

# Distributed Estimation and Control for LTI systems under Finite-Time Agreement

Camilla Fioravanti<sup>1</sup>, Evagoras Makridis<sup>2</sup>, Gabriele Oliva<sup>1\*</sup>, Maria Vrakopoulou<sup>2</sup>, and Themistoklis Charalambous<sup>2</sup>

**Abstract**—This paper considers a strongly connected network of agents, each capable of partially observing and controlling a discrete-time linear time-invariant (LTI) system that is jointly observable and controllable. Additionally, agents collaborate to achieve a shared estimated state, computed as the average of their local state estimates. Recent studies suggest that increasing the number of average consensus steps between state estimation updates allows agents to choose from a wider range of state feedback controllers, thereby potentially enhancing control performance. However, such approaches require that agents know the input matrices of all other nodes, and the selection of control gains is, in general, centralized. Motivated by the limitations of such approaches, we propose a new technique where: (i) estimation and control gain design is fully distributed and finite-time, and (ii) agent coordination involves a finite-time exact average consensus subroutine, allowing arbitrary selection of the convergence rate of the overall asymptotic estimation process despite the estimator’s distributed nature. We verify our methodology’s effectiveness using illustrative numerical simulations.

**Index Terms**—Distributed Estimation, Distributed Control, Distributed Gain Design, LTI Systems, Finite-time Agreement

## I. INTRODUCTION

IN THE LAST DECADES, several works in the literature focused on solving the distributed estimation and control problem, mainly relying on the development of linear observers [1]–[8]. In [2], the authors establish the necessary conditions for observability in the dependence of the plant model and the underlying communication graph, while in [3] the existence of a distributed observer is guaranteed under the detectability and connectivity conditions. In [4], [5], a state-augmented observer is proposed, and necessary and sufficient conditions for their stabilizability are derived. More recently, in [6], Kalman’s canonical observational decomposition to a multi-sensor environment is introduced.

Although all the aforementioned works provided suitable solutions to the distributed estimation problem, none of them was able to directly control the convergence rate. This issue is overcome in [9], [10], [15], which developed the idea of the observability decomposition for the construction of

**Table I:** Comparison of this work against the state of the art.

Approach	Discrete Time	Consensus	Selection of Estimation Gains	Selection of Control Gains
[9] - 2016	✗	✗	uncoordinated	n/a
[5] - 2017	✓	✗	centralized	n/a
[10] - 2019	✗	✗	uncoordinated	n/a
[11] - 2019	✗	✗	uncoordinated	n/a
[12] - 2019	✓	asymptotic	uncoordinated	n/a
[13] - 2020	✗	✗	✗	✗
[14] - 2022	✓	asymptotic	centralized	uncoordinated
<b>This work</b>	✓	<b>finite-time</b>	<b>coordinated</b>	<b>coordinated</b>

a decomposed diffusive coupling. Specifically, in [10] the authors exploit linear matrix inequality (LMI) techniques to compute the observer gains, while in [15] the distributed observer design is improved by eliminating the dependence on the communication topology among the observers. In [12], the estimation error of discrete-time, jointly observable LTI system whose observations are distributed across a time-varying network, converges exponentially fast to zero at a fixed but arbitrarily chosen rate, assuming that the graph is strongly connected. Specifically, the agents are allowed to perform a given number of rounds of distributed agreement in between state updates. The continuous-time counterpart was presented in [11]. In [13], the authors proposed a distributed observer-based control system where each agent can independently adjust the controlled output to any desired set point value for a continuous-time system. However, although the solutions presented in [9]–[13], [15] succeed in the convergence rate assignment and can be implemented in a distributed manner, they all always require a degree of centralized coordination for the design of the estimation gains. In Table I, we analyze the significant aspects of distributed gains design shown by the aforementioned works. Interestingly, only the approaches in [13], [14] and in this paper also account for the control of the system, while all other approaches only focus on state estimation. Indeed, our approach is the only one that allows selecting the local controller gains. In contrast to these approaches, in our paper, we aim to develop a strategy where agents’ local estimation and control gains are designed and executed in a fully distributed manner. In particular, we took inspiration from [14], in which a distributed estimation and control scheme for LTI systems is developed whereby agents are allowed to perform a given number of asymptotic average consensus iterations between successive time updates. Additionally, the authors show how the frequency of information exchange between neighboring agents (consensus steps)

<sup>1</sup>Department of Engineering, University Campus Bio-Medico of Rome, Via Alvaro del Portillo, 21 - 00128 Roma, Italy. E-mails: {c.fioravanti, g.oliva}@unicampus.it.

<sup>2</sup>Department of Electrical and Computer Engineering, School of Engineering, University of Cyprus, 2109 Aglantzia, Nicosia, Cyprus. E-mails: {surname.name}@ucy.ac.cy.

The work of E. Makridis and T. Charalambous has been partly funded by MINERVA, European Research Council (ERC) under the EU’s Horizon 2022 research and innovation programme (Grant Agreement No. 101044629).

\* corresponding author.

affects the performance of the system since it affects the set of admissible controllers. Note that, also in this case, the control and estimation gains are known *a priori* to the agents (e.g., they are computed during a centralized design phase).

The main contributions of this work, which address the aforementioned limitations, are: (i) a novel fully distributed method for designing estimation and control gains through coordination, which, unlike other approaches, enables self-configuration and resilience to node failures in multi-agent systems; (ii) the adoption of an exact finite-time average consensus mechanism<sup>1</sup> (of precomputed required iterations) between estimation steps for precise eigenvalue placement, ensuring arbitrary selection of the convergence rate of the overall asymptotic estimation process. These innovations set our approach apart, particularly in coordinated estimation and control gain selection, and in providing stability assurances. Our simulations demonstrate the expanded set of achievable convergence rates as well as enhanced performance with respect to communication rounds and computation times.

## II. PRELIMINARIES

### A. Notation and Graph Theory

We denote vectors with boldface lowercase letters and matrices with uppercase letters. The transpose of matrix  $A$  and vector  $\mathbf{x}$  are denoted as  $A^\top$ ,  $\mathbf{x}^\top$ , respectively. We refer to the  $(i, j)$ -th entry of a matrix  $A$  by  $A_{ij}$ . We represent by  $\mathbf{0}_n$  and  $\mathbf{1}_n$  vectors with  $n$  entries, all equal to zero and to one, respectively. We use  $\|\cdot\|$  to denote the Euclidean norm.

Let  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$  be a directed graph (digraph) with  $N$  nodes  $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$  and  $e$  edges  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ , where  $(v_j, v_i) \in \mathcal{E}$  captures the existence of a link from node  $v_i$  to node  $v_j$ . A directed graph is *strongly connected* if each node can be reached by every other node via the edges, respecting their orientation. Let the in-neighborhood  $\mathcal{N}_i^{\text{in}}$  of a node  $v_i \in \mathcal{V}$  be the set of nodes  $v_j \in \mathcal{V}$  such that  $(v_i, v_j) \in \mathcal{E}$ ; similarly, the out-neighborhood  $\mathcal{N}_i^{\text{out}}$  of a node  $v_i \in \mathcal{V}$  is the set of nodes  $v_j \in \mathcal{V}$  such that  $(v_j, v_i) \in \mathcal{E}$ . In the case  $\mathcal{N}_i^{\text{in}}$  is equal to  $\mathcal{N}_i^{\text{out}}$  the graph is said to be *undirected* and the neighborhood of a node  $v_i \in \mathcal{V}$  is expressed with  $\mathcal{N}_i$ . The *in-degree*  $d_i^{\text{in}}$  of a node  $v_i$  is the number of its incoming edges, i.e.,  $d_i^{\text{in}} = |\mathcal{N}_i^{\text{in}}|$ ; similarly, the *out-degree*  $d_i^{\text{out}}$  of a node  $v_i$  is the number of its outgoing edges, i.e.,  $d_i^{\text{out}} = |\mathcal{N}_i^{\text{out}}|$ .

### B. Average (Minimum-Time) Consensus in Directed Graphs

The problem of average consensus over digraphs involves a number of nodes in a network represented by a graph  $\mathcal{G}$ , that exchange information to compute the network-wide average of their initial values. We consider a synchronous setting in which each node  $v_j$  updates and sends its information to its neighbors at discrete times  $t_0, t_1, t_2, \dots$  (herein called *consensus iterations*). We use  $\alpha_j[m] \in \mathbb{R}^n$  to denote the information state of node  $v_j$  at time  $t_m$ . We index nodes' information states and any other information at time  $t_m$  by

$m$ . At each consensus iteration  $m$ , each node  $v_j$  maintains its information state  $\alpha_j[m] \in \mathbb{R}^n$  (initialized at a certain value, e.g., a measurement, say  $\alpha_j[0] = \theta_j$ ), an auxiliary scalar variable  $\pi_j[m] \in \mathbb{R}_+$  (initialized at  $\pi_j[0] = 1$ ), and  $\mu_j[m] := \alpha_j[m]/\pi_j[m]$ . The authors in [16] showed that the nodes in an unbalanced digraph can asymptotically reach the exact average consensus by having each node  $v_j \in \mathcal{V}$  to iteratively communicate and update its values according to the so-called *ratio consensus* algorithm:

$$\alpha_j[m+1] = p_{jj}\alpha_j[m] + \sum_{v_i \in \mathcal{N}_j^{\text{in}}} p_{ji}\alpha_i[m], \quad (1a)$$

$$\pi_j[m+1] = p_{jj}\pi_j[m] + \sum_{v_i \in \mathcal{N}_j^{\text{in}}} p_{ji}\pi_i[m], \quad (1b)$$

where  $p_{lj}$  is a nonnegative weight assigned to outgoing information by each agent  $v_j$  as  $p_{lj} = 1/(1 + d_j^{\text{out}})$  if  $v_l \in \mathcal{N}_j^{\text{out}} \cup \{v_j\}$ , otherwise  $p_{lj} = 0$ . The weights assigned based on this strategy form a column-stochastic matrix  $P = \{p_{ji}\} \in \mathbb{R}_+^{N \times N}$  in which (possible) zero-valued entries,  $p_{ji} = 0$ , denote the absence of a (directed) link  $(v_j, v_i)$ . Then, the solution to the average consensus problem can be asymptotically obtained. In particular, we have that [16]

$$\lim_{m \rightarrow \infty} \mu_j[m] = \frac{\sum_{v_j \in \mathcal{V}} \alpha_j[0]}{\sum_{v_j \in \mathcal{V}} \pi_j[0]} = \frac{\sum_{v_j \in \mathcal{V}} \theta_j}{N}, \quad \forall v_j \in \mathcal{V}.$$

The results in [17] have shown that the exact average  $\mu$  can be distributively obtained in a finite and minimum number of steps in strongly connected digraphs. The results are based on the use of the concept of the minimal polynomial associated with the linear dynamics of each of (1a) and (1b), in conjunction with the final value theorem, initially proposed for undirected graphs in [18].

Since the update of each element  $r$ ,  $r \leq n$ , of the information state of node  $v_j$  at time instant  $m$ , denoted by  $\alpha_{jr}[m] \in \mathbb{R}$ , is independent of the update of all the other elements, we will explain how the minimum-time average consensus works for a scalar information state, for simplicity. Consider the vectors of the differences between  $2m+1$  successive discrete-time values at node  $v_j$ ,  $\bar{\alpha}_{jr}^\top = (\bar{\alpha}_{jr}[0], \bar{\alpha}_{jr}[1], \dots, \bar{\alpha}_{jr}[2m])$  and  $\bar{\pi}_{jr}^\top = (\bar{\pi}_j[0], \bar{\pi}_j[1], \dots, \bar{\pi}_j[2m])$ , where  $\bar{\alpha}_{jr}[l] = \alpha_{jr}[l+1] - \alpha_{jr}[l]$ , and  $\bar{\pi}_j[l] = \pi_j[l+1] - \pi_j[l]$  for  $l = 0, \dots, 2m$ . Then, each agent  $v_j$  constructs the Hankel matrices  $\Gamma\{\bar{\alpha}_{2m}^\top\}$  and  $\Gamma\{\bar{\pi}_{2m}^\top\}$ . The construction for  $\Gamma\{\bar{\alpha}_{2m}^\top\}$  (and respectively for  $\Gamma\{\bar{\pi}_{2m}^\top\}$ ) is as follows:

$$\Gamma\{\bar{\alpha}_{2m}^\top\} \triangleq \begin{bmatrix} \bar{\alpha}_{jr}[0] & \bar{\alpha}_{jr}[1] & \dots & \bar{\alpha}_{jr}[m] \\ \bar{\alpha}_{jr}[1] & \bar{\alpha}_{jr}[2] & \dots & \bar{\alpha}_{jr}[m+1] \\ \vdots & \vdots & \ddots & \vdots \\ \bar{\alpha}_{jr}[m] & \bar{\alpha}_{jr}[m+1] & \dots & \bar{\alpha}_{jr}[2m] \end{bmatrix}.$$

It has been shown in [17], [18] that after  $2(M_j + 1)$ , where  $M_j + 1$  is the minimum degree of the minimal polynomial<sup>2</sup>

<sup>1</sup>Finite-time average consensus is a class of distributed algorithms that allows the agents to compute the average of their initial values in a finite number of rounds. See Section II-C for more details.

<sup>2</sup>The minimal polynomial associated with the pair  $[P, e_j^\top]$  is denoted by  $q_j(t) = t^{M_j+1} + \sum_{i=0}^{M_j} \alpha_i^{(j)} t^i$ , where  $\alpha_i^{(j)} \in \mathbb{R}$  and  $e_j^\top$  is a vector with 1 in the  $j$ -th coordinate and 0's elsewhere, is the monic polynomial of minimum degree  $M_j + 1$  that satisfies  $e_j^\top q_j(P) = 0$ .

associated with the matrix pair  $[P, e_j^\top]$ , the kernel of the Hankel matrices  $\Gamma\{\bar{\alpha}_{2m}^\top\}$  and  $\Gamma\{\bar{\pi}_{2m}^\top\}$  for arbitrary initial conditions  $\alpha[0]$  and  $\pi[0]$  except a set of initial conditions with Lebesgue measure zero become defective. Once they become defective, node  $v_j$  computes  $M_j$  among others and has all the information needed to compute the final value of the overall system and, hence, compute the exact average in a finite number of steps. For more details, see [17].

### C. Distributed Termination and Synchronization

In [19] a distributed termination mechanism is proposed to enhance an existing finite-time distributed algorithm to allow the nodes to agree on when to terminate their iterations, provided they have all computed their exact average. Specifically, their proposed method is based on the fact that the finite-time consensus algorithm in [17] allows nodes in the network running iterations (1a) and (1b) to compute an upper bound of their *eccentricity*<sup>3</sup> and use this information for deciding when to terminate the process. The procedure is as follows:

- Once iterations (1a) and (1b) are initiated, each node  $v_j$  also initiates two counters  $c_j$ ,  $c_j[0] = 0$ , and  $r_j$ ,  $r_j[0] = 0$ . Counter  $c_j$  increments by one at every time step, i.e.,  $c_j[m+1] = c_j[m] + 1$ . Counter  $r_j$  updates are described next.
- Alongside iterations (1a) and (1b) a max-consensus algorithm [20] is initiated as well, given by

$$\phi_j[m+1] = \max_{v_i \in \mathcal{N}_j^{\text{in}} \cup \{v_j\}} \{ \max\{\phi_i[m], c_i[m]\} \}, \quad (2)$$

with  $\phi_j[0] = 0$ . Every time step  $m$  for which  $\phi_j[m+1] = \phi_j[m]$ ,  $r_j$  increments by one, but if, however, at any step  $m'$ ,  $\phi_j[m'+1] \neq \phi_j[m']$ , then  $r_j$  is set to zero, i.e.,  $r_j[m+1] = 0$  if  $\phi_j[m+1] \neq \phi_j[m]$ , and  $r_j[m+1] = r_j[m] + 1$ , otherwise.

- Once the square Hankel matrices  $\Gamma\{\bar{\alpha}_{M_j}^\top\}$  and  $\Gamma\{\bar{\pi}_{M_j}^\top\}$  for node  $v_j$  lose rank, node  $v_j$  saves the count of the counter  $c_j$  at that time step, denoted by  $m_j^o$ , as  $c_j^o$ , i.e.,  $c_j^o \triangleq c_j[m_j^o]$ , and it stops incrementing the counter, i.e.,  $\forall m' \geq m_j^o$ ,  $c_j[m'] = c_j[m_j^o] = c_j^o$ . Note that  $c_j^o = 2(M_j + 1)$ .
- Node  $v_j$  terminates (1a) and (1b) when  $r_j$  reaches  $c_j^o$ .

The main idea of this approach is that  $2(M_j + 1)$  serves as an upper bound on the maximum distance of any other node to node  $v_j$ . This quantity is not known initially but becomes known to node  $v_j$  through the finite-time consensus algorithm [17]. Therefore, the algorithm in [19] is used for computing distributively the number of steps of the consensus iterations  $\bar{m}$  needed by all nodes to complete the finite-time consensus; specifically, we have  $\bar{m} = 2 \max_{v_j \in \mathcal{V}} \{2(M_j + 1)\} - 1$ . Hence, after  $\bar{m}$  steps, each node can terminate the iterations.

*Remark 1:* Note that, as a byproduct of the distributed termination mechanism, an upper bound on the diameter of the network,  $D' = \max_{v_j \in \mathcal{V}} M_j$ , can be acquired once a node has completed the execution of the termination mechanism.

<sup>3</sup>The eccentricity of a node  $v_j \in \mathcal{V}$  is the length of the maximum distance between  $v_j$  and any other node  $v_i \in \mathcal{V}$ .

## III. PROBLEM STATEMENT

The problem considered in this paper is based on the setup used in [13], [14] in which a network of  $N$  agents aims at estimating and controlling a discrete-time LTI system, characterized by

$$\begin{cases} \mathbf{x}[k+1] = A\mathbf{x}[k] + \sum_{j=1}^N B_j \mathbf{u}_j[k], & \mathbf{x}[0] \in \mathbb{R}^n, \\ \mathbf{y}_i[k] = C_i \mathbf{x}[k], & \forall v_i \in \mathcal{V}, \end{cases} \quad (3)$$

where  $A \in \mathbb{R}^{n \times n}$ ,  $B_j \in \mathbb{R}^{n \times q_j}$ ,  $C_i \in \mathbb{R}^{p_i \times n}$ ,  $\mathbf{x}[k] \in \mathbb{R}^n$  is the state vector, and  $\mathbf{u}_j[k] \in \mathbb{R}^{q_j}$ ,  $\mathbf{y}_i[k] \in \mathbb{R}^{p_i}$  are, respectively, the  $j$ -th control input and the  $i$ -th system output at discrete times  $t_k$  (information states and any other information at time  $t_k$  is indexed by  $k$ ), herein called *estimation-control iterations*.

Let us consider  $q = \sum_{i=1}^N q_i$  and  $p = \sum_{i=1}^N p_i$ , and let us define  $B = [B_1, \dots, B_N] \in \mathbb{R}^{n \times q}$  and  $C = [C_1^\top, \dots, C_N^\top]^\top \in \mathbb{R}^{p \times n}$ . In the remainder of the paper, we make the following assumption.

*Assumption 1:* The system in Eq. (3) is jointly controllable and jointly observable, i.e., the pair  $(A, B)$  is controllable and the pair  $(A, C)$  is observable.

Assumption 1 not only implies that each agent can only partially observe the LTI system output and only partially control the LTI system state, but that there exist controllers and observers such that the system is stable and the estimation error converges to zero. Hence, even though each agent might not be able to fully control or observe the system on its own, all the nodes can do so collaboratively by exchanging information over the strongly connected graph  $\mathcal{G}$ .

The problem is to design the estimation dynamics and feedback control law for each agent, given its own set of information, so that jointly the agents stabilize the LTI system. Towards this end, Savas *et al.* [14] proposed a distributed state estimation scheme with the following structure:

$$\begin{aligned} \hat{\mathbf{x}}_i[k+1] = & A\tilde{\mathbf{x}}_i[k] + L_i(\mathbf{y}_i[k] - C_i\tilde{\mathbf{x}}_i[k]) \\ & + \sum_{j=1}^N B_j K_j \tilde{\mathbf{x}}_i[k] + \sum_{j \in \mathcal{N}_i} W_{ij}(\tilde{\mathbf{x}}_j[k] - \tilde{\mathbf{x}}_i[k]), \end{aligned} \quad (4)$$

where  $\tilde{\mathbf{x}}_i[k]$  is the result of  $m$ -steps of consensus over the graph topology, using the state estimates  $\hat{\mathbf{x}}_i[k]$  as the initial conditions;  $L_i \in \mathbb{R}^{n \times p_i}$ ,  $W_{ij} \in \mathbb{R}^{n \times n}$  are suitable gains that make the estimation error dynamics asymptotically stable, while the control input is chosen to be in the form  $\mathbf{u}_i = K_i \tilde{\mathbf{x}}_i[k]$ , with  $K_i \in \mathbb{R}^{q_i \times n}$ . Note that in [14] matrices  $K_i$  are designed offline, in a centralized fashion, so that  $A + \sum_{i=1}^N B_i K_i$  is Schur stable. Moreover, the authors consider a scenario where the agents are able to run a certain number  $m$  of consensus iterations via a linear consensus algorithm for balanced graphs on their estimated states, between two consecutive estimation-control iterations. Specifically, in [14] it is shown that, for sufficiently large  $m$ , there are gains  $L_i, W_{ij}$  which stabilize the overall control and estimation process (i.e., there is a separation between estimation and control).

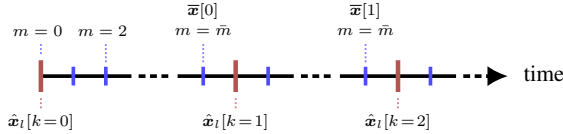
In this paper, we assume that the agents execute, instead, a

*finite-time consensus* procedure (as described in Sections II-B and II-C), between consecutive estimation-control iterations. Under this assumption, we show that the distributed state estimation process is simplified, enabling the agents to arbitrarily place the eigenvalues for the estimation error dynamics, but also allowing for a larger set of controllers, if implemented under the framework of [14]. Given the existence of this set of controllers, we develop an initialization procedure that allows the agents to coordinate in order to choose appropriate gains  $K_i$  and  $L_i$  in a completely distributed way and in finite time.

#### IV. FINITE-TIME AVERAGE CONSENSUS AS AN ENABLER FOR DISTRIBUTED ESTIMATION AND CONTROL

In this section, we propose the adoption of the finite-time exact ratio consensus iterations discussed in Section II-B, which operate in-between two successive estimation-control time steps as shown in Fig. 1. In particular, at each iteration  $k$ , each node  $v_i$  receives the local state estimates of its in-neighbors, *i.e.*,  $\hat{x}_j[k]$  for all  $v_j \in \mathcal{N}_i^{\text{in}}$  and runs the finite-time consensus procedure as discussed in Sections II-B and II-C. The following assumption ensures that all nodes have computed the average of the network-wide state estimate in  $2(\max_{v_j \in \mathcal{V}} M_j + 1)$  consensus iterations, and terminated the finite-time consensus procedure after at most  $\bar{m}$  consensus iterations.

*Assumption 2:* At least  $\bar{m}$  consensus steps can be completed between two successive estimation-control time steps. Hence, under Assumption 2 each node  $v_i$  at time step  $k$  knows the common state estimate that is the average of the network-wide state estimate,  $\bar{x}[k]$ , *i.e.*,  $\bar{x}[k] = \frac{1}{N} \sum_{l=1}^N \hat{x}_l[k]$ . Notice



**Figure 1:** Finite-time average consensus iterations  $m$  (blue ticks) within distributed estimation and control iterations  $k$  (red ticks).

that, due to such a procedure, the agents are able to reach an agreement on the estimate  $\bar{x}[k]$  of the state at time  $k$ ; as a result, the systems' dynamics can be equivalently modeled as

$$\mathbf{x}[k+1] = A\mathbf{x}[k] + \sum_{j=1}^N B_j K_j \bar{x}[k] \quad (5)$$

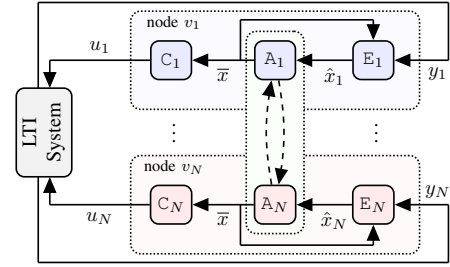
while since  $\hat{x}_i[k] - \hat{x}_j[k] = \bar{x}[k] - \bar{x}[k] = \mathbf{0}_n$ , the local estimator in Eq. (4) reduces to

$$\hat{x}_i[k+1] = A\bar{x}[k] + L_i(y_i[k] - C_i\bar{x}[k]) + \sum_{j=1}^N B_j K_j \bar{x}[k].$$

We define the estimation error for the  $i$ -th agent as  $e_i[k] = \mathbf{x}[k] - \hat{x}_i[k]$ , and the average estimation error

$$\bar{e}[k] = \frac{1}{N} \sum_{l=1}^N e_l[k] = \frac{1}{N} \sum_{l=1}^N (\mathbf{x}[k] - \hat{x}_l[k]) = \mathbf{x}[k] - \bar{x}[k].$$

An instance of our proposed distributed estimation and control scheme (which extends the approach in [14]) for the problem in Section III, considering a simple network, is shown in Fig. 2. In particular, the operations undertaken by each node  $v_i$  in-between state updates essentially amount to an estimation task (represented by the symbol  $E_i$  in the figure), a finite-time agreement task (represented by the symbol  $A_i$ ) and a control task (represented by the symbol  $C_i$ ). The figure shows the relations among such tasks for each node. Interestingly, as discussed next, the nodes, besides being coupled by the physical process, directly interact by exchanging information only within the finite-time agreement task, while the estimation and control tasks are performed locally, based on the agreed-upon estimate computed during the finite-time agreement procedures.



**Figure 2:** Proposed distributed estimation and control scheme.

In the next theorem, we show that finite-time consensus in-between estimation steps, allows arbitrary eigenvalue selection for the error dynamics<sup>4</sup>.

*Theorem 1:* Let Assumption 1 hold. Then, the local estimation errors,  $e_i[k]$ , can be brought to zero asymptotically, by selecting suitable gain matrices  $L_i$  that assign arbitrary eigenvalues to  $A - \frac{1}{N} \sum_{i=1}^N L_i C_i$ .

*Proof:* In order to prove the result, we observe that the dynamics of the network-wide average estimation error are described by

$$\begin{aligned} \bar{e}[k+1] &= \frac{1}{N} \sum_{l=1}^N (\mathbf{x}[k+1] - \hat{x}_l[k+1]) \\ &= \frac{1}{N} \sum_{l=1}^N (A\mathbf{x}[k] - A\bar{x}[k] - L_l(y_l[k] - C_l\bar{x}[k])) \\ &= \frac{1}{N} \sum_{l=1}^N (A - L_l C_l)(\underbrace{\mathbf{x}[k] - \bar{x}[k]}_{\bar{e}[k]}) = \left(A - \frac{1}{N} \mathcal{L} C\right) \bar{e}[k], \end{aligned}$$

where  $\mathcal{L} = [L_1, \dots, L_N]$ , and, thus,  $\mathcal{L} C = \sum_{l=1}^N L_l C_l$ . Since, by Assumption 1, the pair  $(A, C)$  is observable, we have that also  $(A, C/N)$  is observable and thus, by definition, there is a gain  $\mathcal{L}$  which brings the error  $\bar{e}$  to zero by assigning arbitrary eigenvalues.

Let us now consider the dynamics of the local estimation

<sup>4</sup>Even if the agents execute finite-time average consensus in-between estimation steps, the overall estimation process converges asymptotically.

errors  $e_i[k]$ . In particular, we have that

$$\begin{aligned} e_i[k+1] &= \mathbf{x}[k+1] - \hat{\mathbf{x}}_i[k+1] \\ &= \mathbf{A}\mathbf{x}[k] - \mathbf{A}\bar{\mathbf{x}}[k] - L_i(\mathbf{y}_i[k] - C_i\bar{\mathbf{x}}[k]) \\ &= (\mathbf{A} - L_i C_i)(\mathbf{x}[k] - \bar{\mathbf{x}}[k]) = (\mathbf{A} - L_i C_i) \bar{\mathbf{e}}[k]. \end{aligned}$$

Therefore, being the local estimation errors  $e_i[k+1]$  essentially a scaling of  $\bar{\mathbf{e}}[k]$  (for which we already proved convergence to zero) by a constant matrix  $\mathbf{A} - L_i C_i$ , we have that also the terms  $e_i[k]$  converge to zero. This completes our proof.  $\blacksquare$

*Remark 2:* In [9]–[12] the gains  $L_i$  are designed locally. However, due to the presence of multiple input signals, this strategy is not guaranteed to work for the design of the control gains  $K_i$ . Consider, for instance, the case where  $N = 2$ ,

$$\mathbf{A} = \begin{bmatrix} 1 & -2 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0.1 & -0.1 & 0.5 & 0.1 \\ 0.2 & -0.1 & 0.5 & 0.1 \end{bmatrix}, \quad \mathbf{B}_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}.$$

Using the approach in [9]–[12] to choose  $K_i$  we get

$$\begin{aligned} K_1 &= [2.7788 \quad -2.0033 \quad 0.0436 \quad 1.5033], \\ K_2 &= [-1.7909 \quad 4.0311 \quad -0.1091 \quad -5.0182]. \end{aligned}$$

However, the eigenvalues of  $\mathbf{A} + \mathbf{B}_1 K_1 + \mathbf{B}_2 K_2$  are unstable, being  $\{0.6515 \pm 2.8137i, -2.7102, 0.5073\}$ . Notice that a similar problem would arise if the gains  $L_i$  in our scheme were computed as in [9]–[12], as  $\mathbf{A} - \frac{1}{N} \sum_{i=1}^N L_i C_i$  would not necessarily have roots inside the unit disc.

## V. DISTRIBUTED SELECTION OF THE GAINS

In this section, we develop an initialization procedure that allows the agents to select suitable gains matrices  $K_i$  and  $L_i$  in finite time and in a completely distributed way. Notice that, for the sake of brevity, in the remainder of this section we address the problem of choosing the control gains  $K_i$ ; the application of the envisaged procedure to the selection of the observer gains  $L_i$  is straightforward.

Specifically, the overall initialization procedure is comprised of three consecutive initialization steps, described next:

- P1) Run iterations of the finite-time average consensus algorithm twice, as in Sections II-B and II-C, to obtain an upper bound  $D'$  on the network diameter  $D$  (see Remark 1) and  $\bar{m}$  [19].
- P2) Perform a distributed leader election procedure, e.g., a max-consensus procedure for  $D'$  steps (computed already in procedure P1) over the agents' identifiers or by just selecting a number from a continuous fixed range of values [20], [21].
- P3) Execute the procedure for the distributed selection of the gains discussed next, in Section V-B; it is based on the methodology discussed in Section V-A, which requires a leader, already elected in procedure P2.

### A. Single Eigenvalue Placement

This subsection reviews a control technique for a system in the form  $\mathbf{z}[k+1] = \mathcal{A}\mathbf{z}[k] + \mathcal{B}\mathbf{u}[k]$ , with  $\mathcal{A} \in \mathbb{R}^{n \times n}$  and  $\mathcal{B} \in \mathbb{R}^{n \times q}$  that allows a single eigenvalue  $\lambda_i$  of the dynamical

matrix  $\mathcal{A}$  to be modified while leaving all other eigenvalues unchanged. In particular, in the following we assume that  $\lambda_i$  is a controllable eigenvalue of  $\mathcal{A}$ , i.e.,  $\text{rank}([\mathcal{A} - \lambda_i I_n \quad \mathcal{B}]) = n$  as per the Popov-Belevitch-Hautus test (e.g., see [22]). Moreover, let us consider a left eigenvector  $\mathbf{w}_i(\mathcal{A}) \in \mathbb{R}^n$  associated to  $\lambda_i$ , i.e.,  $\mathbf{w}_i^\top(\mathcal{A})\mathcal{A} = \lambda_i \mathbf{w}_i^\top(\mathcal{A})$ . Let us now consider the dynamics  $\mathbf{x}[k+1] = \mathcal{A}\mathbf{x}[k] + \mathcal{B}\mathbf{u}[k]$  and let us initially assume, for simplicity of exposition, that  $\mathcal{B} \in \mathbb{R}^{n \times 1}$ , i.e., that  $\mathbf{u}[k]$  is scalar. Premultiplying the systems' dynamics by  $\mathbf{w}_i^\top(\mathcal{A})$ , we highlight the dynamics along the eigenspace spanned by  $\mathbf{w}_i^\top(\mathcal{A})$  (e.g., see [22]), thus obtaining

$$\begin{aligned} \mathbf{w}_i^\top(\mathcal{A})\mathbf{z}[k+1] &= \mathbf{w}_i^\top(\mathcal{A})\mathcal{A}\mathbf{z}[k] + \mathbf{w}_i^\top(\mathcal{A})\mathcal{B}\mathbf{u}[k] \\ &= \lambda_i \mathbf{w}_i^\top(\mathcal{A})\mathbf{z}[k] + \mathbf{w}_i^\top(\mathcal{A})\mathcal{B}\mathbf{u}[k]. \end{aligned}$$

Suppose that, as a result of the control action, we want to replace  $\lambda_i$  with  $\bar{\lambda}_i$ , i.e., we want to choose  $\mathbf{u}[k]$  such that

$$\mathbf{w}_i^\top(\mathcal{A})\mathbf{z}[k+1] = \bar{\lambda}_i \mathbf{w}_i^\top(\mathcal{A})\mathbf{z}[k].$$

This is possible by choosing  $\mathbf{u}[k] = \mathcal{K}\mathbf{z}[k]$  with

$$\mathcal{K} = \frac{\bar{\lambda}_i - \lambda_i}{\mathbf{w}_i^\top(\mathcal{A})\mathcal{B}} \mathbf{w}_i^\top(\mathcal{A}). \quad (6)$$

Notice that the above selection for  $\mathcal{K}$  is well-posed; in fact, since  $\lambda_i$  is controllable then  $[\mathcal{A} - \lambda_i I_n \quad \mathcal{B}]$  is full rank and thus  $\mathbf{w}_i^\top(\mathcal{A})[\mathcal{A} - \lambda_i I_n \quad \mathcal{B}] \neq 0$ , from which we have that

$$[\mathbf{w}_i^\top(\mathcal{A})\mathcal{A} - \lambda_i \mathbf{w}_i^\top(\mathcal{A}) \quad \mathbf{w}_i^\top(\mathcal{A})\mathcal{B}] = [0 \quad \mathbf{w}_i^\top(\mathcal{A})\mathcal{B}] \neq 0,$$

which implies that  $\mathbf{w}_i^\top(\mathcal{A})\mathcal{B} \neq 0$ . The above  $\mathcal{K}$  is such that  $\mathcal{A} + \mathcal{B}\mathcal{K}$  has the same eigenvalues as  $\mathcal{A}$ , except for the fact that the eigenvalue  $\lambda_i(\mathcal{A})$  is replaced by the desired  $\bar{\lambda}_i$ . Note that this is similar to a Hotelling deflation [23], since we only modify the direction spanned by one eigenvector, without modifying the other directions.

Notably, the procedure can be iterated. Assuming that we also want to replace  $\lambda_j$  with  $\bar{\lambda}_j$ , it is sufficient to set  $\mathbf{u}[k] = \mathcal{K}\mathbf{z}[k] + \mathbf{r}[k]$ , where  $\mathcal{K}$  is as above, while  $\mathbf{r}[k] = \mathcal{H}\mathbf{z}[k]$ , with

$$\mathcal{H} = \frac{\bar{\lambda}_j - \lambda_j}{\mathbf{w}_j^\top(\mathcal{A} + \mathcal{B}\mathcal{K})\mathcal{B}} \mathbf{w}_j^\top(\mathcal{A} + \mathcal{B}\mathcal{K}). \quad (7)$$

Notice that the left eigenvector used to replace  $\lambda_j$  is the one obtained after applying  $\mathcal{K}$ , i.e., considering the matrix  $\mathcal{A} + \mathcal{B}\mathcal{K}$ .

Let us now consider the case where  $\mathcal{K} \in \mathbb{R}^{n \times q}$  with  $q > 1$ . In this case, assuming that a matrix  $\mathcal{K}^{(j)}$  has been selected with the above approach for each column  $\mathcal{K}^{(j)}$  of  $\mathcal{B}$ , it is sufficient to set the overall  $\mathcal{K} \in \mathbb{R}^{q \times n}$  as  $\mathcal{K} = [\mathcal{K}^{(1)\top} \quad \dots \quad \mathcal{K}^{(q)\top}]^\top$ .

### B. Proposed Protocol for Choosing the Gains

The main idea of our protocol is to have the agents choose  $K_i$  one after another, using the approach discussed in Section V-A. In the beginning, a leader is elected in a distributed way<sup>5</sup>. The leader initiates a token-passing procedure where a token is circulated among the agents, and the token contains

<sup>5</sup>Leader election could be implemented via max-consensus over the agents' identifiers [24], [25].

a matrix  $F$  to be described next. In general, when a node  $v_i$  receives a message  $F$  for the first time, it checks if  $A + F$  is Schur stable. If  $A + F$  is not Schur stable, it computes a gain  $K_i$  that stabilizes all the eigenvalues it can (as done in Section V-A), it adds its own term  $B_i K_i$  to the message and it sends the updated message  $F + B_i K_i$  to an out-neighbor that has not yet been visited. Notice that some of the agents might not be able to control any eigenvalue, and in that case, they simply choose  $K_i = 0_{n \times q_i}$ . If, conversely, upon receiving the message for the first time, agent  $i$  realizes that  $A + F$  is Schur stable, the message becomes read-only and starts being flooded in the network. A last case is when a message that is not read-only is received by a node that has already received it; in this case, the message is simply forwarded to a not-yet-visited out-neighbor. Notice that the proposed approach does not require a stopping criterion. In fact, since the system is jointly controllable by all agents, and due to the approach adopted for the selection of the gains, eventually, all unstable eigenvalues will be placed and one of the agents will thus receive  $F$  such that  $A + F$  is Schur stable; as a consequence, the read-only phase will begin. Also in this case, since the graph is strongly connected, and since each agent broadcasts the message to all its neighbors and then stops participating, the read-only message will reach all nodes in finite time. Algorithm 1 summarizes the above procedure.

---

**Algorithm 1** Choice of  $K_i$  (from the perspective of agent  $i$ )

---

```

procedure INITIALIZATION()
  Assign a priority to each out-neighbor;
  Declare  $i$  to be not yet visited;
   $K_i \leftarrow 0_{n \times q_i}$ ;
procedure ONLEADERELECTED()
  Create empty token  $F = 0$ ;
  Trigger OnReceiveToken( $F$ );
procedure ONRECEIVETOKEN( $F$ )
  if already visited then
    Forward  $F$  to an out-neighbor6;
  else
    if  $A + F$  is not Schur stable then
      Find  $K_i$  that stabilizes controllable eigenvalues7;
       $F \leftarrow F + K_i B_i$ ;
      Send  $F$  to an out-neighbor6;
      Declare  $i$  to be visited;
    else
      Declare  $F$  read-only;
      Broadcast  $F$  to all out-neighbors;
      Stop Participating;
    end
  end
procedure ONRECEIVEREADONLYMESSAGE( $F$ )
  Broadcast  $F$  to all out-neighbors;
  Stop Participating;

```

---

The following theorem establishes the correctness of the proposed algorithm and characterizes the associated time and message complexity.

<sup>6</sup>According to the priorities.

<sup>7</sup>The matrix  $K_i$  can be computed using the approach in Section V-A, i.e., via Eqs.(6)–(7).

**Theorem 2:** Let  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$  be strongly connected, let Assumption 1 hold true, and assume that the algebraic and geometric multiplicities of the unstable eigenvalues of  $A$  are the same. Then, Algorithm 1 is *correct*, i.e., each agent computes  $K_\ell$  such that  $A + \sum_{\ell=1}^N B_\ell K_\ell$  is Schur stable. Moreover, the time complexity and the message complexity of Algorithm 1 amount to  $\mathcal{O}(N^2)$ . Finally, at each step,  $\mathcal{O}(N)$  messages are transmitted.

*Proof:* In order to prove our statement, let us first show that, over a strongly connected graph  $G$ , the message will be received by every agent (hence each agent will add its contribution  $B_i K_i$ ) in at most  $\mathcal{O}(N^2)$  steps. To this end, let us consider the worst case in terms of graph topology and priorities and let us use  $v_i$  to denote the agents with the order they are first reached by the token (hence, the leader is  $v_1$ ). Moreover let  $\omega_i$  denote the number of steps required to visit node  $v_i$  from the time the token is first received by node  $v_{i-1}$  (in this view,  $\omega_1 = 0$ ). Over a strongly connected graph, in the worst case, the token always goes back to  $v_1$  after visiting a new node. Therefore, node  $v_i$  is visited after  $\omega_i = 2i - 3$  steps; in fact, it takes  $i - 2$  steps to go back to  $v_1$  and additional  $i - 1$  steps to reach  $v_i$  from  $v_1$ . Hence, the total message complexity is  $\sum_{i=2}^N \omega_i = \sum_{i=2}^N (2i - 3) = N^2 - 4N + 2 = \mathcal{O}(N^2)$ .

At this point, we observe that since by Assumption 1 the pair  $(A, B)$  is controllable, and since the gain matrices  $K_i$  are chosen iteratively, each time modifying only the eigenspace spanned by the eigenvalues that are unstable and controllable by the  $i$ -th agent alone, by construction  $A + BK$  is Schur stable, with  $K = [K_1^\top, \dots, K_N^\top]^\top$ . Correctness follows by noting that, when the last agent receives the message for the last time and updates  $F$ , it holds  $A + BK = A + \sum_{\ell=1}^N B_\ell K_\ell = A + F$ . Notice that, since the token passing requires  $\mathcal{O}(N^2)$  steps in the worst case, the message is declared as read-only in  $\mathcal{O}(N^2)$  steps, and the read-only message is received by all agents in additional  $\mathcal{O}(N)$  steps (i.e., because the message is flooded over the strongly connected graph). As a consequence, the time complexity of Algorithm 1 is  $\mathcal{O}(N^2)$ . Let us now discuss the complexity of message exchanges. In particular, during the initial phase in which the nodes transmit the token on a point-to-point basis,  $\mathcal{O}(N^2)$  messages are transmitted. Then, when the token becomes read-only, the message is transmitted once for each link, i.e.,  $\mathcal{O}(|E|)$  times. Therefore, overall, the message complexity of Algorithm 1 is  $\mathcal{O}(N^2 + |E|) = \mathcal{O}(N^2)$ . To conclude we observe that, in the worst case of a star topology where the star-center broadcasts the read-only message,  $\mathcal{O}(N)$  messages are transmitted at each time step. ■

**Remark 3:** As a byproduct of Algorithm 1, each node will know  $\sum_{\ell=1}^N B_\ell K_\ell$ , which can be used within Eq. (5), thus relaxing the assumption in [14], in which each agent  $v_i$  is required to know  $\sum_{j \in \mathcal{N}_i^{\text{in}}} B_j K_j$  a priori. It is worth mentioning that the input matrices  $B_i$  are never directly shared; instead, the nodes accumulate  $B_i K_i$  in the token.

**Remark 4:** Algorithm 1 can also be applied to the selection of the gains  $L_i$ . In fact, it is sufficient to apply the algorithm to the selection of gain matrices  $\tilde{K}_i$  with respect to the pairs



$(A^\top, -\frac{1}{n}C_i^\top)$ , and then set  $L_i = \tilde{K}_i^\top$ .

## VI. ILLUSTRATIVE EXAMPLES

Consider a directed network consisting of four nodes, described by matrix  $P$ , where each node has access to the dynamical matrix  $A$ .

$$P = \begin{bmatrix} 1/3 & 0 & 1/4 & 1/3 \\ 1/3 & 1/2 & 1/4 & 0 \\ 0 & 1/2 & 1/4 & 1/3 \\ 1/3 & 0 & 1/4 & 1/3 \end{bmatrix}, A = \begin{bmatrix} 1 & 0.5 & 0 & 0 & 3 & 0 & 0 & 0 \\ 0.5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.8 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.6 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.7 & 0.1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.7 \end{bmatrix}.$$

Moreover, each node in the network has sensing, and computational capabilities to run the finite-time exact average consensus algorithm (for this setup  $\bar{m} = 11$  steps) and compute its local state estimates, and control input signals. More specifically, each node can sense only one state, and apply only one control signal to the system. The local input matrices  $B_i$  for each node  $v_i$  are  $B_1 = [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^\top$ ,  $B_2 = [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0]^\top$ ,  $B_3 = [0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0]^\top$ , and  $B_4 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1]^\top$ . Similarly, the local measurement matrices  $C_i$  are  $C_1 = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$ ,  $C_2 = [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0]$ ,  $C_3 = [0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0]$ , and  $C_4 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0]$ .

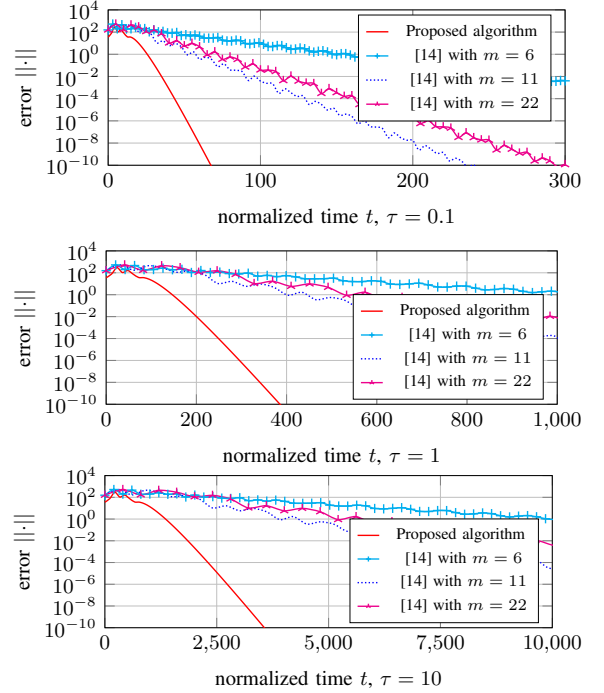
Let  $\rho_i$  and  $\chi_i$  denote, respectively, the controllability and observability indices for the  $i$ -th agent, i.e., the rank of the controllability and observability matrices for the pairs  $(A, B_i)$  and  $(A, C_i)$ , respectively. We have that  $\rho_1 = 4$ ,  $\rho_2 = 2$ ,  $\rho_3 = 6$ ,  $\rho_4 = 2$ , while  $\chi_1 = 4$ ,  $\chi_2 = 2$ ,  $\chi_3 = 2$ ,  $\chi_4 = 6$ , i.e., the system is not independently observable or controllable by the single agents. However, the system is jointly observable and controllable as the rank of both the overall controllability and observability matrices is full.

Let us now consider the initialization phase, and let us assume that  $v_1$  is elected as leader and that the token visits the agents in order from  $v_1$  to  $v_4$ . The agents choose  $L_i$  and  $K_i$  via Algorithm 1, with the aim to place the observer eigenvalues at  $\{0.20, 0.21, \dots, 0.27\}$  and the controller eigenvalues at  $\{0.60, 0.61, \dots, 0.67\}$ . The local observer and control gains chosen by each agent  $v_i$  correspond to the  $i$ -th row of  $K$  and column of  $L$ , respectively, where

$$K = \begin{bmatrix} -1.0916 & 0 & 0.0003 & 0 \\ -1.0114 & 0 & 0.0003 & 0 \\ 0 & -1.0664 & 0 & 0 \\ 0 & -0.7300 & 0 & 0 \\ 31.9630 & 0 & -0.8021 & 0 \\ 32.3204 & 0 & -0.5786 & 0 \\ -0.0350 & 0 & 0 & 0 \\ -0.0247 & 0 & 0 & 0 \end{bmatrix}^\top, L = \begin{bmatrix} 12.5600 & 0 & 0 & 0 \\ -8.1221 & 0 & 0 & 0 \\ 0 & 6.0400 & 0 & 0 \\ 0 & 7.7600 & 0 & 0 \\ 7.0165 & 0 & 0 & 0 \\ 7.6708 & 0 & 0 & 0 \\ -0.1299 & 0 & 0 & 3.4800 \\ -5.7761 & 0 & 0 & 8.3680 \end{bmatrix}.$$

Here it is worth mentioning that, although all agents participate in all procedures involved in estimating and stabilizing the states of the system, the control gain of  $v_4$  and the observer gain of  $v_3$  are zero. In the first case, this occurs since when agent  $v_4$  receives the token the matrix already has the desired eigenvalues, while in the second case, this is due to the fact that agent  $v_3$  is only able to change some of the eigenvalues

that have already been changed, hence there is a need to resort to the observer gain chosen by  $v_4$ .



**Figure 3:** Comparison of the estimation error between the proposed approach (solid line,  $\bar{m} = 11$ ) and the one in [14] when  $m \in \{6, 11, 22\}$ , for different communication duration  $\tau \in \{0.1, 1, 10\}$ .

To conclude, in Fig. 3 we provide a comparison<sup>8</sup> with the approach in Savas *et al.* [14], which is based on  $m$  repeated rounds of asymptotic consensus in-between state updates. Following the approach in [26], by normalization of the time axis, we assume that the local computations executed at each estimation-control iteration  $k$  are performed in one unit of time, denoted by  $t$ . Moreover, we assume that each communication round within the asymptotic or time-varying consensus procedures requires a time equal to  $\tau$  (which may be smaller, equal, or greater than 1). In particular, the upper plot of Fig. 3 considers the case where  $\tau = 0.1$  (i.e., communication is faster than local computation), the middle plot of Fig. 3 considers the case where  $\tau = 1$  (i.e., communication and computation are equally fast), while the case  $\tau = 10$  (i.e., communication is slower than computation) is reported in the lower plot of Fig. 3. In more detail, the figure considers the problem discussed in the previous example and compares the norm of the average error of our approach with the one in [14] relying, for fairness of comparison, on the same control and estimation gains chosen via our distributed initialization algorithm. The

<sup>8</sup>Notice that also [11] relies on rounds of distributed averaging; however, such an approach only focuses on estimation, while no control action is considered. For this reason, and since in [11] the estimation gains are designed in a way that would not be suitable for our approach (and vice versa), we opted not to compare them.

gains  $W_{ij}$  were computed<sup>9</sup> as in [14]. Interestingly, while the proposed approach requires  $\bar{m} = 11$  communication rounds for each state update, the approach in [14], for the considered instance, exhibits a converging behavior only for  $m \geq 6$  asymptotic average consensus rounds. In this view, the figure reports the evolution of the error for the approach in [14] considering  $m \in \{6, 11, 22\}$ . According to the figure, in all three considered scenarios for the computation/communication time, the proposed algorithm consistently outperforms the one in [14]. The comparison is particularly favorable to our algorithm when the communication time is comparable to or larger than the local computation time, *i.e.*,  $\tau \in \{1, 10\}$ .

Interestingly, the normalized time  $t$  required for the case where  $m = 22$  is larger than the one required for  $m = 11$ ; in fact, we observe that for these two cases, the performance is comparable in terms of estimation-control iterations  $k$ , but setting  $m = 22$  requires twice as much communication rounds as for  $m = 11$ . Hence, in the case of asymptotic average-consensus rounds, increasing  $m$  may not correspond to a performance increase.

## VII. CONCLUSIONS AND FUTURE DIRECTIONS

This paper introduces a novel methodology for distributed LTI system estimation and control in discrete time. It incorporates a finite-time average consensus mechanism between estimation steps, allowing for arbitrary eigenvalue placement in estimation error dynamics and overcoming the limitations of previous approaches. A novel, fully distributed method for computing control gains enhances the system's self-configuration and resilience. Simulation results demonstrate improved performance over other methods that are based on asymptotic consensus rounds.

Future research could explore: dealing with consensus mechanisms under unmet assumptions (e.g., when Assumption 2 is not satisfied), addressing communication challenges like delays, packet losses, and varying topologies, developing procedures for simultaneous gain design across nodes, and extending the framework to stochastic linear systems.

## REFERENCES

- [1] F. F. Rego, A. M. Pascoal, A. P. Aguiar, and C. N. Jones, "Distributed state estimation for discrete-time linear time invariant systems: A survey," *Annual Reviews in Control*, vol. 48, pp. 36–56, 2019.
- [2] V. Ugrinovskii, "Conditions for Detectability in Distributed Consensus-Based Observer Networks," *IEEE Transactions on Automatic Control*, vol. 58, no. 10, pp. 2659–2664, 2013.
- [3] H. Zhu, K. Liu, J. Lü, Z. Lin, and Y. Chen, "On the cooperative observability of a continuous-time linear system on an undirected network," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2014, pp. 2940–2944.
- [4] S. Park and N. C. Martins, "Necessary and sufficient conditions for the stabilizability of a class of LTI distributed observers," in *IEEE Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 7431–7436.

<sup>9</sup>In [14], the gains  $W_{ij}$  amount to the sum of two terms  $W_{ij}^E = p_{ij}A$  and  $W_{ij}^C$ . In particular, the term  $W_{ij}^C$  is computed by solving a centralized optimization problem featuring a positive definiteness constraint. To compute the terms  $W_{ij}^E$ , we resorted to an approximated solver, namely MIDACO [27]. The interested reader is referred to [14] for a complete discussion on how to compute the gains  $W_{ij}$ .

- [5] —, "Design of distributed LTI observers for state omniscience," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 561–576, 2016.
- [6] A. Mitra and S. Sundaram, "Distributed observers for LTI systems," *IEEE Transactions on Automatic Control*, vol. 63, no. 11, pp. 3689–3704, 2018.
- [7] M. Khajenejad, S. Brown, and S. Martínez, "Distributed interval observers for bounded-error LTI systems," in *American Control Conference (ACC)*. IEEE, 2023, pp. 2444–2449.
- [8] T. Liu and J. Huang, "Distributed Exponential State Estimation for Discrete-time Linear Systems over Jointly Connected Switching Networks," *IEEE Transactions on Automatic Control*, 2023.
- [9] T. Kim, H. Shim, and D. D. Cho, "Distributed luenberger observer design," in *IEEE Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 6928–6933.
- [10] W. Han, H. L. Trentelman, Z. Wang, and Y. Shen, "A simple approach to distributed observer design for linear systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 1, pp. 329–336, 2018.
- [11] L. Wang, J. Liu, and A. S. Morse, "A distributed observer for a continuous-time linear system," in *American Control Conference (ACC)*. IEEE, 2019, pp. 86–91.
- [12] L. Wang, J. Liu, A. S. Morse, and B. D. Anderson, "A distributed observer for a discrete-time linear system," in *IEEE Conference on Decision and Control (CDC)*, 2019, pp. 367–372.
- [13] L. Wang, D. Fullmer, F. Liu, and A. S. Morse, "Distributed control of linear multi-channel systems: summary of results," in *American Control Conference (ACC)*. IEEE, 2020, pp. 4576–4581.
- [14] A. J. Savas, S. Park, H. V. Poor, and N. E. Leonard, "On Separation of Distributed Estimation and Control for LTI Systems," in *IEEE Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 963–968.
- [15] T. Kim, C. Lee, and H. Shim, "Completely decentralized design of distributed observer for linear systems," *IEEE Transactions on Automatic Control*, vol. 65, no. 11, pp. 4664–4678, 2019.
- [16] A. D. Dominguez-Garcia and C. N. Hadjicostis, "Coordination and control of distributed energy resources for provision of ancillary services," in *IEEE International Conference on Smart Grid Communications*, 2010, pp. 537–542.
- [17] T. Charalambous, Y. Yuan, T. Yang, W. Pan, C. N. Hadjicostis, and M. Johansson, "Distributed finite-time average consensus in digraphs in the presence of time delays," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 4, pp. 370–381, 2015.
- [18] Y. Yuan, G.-B. Stan, M. Barahona, L. Shi, and J. Gonçalves, "Decentralised minimal-time consensus," *Automatica*, vol. 49, no. 5, pp. 1227–1235, 2013.
- [19] T. Charalambous and C. N. Hadjicostis, "When to stop iterating in digraphs of unknown size? an application to finite-time average consensus," in *European Control Conference (ECC)*, 2018, pp. 1–7.
- [20] J. Cortés, "Distributed algorithms for reaching consensus on general functions," *Automatica*, vol. 44, pp. 726–737, March 2008.
- [21] B. M. Nejad, S. A. Attia, and J. Raisch, "Max-consensus in a max-plus algebraic setting: The case of fixed communication topologies," in *International Symposium on Information, Communication and Automation Technologies*. IEEE, 2009, pp. 1–7.
- [22] D. G. Luenberger, *Introduction to dynamic systems: theory, models, and applications*. Wiley New York, 1979, vol. 1.
- [23] L. Mackey, "Deflation methods for sparse pca," *Advances in neural information processing systems*, vol. 21, 2008.
- [24] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.
- [25] G. Muniraju, C. Tepedelenlioglu, and A. Spanias, "Analysis and design of robust max consensus for wireless sensor networks," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, no. 4, pp. 779–791, 2019.
- [26] K. Scaman, F. Bach, S. Bubeck, Y. T. Lee, and L. Massoulié, "Optimal algorithms for smooth and strongly convex distributed optimization in networks," in *international conference on machine learning*. PMLR, 2017, pp. 3027–3036.
- [27] M. Schlueter, S. O. Erb, M. Gerdts, S. Kemble, and J.-J. Rückmann, "Midaco on minlp space applications," *Advances in Space Research*, vol. 51, no. 7, pp. 1116–1131, 2013.