar{C}^i = [(C^i)^\top, (C^{j_1})^\top, \ldots, (C^{j_{M^i}})^\top]^\top$, $\{j_1, \ldots, j_{M^i}\} \in \mathcal{N}^i$. Therefore, with the aim to enhance collective observability [6] by the network, it is proposed that each node $i \in \mathcal{N}$ exploits measurements received from its *ℓ-step neighbourhood* $\mathcal{N}_\ell^i$. This section details how information coming from $\mathcal{N}_\ell^i$ will be considered in the formulation of the DMHE, by choosing $\ell = N$, where N is the length of the horizon of past information considered for state estimation by the algorithm.

The communication network uses a single-hop routing protocol, in which it is assumed that there is no delay or packet losses. Moreover, a time synchronisation is required for all sensors in the network, to make them able to exchange data with their neighbours at each time instant.

Each node $i \in \mathcal{N}$ keeps information received from each of its *in-neighbours nodes* in a time-sliding batch of size N and relays this information to *out-neighbours nodes* at the next time instant. Old information (i.e. received from time instant $t_r < t - N$) is removed from the batch. Therefore, at time t each node disposes of past measurements from nodes in its ℓ-step neighbourhood over the time window $[t - N, t]$. Since dealing with a single-hop routing protocol, each sensor $i \in \mathcal{N}$ receives information only from its neighbours $j \in \mathcal{N}^i$ at each time t. Let $\bar{y}^i$ denote the measurements collected by sensor i from all the nodes $j \in \mathcal{N}^i$. Then, at the time instant t, these collected measurements from nodes j are

$$\bar{y}^i_{[t-N,\ldots,t]} = \begin{bmatrix} y_{t-N}^{j_1} & \cdots & y_t^{j_1} \\ \vdots & \ddots & \vdots \\ y_{t-N}^{j_{M^i}} & \cdots & y_t^{j_{M^i}} \end{bmatrix}, \quad \{j_1, \ldots, j_{M^i}\} \in \mathcal{N}^i.$$

At the same time, each sensor $j \in \mathcal{N}^i$ has data collected from its own neighbours $z \in \mathcal{N}^j$, with $z \neq i$, from the previous time step $t - 1$, i.e.

$$\bar{y}^j_{[t-N,\ldots,t-1]} = \begin{bmatrix} y_{t-N}^{z_1} & \cdots & y_{t-1}^{z_1} \\ \vdots & \ddots & \vdots \\ y_{t-N}^{z_{M^j}} & \cdots & y_{t-1}^{z_{M^j}} \end{bmatrix}, \quad \{z_1, \ldots, z_{M^j}\} \in \mathcal{N}^j.$$

that they can share with sensor i. This philosophy can be reiterated back in time, and so along the communication links in $\mathcal{N}_{t-k}^i$, $\forall k = t - N, \ldots, t - 1$, with a maximum of $N = \ell$ back steps.

To summarise, the node i has the collection of data $\left\{\bar{y}^i_{[t-N,\ldots,t]}, \bar{y}^j_{[t-N,\ldots,t-1]}, \ldots, \bar{y}^z_{t-N}\right\}$, with $j, \ldots, z \in \mathcal{N}_\ell^i$, which is useful in the local MHE optimisation problem to improve the accuracy of the estimates.

B. Problem formulation

We define as a *Poorly-Observing Sensor Network* a network containing at least one node $i \in \mathcal{N}$ having any of the following characteristics

- it has no sensing capabilities, i.e. $C^i = 0$;
- it has sensing capabilities and can provide a measurement on the state of the system, i.e. $C^i \neq 0$, but the pair (A, C^i) still remains non detectable;
- nodes in its neighbourhood are such that the pair $(A, \bar{C}^i)$ is non detectable.

The problem addressed in this paper, namely *Distributed State Estimation over a Poorly-Observing Sensor Network* can be stated as follows.

Given the discrete-time LTI system (1), the sensor network $\mathcal{G}$ with linear sensors as in (2), under the assumptions that:

- the pair (A, C) is observable, where $C = \text{col}(C^i)$ with $i \in \mathcal{N}$ is the collective output matrix, i.e. $C = [(C^1)^\top, \ldots, (C^M)^\top]^\top$;
- the graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is strongly connected, i.e. every node is reachable from every other node.

The role of each sensor $i \in \mathcal{N}$, at each time t, is to (possibly) get measurement on the system, to exchange information among neighbour nodes $\mathcal{N}^i$ and to process locally available information in order to determine a local estimate $\hat{x}_t^i$ of the real state of the system x_t.

III. Proposed DMHE technique

This section presents the proposed DMHE approach. It extends the one of [10] with consensus on the arrival costs, by accounting for information from ℓ-step neighbourhoods and taking advantage of a pre-estimating observer to reduce computation time.

A. Local optimisation problem

At time t, let $\hat{x}_{t-N|t}^i, \ldots, \hat{x}_{t|t}^i$ be the sequence of estimates of the state of system (1) to be computed by each sensor $i \in \mathcal{N}$ over a given past horizon of length $N \geqslant 1$. The estimate of the state x_t to be provided by each sensor at time t corresponds to $\hat{x}_t^i = \hat{x}_{t|t}^i$. To do so, a local minimisation problem can be formulated for each sensor i as follows

$$\hat{x}_{t-N|t}^i = \arg\min_{\hat{x}_{t-N}^i} \quad J_t^i \tag{3}$$

$$\text{s.t.} \quad \hat{x}_{k+1}^i = A\hat{x}_k^i + L^i\hat{v}_k^i + \sum_{j\in\mathcal{N}_{t-k}^i} L^j\hat{v}_k^j, \tag{4}$$

$$\hat{v}_k^j = C^j\hat{x}_k^i - y_k^j, \quad j \in \{i\} \cup \mathcal{N}_{t-k}^i, \tag{5}$$

$$\hat{x}_k^i \in \mathcal{X}, \tag{6}$$

$$\hat{v}_k^j \in \mathcal{V}^j, \quad j \in \{i\} \cup \mathcal{N}_{t-k}^i, \tag{7}$$

$$\forall k = t-N, \ldots, t$$

The sequence of state estimates $\hat{x}_{t-N+1|t}^i, \ldots, \hat{x}_{t|t}^i$ is then computed from the optimal solution $\hat{x}_{t-N|t}^i$ and using (4).

The main difference in this formulation is that a Luenberger observer is used in (4) instead of the state equation of the system, as classically used in MHE formulations and in [10], which requires to consider the disturbance sequence over the past horizon as additional optimisation parameters. This reduces the computation cost, while simultaneously preserving the accuracy of the state estimate. Under the assumption that the gain L^i is computed such that

$$\Phi^i = A - L^iC^i, \ \ \forall i \in \mathcal{N} \tag{8}$$

is Schur stable, then, in order to mitigate the effects on the estimation errors at certain frequencies or to increase robustness for each frequency, the gain L^i can be computed off-line according to some criteria, for example $\mathcal{H}_2$, $\mathcal{H}_\infty$ [17, p. 293]. Note that the assumption of (8) being Schur can be satisfied only when the pair (A, C^i) is observable, otherwise, as *extrema ratio*, it is sufficient to design L^i so as to keep spectrum radius of Φ^i as low as possible.

Another difference w.r.t. [10] is that the optimisation problem (3) uses the set $\mathcal{N}_\ell^i$ instead of $\mathcal{N}^i$, leading to improve the estimation accuracy. In fact, $\mathcal{N}_\ell^i$ appears also in the objective function J_t^i defined as

$$\begin{aligned} J_t^i = &\Gamma_t^i(\hat{x}_{t-N|t}^i) + \sum_{k=t-N}^{t} \left\| y_k^i - C^i\hat{x}_k^i \right\|_{R^i}^2 \\ &+ \sum_{k=t-N}^{t} \sum_{j\in\mathcal{N}_{t-k}^i} \left\| y_k^j - C^j\hat{x}_k^i \right\|_{R^j}^2 \end{aligned} \tag{9}$$

where the weight matrices R^i (resp. R^j) can be chosen as the inverse of the covariance matrix of the measurement noise. The first term is the so called *initial penalty* function $\Gamma_t^i(\cdot)$, known in the MHE environment as *arrival cost*. It is assumed to be non negative and it summarises the effect of the past measurements, before time $t-N$. Further details on the arrival cost are provided in Section III-B, because it plays a major role in the convergence and the performance of the algorithm.

Note that, when the current time instant t is $t \leqslant N$ then the horizon length N is set to $N = t$.

Remark 1: The local optimisation problem can be formulated using information coming only from neighbour sensors, in other words $\ell = 1$. For later comparisons, we denote by DMHE$_{pre}^{\ell}$ the minimisation problem (3) having $\mathcal{N}_\ell^i$, with $\ell = N$, and by DMHE$_{pre}^{1}$ the one with $\mathcal{N}_\ell^i$ equal to $\mathcal{N}^i$, i.e. $\ell = 1$. Indeed, DMHE$_{pre}^{1}$ could be a combination of the methods of [10] and [8].

B. Fused arrival cost

The objective function (9) contains the arrival cost term $\Gamma_t^i(\cdot)$. In the MHE approach, it usually penalizes deviations from some *a priori* information $\bar{x}_{t-N}^i$ on the state at the beginning of the horizon as detailed in [14] and can be formulated as in [10] by

$$\Gamma_t^i(x) = \left\| x - \bar{x}_{t-N}^i \right\|_{P_{t-N}^i}^2, \tag{10}$$

with P_{t-N}^i a positive definite weight matrix. The *a priori* state $\bar{x}_{t-N}^i$ can be computed as a one step prediction from the solution of the optimisation problem at the previous instant. In classical (D)MHE approaches, this prediction is done using the state equation. In the proposed algorithm with pre-estimation formulation, this prediction is computed as

$$\bar{x}_{t-N}^i = (A - L^iC^i)\hat{x}_{t-N-1|t-1}^i. \tag{11}$$

The matrix P_{t-N}^i is defined as the inverse of the covariance matrix of the prediction $\bar{x}_{t-N}^i$ and can be computed recursively, as we explain later on, initialised as P_0^i in order to quantify the confidence on the initial a priori information $\bar{x}_0^i$.

In a distributed setting, this cost has an essential role to propagate information amongst the sensors in the network in order to ensure convergence of the state estimations to the real state of the system, since the local observability for one node or observability amongst the neighbourhood depends on the network topology and sensing capabilities. As in [10], the idea is then to fuse the arrival costs of the neighbourhood

$\mathcal{N}^i$ in a convex combination

$$\Gamma_t^i(x) = \pi^{i,i} \left\| x - \bar{x}_{t-N}^i \right\|_{P_{t-N}^i}^2 + \sum_{j \in \mathcal{N}^i} \pi^{i,j} \left\| x - \bar{x}_{t-N}^j \right\|_{P_{t-N}^j}^2 , \tag{12}$$

where all the weights $\pi^{i,i}$ and $\pi^{i,j}$ are strictly positive and fulfil the condition

$$\pi^{i,i} + \sum_{j \in \mathcal{N}^i} \pi^{i,j} = 1, \quad \forall i \in \mathcal{N}. \tag{13}$$

Consequently, the initial penalty function is defined as a *consensus on the arrival costs* by means of relation (12) ensuring that the local arrival cost is a weighted average of the local arrival costs from neighbours.

In the following, the covariance matrix P_{t-N}^i is recursively updated using only local information available at time t to sensor i. Consider the observability matrix F^i associated to the pair $(A - L^iC^i, C^i)$ along the horizon length N and its relative collective output weight matrix

$$F^i = \begin{bmatrix} C^i \\ C^i(A - L^iC^i) \\ \vdots \\ C^i(A - L^iC^i)^N \end{bmatrix}, \ \Psi^i = \text{diag}(\underbrace{R^i, \ldots, R^i}_{N+1 \text{ times}}).$$

Then a preliminary consensus weight matrix can be computed by using only data locally available at node i from the previous time instant

$$\Omega_{t-N-1}^i = \pi^{i,i} P_{t-N-1}^i + \sum_{j \in \mathcal{N}^i} \pi^{i,j} P_{t-N-1}^j + \left(F^i\right)^\top \Psi^i F^i. \tag{14}$$

Following [10], we now introduce a scalar α such that $0 < \alpha < 1$ and a positive definite matrix S for any t, and adapt the equations to the proposed algorithm with pre-estimation. Then the updated consensus covariance matrix is defined by

$$P_{t-N}^i = \frac{\alpha}{8} \left[A_L^i \left(\Omega_{t-N-1}^i\right)^{-1} \left(A_L^i\right)^\top + S^{-1} \right]^{-1}, \tag{15}$$

where $A_L^i = A - L^iC^i$.

C. DMHE algorithm

Finally, we can describe the *modus operandi* of the proposed distributed algorithm. First of all, it is worth to mention that the steps of the algorithm could be run in a parallel scheme by each sensor $i \in \mathcal{N}$, after they have sent and received the information from the neighbours at each time t, with the assumptions on the network and communication protocol provided in Section II-A.

The steps of the DMHE_{pre}^{ℓ} procedure are described in Algorithm 1. To get the one-step DMHE_{pre}^{1} procedure, it is sufficient to remove the step 10 and to use $\mathcal{N}^i$ instead of $\mathcal{N}_\ell^i$.

Remark 2: All nodes are synchronised at the step 12, since each sensor i needs the data from its neighbours $j \in \mathcal{N}^i$. This is the only communication step.

Remark 3: The path length ℓ of the ℓ-step neighbourhood $\mathcal{N}_\ell^i$ can also be chosen lower than the horizon length N of the DMHE, i.e. $1 \leqslant \ell \leqslant N$. One can design then the DMHE_{pre}^{ℓ} in order to have a good trade-off between the accuracy of the estimation and the amount of data exchanged in the network. This trade-off depends on the observability conditions and the network topology.

Algorithm 1 DMHE_{pre}^{ℓ} procedure

1: **Off-line:** $\forall i \in \mathcal{N}$
2: **compute** the Luenberger gain L^i
3: **store** the *a priori* initial estimation $\hat{x}_{0|0}^i = \hat{x}_0$ of x_0 and the covariance matrix $P_0^i = P_0$ of x_0
4: **receive** from the neighbours $j \in \mathcal{N}^i$: L^j, C^j, R^j, $\mathcal{V}^j$
5: **Initialization:** $\forall i \in \mathcal{N}$, at the first time step $t = 0$
6: **collect** a first local measurement y_0^i
7: **receive** from the neighbours $j \in \mathcal{N}^i$ their measurements y_0^j
8: **Online:** $\forall i \in \mathcal{N}$, $\forall t > 0$
9: **collect** the local measurement y_t^i
10: **gather** past information received at time $t-1$ from $j \in \mathcal{N}^i$, as in Section II-A
11: **compute** the prediction $\bar{x}_{t-N}^i$ and the consensus weight matrix P_{t-N}^i according to (11) and (15), resp.
12: **receive** from the neighbours $j \in \mathcal{N}^i$ the collected, gathered and computed data in the steps 9, 10 and 11
13: **compute** the fused arrival cost Γ_t^i according to (12)
14: **solve** the local MHE, minimising J_t^i as in (9) subject to the constraints (4)-(7)
15: **store** the solution $\hat{x}_{t-N|t}^i$ and the estimate $\hat{x}_{t|t}^i$

IV. Simulations

In this section an evaluation of the proposed DMHE algorithm is provided via simulations examples. To compare it with existing results in literature, the scenario in [10] is considered. The goal is to track a 2D moving target using a sensor network, that could model, for example, a distributed camera network. As illustrated in Fig.1, the network is composed of 100 sensors randomly disposed with a uniform distribution on a plane of $[-500, 500] \times [-500, 500]$ m, in which only the green nodes are active sensors (i.e. with sensing capabilities), while the white nodes are inactive sensors (i.e. null output matrix). In the following it is considered that a communication link between two nodes exists if the distance between them is less than a given communication radius equal to 160 m.

The 2D moving target is modelled as a double integrator system. The state of the system is represented by $x = [p_x \ p_y \ v_x \ v_y]^\top$ which corresponds to the Cartesian coordinates of its position and velocity vectors. The dynamics of the target are described by model (1) with $A = \begin{bmatrix} I_2 & T_sI_2 \\ 0 & I_2 \end{bmatrix}$, where $T_s = 1$ s is the sampling time used for discretization of the continuous-time dynamics of the target. The input disturbance w_t in (1) is a four dimensional vector and is assumed to be modelled by a noise vector with uniform distribution

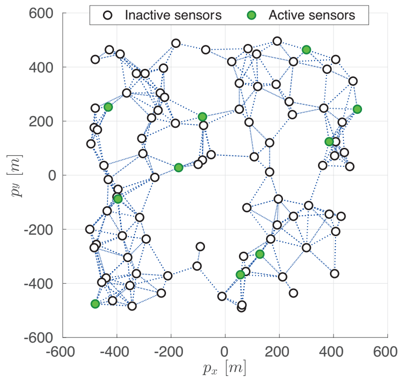

Fig. 1. Topology of the sensor network composed by 100 nodes.

in $\mathcal{W} = [-0.5, 0.5] \times [-0.5, 0.5] \times [-0.5, 0.5] \times [-0.5, 0.5]$. The 10 active sensors provide measurements of the target's position in conformity with the matrix $C^i = [I_2 \quad \mathbf{0_{2,2}}]$ while the remaining 90 inactive sensors have no ability to measure, i.e. their output matrix is $C^i = \mathbf{0}_{2,4}$. The measurement noise v_t^i of each sensor i is a two dimensional vector with a uniform distribution in $\mathcal{V}^i = [-10, 10] \times [-10, 10]$.

Two simulation cases are further analysed.

Case 1. A first simulation considers a horizon length $N = 4$; the *a priori* information about the state $\hat{x}_0^i$ is set equal to $[0 \ 0 \ 0 \ 0]^\top$ for each node; the initial arrival cost weight matrix is $P_0^i = \text{diag}(10^{-5}, 10^{-5}, 1, 1)$, taking into account different magnitudes of the states; the matrices Q and R^i are set for each sensor i as the inverse of the covariance matrices of w_t and v_t^i, respectively. Q is used in [6], [8] and [10] for (D)MHE algorithms without pre-estimation. This weight matrix penalises the norm on the sequence of disturbance input terms in the cost function of these algorithms. All these parameters are identically set in all the considered algorithms. The consensus weights $\pi^{i,j}$ are chosen to be equal among the neighbourhood, satisfying (13).

In order to compare the proposed DMHE$^1_{pre}$ (namely Distributed Moving Horizon Estimation with pre-estimation) and DMHE$^\ell_{pre}$ (namely DMHE with ℓ-neighbourhood diffusion and pre-estimation) algorithms with existing techniques, the simulation has been run also for the centralised MHE of [14] and the DMHE of [6], [8] and [10]. To avoid confusion, the DHME in [8] is an extension of the one of [6] using the pre-estimation strategy and it is called DMHE$_{pre}$ in [8].

The performance metrics that have been taken into account are the *Position Root Mean Square Error* (PRMSE) averaged over the M =100 nodes of the network, denoted by $\text{PRMSE}(t) = \frac{1}{M} \sum_{i \in \mathcal{N}} \left\| C^i \left(x_t - \hat{x}^i_{t|t} \right) \right\|$, and the *computation time* $\tau(t)$ averaged also over the entire network.

Figure 2 shows the time behaviour of the PRMSE of all considered algorithms. The proposed DMHE$^1_{pre}$ technique offers similar results as the DMHE of [10], with a faster convergence (about 4 seconds) with respect to the DMHE of [6] and [8] (about 18 seconds to converge). Further, we can notice that DMHE$^\ell_{pre}$ ensures improved performance in terms of convergence time among the considered distributed algorithms.

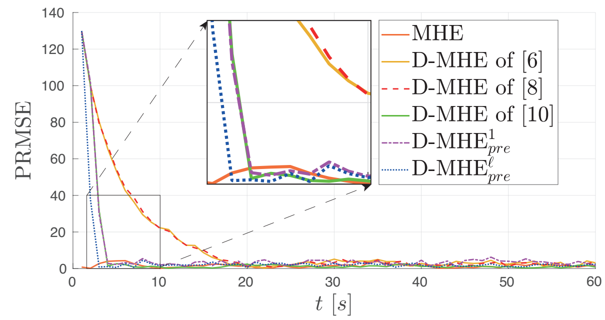

Fig. 2. PRMSE time behaviour comparison.

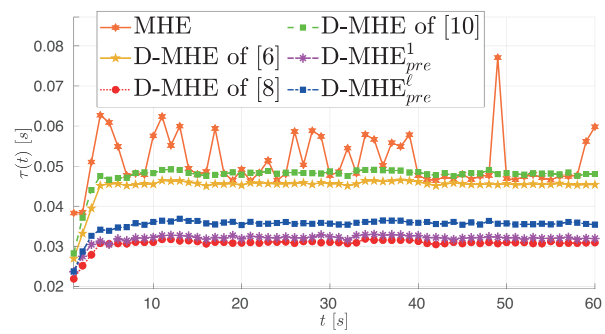

Fig. 3. Computation time comparison.

Evaluating the computation times in Fig. 3 shows that the algorithms with pre-estimation (red, purple and blue dots) are always less computationally demanding compared to their respective version without pre-estimation (yellow and green dots). Indeed, adding the pre-estimation reduces the computation time of about 30%. In particular, the proposed DMHE$^1_{pre}$ technique (purple dots) converges faster and has comparable performances with the DMHE of [8]. Moreover, it is worth to notice that DMHE$^\ell_{pre}$ has the best convergence time from all the considered approaches (see Fig. 2) and needs almost the same computation time as DMHE$^1_{pre}$ and DMHE of [8].

Case 2. A second simulation of nine trials has been performed using the same parameters but changing the fixed window size $N = \{2, 3, \ldots, 10\}$, to the end of evaluating how the horizon length affects the performance of the considered DMHE algorithms. In addition, the initial state of the system x_0 is randomly generated with uniform distribution over the plan $[-500, 500] \times [-500, 500]$ m and in velocity $[-1, 1] \times [-1, 1]$ m/s. The simulation duration is chosen to be $t_f = 20s$.

To emphasise the influence of the horizon length N on the estimations, Fig. 4 shows the evolution of the sum of the PRMSE, i.e. $\sum_{t \in (0, t_f]} \text{PRMSE}(t)$, of each algorithm with respect to N. As for $N = 4$, the proposed technique DMHE$^\ell_{pre}$ has always the best performance with respect to the distributed algorithms. Moreover, even considering information belonging to neighbours, i.e. $\mathcal{N}^i_\ell = \mathcal{N}^i$, the DMHE$^1_{pre}$ method has comparable results in terms of PRMSE with the

DMHE of [10]. In fact, this is noticeable in the zoom part on Fig. 4 because the PRMSEs are one above the other.

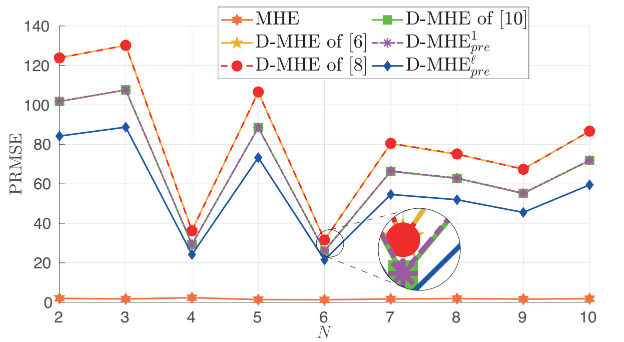

Fig. 4. Comparison of the sum of PRMSE for a different horizon length N.

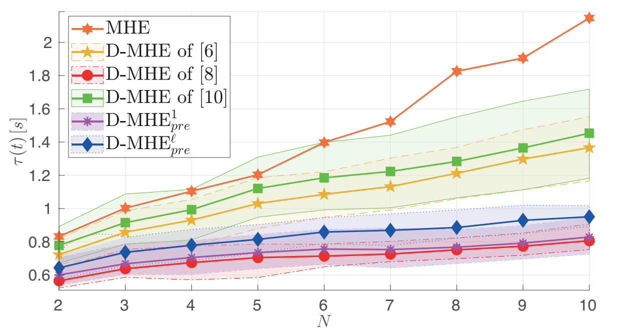

Fig. 5. Comparison of the sum of computation time for a different horizon length N.

Finally, the Fig. 5 points out the differences amongst the sum of the computation time τ, i.e. $\sum_{t\in(0,t_f]}\tau(t)$, of all algorithms when changing the horizon length N. As expected, the algorithms with pre-estimation are less computation demanding for every N since their local optimisation problems involves less optimisation parameters. Another significant aspect to observe in Fig. 5 is that the difference on τ amongst algorithms with and without pre-estimation increases with N. In addition, the Fig. 5 shows also the bounds representing the minimum and maximum computation time of the DMHE algorithms. It can be noticed that these bounds are tighter and less varying w.r.t. N for the algorithms with pre-estimation.

To summarise, the numerical simulations have shown that the proposed DMHE algorithm, DMHE$^\ell_{pre}$, with pre-estimation and ℓ-step neighbourhood information diffusion, is able to solve the considered distributed estimation problem while, at the same time, it turns out to be lower computation demanding and gives better estimation accuracy w.r.t. other existing methods [6], [10].

V. Conclusions and Perspectives

The proposed ℓ-step neighbourhood Distributed Moving Horizon Estimation (DMHE) algorithm is able to solve the Distributed State Estimation problem for a linear system over a poorly-observing sensor network. In particular, the simulation results have shown that the proposed DMHE technique with pre-estimation DMHE$^1_{pre}$ is able to converge with analogous performance with respect to the DMHE of [10] and, simultaneously, to reduce by a significant factor the computation time. The best result comes from the ℓ-step neighbourhood DMHE algorithm DMHE$^\ell_{pre}$ that, spreading out information from neighbourhood to neighbourhood, both improves accuracy (in terms of the Position Root Mean Square Error) and reduces computation time.

Current work concerns stability and robustness analysis of the proposed approach. Further developments focus on implementing the proposed ℓ-step neighbourhood Distributed Moving Horizon Estimation on a real application with multi-sensor system.

References

[1] P. Segovia, V. Puig, E. Duviella, and L. Etienne, "Distributed model predictive control using optimality condition decomposition and community detection," *Journal of Process Control*, vol. 99, pp. 54–68, 2021.

[2] A. Bono, G. Fedele, and G. Franzè, "A distributed model predictive control strategy for vehicle teams in uncertain narrowed environments," in *24th International Conference on Emerging Technologies and Factory Automation*, 2019, pp. 927–932.

[3] D. Ding, Q.-L. Han, Z. Wang, and X. Ge, "A survey on model-based distributed control and filtering for industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 2483–2499, 2019.

[4] C. Conte, C. N. Jones, M. Morari, and M. N. Zeilinger, "Distributed synthesis and stability of cooperative distributed model predictive control for linear systems," *Automatica*, vol. 69, pp. 117–125, 2016.

[5] A. Simonetto, D. Balzaretti, and T. Keviczky, "A distributed moving horizon estimator for mobile robot localization problems," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 8902–8907, 2011.

[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] A. Venturino, S. Bertrand, C. Stoica Maniu, T. Alamo, and E. F. Camacho, "Distributed moving horizon estimation with pre-estimating observer," in *24th International Conference on System Theory, Control and Computing*, 2020, pp. 174–179.

[9] J. Zeng and J. Liu, "Distributed moving horizon estimation subject to communication delays and losses," in *American Control Conference*, 2015, pp. 5533–5538.

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

[11] Y. Wu, Z. Wang, and Z. Huang, "Distributed fault detection for nonlinear multi-agent systems under fixed-time observer," *Journal of the Franklin Institute*, vol. 356, no. 13, pp. 7515–7532, 2019.

[12] K. Gheitasi, M. Ghaderi, and W. Lucia, "A novel networked control scheme with safety guarantees for detection and mitigation of cyber-attacks," in *18th European Control Conference*, 2019, pp. 1449–1454.

[13] A. J. Gallo, M. S. Turan, F. Boem, T. Parisini, and G. Ferrari-Trecate, "A distributed cyber-attack detection scheme with application to DC microgrids," *IEEE Transactions on Automatic Control*, vol. 65, no. 9, pp. 3800–3815, 2020.

[14] K. R. Muske, J. B. Rawlings, and J. H. Lee, "Receding horizon recursive state estimation," in *American Control Conference*, 1993, pp. 900–904.

[15] D. Sui and T. A. Johansen, "Linear constrained moving horizon estimator with pre-estimating observer," *Systems & Control Letters*, vol. 67, pp. 40–45, 2014.

[16] R. Suwantong, S. Bertrand, D. Dumur, and D. Beauvois, "Stability of a nonlinear moving horizon estimator with pre-estimation," in *American Control Conference*, 2014, pp. 5688–5693.

[17] G.-R. Duan and H.-H. Yu, *LMIs in control systems: analysis, design and applications*. CRC press, 2013.