Decentralized State Estimation for Heterogeneous Multi-Agent Systems

Francesca Boem, Lorenzo Sabattini and Cristian Secchi

Abstract— **The paper proposes a decentralized state estimation method for the control of multi-agent networked systems, where the goal is the tracking of arbitrary setpoint functions. The cooperative agents are partitioned into independent robots, providing the control inputs, and dependent robots, controlled by local interaction laws. The proposed state estimation algorithm allows the independent robots to estimate the state of the dependent robots in a completely decentralized way. To do that, it is necessary for each independent robot to estimate the control input components computed by the other independent robots, without requiring communication among the independent robots. The decentralized state estimator, including an input estimator, is developed and the convergence properties are studied. Simulation results show the effectiveness of the proposed approach.**

I. Introduction

The paper considers cooperative dynamic behaviors of multi-agent networked systems, where the goal is the tracking of arbitrary setpoint functions. Similarly as in [1], the networked systems are partitioned into *independent robots*, providing control inputs, and *dependent robots*, controlled through local interaction. In this paper, we propose a decentralized estimation method to let the independent robots estimate the state of the dependent robots. Then, the estimate is used for control purposes.

Decentralized control of networked systems has been widely addressed in the last few years. Main application fields include, for instance, multi-robot systems [2], distributed sensor networks [3] and interconnected manufacturing equipments [4]. Generally speaking, the aim of decentralized control strategies is implementing local interaction rules to regulate the state of the overall system to some desired configuration. Along these lines, mainly investigated coordinated behaviors include aggregation, swarming, formation control, coverage and synchronization [2], [5]–[7].

While they constitute fundamental basic low level objectives to be achieved in multi-agent systems, these coordinated behaviors are still far away from several interesting real world applications. Consider, as a motivating example, a group mobile robots that coordinate their motion for cooperatively executing a production task. For instance, we can imagine a team of mobile robots, each of which is equipped with a different tool, that cooperate to produce a particular object. Clearly, the operations to be performed need to be coordinated, and depend on the production cycle of the particular object to be produced.

This example clearly shows that the application of multi-agent systems to real world problems requires the development of sophisticated control strategies, for obtaining high level complex objectives. A few ideas that move towards these objectives have recently appeared in the literature. As shown in [8], it is possible to model a networked system in such a way that the classical notions of controllability and observability of LTI systems are applicable. Along these lines, in [1], [9] a decentralized methodology was introduced to solve a *tracking* problem for networked systems. Specifically, the well known Francis' *regulator equations* were exploited to design a control law to make a networked system follow a predefined setpoint. In particular, periodic setpoints were defined for each follower agent by means of an exosystem, as the linear combination of a given number of harmonics. The aforementioned control law requires feedback of the state of the dependent robots: in order to implement it in a decentralized manner, a decentralized estimation scheme was introduced in [1], [10]. In particular, this estimation scheme assumes that each independent robot has a partial knowledge of the output of the system, and that a connected communication graph exists among the independent robots. A consensus-based system is then implemented for the independent robots to obtain a complete knowledge of the output of the system, that is then exploited in a standard state observer, that provides an estimate of the state of the dependent robots.

While several strategies can be found in the literature for obtaining preservation of the connectivity of the communication graph among the independent robots [11]–[15], they introduce constraints on the admissible motion. Thus, in this paper we introduce a completely decentralized estimation procedure that does not require communication among the independent robots.

The research on distributed and decentralized state estimation is a fertile and extensive field, with a lot of remarkable contributions. For a survey on distributed estimation see for example [16]. Decentralized estimation is often exploited for control purpose, since it allows for the implementation of control strategies based on global quantities relying only on locally available information [17], [18]. Two main different approaches have been proposed in the state of the art to the problem of distributed state estimation: the diffusion mechanism [19], [20], where the diffusion of the local estimations in neighbors is obtained after incremental update, and the consensus strategies [21], applied to obtain average observations or estimations at each iteration. Moreover, an

F. Boem is with the Dept. of Electrical and Electronic Engineering at the Imperial College London, UK. (f.boem@imperial.ac.uk)

L. Sabattini and C. Secchi are with the Department of Sciences and Methods for Engineering (DISMI), University of Modena and Reggio Emilia, Italy (lorenzo.sabattini, cristian.secchi @unimore.it)

important branch of research on distributed estimation is represented by distributed Kalman filters [22] and their combination with the diffusion mechanism [23], [24]. See [25] for a survey. An interesting new field is the link between distributed/decentralized estimation and distributed monitoring (see as example [26], [27] and [28]).

The contribution of the paper is the design of a decentralized state estimation method for control purposes. The proposed estimation method allows each independent robot to estimate the state of the dependent robots. Some local conditions on the filter matrix of the state estimator are derived, to guarantee the convergence of the estimation error to zero. The main novelty of this paper is the lack of information available at each independent robot, which is not able to communicate to the other independent robots, and the application scenario, requiring each node to estimate also the control input components computed by the other independent robots. The considered problem represents therefore a more challenging scenario than previous works.

The paper is organized as follows. In Section II, the notation used throughout the paper is presented. The considered problem is introduced in Section III. In Section IV the decentralized state estimator is designed and the related estimation error is analyzed in Section V. Then, some convergence conditions are derived in Section VI. The effectiveness of the proposed method is shown in simulation in Section VII. Some final remarks can be found in Section VIII.

II. Notation and mathematical operators

In this section we define some symbols that will be used throughout the paper.

The symbol $\mathbb{I}_\rho$ will be used to indicate the identity matrix in $\mathbb{R}^{\rho\times\rho}$, while the symbols $\mathbb{O}_\rho$ and $\mathbb{O}_{\rho,\sigma}$ will be used to indicate a square and a rectangular zero matrix in $\mathbb{R}^{\rho\times\rho}$ and in $\mathbb{R}^{\rho\times\sigma}$, respectively.

Moreover, we will use $v[i]$ to denote the $i-$th component of vector v, and $\Psi[i]$ to denote the $i-$th block of a block diagonal matrix Ψ.

Given a list of vectors $\chi_i \in \mathbb{R}^\rho$, $i = 1, \dots, \sigma$, we use the symbol $\mathrm{col}(\cdot)$ to denote the vector $\bar{\chi} \in \mathbb{R}^{\rho\sigma}$, namely

$$\bar{\chi} = \mathrm{col}(\chi_i, i = 1, \dots, \sigma)$$

that is obtained stacking all the vectors χ_i, $i = 1, \dots, \sigma$.

The symbol $\cdot \otimes \cdot$ will be used throughout the paper to indicate the Kronecker product. This operator exhibits the following *mixed product property* [29]:

Property 1: Let $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$ be matrices of opportune dimension, defined in such a way that $\mathcal{AC}$ and $\mathcal{BD}$ exist. Then,

$$(\mathcal{A} \otimes \mathcal{B})(\mathcal{C} \otimes \mathcal{D}) = \mathcal{AC} \otimes \mathcal{BD} \tag{1}$$

III. Problem formulation

Consider a system of N interconnected agents, whose interconnection is modeled by means of a connected undirected graph $\mathcal{G}$.

Let $x_i \in \mathbb{R}^m$ be the state of the $i-$th agent: without loss of generality, we can consider the case where the state corresponds to each agents position. Then, let the agents be interconnected according to the well known (weighted) consensus protocol [21]:

$$\dot{x}_i = -\sum_{j\in\mathcal{N}_i} w_{ij}(x_i - x_j) \tag{2}$$

where $w_{ij} > 0$ is the edge weight, and $\mathcal{N}_i$ is the set of the neighbors of the $i-$th agent, that is the set of the agents that are interconnected to the $i-$th one. Without loss of generality, in the following we will consider the scalar case, thus considering $x_i \in \mathbb{R}$. It is possible to extend all the results to the multi-dimensional case, considering each component separately. Let $x = [x_1, \dots, x_N]^\top \in \mathbb{R}^N$ be the state of the multi-agent system, collecting the single agents states. The relation defined in (2) can be rewritten as:

$$\dot{x} = -\mathcal{L}(\mathcal{G})x, \tag{3}$$

where $\mathcal{L}(\mathcal{G})$ is the Laplacian matrix associated to the graph $\mathcal{G}$; under the consensus protocol, the states of the agents converge to a common value, for a connected undirected graph $\mathcal{G}$ [21]. Let us now divide the agents into a few *independent robots*, to whom it is possible to inject a control action so that $x_i = u_i$[1], and the *dependent robots*, whose state evolves according to the consensus protocol (2). The goal is to control the states of the networked agents.

We will hereafter make the following assumption:

Assumption 1: No edge exists among the independent robots.

Namely, each independent robot is connected to one (or more) dependent robots; connections exist among the dependent robots; no exchange of information occurs among the independent robots.

An example of interconnection topology defined according to Assumption 1 is depicted in Fig. 1. In the picture, independent robots are highlighted with red dashed ellipses, while dependent robots are highlighted with green solid ellipses: black lines represent interconnections among the robots.

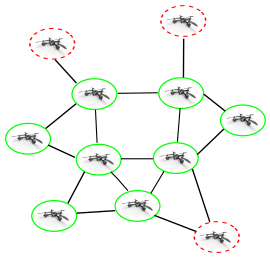


Fig. 1. Independent robots are highlighted with red dashed ellipses, while dependent robots are highlighted with green solid ellipses. Black lines represent interconnections among the robots: under Assumption 1, no edge exist among the independent robots

[1]For ease of notation, in this paper, we consider the independent robots as agents whose position can be directly controlled. Even though this assumption is not realistic, when dealing with real-robotic systems, it allows us to keep the notation simple, which improves the readability of the paper. The desired behavior can be obtained on real independent robots by means of a sufficiently fast and accurate position control.

Let N_I be the number of independent robots, and $N_D = N - N_I$ be the number of dependent robots. It is always possible to index the agents in such a way that the first N_D agents are the dependent robots and the last N_I are the independent robots. Therefore, under Assumption 1, it is possible to decompose the Laplacian matrix $\mathcal{L}(\mathcal{G})$ as:

$$\mathcal{L}(\mathcal{G}) = -\begin{bmatrix} A & B \\ B^\top & C \end{bmatrix}, \tag{4}$$

where $A = A^\top \in \mathbb{R}^{N_D \times N_D}$ represents the interconnection among the dependent robots, and $B \in \mathbb{R}^{N_D \times N_I}$ represents the interconnection among dependent and independent robots. Since, under Assumption 1, no interconnection exists among the independent robots, the corresponding lower-right block of the Laplacian matrix, namely $C \in \mathbb{R}^{N_I \times N_I}$, is a diagonal matrix, whose $i-$th diagonal entry is equal to the sum of the weights of the edges that connect the $i-$th independent robot with its neighboring dependent robots.

Let us define $x_D \in \mathbb{R}^{N_D}$ as the dependent robots state vector, that is $x_D = [x_1, \ldots, x_{N_D}]^\top$ and $u \in \mathbb{R}^{N_I}$ as the input vector: $u = [x_{N_D+1}, \ldots, x_N]^\top$. Moreover, let $y \in \mathbb{R}^{N_I}$ be the output vector, that is the vector containing the state variables measured by the independent robot: in particular, we assume each independent robot to be able to measure the state of the dependent robots it is connected with. The dynamics of the networked system can then be modeled as a LTI system as

$$\begin{aligned} \dot{x}_D &= Ax_D + Bu \\ y &= B^\top x_D, \end{aligned} \tag{5}$$

where each independent robot can measure only one component of the output y. Therefore, the output equation for each $i-$th independent robot can be rewritten as:

$$y[i] = b_i^\top x_D \tag{6}$$

for each component $i = 1, \ldots, N_I$, being $b_i^\top$ the $i-$th row of $B^\top$. In [10], the well known Francis' regulator equations are used to design a decentralized control law to make the networked system follow a predefined setpoint. In particular, periodic setpoints are defined for each follower agent by means of an exosystem, as the linear combination of n harmonics, that is a linear combination of the elements of the following vector:

$$\xi(t) = \left[1 \ \sin\left(\frac{2\pi}{T}t\right) \ \cos\left(\frac{2\pi}{T}t\right) \ \ldots \ \sin\left(n\frac{2\pi}{T}t\right) \ \cos\left(n\frac{2\pi}{T}t\right)\right]^T \tag{7}$$

A periodic setpoint can then be defined as a linear combination of the components of ξ, defining a matrix $\mathcal{J} \in \mathbb{R}^{N_D \times \bar{n}}$ such that

$$x_s(t) = \mathcal{J}\xi(t) \tag{8}$$

where $\bar{n} = (2n + 1)$.

It is possible to define a linear exosystem

$$\dot{\xi}(t) = G\xi(t) \tag{9}$$

where $G \in \mathbb{R}^{\bar{n} \times \bar{n}}$ is an opportunely defined marginally stable matrix [1], that, initialized with

$$\xi(0) = [1, 0, 1, \ldots, 1]^T$$

yields (7) as a solution.

The control law is designed as:

$$u = Fx_D + (\Gamma - F\Pi)\xi \tag{10}$$

where matrices Γ and Π are opportunely defined as in [1] from the solution of the regulator equations, and $F \in \mathbb{R}^{N_I \times N_D}$ is defined so that $(A + BF)$ is Hurwitz stable.

It is worth remarking that, as known from basic linear control theory, Γ, Π, F can always be found if the pair (A, B) is controllable. As shown in [30], given a connected undirected graph $\mathcal{G}$, utilizing randomly chosen edge weights it is possible to ensure controllability of the pair (A, B) with probability one. On these lines, we will hereafter assume controllability of the pair (A, B).

Since the state x_D is not completely available to the independent robots, we derive a decentralized state observer, so that the following control law can be implemented

$$u = F\hat{x}_D + (\Gamma - F\Pi)\xi \tag{11}$$

where $\hat{x}_D$ is an estimate of the state x, and we demonstrate that asymptotic tracking of periodic setpoint trajectories is achieved.

IV. The decentralized state observer

In order to make each independent robot able to implement the control strategy in (11) exploiting only locally available information, it is necessary to derive a state estimator of the state x_D. Moreover, since each independent robot is not able to communicate to other independent robots, it is necessary to estimate the control input components computed by the other independent robots.

We assume that all the matrices (system and regulator matrices) are known and constant. Hence, each independent robot i estimates the input vector using its own state estimate $\hat{x}_i$ as:

$$\hat{u}_i = F\hat{x}_i + (\Gamma - F\Pi)\xi, \tag{12}$$

where the state estimate $\hat{x}_i$ dynamics are computed as:

$$\dot{\hat{x}}_i = A\hat{x}_i + B\hat{u}_i - k_i(y[i] - b_i^\top \hat{x}_i) \tag{13}$$

with k_i being a column vector containing the $i-$th row of the filter matrix K that will be defined in the following sections to guarantee the desired convergence properties. In fact, it is worth noting that the estimation scheme (13) is implemented by each independent robot based only on locally available data, that is without knowledge of the other independent robots' state estimates. For this reason, standard estimation schemes can not be adopted in this case.

V. Estimation error analysis

For analysis purposes, we consider an extended formulation of the state vector estimate $\hat{x}_E$, collecting the state estimates of the independent robots, that is

$$\hat{x}_E = \text{col}(\hat{x}_i, i = 1, \dots, N_I).$$

The dynamics of the extended estimator can be described as:

$$\dot{\hat{x}}_E = A_E\hat{x}_E + B_E\hat{u}_E - K_E(y - D_E\hat{x}_E), \quad (14)$$

where $A_E = \mathbb{I}_{N_I} \otimes A$ is a block matrix having non-null blocks only on the diagonal, equal to A; $B_E = \mathbb{I}_{N_I} \otimes B$ is defined in an analogous way; $\hat{u}_E$ can be computed similarly to $\hat{x}_E$ as

$$\hat{u}_E = \text{col}(\hat{u}_i, i = 1, \dots, N_I),$$

K_E is a $N_I N_I \times N_I$ block matrix having on the diagonal the column vectors k_i; similarly, D_E is a $N_I \times N_I N_D$ block matrix having on the diagonal the rows $b_i^\top$. Since (12) can be rewritten in the extended form

$$\hat{u}_E = F_E\hat{x}_E + \Omega_E\xi_E, \quad (15)$$

where ξ_E is a vector collecting N_I times the state vector of the exosystem ξ, $F_E = \mathbb{I}_{N_I} \otimes F$ and $\Omega_E = \mathbb{I}_{N_I} \otimes (\Gamma - F\Pi)$ are diagonal block matrices having the matrix F and $(\Gamma - F\Pi)$ repeated on the diagonal respectively, (14) becomes

$$\dot{\hat{x}}_E = (A_E + B_E F_E + K_E D_E)\hat{x}_E + B_E\Omega_E\xi_E - K_E y. \quad (16)$$

Define now the extended estimation error $\epsilon \in \mathbb{R}^{N_I N_D}$ as follows:

$$\epsilon = \hat{x}_E - x_E \quad (17)$$

where x_E is a vector collecting N_I times the state vector x_D. We now analyze the dynamics of the extended estimation error $\dot{\epsilon} = \dot{\hat{x}}_E - \dot{x}_E$, using (16) and the extended version of (5):

$$\dot{x}_E = A_E x_E + B_E u_E \quad (18)$$

with

$$u_E = \tilde{F}_E\hat{x}_E + \Omega_E\xi_E, \quad (19)$$

where $\tilde{F}_E$ is a block matrix, where each $(i, j)-$th block (with $i, j = 1, \dots, N_I$) is a null matrix except from the $j-$th row of matrix F.

The dynamics of the extended estimation error are given by:

$$\dot{\epsilon} = (A_E + B_E F_E + K_E D_E)\hat{x}_E + B_E\Omega_E\xi_E - K_E y - A_E x_E - B_E\tilde{F}_E\hat{x}_E - B_E\Omega_E\xi_E. \quad (20)$$

We can observe that the output can be rewritten using extended vectors as

$$y = D_E x_E. \quad (21)$$

Moreover, since $\tilde{F}_E x_E - F_E x_E = 0$, it holds

$$B_E F_E\hat{x}_E - B_E\tilde{F}_E\hat{x}_E = B_E(F_E - \tilde{F}_E)\hat{x}_E = B_E[F_E\hat{x}_E - \tilde{F}_E\hat{x}_E - F_E x_E + \tilde{F}_E x_E] = B_E[F_E - \tilde{F}_E]\epsilon. \quad (22)$$

So we have

$$\dot{\epsilon} = (A_E + B_E(F_E - \tilde{F}_E) + K_E D_E)\epsilon. \quad (23)$$

Matrix $K_E D_E$ is a block diagonal matrix where each block on the diagonal is based on the outer product $k_i b_i^\top$. Matrix $\tilde{F}_E$ can be computed as:

$$\tilde{F}_E = \tilde{I}_E F_E,$$

where $\tilde{I}_E$ is a $N_I m \times N_I m$ block matrix, each block (i, j) having one single element different from 0 in correspondence to the (j, j) diagonal element. Therefore, (23) can be rewritten as

$$\dot{\epsilon} = (A_E + K_E D_E + B_E(\mathbb{I}_{N_I m} - \tilde{I}_E)F_E)\epsilon = (\Lambda + \tilde{B}F_E)\epsilon, \quad (24)$$

with $\Lambda = A_E + K_E D_E$ is a block diagonal matrix, being each $i-$th block equal to $A + k_i b_i^\top$, and $\tilde{B} = B_E(\mathbb{I}_{N_I m} - \tilde{I}_E)$.

VI. Convergence analysis

In this section we analyze the convergence properties of the proposed estimation scheme. In particular, the following Theorem provides a methodology for defining matrix K in (13) in such a way that the estimation error asymptotically converges to zero.

Theorem 1: Consider state estimation scheme defined in (13), and let $\lambda_{i,min}$ be the minimum eigenvalue of $A + k_i b_i^\top + BF$, $\forall i = 1, \dots, N_I$. If, $\forall i, \dots, N_I$, k_i is defined in such a way that the following holds

$$\lambda_{i,min} \leq -\left\| (\mathbb{I}_{N_I} \otimes B)\,\tilde{F}_E \right\|, \quad (25)$$

then the estimation error (24) converges asymptotically to zero.

Proof: We need to prove that (24) is asymptotically stable. More specifically, we have that:

- A, B and related matrices are defined by the system topology;
- F and related matrices are defined by the control law, in order to ensure the desired tracking performances.

We define matrix K in order to guarantee the convergence of the estimation error. Let us consider the Lyapunov function

$$V = \frac{1}{2}\epsilon^\top\epsilon\,. \quad (26)$$

To guarantee that the estimation error goes to zero, we need to ensure that the derivative of the Lyapunov function is negative (semi)definite, namely

$$\dot{V} = \frac{\partial V}{\partial \epsilon}\dot{\epsilon} = \epsilon^\top\dot{\epsilon} \leq 0 \quad (27)$$

Let us consider (24) and let matrix Ψ be defined as follows:

$$\Psi = \Lambda + \mathbb{I}_{N_I} \otimes (BF)\,. \quad (28)$$

Then, it is possible to rewrite (24) as:

$$\dot{\epsilon} = \left(\Psi - (\mathbb{I}_{N_I} \otimes B)\,\tilde{F}_E\right)\epsilon \quad (29)$$

by noting that

$$B_E F_E = (\mathbb{I}_{N_I} \otimes B)\,(\mathbb{I}_{N_I} \otimes F) \quad (30)$$

can be rewritten as

$$B_E F_E = \mathbb{I}_{N_I} \otimes (BF) \tag{31}$$

according to (1) (Property 1). It is worth noting that matrix Ψ is a block diagonal matrix, whose $i-$th block $\Psi[i]$ is equal to

$$\Psi[i] = A + BF + k_i b_i^\top \tag{32}$$

Let us now consider the Lyapunov derivative (27): it can be rewritten as

$$\begin{aligned} \dot{V} &= \epsilon^\top \left(\Psi\epsilon - (\mathbb{I}_{N_I} \otimes B)\, \tilde{F}_E \epsilon \right) \\ &= \epsilon^\top \Psi \epsilon - \epsilon^\top \left((\mathbb{I}_{N_I} \otimes B)\, \tilde{F}_E \right) \epsilon \\ &\leq \epsilon^\top \Psi \epsilon + \left\| (\mathbb{I}_{N_I} \otimes B)\, \tilde{F}_E \right\| \epsilon^\top \epsilon \end{aligned} \tag{33}$$

It is worth noting that the value of

$$\left\| (\mathbb{I}_{N_I} \otimes B)\, \tilde{F}_E \right\|$$

can be computed, once B and F have been defined.

In order to prove the theorem, it is therefore necessary to define K so that the following holds:

$$\epsilon^\top \Psi \epsilon \leq - \left\| (\mathbb{I}_{N_I} \otimes B)\, \tilde{F}_E \right\| \epsilon^\top \epsilon \tag{34}$$

From the theory [31], it holds for a symmetric matrix $\mathcal{A} = \mathcal{A}^\top$:

$$x^\top \mathcal{A} x \leq \lambda_{min}(\mathcal{A}) x^\top x,$$

being $\lambda_{min}(\mathcal{A})$ the minimum eigenvalue of $\mathcal{A}$; moreover, $x^\top \mathcal{A} x = x^\top ((\mathcal{A} + \mathcal{A}^\top)/2) x$. Then we can write:

$$\epsilon^\top \Psi \epsilon = \epsilon^\top ((\Psi + \Psi^\top)/2) \epsilon \leq \lambda_{min}((\Psi + \Psi^\top)/2) \epsilon^\top \epsilon. \tag{35}$$

Since $\lambda_{min}((\Psi + \Psi^\top)/2) \leq \lambda_{min}(\Psi)$ [32], we have the following condition

$$\epsilon^\top \Psi \epsilon \leq \lambda_{min} \epsilon^\top \epsilon, \tag{36}$$

with λ_{min} the minimum eigenvalue of Ψ. We can rewrite condition (34) as

$$\lambda_{min} \epsilon^\top \epsilon \leq - \left\| (\mathbb{I}_{N_I} \otimes B)\, \tilde{F}_E \right\| \epsilon^\top \epsilon \tag{37}$$

Then, we obtain

$$\lambda_{min} \leq - \left\| (\mathbb{I}_{N_I} \otimes B)\, \tilde{F}_E \right\| \tag{38}$$

that can be computed locally since Ψ is a block diagonal matrix: we have, $\forall i$,

$$\lambda_{i,min} \leq - \left\| (\mathbb{I}_{N_I} \otimes B)\, \tilde{F}_E \right\|,$$

being $\lambda_{i,min}$ the minimum eigenvalue of $A + k_i b_i^\top + BF$. Therefore, k_i has to be designed so that (25) holds. ■

In the following, we show a constructive algorithm to define matrix K such that the condition in Theorem 1 holds. This algorithm implies the design of the control matrix F. In the case that the obtained matrix F does not satisfy the initial requirement that $(A + BF)$ is Hurwitz stable, other procedures have to be followed. Even if we have not theoretical results about this, in all the developed simulation examples the obtained matrix F satisfies the required conditions.

A. Constructive algorithm

Let (A, B) be a controllable pair and let b_i be a non null column of B. In order to find k_i satisfying (25) we can use pole placement techniques. It is possible to demonstrate that there always exists k_i such that $A + k_i b_i^\top + BF$ has the desired set of eigenvalues. In order to guarantee that this problem is feasible, we have to check controllability of the couple $\left((A + BF)^\top, b_i\right)$, by noting that

$$eig[A + k_i b_i^T + BF] = eig[(A + BF)^\top + b_i k_i^\top].$$

Since (A, B) is controllable by assumption, using the results in Heymann's Lemma [33], it is always possible to find a matrix F such that the pair $(A + BF, b_i)$ is controllable. Consequently it is always possible to find a row $k_i^\top$ such that the matrix $A + BF + k_i b_i^\top$ has the desired set of eigenvalues. Controllability is a structural property for LTI systems and, therefore, it is invariant with respect to static feedback. Thus, the pair $(A + BF + k_i b_i^\top, b_i)$ is controllable. We can always write $b_i k_i^\top = BK_i$ where all the rows of K_i are null but the $i-$th one, that is equal to $k_i^\top$. Setting $K = F + K_i$, we have that the pair $(A + BK, b_i)$ is controllable and $A + BK$ has the desired spectrum. The final step is to check that $(A + BF)$, with the obtained matrix F, is Hurwitz stable.

VII. Simulations

Several simulations have been carried out in order to evaluate the performance of the proposed control strategy. In the simulations, we considered single integrator agents moving in a three dimensional environment, namely $x_i \in \mathbb{R}^3$, $\forall i = 1, \ldots, N$. Let $(\mathsf{x}, \mathsf{y}, \mathsf{z})$ represent the global reference frame.

Specifically, different graph topologies (similar to Fig. 1) have been considered, with variable number of dependent and independent robots. The results of a representative simulation run are reported in Fig. 2

In order to evaluate the performance of the proposed estimation and control strategy, we computed both the estimation error $\epsilon(t)$ defined in (17) and the tracking error $e(t)$ defined as follows:

$$e\,(t) = x_D\,(t) - x_s\,(t) \tag{39}$$

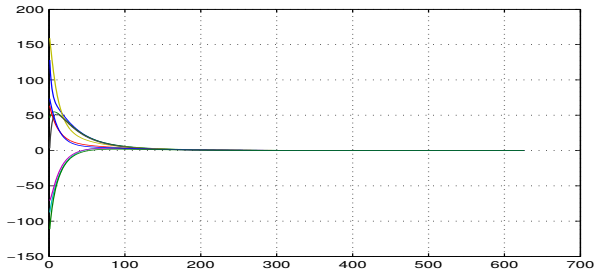
where the setpoint $x_s\,(t)$ is computed as in (8).

Specifically, Fig. 2(a) shows the evolution of the estimation error $\epsilon(t)$, while Fig. 2(b) shows the evolution of the tracking error $e(t)$. Due to space limitations, the plot depicts only the components along the x-axis, and only the estimation error for the first independent robot.

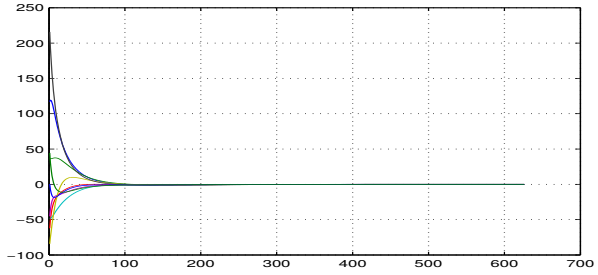
As expected, the estimation error quickly goes to zero and, subsequently, the tracking error goes to zero as well.

VIII. Conclusions

This paper proposes a decentralized state estimation method. The purpose is the control of multi-agent networked systems in order to track arbitrary setpoint trajectories. The proposed state estimation algorithm is designed allowing each independent robot to estimate the input of the other independent robots and the state of the dependent robots,



(a) Estimation error



(b) Tracking error

Fig. 2. 3 independent robots, 9 dependent robots

without requiring communication among the independent robots. As a future work, we aim at investigating the case that the measurements taken by the independent robots are affected by noise. Furthermore, we would like to analyze the scenario where the network topology may change along the time.

REFERENCES

[1] L. Sabattini, C. Secchi, M. Cocetti, A. Levratti, and C. Fantuzzi, "Implementation of coordinated complex dynamic behaviors in multi-robot systems," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 1018–1032, aug. 2015.

[2] L. Sabattini, C. Secchi, and C. Fantuzzi, "Arbitrarily shaped formations of mobile robots: artificial potential fields and coordinate transformation," *Autonomous Robots (Springer)*, vol. 30, no. 4, pp. 385–397, may 2011.

[3] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *Communications Magazine, IEEE*, vol. 40, no. 8, pp. 102–114, 2002.

[4] P. R. Wurman, R. D'Andrea, and M. Mountz, "Coordinating hundreds of cooperative, autonomous vehicles in warehouses," *AI Magazine*, vol. 29, no. 1, pp. 9–20, 2008.

[5] A. Ganguli, J. Cortes, and F. Bullo, "Multirobot rendezvous with visibility sensors in nonconvex environments," *Robotics, IEEE Transactions on*, vol. 25, no. 2, pp. 340 –352, april 2009.

[6] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," 2004.

[7] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi–agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[8] M. Egerstedt, S. Martini, M. Cao, K. Camlibel, and A. Bicchi, "Interacting with networks: How does structure relate to controllability in single-leader, consensus networks?" *Control Systems, IEEE*, vol. 32, no. 4, pp. 66–73, 2012.

[9] M. Cocetti, L. Sabattini, C. Secchi, and C. Fantuzzi, "Decentralized control strategy for the implementation of cooperative dynamic behaviors in networked systems," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 5902–5907.

[10] L. Sabattini, C. Secchi, and C. Fantuzzi, "Cooperative dynamic behaviors in networked systems with decentralized state estimation," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, Sept 2014, pp. 3782–3787.

[11] L. Sabattini, N. Chopra, and C. Secchi, "Decentralized connectivity maintenance for cooperative control of mobile robotic systems," *The International Journal of Robotics Research (SAGE)*, no. 12, pp. 1411–1423, Oct. 2013.

[12] L. Sabattini, C. Secchi, N. Chopra, and A. Gasparri, "Distributed control of multi-robot systems with global connectivity maintenance," *IEEE Transactions on Robotics*, no. 5, pp. 1326 – 1332, Oct. 2013.

[13] P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff, "A passivity-based decentralized strategy for generalized connectivity maintenance," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 299–323, 2013.

[14] A. Ajorlou, A. Momeni, and A. G. Aghdam, "A class of bounded distributed control strategies for connectivity preservation in multi–agent systems," *IEEE Transactions on Automatic Control*, vol. 55, pp. 2828–2833, 2010.

[15] M. M. Zavlanos, M. B. Egerstedt, and G. J. Pappas, "Graph-theoretic connectivity control of mobile robot networks," *Proceedings of the IEEE*, vol. 99, pp. 1525–1540, 2011.

[16] F. Garin and L. Schenato, "A survey on distributed estimation and control applications using linear consensus algorithms," in *Networked Control Systems*. Springer, 2010, vol. 406, pp. 75–107.

[17] P. Yang, R. A. Freeman, and K. M. Lynch, "Multi-agent coordination by decentralized estimation and control," *IEEE Transactions on Automatic Control*, vol. 53, no. 11, pp. 2480–2496, 2008.

[18] D. Viegas, P. Batista, P. Oliveira, C. Silvestre, and C. L. P. Chen, "Distributed state estimation for linear multi-agent systems with time-varying measurement topology," *Automatica*, vol. 54, pp. 72–79, 2015.

[19] A. Speranzon, C. Fischione, K. Johansson, and A. Sangiovanni-Vincentelli, "A distributed minimum variance estimator for sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 609–621, 2008.

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

[21] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[22] R. Olfati-Saber, "Distributed kalman filtering for sensor networks," in *Decision and Control, 46th IEEE Conf. on*, 2007, pp. 5492–5498.

[23] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, "Distributed kalman filtering based on consensus strategies," *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 4, pp. 622–633, 2008.

[24] R. Olfati-Saber, "Kalman-consensus filter : Optimality, stability, and performance," in *Decision and Control, 48th IEEE Conf. on, held jointly with 28th Chinese Control Conference*, 2009, pp. 7036–7042.

[25] M. S. Mahmoud and H. M. Khalid, "Distributed kalman filtering: a bibliographic review," *Control Theory & Applications, IET*, vol. 7, no. 4, 2013.

[26] E. Franco, R. Olfati-Saber, T. Parisini, and M. Polycarpou, "Distributed fault diagnosis using sensor networks and consensus-based filters," in *Decision and Control, 45th IEEE Conf. on*, 2006, pp. 386 –391.

[27] F. Boem, R. Ferrari, T. Parisini, and M. Polycarpou, "Distributed fault detection and isolation of continuous-time nonlinear systems," *European Journal of Control*, vol. 5-6, pp. 603–620, 2011.

[28] F. Boem, Y. Xu, C. Fischione, and T. Parisini, "Distributed fault detection using sensor networks and pareto estimation," in *European Control Conference*, July 2013, pp. 932–937.

[29] A. J. Laub, *Matrix analysis for scientists and engineers*. Siam, 2005.

[30] L. Sabattini, C. Secchi, and C. Fantuzzi, "Controllability and observability preservation for networked systems with time varying topologies," in *Proceedings of the IFAC World Congress*, Cape Town, South Africa, aug. 2014, pp. 1837 – 1842.

[31] C. D. Meyer, *Matrix analysis and applied linear algebra*. Siam, 2000.

[32] F. Zhang, *Matrix theory: basic results and techniques*. Springer Science & Business Media, 2011.

[33] M. Heymann and W. Wonham, "Comments "on pole assignment in multi-input controllable linear systems"," *IEEE Transactions on Automatic Control*, vol. 13, no. 6, pp. 748–749, December 1968.