Decentralized State Estimation for the Control of Network Systems☆

Francesca Boem[a,*], Lorenzo Sabattini[b], Cristian Secchi[b]

[a] *Dept. of Electrical and Electronic Engineering, Imperial College London, UK*
[b] *Dept. of Sciences and Methods for Engineering (DISMI), University of Modena and Reggio Emilia, Italy*

Abstract

The paper proposes a decentralized state estimation method for the control of network systems, where a cooperative objective has to be achieved. The nodes of the network are partitioned into independent nodes, providing the control inputs, and dependent nodes, controlled by local interaction laws. The proposed state estimation algorithm allows the independent nodes to estimate the state of the dependent nodes in a completely decentralized way. To do that, it is necessary for each independent node of the network to estimate the control input components computed by the other independent nodes, without requiring communication among the independent nodes. 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.

Keywords: Network Systems, Multi-Agent, Decentralized State Estimation, Cooperative behavior.

1. Introduction

The paper considers the control of network systems problem, where the goal is to achieve some desired dynamic cooperative behavior. Network systems

☆Preliminary results are presented in [1].

*Corresponding author

Email addresses: `f.boem@imperial.ac.uk` (Francesca Boem), `lorenzo.sabattini@unimore.it` (Lorenzo Sabattini), `cristian.secchi@unimore.it` (Cristian Secchi)

include, as example, sensor networks [2], networks of autonomous agents (such as robots and UAVs) [3], biological networks [4], transportation networks [5], microgrids [6], social [7] and economic networks.

Decentralized control of network systems has been widely addressed in the last few years, mainly in the application fields of multi-robot systems [3], distributed sensor networks [2], and interconnected manufacturing equipments [8]. Controllability issues of network systems are analyzed in [9].

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. In fact, mainly investigated coordinated behaviors include aggregation, swarming, formation control, coverage and synchronization [10, 11, 12, 3, 13, 14, 15]. 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.

In this paper, we consider *heterogeneous* networks, composed of nodes that are assigned with different roles. In particular, the nodes are partitioned into *independent nodes*, providing control inputs, and *dependent nodes*, controlled through local interaction. In this paper, in order to control the state of the entire network, we propose a decentralized estimation method to let the independent nodes estimate the state of the dependent nodes. In particular, both the estimation and the control phases are computed locally at each node in a decentralized way.

As a motivating example, consider the problem analyzed in [16, 17], where a team of mobile robots is controled to implement a cooperative dynamic behavior, modeled as the cooperative tracking of desired periodic trajectories. The motivation behind this work was to provide a model for cooperative operations, similar to those performed by groups of human operators, such as the production cycle for a certain object, or the construction of a building. The team of robots was partitioned into two groups: a (small) set of independent robots was used as control inputs for the (large) set of dependent robots, evolving according to local interaction. In order to achieve such a cooperative behavior, the input for the system depends on the state of the entire team.

Another example is represented by microgrids [6] for energy production and delivery. In these systems, distributed generators adapt their production rate (i.e. the input for the system) based on the need of the loads (i.e. the state of the system), which are connected by means of a network. Efficient management of microgrids requires knowledge of the state of the entire network.

In these application examples, independent nodes (i.e. the input points for the network system) have usually access only to a subset of the state variables of the system: namely, they can only measure the state of their neighboring nodes. It is worth noting that, if a connected communication network exists among the control nodes, information can be shared among them. However, this is not always feasible, nor reliable, and can cause drawbacks. In particular, when considering static nodes (e.g. the generators in a microgrid) having a connected communication network to exchange information among the nodes requires significant infrastructure, which may not be feasible, and may raise security and privacy issues. On the other hand, when considering mobile nodes (e.g. mobile robots), several strategies exist in the literature to guarantee connectivity preservation [18, 19, 20, 21, 22], but they introduce constraints on the admissible trajectories of the nodes, that may be undesirable.

To address this kind of problems, in this paper we introduce a completely decentralized estimation procedure: without requiring any communication among the independent nodes, the proposed method provides a reliable estimation of the entire state of the network system that can be used for control purposes.

The paper is organized as follows. A review of the relevant literature and the main contributions of the paper are provided in Section 2. In Section 3, the notation used throughout the paper is presented. The considered problem is introduced in Section 4. In Section 5 the decentralized state estimator is designed and the related estimation error is analyzed in Section 6. Then, some convergence conditions are derived in Section 7. The effectiveness of the proposed method is shown in simulation in Section 9. Some final remarks can be found in Section 10.

2. Related work

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, [23]. Decentralized estimation is often exploited for control purposes, since it allows the implementation of control strategies based on global quantities relying only on locally available information [24, 25]. Two main different approaches have been proposed in the state of the art to the problem of distributed state estimation: the diffusion mechanism [26, 27], where the diffusion of the local estimations in neighbors is obtained after incremental update, and the consensus strategies [28], applied to obtain average observations or estimations at each iteration. Moreover, an important branch of research on distributed estimation is represented by distributed Kalman filters [29] and their combination with the diffusion mechanism [30, 31]. See [32] for a survey. An interesting new field is the link between distributed/decentralized estimation and distributed monitoring (see as example [33], [34] and [35]).

The contribution of the paper is the design of a decentralized state estimation method to control network systems. The proposed estimation method allows each independent node to estimate the state of the dependent nodes. 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 node, which is not able to communicate to the other independent nodes: in this manner, it is not necessary to assume the presence of any direct communication among the independent nodes. Moreover, each independent node not only needs to estimate the state of the other nodes, but also the control input components computed by the other independent nodes. The considered problem represents therefore a more challenging scenario than previous works. In this paper, we extend the results obtained in [1] for multi-agent systems controlled by consensus interactions, to the general framework of network systems. Furthermore, in this paper i) we provide extended proofs of the theoretical results; ii) we describe several simulations performed in different scenarios; and iii) we analyze the effects of the measurement noise and of the lack of communication between independent nodes on the performance of the proposed estimation and control schemes.

A similar estimation problem is considered in [36] for multi-robot systems, where an observer-controller scheme is presented for fault diagnosis, but differently from our proposed method, the observer uses information communicated from neighbours in a distributed way and the convergence proof uses the strongly connected communication graph assumption.

Many works in the literature consider single-leader networks (see [37] as example), while here the leaders can be many and have to cooperate without communication.

In [38], the problem of estimating the state of a multi-agent system based on asynchronous and noisy measurements is considered, but the estimation task is not carried out by the agents.

3. 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 Ψ_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* [39]:

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,*

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

Given a random variable $\psi \in \mathbb{R}$, we will denote with $\mathbb{E}\psi$ its expected value.

4. Problem formulation

Consider a set $\mathcal{V}$ of N nodes, 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 node and consider the following single-integrator kinematic model:

$$\dot{x}_i = \mu_i \tag{2}$$

where $\mu_i \in \mathbb{R}^m$ is each node's control input.

In order to keep the notation simple, in the following we will consider the scalar case, thus assuming $x_i \in \mathbb{R}$ and that the results can be extended to the multi-dimensional case, considering each component separately.

Let us now divide the nodes into two sets: a (small) set $\mathcal{V}_I \subset \mathcal{V}$ of *independent nodes*, to whom it is possible to inject an external control action, and a set $\mathcal{V}_D = \mathcal{V} \setminus \mathcal{V}_I$ of *dependent nodes*, whose state evolves according to a local control action, based on the state of the neighboring nodes. Let N_I be the number of independent nodes, and $N_D = N - N_I$ be the number of dependent nodes. We consider the challenging scenario where there is no communication between independent nodes.

Let $x \in \mathbb{R}^N$ be the state of the network system, collecting the nodes' states x_i, $i = 1, \ldots, N$, ordered putting first the N_D dependent nodes and then the N_I independent nodes. We will hereafter consider the nodes interconnected in such a way that the dynamics of the network system can be written as follows:

$$\dot{x} = \begin{bmatrix} A & B \\ \mathbb{O}_{N_I,N_D} & \mathbb{O}_{N_I,N_I} \end{bmatrix} x + \begin{bmatrix} \mathbb{O}_{N_D,N_I} \\ \mathbb{I}_{N_I} \end{bmatrix} \nu \tag{3}$$

where $\nu \in \mathbb{R}^{N_I}$ is the input vector, $A \in \mathbb{R}^{N_D \times N_D}$ represents the interconnection among the dependent nodes, and $B \in \mathbb{R}^{N_D \times N_I}$ represents the interconnection among dependent and independent nodes. This represents a generic interconnection among the nodes: for instance, as shown in [40], it can be obtained from the consensus interconnection.

Define now $x_D = [x_1, \ldots, x_{N_D}]^\top \in \mathbb{R}^{N_D}$ as the state vector of the dependent nodes, and $x_I = [x_{N_D+1}, \ldots, x_N]^\top \in \mathbb{R}^{N_I}$ as the state vector of the independent nodes. According to (3), the dynamics of the independent nodes can be written as the following single-integrator model:

$$\dot{x}_I = \nu \tag{4}$$

We introduce now $u \in \mathbb{R}^{N_I}$ as the desired input that we would like to introduce into the system by means of the independent nodes, in order to achieve some cooperative objective. For this purpose, we define the input ν as follows:

$$\nu = \dot{u} - \mathcal{H}\,(x_I - u) \tag{5}$$

with $\mathcal{H} \in \mathbb{R}^{N_I \times N_I}$. It is possible to show that, for any choice of $\mathcal{H} > 0$, under the control law (5), the dynamics of the independent nodes exponentially converge to the desired input u. In particular, matrix $\mathcal{H}$ can be designed for making such a convergence arbitrarily fast. Hence, in order to keep the notation simple, we will hereafter assume that the independent nodes are controlled in such a way that

$$x_I \approx u \tag{6}$$

If, for instance, nodes are represented by mobile robots whose state is the position, this corresponds to imposing a sufficiently fast and accurate position control.

Hence, from (3) and (6), the dynamics of the dependent nodes can be written as follows:

$$\dot{x}_D = Ax_D + Bu \tag{7}$$

Each independent node is then used to inject the control input u_i, $i = 1, \ldots, N_I$, at different points of the network according to matrix B. To achieve this objective, we assume that each independent node is able to measure the state of the dependent nodes it is connected with, according to matrix $B^\top$.

$$y = B^\top x_D, \tag{8}$$

where $y \in \mathbb{R}^{N_I}$ is the output vector, that is the vector containing the state variables measured by the independent nodes, where each independent node can measure only its corresponding components of the output y. Therefore, the output equation for each $i-$th independent node can be rewritten as:

$$y_i = b_i^\top x_D \tag{9}$$

for each component $i = 1, \ldots, N_I$, being $b_i^\top$ the $i-$th row of $B^\top$.

The interconnection between the N nodes can be modeled by means of a connected undirected graph $\mathcal{G}$. We will hereafter make the following assumption:

Assumption 1. *$\mathcal{G}$ is a connected undirected graph.*

Furthermore, in the scenario we are considering there is no communication between independent nodes, that is, no edge exists among the independent nodes. Namely, information can be shared among the robots, and each independent node is connected to one (or more) dependent nodes. The graph then includes connections among the dependent nodes, while no exchange of information occurs among the independent nodes.

Furthermore, we make the following additional assumption:

Assumption 2. *The pair (A, B) is controllable.*

According to (3), matrices A, B are related to the interconnection structure. Controllability is typically guaranteed, for connected networks, under mild assumptions [41].

The goal is to control the state of the network system to achieve some cooperative objective. For this purpose, we design the control law as follows:

$$u = F x_D + \zeta \tag{10}$$

where the first term Fx_D is a stabilizing term, being $F \in \mathbb{R}^{N_I \times N_D}$ defined so that $(A+BF)$ is Hurwitz stable, and ζ is a general vector, an exogenous signal, known by the independent nodes, designed based on the cooperative objective we want to achieve.

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

$$u = F\hat{x}_D + \zeta \tag{11}$$

where $\hat{x}_D$ is an estimate of the state x_D, and we demonstrate that the state of the system converges to the desired configuration.

5. The decentralized state observer

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

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

$$\hat{u}_i = F\hat{d}_i + \zeta, \tag{12}$$

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

$$\dot{\hat{d}}_i = A\hat{d}_i + B\hat{u}_i - K_i(y_i - b_i^\top \hat{d}_i) \tag{13}$$

with $K_i \in \mathbb{R}^{N_D \times 1}$ being a matrix containing the weights related to the $i-$th node 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 node based only on locally available data, that is without knowledge of the other independent nodes' state estimates. For this reason, standard estimation schemes cannot be adopted in this case.

6. 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

$$\hat{x}_E = \mathrm{col}(\hat{d}_i, i = 1, \ldots, N_I).$$

The dynamics of the extended estimator can be described as:

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

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

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

K_E is a $N_I N_D \times N_I$ 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

$$\hat{u}_E = F_E \hat{x}_E + \Omega_E, \tag{15}$$

where $F_E = \mathbb{I}_{N_I} \otimes F$ is a diagonal block matrix having the matrix F repeated on the diagonal and $\Omega_E = \mathbb{I}_{N_I} \otimes \zeta$, (14) becomes

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

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

$$\epsilon = \hat{x}_E - x_E \tag{17}$$

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 (7):

$$\dot{x}_E = A_E x_E + B_E u_E \tag{18}$$

with

$$u_E = \tilde{F}_E \hat{x}_E + \Omega_E, \tag{19}$$

where $\tilde{F}_E$ is a block matrix, where each $(i, j)-$th block (with $i, j = 1, \ldots, N_I$) is a null matrix except from the $j-$th row of matrix F. An example of F, $\tilde{F}_E$ and F_E is provided in Appendix.

The dynamics of the extended estimation error are given by:

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

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

$$y = D_E x_E. \tag{21}$$

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

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

So we have

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

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

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

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

7. Convergence analysis

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

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

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

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

Proof. We need to prove the asymptotically stability of the system described by (24). 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 convergence performances.

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

$$V = \frac{1}{2}\epsilon^\top \epsilon\ . \tag{26}$$

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

$$\dot{V} = \frac{\partial V}{\partial \epsilon}\dot{\epsilon} = \epsilon^\top \dot{\epsilon} \leq 0 \tag{27}$$

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

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

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

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

by noting that

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

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

$$\Psi[i] = A + BF + K_i D_i \tag{32}$$

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

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

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:

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

From standard matrix theory [42], it holds for a symmetric matrix $\mathcal{A} = \mathcal{A}^\top$:

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

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 $\lambda_{min}((\Psi + \Psi^\top)/2) \leq \lambda_{min}(\Psi)$ [43], 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 $A + K_i D_i + BF$. Therefore, K_i has to be designed so that (25) holds.

Remark. It is worth noting that matrix K_i can be locally computing, only assuming that each independent node knows matrices F and B. In the following, we show a constructive algorithm to design matrix K so that the condition in Theorem 1 holds.

7.1. Constructive algorithm

In this section we provide a constructive method to define k_i so that (25) is satisfied, while guaranteeing that $(A + BF)$ is Hurwitz stable: this ensures both that the estimation error vanishes, and that the state has the desired convergence properties.

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

Algorithm 1 Constructive algorithm for node i to find k_i

1: Given (A, B) controllable

2: Heymann's lemma:

find $F \Rightarrow (A + BF, b_i)$ controllable

3: **if** $(A + BF)$ is Hurwitz stable **then**

4: Pole placement:

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

5: **end**

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 [44], 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, which 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. The constructive algorithm is summarized in Algorithm 1.

Remark 1. *Defining matrix F as in Line 2 of Algorithm 1, that is, according to the Heymann's Lemma, does not formally guarantee Hurwitz stability of matrix $(A + BF)$. Therefore, it is necessary to check this condition, as defined in Line 3 Algorithm 1. In the case that the obtained matrix F does not satisfy this requirement, other procedures have to be followed. In all the developed simulation examples the obtained matrix F satisfies the required condition.* $\diamond$

8. Measurement noise

For a more complete analysis, it is possible to consider the presence of measurement noises, by including an additive term in the output equation (9), thus obtaining:

$$y_i = b_i^\top x_D + \eta[i],$$

where $\eta[i]$ is a random noise for the i-th node with mean value equal to $\bar{\eta}_i$, and standard deviation equal to $\bar{\sigma}_i$. Therefore, we can rewrite the output equation using the extended formulation and we obtain: $y = D_E x_E + \eta$, where η is a vector collecting all the components of $\eta[i]$, $i = 1, \dots, N_I$, ordered according to their index. The dynamics of the extended estimation error (17) becomes

$$\dot{\epsilon} = (A_E + B_E(F_E - \tilde{F}_E) + K_E D_E)\epsilon - K_E \eta. \tag{39}$$

We have the following results.

Proposition 1. *If the measurement noise is a zero-mean noise, then the estimation error mean converges to zero.*

Proof. The mean of the estimation error can be described by the following dynamic model:

$$\dot{\mathbb{E}}\epsilon = (A_E + B_E(F_E - \tilde{F}_E) + K_E D_E)\mathbb{E}\epsilon - K_E \mathbb{E}\eta,$$

Since, by assumption, we consider a zero-mean measurement noise, then

$$\dot{\mathbb{E}}\epsilon = (A_E + B_E(F_E - \tilde{F}_E) + K_E D_E)\mathbb{E}\epsilon,$$

that converges to zero, thanks to the result in Theorem 1.

Proposition 2. *Given $\bar{\eta}$ the mean of the measurement noise, different from zero, then the estimation error mean converges to a bounded value.*

Proof. The mean of the estimation error can be described in this case by the following equation:

$$\dot{\mathbb{E}}\epsilon = (A_E + B_E(F_E - \tilde{F}_E) + K_E D_E)\mathbb{E}\epsilon - K_E \mathbb{E}\bar{\eta},$$

which converges to a bounded value, since it represents an asymptotically stable linear system with bounded input.

In the case that the measurement noise mean is known, we can trivially consider it in the estimator formulation.

9. Simulation results

9.1. The considered problem

As a case study, we consider a multi-agent system composed of N interconnected mobile robots, partitioned into two groups: a small group of independent robots, and a large group of dependent robots. The objective of the system is to solve a tracking problem: namely, a set of periodic setpoint trajectories is designed, and the independent robots are controlled in such a way that the dependent robots asymptotically track those trajectories.

For this purpose, we consider the following interconnection:

$$\begin{cases} \dot{x}_i = -\sum_{j \in \mathcal{N}_i} w_{ij}(x_i - x_j) & \text{if } i \in \mathcal{V}_D \\ \dot{x}_h = \nu_h & \text{if } h \in \mathcal{V}_I \end{cases} \tag{40}$$

where x_i represents the position of the i-th robot, $w_{ij} > 0$ are the edge weights, and $\mathcal{N}_i$ is the set of the neighbors of the $i-$th robot, that is the set of the robots that are interconnected to the $i-$th one. Moreover, define $\nu_h \in \mathbb{R}^m$ as a control input. It is worth noting that the dependent robots are interconnected with their neighbors with the standard, well known, (weighted) consensus protocol [28]. Without loss of generality, in the following we will consider the scalar case, thus assuming $x_i \in \mathbb{R}$. It is possible to extend all the results to the multi-dimensional case, considering each component separately.

Define $\mathcal{L}(\mathcal{G})$ as the Laplacian matrix associated to the graph $\mathcal{G}$. Therefore, as shown in [40], it is possible to decompose the Laplacian matrix $\mathcal{L}(\mathcal{G})$ as:

$$\mathcal{L}(\mathcal{G}) = -\begin{bmatrix} A & B \\ B^\top & C \end{bmatrix}, \tag{41}$$

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. The dynamics of the dependent robots can be written as in (7), namely

$$\dot{x}_D = Ax_D + Bu$$

We would like to note that this equation represents the standard model of a LTI system: standard linear control techniques, based on the Francis' regulator equations, were exploited in [16] for solving the tracking problem.

In particular, periodic setpoints are defined for each independent 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:

$$\xi\left(t\right) = \left[1 \ \sin\left(\frac{2\pi}{T}t\right) \ \cos\left(\frac{2\pi}{T}t\right) \ \ldots \ \sin\left(n\frac{2\pi}{T}t\right) \ \cos\left(n\frac{2\pi}{T}t\right)\right]^\top . \tag{42}$$

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

$$x_s\left(t\right) = \mathcal{J}\xi\left(t\right) \tag{43}$$

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

It is possible to define a linear exosystem

$$\dot{\xi}\left(t\right) = G\xi\left(t\right) \tag{44}$$

where $G \in \mathbb{R}^{\bar{n} \times \bar{n}}$ is an opportunely defined marginally stable matrix [16], that, initialized with

$$\xi\left(0\right) = \left[1, 0, 1, \ldots, 1\right]^\top$$

yields (42) as a solution.

The input u was then defined as follows:

$$u = Fx_D + (\Gamma - F\Pi)\xi \tag{45}$$

where $\Gamma \in \mathbb{R}^{N_I \times \bar{n}}$ and $\Pi \in \mathbb{R}^{N_D \times \bar{n}}$ are obtained as the solution of the regulator equations that, in this example, can be written as follows:

$$\begin{cases} A\Pi + B\Gamma & = & \Pi G \\ \Pi - \mathcal{J} & = & \mathbb{O}_{N_D,\bar{n}} \end{cases} \tag{46}$$

Matrix $F \in \mathbb{R}^{N_I \times N_D}$ is chosen in such a way 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 [41], 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.

The control strategy (45) requires the full knowledge of the state vector x_D, which is a centralized quantity. In [16], the presence of a communication graph among the independent robots was assumed, and a state observer was then designed, based on information exchange among the independent robots.

According to the considered problem, here we assume that no communication exists among the independent robots. An example of interconnection topology defined with no communication among the independent nodes is depicted in Fig. 1. In the picture, independent nodes are highlighted with red dashed ellipses, while dependent nodes are highlighted with green solid ellipses: black lines represent interconnections among the nodes.

Exploiting the procedure introduced in this paper, it is possible to design a decentralized state estimation system, that does not require communication among the independent nodes. Subsequently, the control law (45) can be implemented in a decentralized manner, as follows:

$$u_i = \alpha_i^\top F \hat{d}_i + (\Gamma - F\Pi)\xi \tag{47}$$

where $\alpha_i^\top$ selects the component related to agent i.

In the simulations we consider 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. The results presented in the previous sections are easily extended to this multi-dimensional case, considering each component separately.

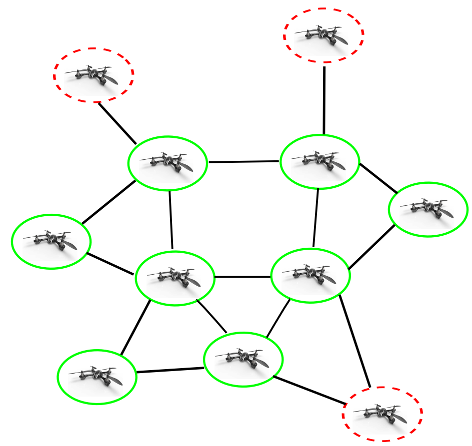


Figure 1: Independent nodes are highlighted with red dashed ellipses, while dependent nodes are highlighted with green solid ellipses. Black lines represent interconnections among the nodes: in the considered scenario, no edge exist among the independent nodes.

9.2. Estimation procedure

Several simulations have been carried out in order to evaluate the performance of the proposed control strategy. The results of some remarkable examples are summarized in Figs. 2, 3 and 4, for different network topologies and different numbers of dependent and independent robots.

The interconnection topology among the robots is depicted in Figs. 2(a), 3(a) and 4(a). In the pictures, Di indicates the i-th dependent robot, and Ij indicates the j-th independent robot. Red and green lines are used to represent dependent-dependent and independent-dependent robots interconnections, respectively.

The three-dimensional setpoint trajectories $x_s\,(t)$ are represented in Figs. 2(b), 3(b) and 4(b): each colored line represents the setpoint for one of the dependent robots. The objective of the control system is then to make each dependent robot track one of the setpoint trajectories. This is obtained by means of the independent robots, that act as the control input for the system.

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:

$$e(t) = x_D(t) - x_s(t) \tag{48}$$

where the setpoint $x_s(t)$ is computed as in (43).

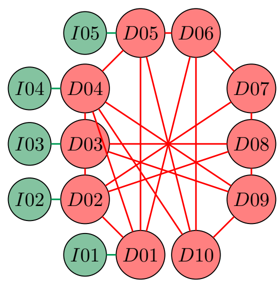
Specifically, Figs. 2(c), 3(c) and 4(c) show the evolution of the estimation error $\epsilon(t)$, while Figs. 2(d), 3(d) and 4(d) show the evolution of the tracking error $e(t)$. Due to space limitations, the plot depicts only the components along the x-axis, and only the estimation error for the first independent agent. Similar results are obtained in the other cases.

As expected, the estimation error asymptotically vanishes, and as a result the dependent robots correctly track the desired setpoint trajectories.

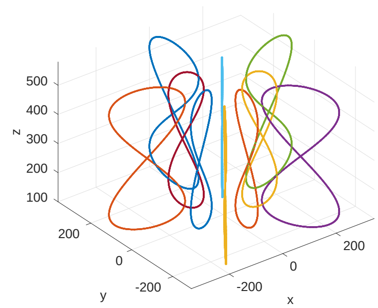
As expected, the estimation error quickly goes to zero and, subsequently, the tracking error goes to zero as well.

9.3. Analysis of the effect of the lack of communication among independent robots

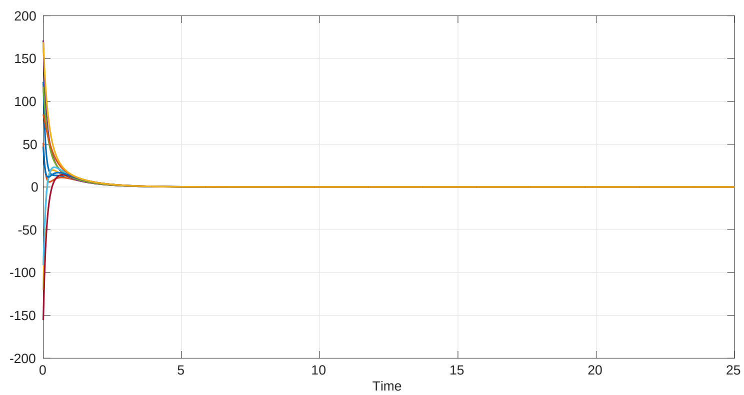
Simulations have been carried out in order to assess the degradation of the performance introduced by the decentralized estimation scheme proposed in this paper due to the fact that no communication exists between independent nodes. For comparison purposes, we considered the estimation scheme originally proposed in [16], in which communication among the independent robots was assumed: in particular, each independent robot had access to the entire output vector, and was then able to implement a standard Luenberger state observer. We consider as a representative example the case described in Fig. 3, considering $N_D = 9$ dependent robots and $N_I = 3$ independent robots. Fig. 5 shows the evolution of the tracking error (along the x-axis) in the two cases: considering communication among the independent robots, and utilizing the decentralized observer introduced in this paper (see Figs. 5(a) and 5(b), respectively). It is worth noting that the performance of the system is very similar, in the two cases: hence, it is possible to conclude that, in practical cases, utilizing the



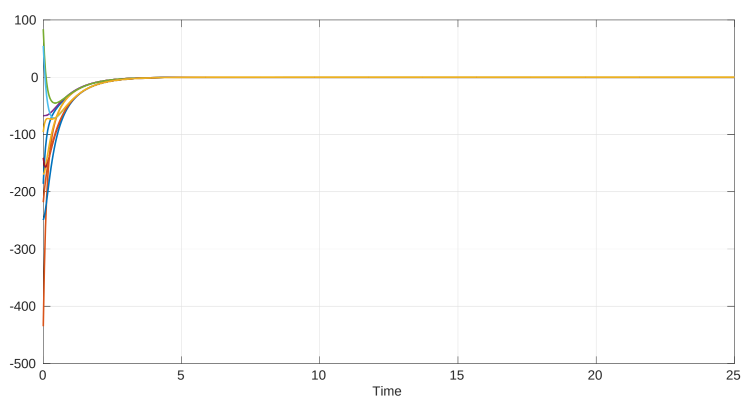
(a) Interconnection topology



(b) Setpoint trajectories $x_s(t)$, defined as in (43)

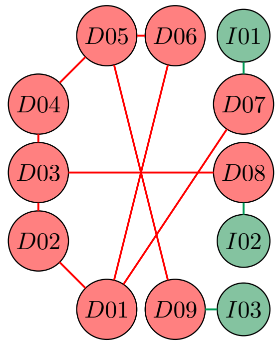


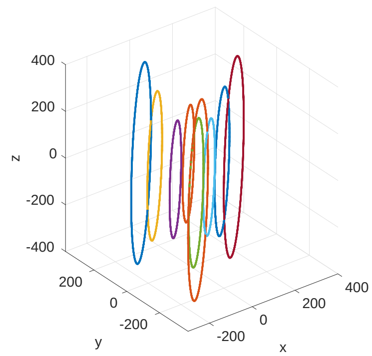
(c) Estimation error $\epsilon(t)$, x component, defined as in (17)



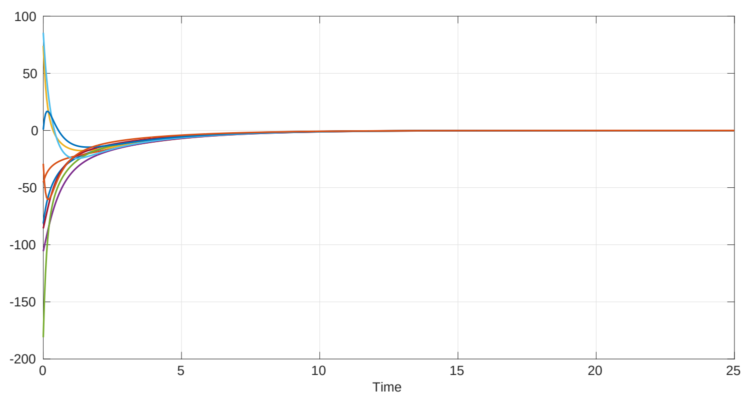
(d) Tracking error $e(t)$, x component, defined as in (48)

Figure 2: Simulation performed with $N_D = 10$ dependent robots, and $N_I = 5$ independent robots

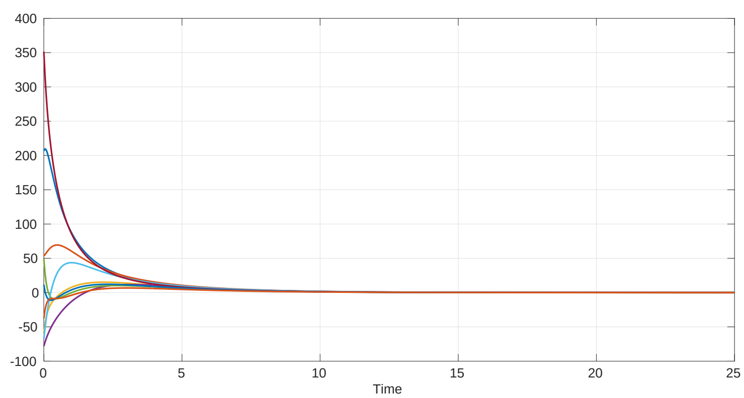




(a) Interconnection topology (b) Setpoint trajectories $x_s(t)$, defined as in (43)

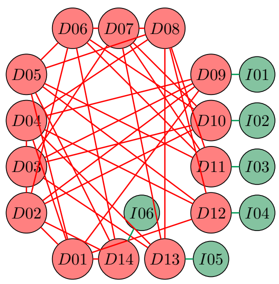


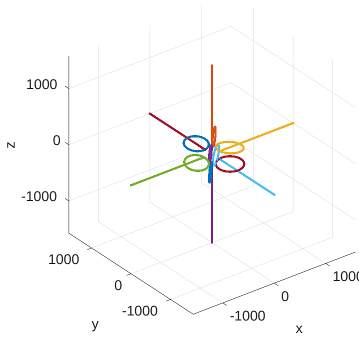
(c) Estimation error $\epsilon(t)$, x component, defined as in (17)



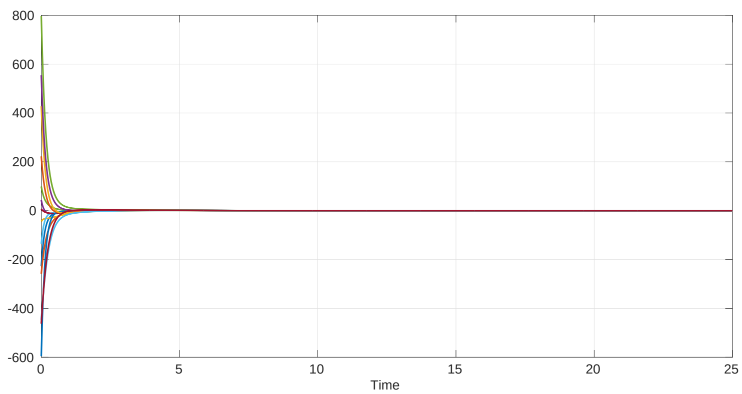
(d) Tracking error $e(t)$, x component, defined as in (48)

Figure 3: Simulation performed with $N_D = 9$ dependent robots, and $N_I = 3$ independent robots

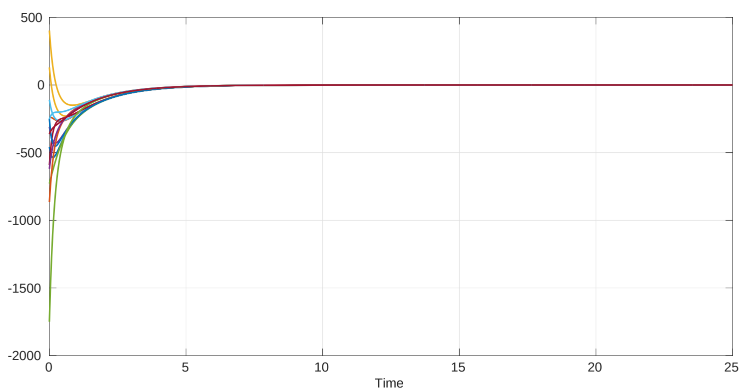




(a) Interconnection topology (b) Setpoint trajectories $x_s(t)$, defined as in (43)



(c) Estimation error $\epsilon(t)$, x component, defined as in (17)



(d) Tracking error $e(t)$, x component, defined as in (48)

Figure 4: Simulation performed with $N_D = 14$ dependent robots, and $N_I = 6$ independent robots

decentralized estimation scheme proposed in this paper has a negligible effect on the system, and does not significantly degrade the performance.

9.4. Measurement noise

The presence of measurement noise has been considered as well. In particular, as described in Section 8, a Gaussian measurement noise has been considered, with mean value equal to $\bar{\eta}$, and standard deviation equal to $\bar{\sigma}$.

Results are reported for one representative example, corresponding to the interconnection topology depicted in Fig. 4(a), including $N_D = 14$ dependent robots, controlled by $N_I = 6$ dependent robots having the objective to implement the setpoint trajectories depicted in Fig. 4(b).

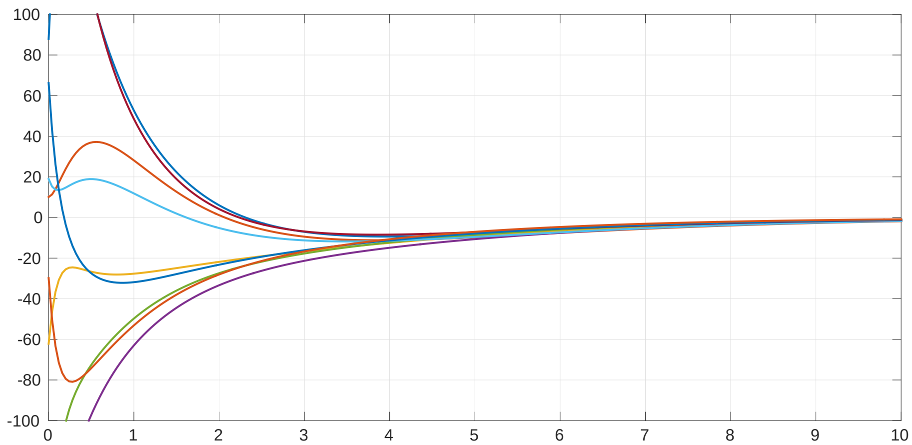
Two different measurement noises were considered: Fig. 6 considers a zero-mean Gaussian noise, while Fig. 7 considers a Gaussian measurement noise with mean value $\bar{\eta} = 10$. In both cases, the standard deviation is $\bar{\sigma} = 1$.

As expected, in the zero-mean case, both the estimation error (depicted in Fig. 6(a)) and the tracking error (depicted in Fig. 6(b)) asymptotically converge to zero, while they converge to a bounded value in the non-zero-mean case (see Figs. 7(a) and 7(b) respectively).

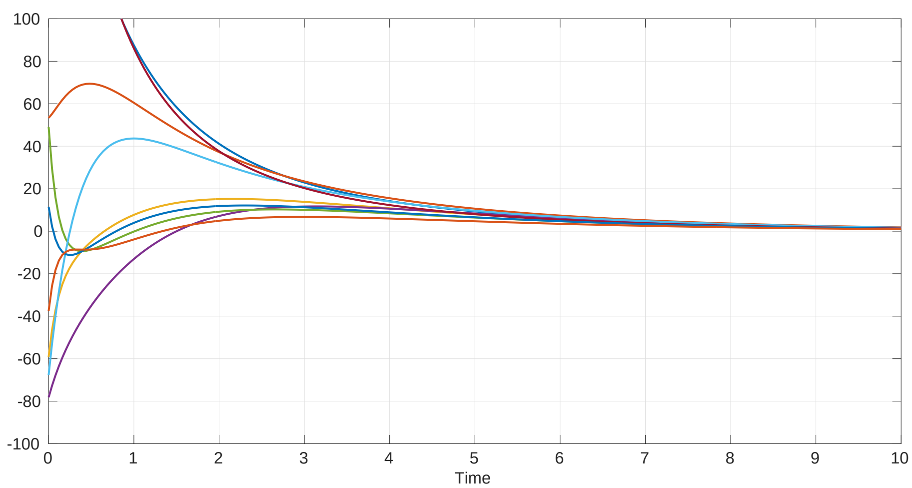
10. Conclusions

This paper proposes a decentralized state estimation method. The purpose is the control of network systems in order to track arbitrary setpoint trajectories. The proposed state estimation algorithm is designed allowing each independent node to estimate the input of the other independent nodes and the state of the dependent nodes, without requiring communication among the independent nodes. Conditions are derived to formally guarantee that estimation error asymptotically converges to zero. The presence of measurement noises has been investigated as well.

The proposed estimation method has been exploited in [45] to implement a decentralized fault diagnosis scheme, allowing to detect and isolate faults in the considered multi-agent systems.

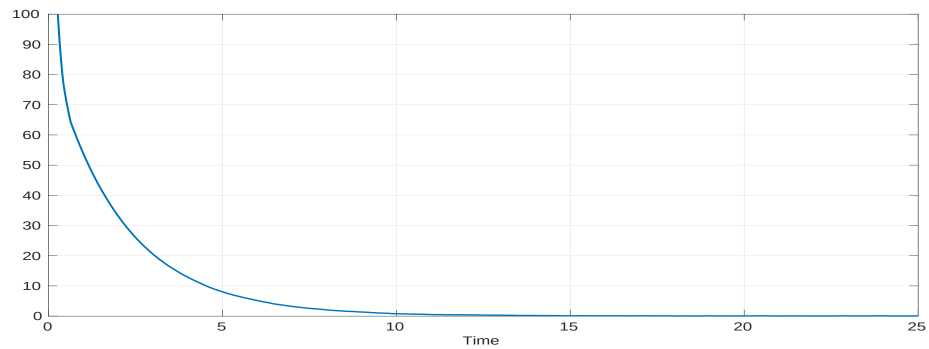


(a) Tracking error $e(t)$, x component, defined as in (48), achieved considering communication among the independent robots.

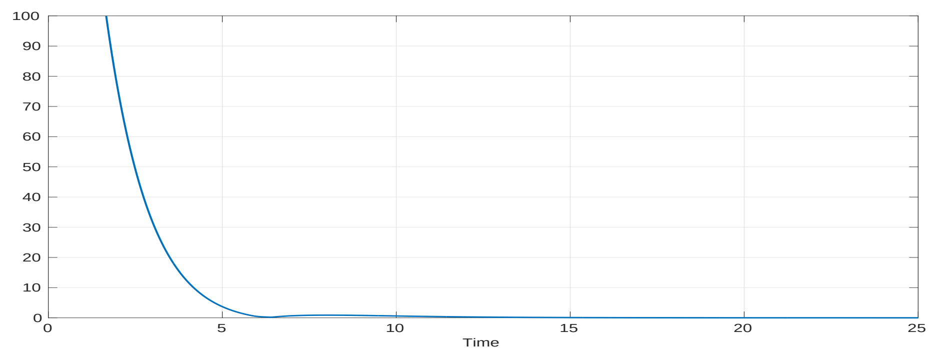


(b) Tracking error $e(t)$, x component, defined as in (48), achieved utilizing the decentralized estimation scheme introduced in this paper

Figure 5: Simulation performed with $N_D = 9$ dependent robots, and $N_I = 3$ independent robots



(a) Estimation error (average absolute value)



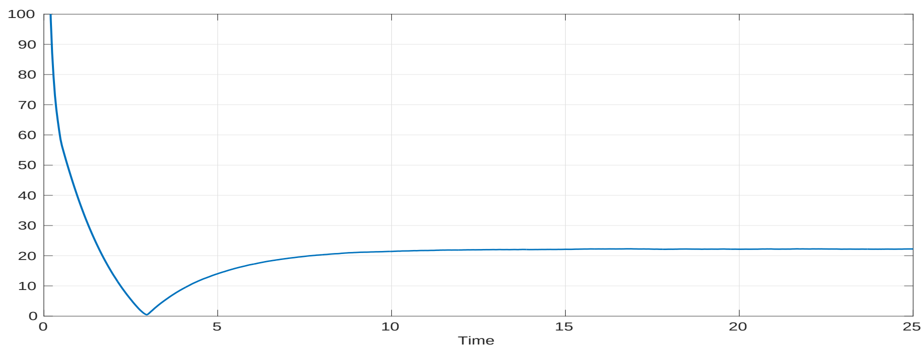
(b) Tracking error (average absolute value)

Figure 6: Simulation performed with $N_D = 14$ dependent robots, and $N_I = 6$ independent robots, with measurement noise (mean $\bar{\eta} = 0$, standard deviation $\bar{\sigma} = 1$)

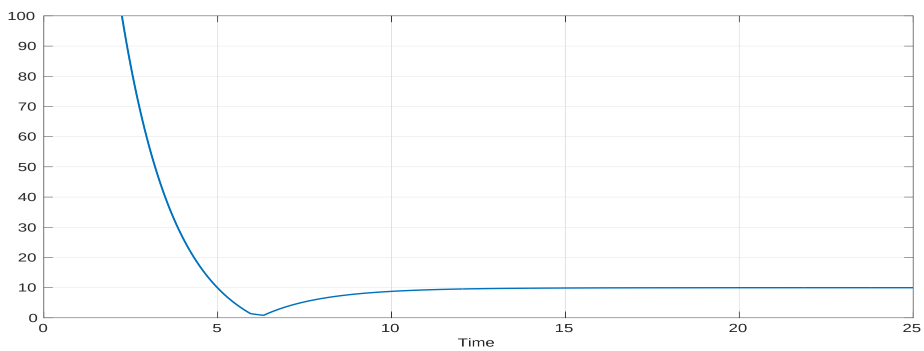
As a future work, we would like to analyze the scenario where the network topology may change over time.

Acknowledgements

The dissemination of this work is being supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 739551 (KIOS CoE). This work was also partially supported by the research project *Stability and Control of Power Networks with Energy Storage* (STABLE-NET) which is funded by the RCUK Energy Programme (contract no: EP/L014343/1).



(a) Estimation error (average absolute value)



(b) Tracking error (average absolute value)

Figure 7: Simulation performed with $N_D = 14$ dependent robots, and $N_I = 6$ independent robots, with measurement noise (mean $\bar{\eta} = 10$, standard deviation $\bar{\sigma} = 1$)

Appendix

In the appendix we show an example that clarifies the structure of matrices F_E and $\tilde{F}_E$.

Consider, as an example, the case where $N_I = 2$ and $N_D = 3$. In this case, we have $F \in \mathbb{R}^{N_I \times N_D}$, defined as follows:

$$F = \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \end{bmatrix} \tag{49}$$

Matrix $F_E = \mathbb{I}_{N_I} \otimes F$ is then defined as follows:

$$F_E = \left[\begin{array}{ccc|ccc} F_{11} & F_{12} & F_{13} & 0 & 0 & 0 \\ F_{21} & F_{22} & F_{23} & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & F_{11} & F_{12} & F_{13} \\ 0 & 0 & 0 & F_{21} & F_{22} & F_{23} \end{array}\right] \tag{50}$$

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

$$\tilde{F}_E = \left[\begin{array}{ccc|ccc} F_{11} & F_{12} & F_{13} & 0 & 0 & 0 \\ 0 & 0 & 0 & F_{21} & F_{22} & F_{23} \\ \hline F_{11} & F_{12} & F_{13} & 0 & 0 & 0 \\ 0 & 0 & 0 & F_{21} & F_{22} & F_{23} \end{array}\right] \tag{51}$$

Let $x_E = \left[x_D^\top \, x_D^\top\right]^\top$, where $x_D = [x_1 \, x_2 \, x_3]^\top$. Then, the following equality holds:

$$F_E x_E = \tilde{F}_E x_E \tag{52}$$

References

[1] F. Boem, L. Sabattini, C. Secchi, Decentralized state estimation for heterogeneous multi-agent systems, in: Proceedings of the IEEE Conference on Decision and Control, 2015, pp. 4121–4126.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, A survey on sensor networks, Communications Magazine, IEEE 40 (8) (2002) 102–114.

[3] L. Sabattini, C. Secchi, C. Fantuzzi, Arbitrarily shaped formations of mobile robots: artificial potential fields and coordinate transformation, Autonomous Robots (Springer) 30 (4) (2011) 385–397. `doi:10.1007/s10514-011-9225-4`.

[4] D.-H. Le, Y.-K. Kwon, A coherent feedforward loop design principle to sustain robustness of biological networks, Bioinformatics 29 (5) (2013) 630–637.

[5] I. Karafyllis, M. Papageorgiou, Global exponential stability for discrete-time networks with applications to traffic networks, IEEE Transactions on Control of Network Systems 2 (1) (2015) 68–77. doi:10.1109/TCNS.2014.2367364.

[6] A. Parisio, E. Rikos, L. Glielmo, A model predictive control approach to microgrid operation optimization, IEEE Transactions on Control Systems Technology 22 (5) (2014) 1813–1827.

[7] V. Srivastava, N. Leonard, Collective decision-making in ideal networks: The speed-accuracy tradeoff, IEEE Transactions on Control of Network Systems 1 (1) (2014) 121–132. doi:10.1109/TCNS.2014.2310271.

[8] P. R. Wurman, R. D'Andrea, M. Mountz, Coordinating hundreds of cooperative, autonomous vehicles in warehouses, AI Magazine 29 (1) (2008) 9–20.

[9] F. Pasqualetti, S. Zampieri, F. Bullo, Controllability metrics, limitations and algorithms for complex networks, IEEE Transactions on Control of Network Systems 1 (1) (2014) 40–52. doi:10.1109/TCNS.2014.2310254.

[10] A. Ganguli, J. Cortes, F. Bullo, Multirobot rendezvous with visibility sensors in nonconvex environments, IEEE Transactions on Robotics 25 (2) (2009) 340 –352.

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

[12] R. Olfati-Saber, J. A. Fax, R. M. Murray, Consensus and cooperation in networked multi–agent systems, Proceedings of the IEEE 95 (1) (2007) 215–233.

[13] A. Adaldo, D. Liuzza, D. V. Dimarogonas, K. H. Johansson, Control of multi-agent systems with event-triggered cloud access, in: European Control Conference, 2015.

[14] F. Priscoli, A. Isidori, L. Marconi, A. Pietrabissa, Leader-following coordination of nonlinear agents under time-varying communication topologies, IEEE Transactions on Control of Network Systems 2 (4) (2015) 393–405. doi:10.1109/TCNS.2015.2426752.

[15] J. Klotz, Z. Kan, J. Shea, E. Pasiliao, W. Dixon, Asymptotic synchronization of a leader-follower network of uncertain euler-lagrange systems, IEEE Transactions on Control of Network Systems 2 (2) (2015) 174–182. doi:10.1109/TCNS.2014.2378875.

[16] L. Sabattini, C. Secchi, M. Cocetti, A. Levratti, C. Fantuzzi, Implementation of coordinated complex dynamic behaviors in multi-robot systems, IEEE Transactions on Robotics 31 (4) (2015) 1018–1032.

[17] M. Cocetti, L. Sabattini, C. Secchi, 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.

[18] L. Sabattini, N. Chopra, C. Secchi, Decentralized connectivity maintenance for cooperative control of mobile robotic systems, The International Journal of Robotics Research (SAGE) (12) (2013) 1411–1423. doi:10.1177/0278364913499085.

[19] L. Sabattini, C. Secchi, N. Chopra, A. Gasparri, Distributed control of multi-robot systems with global connectivity maintenance, IEEE Transactions on Robotics (5) (2013) 1326 – 1332. doi:10.1109/TRO.2013.2267971.

[20] P. Robuffo Giordano, A. Franchi, C. Secchi, H. H. Bülthoff, A passivity-based decentralized strategy for generalized connectivity maintenance, The International Journal of Robotics Research 32 (3) (2013) 299–323.

[21] A. Ajorlou, A. Momeni, A. G. Aghdam, A class of bounded distributed con-

trol strategies for connectivity preservation in multi–agent systems, IEEE Transactions on Automatic Control 55 (2010) 2828–2833.

[22] M. M. Zavlanos, M. B. Egerstedt, G. J. Pappas, Graph-theoretic connectivity control of mobile robot networks, Proceedings of the IEEE 99 (2011) 1525–1540.

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

[24] P. Yang, R. A. Freeman, K. M. Lynch, Multi-agent coordination by decentralized estimation and control, IEEE Transactions on Automatic Control 53 (11) (2008) 2480–2496.

[25] D. Viegas, P. Batista, P. Oliveira, C. Silvestre, C. L. P. Chen, Distributed state estimation for linear multi-agent systems with time-varying measurement topology, Automatica 54 (2015) 72–79.

[26] A. Speranzon, C. Fischione, K. Johansson, A. Sangiovanni-Vincentelli, A distributed minimum variance estimator for sensor networks, IEEE Journal on Selected Areas in Communications 26 (4) (2008) 609–621.

[27] F. S. Cattivelli, A. H. Sayed, Diffusion strategies for distributed kalman filtering and smoothing, IEEE Transactions on Automatic Control 55 (9) (2010) 2069–2084.

[28] R. Olfati-Saber, J. A. Fax, R. M. Murray, Consensus and cooperation in networked multi-agent systems, Proceedings of the IEEE 95 (1) (2007) 215–233.

[29] R. Olfati-Saber, Distributed kalman filtering for sensor networks, in: 46th IEEE Conf. on Decision and Control, 2007, pp. 5492–5498. doi:10.1109/CDC.2007.4434303.

[30] R. Carli, A. Chiuso, L. Schenato, S. Zampieri, Distributed kalman filtering based on consensus strategies, IEEE Journal on Selected Areas in Communications 26 (4) (2008) 622–633.

[31] R. Olfati-Saber, Kalman-consensus filter : Optimality, stability, and performance, in: 48th IEEE Conf. on Decision and Control, held jointly with 28th Chinese Control Conference, 2009, pp. 7036–7042. doi:10.1109/CDC.2009.5399678.

[32] M. S. Mahmoud, H. M. Khalid, Distributed kalman filtering: a bibliographic review, Control Theory & Applications, IET 7 (4).

[33] E. Franco, R. Olfati-Saber, T. Parisini, M. Polycarpou, Distributed fault diagnosis using sensor networks and consensus-based filters, in: 45th IEEE Conf. on Decision and Control, 2006, pp. 386 –391. doi:10.1109/CDC.2006.376797.

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

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

[36] F. Arrichiello, A. Marino, F. Pierri, Observer-based decentralized fault detection and isolation strategy for networked multirobot systems, IEEE Transactions on Control Systems Technology 23 (4) (2015) 1465–1476. doi:10.1109/TCST.2014.2377175.

[37] S. Battilotti, C. Califano, Semiglobal leader-following consensus for generalized homogenous agents, in: European Control Conference, 2015, pp. 1011–1016. doi:10.1109/ECC.2015.7330673.

[38] L. Carrillo, W. Russell, J. Hespanha, G. Collins, State estimation of multiagent systems under impulsive noise and disturbances, IEEE

Transactions on Control Systems Technology 23 (1) (2015) 13–26. doi:10.1109/TCST.2014.2309911.

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

[40] M. Egerstedt, S. Martini, M. Cao, K. Camlibel, A. Bicchi, Interacting with networks: How does structure relate to controllability in single-leader, consensus networks?, Control Systems, IEEE 32 (4) (2012) 66–73.

[41] L. Sabattini, C. Secchi, C. Fantuzzi, Controllability and observability preservation for networked systems with time varying topologies, in: Proceedings of the IFAC World Congress, 2014, pp. 1837 − 1842.

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

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

[44] M. Heymann, W. Wonham, Comments ”on pole assignment in multi-input controllable linear systems”, IEEE Transactions on Automatic Control 13 (6) (1968) 748–749.

[45] F. Boem, L. Sabattini, C. Secchi, Decentralized fault diagnosis for heterogeneous multi-agent systems, in: Proceedings of the International Conference on Control and Faul-Tolerant Systems (SysTol), Barcelona, Spain, 2016.