A decentralized controller-observer scheme for multi-robot weighted centroid tracking

Gianluca Antonelli, Filippo Arrichiello, Fabrizio Caccavale, Alessandro Marino*

Abstract— **In this paper a decentralized controller-observer scheme for centroid tracking with a multi-robot system is presented. The key idea is to develop, for each robot, an observer of the collective system's state; each local observer is updated by only using information of the state of the robot and of its neighbors. The local observers' estimations are then used by the individual robots to cooperatively track an assigned time-varying reference for the weighted centroid. Convergence of the scheme is proven for both fixed and switching communication topologies, as well as for directed and undirected communication graphs. Numerical simulations relative to different case studies are illustrated to validate the approach.**

I. Introduction and literature review

Multi-robot systems have been deeply investigated in the last decades, due to their advantages with respect to single robots in terms of, e.g., flexibility, redundancy and fault tolerance. Furthermore, multi-robot systems can take the advantages of distributed systems, i.e., instead of using a single and powerful robot, a team of robots with lower performance can be used to achieve the assigned mission. Autonomous robots can be spread into the environment to increase the coverage range of sensors, actuators and communication devices, such that the overall team can better accomplish the assigned mission in terms of time and efficiency.

However, in the presence of limited communication/sensing capabilities, the common goal has to be achieved in a cooperative way by using only local information. In fact, each robot can only rely on local information, coming from its on-board sensors or received from its direct neighbors, while the goal of the overall team usually depends on the global state of the system. Several recent studies dealt with the development of distributed control approaches for multi-robot systems with the aim of achieving a global task (e.g., controlling the swarm centroid) by using distributed controllers. In such cases, the possibility for the robots to explicitly exchange information with their neighbors strongly influences the control strategy. Research on these aspects posed the challenging problem of *consensus* for multi-agent systems, i.e., reaching an agreement regarding a certain variable depending on the state of all the agents. A wide overview on such problems can be found in recent books [13], [9], while in reference papers [14] and [10] the consensus algorithms are investigated with emphasis on robustness, time-delays and performance guarantee. The work in [8] shows how the consensus can be used to achieve specific behaviors, e.g., formation keeping and rendez-vous, of the multi-agent system. The results in [7] are related to the stability analysis of several decentralized strategies that achieve an emergent behavior. In [1] non-linear protocols are proposed to solve non-linear stationary consensus problem for networks of dynamic agents with fixed topologies. The results in [1] have been further extended in [3] for a more general class of consensus functions.

The above cited papers mainly focus on stationary consensus problems, where the consensus must be reached on a quantity function of the initial states of the agents. On the other hand, in distributed robotics, the mission of a multi-robot system is usually expressed as a time-varying goal (or target) function, e.g., describing the location and shape of a robotic team (formation statistics). The first convincing attempt to design *local* control laws aimed at achieving a target *collective* behavior of a multi-robot team, expressed in terms of formation statistics, can be found in [6] and [18]; noticeably, the approach uses a distributed estimator of the actual collective behavior, which is based on the dynamic average consensus protocol proposed in [17]. However, for these approaches, asymptotic tracking is not guaranteed, unless the goal is constant or has poles in left half plane [17].

Decentralized estimation and control are also investigated in [16] in the framework of linear state feedback control. Moreover, it is worth remarking the work on spatially distributed gradients of collective objective functions [2], [4]. The problem tracking a time-varying reference state for each agent has been investigated in [12], [14].

In this paper, the problem of tracking a certain class of global functions (i.e., the weighted centroid) expressing a time-varying common goal to be achieved cooperatively by a team of mobile robots is addressed. Namely, each robot estimates the global state of the system via a properly designed observer. Then, the estimated state is used by a local controller in charge of achieving asymptotic tracking of a given time-varying reference for the weighted centroid of the team. Convergence of both estimation and tracking errors is proven. It is worth remarking that, as in [18], tracking is achieved by using distributed estimation and control, although here, instead of the common goal function, the whole collective state is estimated by each robot in the team. Although the class of goal functions considered here is limited to the generalized (weighted) centroid, tracking

G. Antonelli, F. Arrichiello, A. Marino are with the Department DAEIMI of the University of Cassino, Via G.Di Biasio 43, 03043 Cassino (FR), Italy {antonelli, f.arrichiello, al.marino}@unicas.it

F. Caccavale is with the Department DIFA of the University of Basilicata, Viale dell'Ateneo Lucano 10, 85100 Potenza, Italy fabrizio.caccavale@unibas.it

*Authors are in alphabetic order.

of time-varying reference goals is guaranteed. Moreover, the estimation of the whole system's state provided by the distributed observer might be used to achieve, e.g., more complex missions or fault diagnosis.

The rest of the paper is organized as follows: Section II presents the system modeling and recalls some basic concepts related to graph theory; Sections III and IV respectively present the problem statement and the proposed solution, while in Section V a stability proof of the controller-observer scheme is presented in the case of undirected graphs with fixed topology; in Section VI, the extension to directed and switching topologies is presented; finally, numerical simulations and some concluding remarks are provided in Sections VII and VIII.

II. Modeling

Consider a system composed by N mobile robots, where the ith agent's state is denoted by $\boldsymbol{x}_i \in \mathbb{R}^n$. It is assumed that each robot is characterized by a single-integrator dynamics

$$\dot{\boldsymbol{x}}_i = \boldsymbol{u}_i \,,$$

where $\boldsymbol{u}_i \in \mathbb{R}^n$ is the input vector. The collective state is given by $\boldsymbol{x} = \begin{bmatrix}\boldsymbol{x}_1^{\mathrm{T}} & \ldots & \boldsymbol{x}_N^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{Nn}$ and the collective dynamics is then expressed as

$$\dot{\boldsymbol{x}} = \boldsymbol{u} \,, \tag{1}$$

where $\boldsymbol{u} = \begin{bmatrix}\boldsymbol{u}_1^{\mathrm{T}} & \ldots & \boldsymbol{u}_N^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{Nn}$ is the collective input vector.

Information exchange between the robots can be modeled as a network of agents described by a graph $\mathcal{G}(\mathcal{E}, \mathcal{V}, \boldsymbol{A})$ characterized by its topology [5], [11], [15], i.e., the set $\mathcal{V}$ of the indexes labeling the N vertices (nodes), a set of edges (arcs) $\mathcal{E} = \mathcal{V} \times \mathcal{V}$, and the $(N \times N)$ adjacency matrix $\boldsymbol{A} = \{a_{ij}\}$, defined as the matrix having $a_{ii} = 0$, $a_{ij} = 1$ if $(j, i) \in \mathcal{E}$, and $a_{ij} = 0$ otherwise. If all the communication links between the robots are bi-directional, the graph is called *undirected* (i.e., $(i, j) \in \mathcal{E} \Rightarrow (j, i) \in \mathcal{E}$), otherwise, the graph is called *directed*. The latter case is considered when, for example, an agent measures the state of other agents (its neighbors) but not all its neighbors can measure its state. Moreover, the graph topology can be assumed either fixed or switching (e.g., communication links may appear or disappear). A directed graph is called *strongly connected* if any two distinct nodes of the graph can be connected via a directed path, i.e., a path that follows the direction of the edges of the graph. An undirected graph is called *connected* if there is an undirected path between every pair of distinct nodes. A node of a directed graph is balanced if its in-degree (i.e., the number of incoming edges) and its out-degree (i.e., the number of outgoing edges) are equal; a directed graph is called *balanced* if each node of the graph is balanced. Any undirected graph is balanced.

It is assumed that the ith agent communicates only with the set of its neighbors $\mathcal{N}_i = \{j \in \mathcal{V} : (j, i) \in \mathcal{E}\}$, and it does not know the topology of the overall communication graph. The $(N \times N)$ Laplacian matrix, $\boldsymbol{L} = \{l_{ij}\}$, is the matrix defined as $l_{ii} = \sum_{j=1, j\neq i}^{N} a_{ij}$, $l_{ij} = -a_{ij}$. The Laplacian always has a zero eigenvalue with corresponding right eigenvector the $N \times 1$ vector of all ones, $\mathbf{1}_N$. Hence, $\mathrm{rank}(\boldsymbol{L}) \leq N - 1$ and $\boldsymbol{L}\mathbf{1}_N = \mathbf{0}_N$, where $\mathbf{0}_N$ is the $(N \times 1)$ null vector. For a balanced directed graph (and thus, for an undirected graph), $\mathbf{1}_N$ is a left eigenvector of $\boldsymbol{L}$, i.e. $\mathbf{1}_N^T\boldsymbol{L} = \mathbf{0}_N^T$. If the graph is strongly connected $\mathrm{rank}(\boldsymbol{L}) = N - 1$. If the graph is undirected, the Laplacian is symmetric and positive semidefinite; moreover, if the graph is connected, 0 is a simple eigenvalue of $\boldsymbol{L}$.

Let ${}^i\hat{\boldsymbol{x}} \in \mathbb{R}^{Nn}$ denote the estimate of the collective system's state computed by agent i. All the state estimates can be stacked into the collective estimate vector, $\hat{\boldsymbol{x}} \in \mathbb{R}^{N^2 n}$, for which a corresponding error can be defined as

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} {}^1\tilde{\boldsymbol{x}} \\ {}^2\tilde{\boldsymbol{x}} \\ \vdots \\ {}^N\tilde{\boldsymbol{x}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x} - {}^1\hat{\boldsymbol{x}} \\ \boldsymbol{x} - {}^2\hat{\boldsymbol{x}} \\ \vdots \\ \boldsymbol{x} - {}^N\hat{\boldsymbol{x}} \end{bmatrix} . \tag{2}$$

Notice that the collective estimation error can be rewritten as

$$\tilde{\boldsymbol{x}} = (\mathbf{1}_N \otimes \boldsymbol{I}_{Nn})\, \boldsymbol{x} - \hat{\boldsymbol{x}} \,, \tag{3}$$

where $\otimes$ denotes the Kronecker product, and $\boldsymbol{I}_{Nn}$ is the $(Nn \times Nn)$ identity matrix.

The $(Nn \times Nn)$ selection matrix, $\boldsymbol{\Pi}_i$, is defined as follow

$$\boldsymbol{\Pi}_i \quad = \quad \mathrm{diag}\begin{Bmatrix}\boldsymbol{O}_n & \cdots & \underbrace{\boldsymbol{I}_n}_{i \text{ node}} & \cdots & \boldsymbol{O}_n\end{Bmatrix}$$

where $\boldsymbol{O}_n$ denotes the $n \times n$ null matrix. Matrix $\boldsymbol{\Pi}_i$ satisfies $\sum_{i=1}^{N} \boldsymbol{\Pi}_i = \boldsymbol{I}_{Nn}$. Finally, the $(N^2 n \times N^2 n)$ matrix $\boldsymbol{\Pi}$ is defined as

$$\boldsymbol{\Pi} = \mathrm{diag}\begin{Bmatrix}\boldsymbol{\Pi}_1 & \ldots & \boldsymbol{\Pi}_N\end{Bmatrix} . \tag{4}$$

III. Problem statement

It is assumed that the task for the system of agents is encoded by the smooth function $\boldsymbol{\sigma} \in \mathbb{R}^n$ (weighted centroid)

$$\boldsymbol{\sigma}(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i \boldsymbol{x}_i \; = \left(\boldsymbol{\alpha}^{\mathrm{T}} \otimes \boldsymbol{I}_n\right) \boldsymbol{x} \,, \tag{5}$$

where $\boldsymbol{\alpha}^{\mathrm{T}} = \begin{bmatrix}\alpha_1 & \ldots & \alpha_N\end{bmatrix} \in \mathbb{R}^N$. The above task function reduces to the centroid of the swarm when $\alpha_1 = \ldots = \alpha_N = 1/N$.

The main design goals are:

- to design, for each agent, a state observer providing an estimate, ${}^i\hat{\boldsymbol{x}} \in \mathbb{R}^{Nn}$, asymptotically convergent to the collective state, $\boldsymbol{x}$, as $t \to \infty$;
- to design, for each agent, a feedback control law,

$$\boldsymbol{u}_i = \boldsymbol{u}_i(\boldsymbol{x}_i, {}^i\hat{\boldsymbol{x}}, \mathcal{N}_i) \,,$$

  such that $\boldsymbol{\sigma}(\boldsymbol{x})$ asymptotically converges to a given (in general time-varying) reference, $\boldsymbol{\sigma}_d(t)$, as $t \to \infty$.

Both the observer and the controller for each robot can only use *local* information, i.e., its state and input, and the states of the robots belonging to its set of neighbors, $\mathcal{N}_i$.

It is assumed that each agent knows in advance the goal, encoded by the function $\boldsymbol{\sigma}_d(t)$, and its first derivative.

IV. Proposed approach

The following control law is considered for the ith agent

$$\boldsymbol{u}_i = \boldsymbol{u}_i({}^i\hat{\boldsymbol{x}}) = \frac{\alpha_i}{\|\boldsymbol{\alpha}\|^2}\left(\dot{\boldsymbol{\sigma}}_d + k_c\left(\boldsymbol{\sigma}_d - \boldsymbol{\sigma}({}^i\hat{\boldsymbol{x}})\right)\right), \qquad (6)$$

where $k_c > 0$ is a gain to be properly selected.
The observer takes the form ($i = 1, \ldots, N$)

$${}^i\dot{\hat{\boldsymbol{x}}} = k_o\left(\sum_{j\in\mathcal{N}_i}\left({}^j\hat{\boldsymbol{x}} - {}^i\hat{\boldsymbol{x}}\right) + \boldsymbol{\Pi}_i\left(\boldsymbol{x} - {}^i\hat{\boldsymbol{x}}\right)\right) + {}^i\hat{\boldsymbol{u}}, \qquad (7)$$

where $k_o > 0$ is a gain to be properly selected and

$${}^i\hat{\boldsymbol{u}} = \begin{bmatrix} \boldsymbol{u}_1({}^i\hat{\boldsymbol{x}}) \\ \boldsymbol{u}_2({}^i\hat{\boldsymbol{x}}) \\ \vdots \\ \boldsymbol{u}_N({}^i\hat{\boldsymbol{x}}) \end{bmatrix}, \qquad (8)$$

with

$$\boldsymbol{u}_j({}^i\hat{\boldsymbol{x}}) = \frac{\alpha_j}{\|\boldsymbol{\alpha}\|^2}\left(\dot{\boldsymbol{\sigma}}_d + k_c\left(\boldsymbol{\sigma}_d - \boldsymbol{\sigma}({}^i\hat{\boldsymbol{x}})\right)\right), \qquad (9)$$

and it represents an estimate, computed by the ith agent, of the collective input. Notice that the observer (7) can be implemented by using only local information, since $\boldsymbol{\Pi}_i$ selects only the ith component of the estimation error ${}^i\tilde{\boldsymbol{x}}$.
The collective estimation dynamics is given by

$$\dot{\hat{\boldsymbol{x}}} = -k_o\left(\boldsymbol{L}\otimes\boldsymbol{I}_{Nn}\right)\hat{\boldsymbol{x}} + k_o\boldsymbol{\Pi}\tilde{\boldsymbol{x}} + \hat{\boldsymbol{u}}, \qquad (10)$$

where $\hat{\boldsymbol{u}} = \begin{bmatrix} {}^1\hat{\boldsymbol{u}}^{\mathrm{T}} & {}^2\hat{\boldsymbol{u}}^{\mathrm{T}} & \cdots & {}^N\hat{\boldsymbol{u}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{N^2n}$. By taking into account eq. (3), the equality $(\boldsymbol{L}\otimes\boldsymbol{I}_{Nn})(\boldsymbol{1}_N\otimes\boldsymbol{I}_{Nn}) = \boldsymbol{L}\boldsymbol{1}_N\otimes\boldsymbol{I}_{Nn}$ and the property of the Laplacian $\boldsymbol{L}\boldsymbol{1}_N = \boldsymbol{0}_N$, the estimation error dynamics can be derived

$$\dot{\tilde{\boldsymbol{x}}} = -k_o\left(\boldsymbol{L}\otimes\boldsymbol{I}_{Nn} + \boldsymbol{\Pi}\right)\tilde{\boldsymbol{x}} + \left(\boldsymbol{1}_N\otimes\boldsymbol{I}_{Nn}\right)\boldsymbol{u} - \hat{\boldsymbol{u}}. \qquad (11)$$

The task tracking error is represented by the n-dimensional vector $\tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}_d - \boldsymbol{\sigma} \in \mathbb{R}^n$, whose dynamics is given by

$$\begin{aligned}
\dot{\tilde{\boldsymbol{\sigma}}} &= \dot{\boldsymbol{\sigma}}_d - \dot{\boldsymbol{\sigma}} = \dot{\boldsymbol{\sigma}}_d - \sum_{i=1}^N \alpha_i\dot{\boldsymbol{x}}_i \\
&= \dot{\boldsymbol{\sigma}}_d - \sum_{i=1}^N \frac{\alpha_i^2}{\|\boldsymbol{\alpha}\|^2}\left(\dot{\boldsymbol{\sigma}}_d + k_c\left(\boldsymbol{\sigma}_d - \boldsymbol{\sigma}({}^i\hat{\boldsymbol{x}})\right)\right) \\
&= -\sum_{i=1}^N \frac{\alpha_i^2}{\|\boldsymbol{\alpha}\|^2}\left(k_c\left(\boldsymbol{\sigma}_d - \boldsymbol{\sigma}({}^i\hat{\boldsymbol{x}}) \pm \boldsymbol{\sigma}(\boldsymbol{x})\right)\right) \\
&= -k_c\tilde{\boldsymbol{\sigma}} - \frac{k_c}{\|\boldsymbol{\alpha}\|^2}\sum_{i=1}^N \alpha_i^2\left(\boldsymbol{\sigma}(\boldsymbol{x}) - \boldsymbol{\sigma}({}^i\hat{\boldsymbol{x}})\right) \\
&= -k_c\tilde{\boldsymbol{\sigma}} - \frac{k_c}{\|\boldsymbol{\alpha}\|^2}\sum_{i=1}^N \alpha_i^2\left(\boldsymbol{\alpha}^{\mathrm{T}}\otimes\boldsymbol{I}_n\right)\left(\boldsymbol{x} - {}^i\hat{\boldsymbol{x}}\right) \\
&= -k_c\tilde{\boldsymbol{\sigma}} - \frac{k_c}{\|\boldsymbol{\alpha}\|^2}\sum_{i=1}^N \alpha_i^2\left(\boldsymbol{\alpha}^{\mathrm{T}}\otimes\boldsymbol{I}_n\right){}^i\tilde{\boldsymbol{x}}.
\end{aligned} \qquad (12)$$

V. Stability proof

Convergence of the overall controller-observer scheme is carried out in the case of a undirected graph with connected and fixed topology. The extension to directed and/or switching topologies will be provided in the next Section.
The overall closed-loop system can be analyzed by resorting to the positive definite and radially unbounded candidate Lyapunov function

$$V(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{\sigma}}) = V_o + V_c = \frac{1}{2}\tilde{\boldsymbol{x}}^{\mathrm{T}}\tilde{\boldsymbol{x}} + \frac{1}{2}\tilde{\boldsymbol{\sigma}}^{\mathrm{T}}\tilde{\boldsymbol{\sigma}}, \qquad (13)$$

where V_o and V_c are the first and second term of the right hand side, respectively.
The time derivative of V_o along the system's trajectories is given by

$$\begin{aligned}
\dot{V}_o \;=\; & -k_o\tilde{\boldsymbol{x}}^{\mathrm{T}}\left(\boldsymbol{L}\otimes\boldsymbol{I}_{Nn} + \boldsymbol{\Pi}\right)\tilde{\boldsymbol{x}} + \\
& \tilde{\boldsymbol{x}}^{\mathrm{T}}\left(\left(\boldsymbol{1}_N\otimes\boldsymbol{I}_{Nn}\right)\boldsymbol{u} - \hat{\boldsymbol{u}}\right).
\end{aligned} \qquad (14)$$

The matrix $\boldsymbol{L}\otimes\boldsymbol{I}_{Nn}$ is symmetric and positive semidefinite, since the communication graph is undirected and connected. In fact, in such a case, $\boldsymbol{L}$ admits $n-1$ positive eigenvalues and one simple zero eigenvalue; thus, $\boldsymbol{L}\otimes\boldsymbol{I}_{Nn}$ has $Nn(N-1)$ positive eigenvalues and Nn zero regular eigenvalues. Moreover, $\boldsymbol{\Pi}$ is a diagonal matrix with Nn non-null (unitary) elements along the main diagonal; thus, it is symmetric and positive semidefinite as well, since it admits Nn eigenvalues equal to 1 and $N^2n - Nn$ zero eigenvalues. Hence, the sum of the two matrices is positive semidefinite as well. Indeed, the sum $\boldsymbol{L}\otimes\boldsymbol{I}_{Nn} + \boldsymbol{\Pi}$ is positive definite, since the intersection of their null spaces is given by the null vector (see Appendix A). Hence, $\dot{V}_o$ can be upper bounded as follows

$$\dot{V}_o \leq -\lambda_o\|\tilde{\boldsymbol{x}}\|^2 + \sum_{i=1}^N {}^i\tilde{\boldsymbol{x}}^{\mathrm{T}}(\boldsymbol{u} - {}^i\hat{\boldsymbol{u}}), \qquad (15)$$

where $\lambda_o = k_o\lambda_m$ and λ_m is the smallest eigenvalue of $(\boldsymbol{L}\otimes\boldsymbol{I}_{Nn} + \boldsymbol{\Pi})$. It is worth noticing that λ_o is function of the Laplacian (i.e., depends on the network topology) and of the gain k_o; thus, for a given network topology, it can be arbitrarily tuned by choosing k_o. In view of (6) and (8), inequality (15) yields

$$\begin{aligned}
\dot{V}_o &\leq -\lambda_o\|\tilde{\boldsymbol{x}}\|^2 + \sum_{i=1}^N\sum_{j=1}^N {}^i\tilde{\boldsymbol{x}}_j^{\mathrm{T}}\left(\boldsymbol{u}_j({}^j\hat{\boldsymbol{x}}) - \boldsymbol{u}_j({}^i\hat{\boldsymbol{x}})\right) \\
&= -\lambda_o\|\tilde{\boldsymbol{x}}\|^2 + \sum_{i=1}^N\sum_{j=1}^N {}^i\tilde{\boldsymbol{x}}_j^{\mathrm{T}}\frac{k_c\alpha_j}{\|\boldsymbol{\alpha}\|^2}\left(\boldsymbol{\alpha}^{\mathrm{T}}\otimes\boldsymbol{I}_n\right)\left({}^i\hat{\boldsymbol{x}} - {}^j\hat{\boldsymbol{x}}\right) \\
&\leq -\lambda_o\|\tilde{\boldsymbol{x}}\|^2 + \frac{k_c\left\|\boldsymbol{\alpha}^{\mathrm{T}}\otimes\boldsymbol{I}_n\right\|}{\|\boldsymbol{\alpha}\|^2}\sum_{i=1}^N\left\|{}^i\tilde{\boldsymbol{x}}\right\|\sum_{j=1}^N|\alpha_j|\left\|{}^i\tilde{\boldsymbol{x}} - {}^j\tilde{\boldsymbol{x}}\right\| \\
&\leq -\lambda_o\|\tilde{\boldsymbol{x}}\|^2 + \frac{k_c\sqrt{n}}{\|\boldsymbol{\alpha}\|}\sum_{i=1}^N\left\|{}^i\tilde{\boldsymbol{x}}\right\|\sum_{j=1}^N|\alpha_j|\left(\left\|{}^i\tilde{\boldsymbol{x}}\right\| + \left\|{}^j\tilde{\boldsymbol{x}}\right\|\right) \\
&\leq -\lambda_o\|\tilde{\boldsymbol{x}}\|^2 + k_c\sqrt{n}\sum_{i=1}^N\left\|{}^i\tilde{\boldsymbol{x}}\right\|\sum_{j=1}^N\left\|{}^i\tilde{\boldsymbol{x}}\right\| + \left\|{}^j\tilde{\boldsymbol{x}}\right\|
\end{aligned}$$

$$\leq -\lambda_o \|\tilde{\boldsymbol{x}}\|^2 + N k_c \sqrt{n} \|\tilde{\boldsymbol{x}}\|^2 + \frac{k_c\sqrt{n}}{2} \sum_{i=1}^{N}\sum_{j=1}^{N}\left(\|{}^i\tilde{\boldsymbol{x}}\|^2 + \|{}^j\tilde{\boldsymbol{x}}\|^2\right)$$
$$= -\left(\lambda_o - 2\rho_o\right)\|\tilde{\boldsymbol{x}}\|^2 , \quad (16)$$

where $\rho_o = N k_c \sqrt{n}$ and the usual Euclidean norm has been used for vectors, while the Frobenius norm has been used for matrices.

The time derivative of V_c along the system's trajectories is given by

$$\begin{aligned}
\dot{V}_c &= \tilde{\boldsymbol{\sigma}}^{\mathrm{T}}\dot{\tilde{\boldsymbol{\sigma}}} \\
&= \tilde{\boldsymbol{\sigma}}^{\mathrm{T}}\left(-k_c\tilde{\boldsymbol{\sigma}} - \frac{k_c}{\|\boldsymbol{\alpha}\|^2}\sum_{i=1}^{N}\alpha_i^2\left(\boldsymbol{\alpha}^{\mathrm{T}}\otimes\boldsymbol{I}_n\right){}^i\tilde{\boldsymbol{x}}\right) \\
&\leq -k_c\|\tilde{\boldsymbol{\sigma}}\|^2 + \frac{k_c\sqrt{n}}{\|\boldsymbol{\alpha}\|}\|\tilde{\boldsymbol{\sigma}}\|\sum_{i=1}^{N}\alpha_i^2\|{}^i\tilde{\boldsymbol{x}}\| \\
&\leq -k_c\|\tilde{\boldsymbol{\sigma}}\|^2 + 2\rho_c\|\tilde{\boldsymbol{\sigma}}\|\|\tilde{\boldsymbol{x}}\| , \qquad (17)
\end{aligned}$$

where $\rho_c = N k_c \sqrt{n}\|\boldsymbol{\alpha}\|/2$.

Hence, the overall time derivative of the candidate Lyapunov function (13) can be upper bounded as follows

$$\dot{V} \leq -\begin{bmatrix}\|\tilde{\boldsymbol{x}}\| \\ \|\tilde{\boldsymbol{\sigma}}\|\end{bmatrix}^T \begin{bmatrix}\lambda_o - \rho_o & -\rho_c \\ -\rho_c & k_c\end{bmatrix}\begin{bmatrix}\|\tilde{\boldsymbol{x}}\| \\ \|\tilde{\boldsymbol{\sigma}}\|\end{bmatrix} \qquad (18)$$

that is negative definite with a proper choice of the design gains k_o and k_c; in detail, by taking into account that $\lambda_o = k_o\lambda_m$, negative definiteness is guaranteed by

$$k_o > N\frac{k_c}{\lambda_m}\left(\sqrt{n} + \frac{Nn\|\boldsymbol{\alpha}\|^2}{4}\right) . \qquad (19)$$

Remark 5.1: Since λ_o can be arbitrarily set via k_o, the above inequality can be always satisfied by suitably choosing k_o and k_c. However, it must be noticed that tuning of the observer and controller gains cannot be performed independently, i.e., a separation property does not hold. Arguably, this is due to the fact that the observer does not know the whole collective input, and thus convergence of the state estimates cannot be achieved independently from the control input structure.

Remark 5.2: Consider the following, slightly more general, class of task functions

$$\boldsymbol{\sigma}(\boldsymbol{x}) = \sum_{i=1}^{N}\alpha_i\boldsymbol{\phi}(\boldsymbol{x}_i) , \qquad (20)$$

where $\boldsymbol{\phi} \in \mathbb{R}^n$ is a smooth vector field, and assume that the same control law (6) is adopted, i.e.,

$$\boldsymbol{u}_i = \frac{\alpha_i}{\|\boldsymbol{\alpha}\|^2}\left(\dot{\boldsymbol{\sigma}}_d + k_c\left(\boldsymbol{\sigma}_d - \sum_{i=1}^{N}\alpha_i\boldsymbol{\phi}({}^i\hat{\boldsymbol{x}}_i)\right)\right) . \qquad (21)$$

It can be easily verified that the above stability proof can be readily extended to this case, provided that the function $\boldsymbol{\phi}(\cdot)$ is globally Lipschitz, i.e.,

$$\|\boldsymbol{\phi}(\boldsymbol{x}_a) - \boldsymbol{\phi}(\boldsymbol{x}_b)\| \leq \rho_x\|\boldsymbol{x}_a - \boldsymbol{x}_b\| , \qquad (22)$$

for some positive constant ρ_x and for all $\boldsymbol{x}_a, \boldsymbol{x}_b$.

VI. Directed and switching topologies

A. Directed Topologies

In the following, it will be shown that stability of the overall closed-loop system is preserved also in the case of directed topologies, provided that the graph is balanced and strongly connected.

In the stability proof reported in Section V, it is required that the matrix $\boldsymbol{L}\otimes\boldsymbol{I}_{Nn} + \boldsymbol{\Pi}$ is positive definite. As shown in Appendix A, this property holds for undirected connected graphs, for which the above matrix is symmetric. In the case of directed graphs, the Laplacian $\boldsymbol{L}$ is not symmetric. However, in such a case a *mirror graph*, $\mathcal{G}_S$, associated to $\mathcal{G}$ can be defined as the undirected graph having the same set of nodes and same set of edges, but considered undirected, as $\mathcal{G}$ [11]. It can be shown that the symmetric part of the Laplacian, $\boldsymbol{L}_S = \dfrac{\boldsymbol{L}+\boldsymbol{L}^{\mathrm{T}}}{2}$, is a valid Laplacian for $\mathcal{G}_S$, if and only if $\mathcal{G}$ is balanced [11]. In addition, if $\mathcal{G}$ is strongly connected, then $\mathcal{G}_S$ is connected.

Since $\dfrac{1}{2}\left((\boldsymbol{L}\otimes\boldsymbol{I}_{Nn}) + (\boldsymbol{L}\otimes\boldsymbol{I}_{Nn})^{\mathrm{T}}\right) + \boldsymbol{\Pi} = \boldsymbol{L}_S\otimes\boldsymbol{I}_{Nn} + \boldsymbol{\Pi}$, equation (14) can be rewritten as

$$\begin{aligned}
\dot{V}_o \;=\; & -k_o\tilde{\boldsymbol{x}}^{\mathrm{T}}\left(\boldsymbol{L}_S\otimes\boldsymbol{I}_{Nn} + \boldsymbol{\Pi}\right)\tilde{\boldsymbol{x}} + \\
& \tilde{\boldsymbol{x}}^{\mathrm{T}}((\boldsymbol{1}_N\otimes\boldsymbol{I}_{Nn})\boldsymbol{u} - \hat{\boldsymbol{u}}) .
\end{aligned} \qquad (23)$$

Moreover, being $\mathcal{G}_S$ connected, the same arguments in Appendix A can be used to prove positive definiteness of $\boldsymbol{L}_S\otimes\boldsymbol{I}_{Nn} + \boldsymbol{\Pi}$. Thus, the same arguments used in Section V lead to prove asymptotic stability.

B. Switching Topologies

It is reasonable to consider a time-varying network topology, e.g., due to the failure of active communication links or to the activation/deactivation of links due to the dynamic displacement of the nodes. In such cases, the network topology can be described via a finite collection of K graphs of order N, $\Gamma = \{\mathcal{G}_1, \ldots, \mathcal{G}_K\}$, each characterized by its adjacency matrix $\boldsymbol{A}_k$, $k \in I = \{1, \ldots, K\}$.

Hence, the adjacency matrix can be modeled as a function of time, i.e, $\boldsymbol{A} = \boldsymbol{A}_{s(t)}$, where $s(\cdot) : t \in \mathbb{R} \rightarrow I$ is a switching signal. In other words, $\boldsymbol{A}_{s(t)}$ is a piecewise continuous function that associates at each time instant one of the finite possible network configurations. In the same way, let be $\boldsymbol{L}_{s(t)}$ the Laplacian matrix corresponding to $\boldsymbol{A}_{s(t)}$ that is, obviously, a piecewise continuous function too.

The Lyapunov function in (13) is a Common Lyapunov Function (CLF) for the overall closed-loop system for any switching signal $s(t)$, provided that each graph in Γ is balanced and strongly connected (in the case of directed topology) or simply connected (in the case of undirected topology) and (19) holds for any t. To this aim, tuning of k_o and k_c could be performed according to the worst case scenario, i.e., by considering the minimum value of λ_m over the finite set of network topologies.

VII. Numerical simulations

In the following, two simulation case studies have been considered. In the first case, the multi-robot system is characterized by an undirected fixed topology, while in the second case a directed switching topology is considered. In both the cases, the vehicles start at a random configuration. The task function is defined as in (5) with $\alpha_i = 1/N, \quad i = 1, 2 \cdots N$; the desired trajectory $\boldsymbol{\sigma}_d(t)$ is given by a cubic spline function interpolating a given set of via points.

A. First case study: fixed topology

As first case study, a fixed undirected topology with 4 vehicles ($N = 4$) moving in a plane ($n = 2$) has been considered. The topology is shown in Figure 1. The parameters k_o and

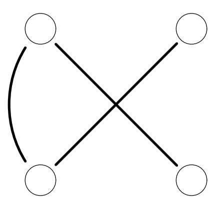

Fig. 1. First case study. Undirected topology.

k_c in (21) and (7) have been set, respectively, to 5 and 3. In Figure 2, the vehicles' paths, and the desired and actual task functions are shown. In Figure 3, the time histories of

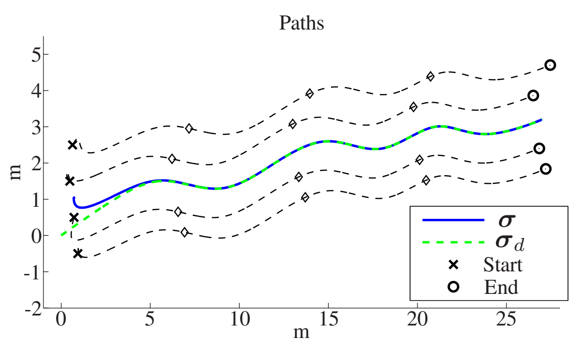

Fig. 2. First case study. Task and vehicles' paths. The crosses represent the vehicles' starting positions, the rounds the ending positions and the diamonds some intermediate configurations.

the average estimation error norm (top) and of the task error norm (bottom) are reported.

B. Second case study: directed switching topology

As second case study, a switching directed topology with 8 vehicles ($N = 8$) moving in the 3D-space ($n = 3$) has been considered. The network topology switches between the three configurations shown in Figure 4. The parameters k_o and k_c in (21) and (7), have been set to 8 and 3, respectively. In Figure 5, the vehicles' path, the desired and actual task functions are shown. In Figure 6, the time histories of the average estimation error norm (top) and of the task error norm (bottom) are reported.

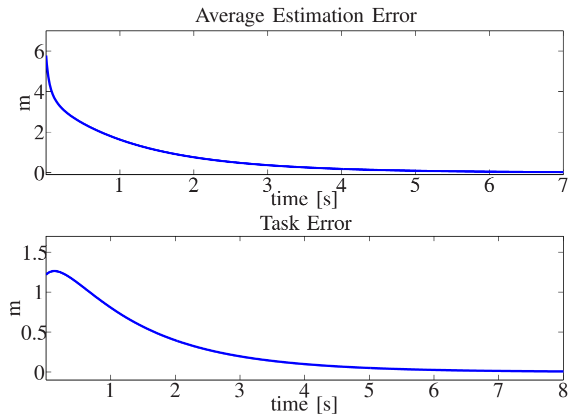

Fig. 3. First case study. Top: average estimation error. Bottom: task error.

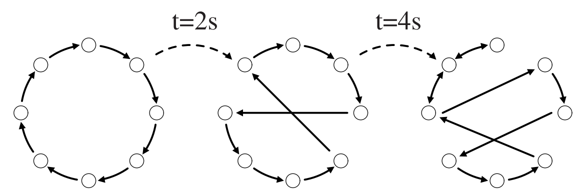

Fig. 4. Second case study. Directed switching topology.

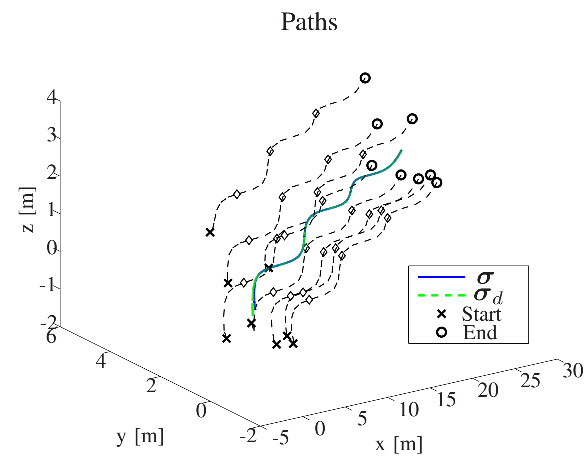

Fig. 5. Second case study. Task and vehicles' paths. Crosses represent the vehicles' starting positions, rounds the ending positions and diamonds intermediate configurations.

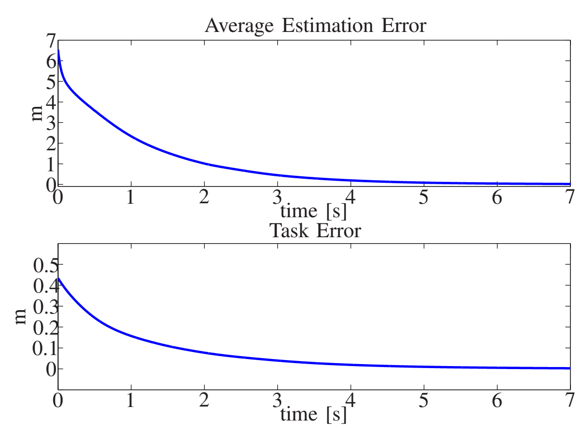

Fig. 6. Second case study. Top: average estimation error. Bottom: task error.

VIII. Conclusions

In this paper, a decentralized controller-observer approach for a multi-robot system has been developed. Each robot estimates the collective state of the system by using only local information. The estimated state is then used by the individual robots to cooperatively track a global assigned time-varying task function. The approach is, then, extended to the case of strongly connected and balanced directed graphs, as well as to switching topologies. Future works will be focused on extending the class of achievable task functions to a wider domain, on how to make the proposed approach robust to dynamic lost or addition of robots to the team, and on how to improve the scalability of the approach reducing the overall information exchange.

Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement n. 231378 (STREP project Co^{3}AUVs) and from the Italian Government, under Grant FIRB - Futuro in ricerca 2008 n. RBFR08QWUV (project NECTAR) and under Grant PRIN 2008 n. 2008EPKCHX (project MEMONET).

Appendix

A. Analysis of the matrix $k_o\left(\boldsymbol{L}\otimes\boldsymbol{I}_{Nn}+\boldsymbol{\Pi}\right)$

The proof in this section applies to undirected connected graph but, as seen in Section VI-A, can easily be extended to the case of directed balanced weakly connected graph.

As stated in Section V, the matrix $\boldsymbol{L}\otimes\boldsymbol{I}_{Nn}$ and $\boldsymbol{\Pi}$ are positive semi-definite matrices in the case of undirected graph. Then, their sum is a positive semi-definite matrix, and is also positive-definite if and only if their kernel subspaces are disjoint, i.e.:

$$\ker\left(\boldsymbol{L}\otimes\boldsymbol{I}_{Nn}\right)\bigcap\ker\left(\boldsymbol{\Pi}\right)=\left\{\boldsymbol{0}_{N^2n}\right\}. \tag{24}$$

Given the Laplacian properties recalled in Section II, it can be easily recognized that, for connected graphs,

$$\operatorname{rank}\left(\boldsymbol{L}\otimes\boldsymbol{I}_{Nn}\right)=\operatorname{rank}\left(\boldsymbol{L}\right)\operatorname{rank}\left(\boldsymbol{I}_{Nn}\right)=Nn(N-1),$$

and $\dim\left(\ker\left(\boldsymbol{L}\otimes\boldsymbol{I}_{Nn}\right)\right)=Nn$. The null space of $\boldsymbol{L}\otimes\boldsymbol{I}_{Nn}$ can be parameterized as follows

$$\ker\left(\boldsymbol{L}\otimes\boldsymbol{I}_{Nn}\right)=\operatorname{span}\left(\boldsymbol{1}_N\otimes\boldsymbol{I}_{Nn}\right). \tag{25}$$

Thus, a vector belonging to $\left(\boldsymbol{L}\otimes\boldsymbol{I}_{Nn}\right)$ has the form

$$\boldsymbol{v}=\left[\boldsymbol{\nu}^{\mathrm{T}}\dots\boldsymbol{\nu}^{\mathrm{T}}\right]^{\mathrm{T}}\in\mathbb{R}^{N^2n},\ \forall\boldsymbol{\nu}\in\mathbb{R}^{Nn}. \tag{26}$$

Moreover, being $\boldsymbol{\Pi}$ a diagonal matrix with Nn non-null (unitary) elements along the main diagonal, $\operatorname{rank}\left(\boldsymbol{\Pi}\right)=Nn$ and $\dim\left(\ker\left(\boldsymbol{\Pi}\right)\right)=Nn(N-1)$. The null space of $\boldsymbol{\Pi}$ can be expressed as

$$\ker\left(\boldsymbol{\Pi}\right)=\operatorname{span}\left(\boldsymbol{I}_{N^2n}-\boldsymbol{\Pi}\right), \tag{27}$$

where $\left(\boldsymbol{I}_{N^2n}-\boldsymbol{\Pi}\right)$ is a diagonal matrix with $Nn(N-1)$ non null elements on the main diagonal.

Thus, a vector belonging to $\ker\left(\boldsymbol{\Pi}\right)$ has the form

$$\begin{aligned}\boldsymbol{v}&=\begin{bmatrix}\boldsymbol{v}_1^{\mathrm{T}} & \dots & \boldsymbol{v}_N^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}\in\mathbb{R}^{N^2n},\\ \boldsymbol{v}_i&=\begin{bmatrix}\boldsymbol{v}_{i,1}^{\mathrm{T}}\dots\boldsymbol{v}_{i,N}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}\in\mathbb{R}^{Nn}:\forall\boldsymbol{v}_{i,j}\in\mathbb{R}^n,\boldsymbol{v}_{i,i}=\boldsymbol{0}_n.\end{aligned} \tag{28}$$

Comparing eqs (26)–(28) it is possible to observe that a non-null vector in $\ker\left(\boldsymbol{L}\otimes\boldsymbol{I}_n\right)$ cannot belong to $\ker\left(\boldsymbol{\Pi}\right)$ and viceversa. This implies that (24) holds and that $\left(\boldsymbol{L}\otimes\boldsymbol{I}_{Nn}+\boldsymbol{\Pi}\right)$ is positive definite. It is worth noticing that in the case of non-connected graphs

$$\operatorname{rank}\left(\boldsymbol{L}\otimes\boldsymbol{I}_n\right)\leq N^2n-2Nn$$

and then

$$\begin{aligned}\operatorname{rank}\left(\boldsymbol{L}\otimes\boldsymbol{I}_n+\boldsymbol{\Pi}\right)&\leq\operatorname{rank}\left(\boldsymbol{L}\otimes\boldsymbol{I}_n\right)+\operatorname{rank}\left(\boldsymbol{\Pi}\right)\\&\leq N^2n-Nn,\end{aligned}$$

where the inequality holds strictly when more then one disconnected component arises in the graph.

References

[1] D. Bauso, L. Giarré, and R. Pesenti. Non-linear protocols for optimal distributed consensus in networks of dynamic agents. *Systems & Control Letters*, 55(11):918–928, 2006.

[2] J. Cortés. Characterizing robust coordination algorithms via proximity graphs and and set-valued maps. In *Proceedings of the American Control Conference*, pages 8–13, 2006.

[3] J. Cortés. Distributed algorithms for reaching consensus on general functions. *Automatica*, 44:726–737, 2008.

[4] J. Cortés, S. Martínez, and F. Bullo. Analysis and design tools for distributed motion coordination. In *Proceedings of the American Control Conference*, pages 1680–1685, 2005.

[5] J.A. Fax and R.M. Murray. Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 49(9):1465–1476, 2004.

[6] R.A. Freeman, P. Yang, and K.M. Lynch. Stability and convergence properties of dynamic consensus estimators. In *Proceedings of IEEE International Conference on Decision and Control*, pages 338–343, December 2006.

[7] A. Jadbabaie, J. Lin, and A.S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, June 2003.

[8] M. Ji and M. Egerstedt. Distributed Coordination Control of Multi-agent Systems While Preserving Connectedness. *IEEE Transactions on Robotics*, 23(4):693–703, 2007.

[9] M. Mesbahi and M. Egerstedt. *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

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

[11] R. Olfati-Saber and R.M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49:1520–1533, 2004.

[12] W. Ren. Multi-vehicle consensus with a time-varying reference state. *Systems & Control Letters*, 56(7-8):474–483, 2007.

[13] W. Ren and R.W. Beard. *Distributed Consensus in Multi-vehicle Cooperative Control*. Communications and Control Engineering. Springer, Berlin, G, 2008.

[14] W. Ren, R.W. Beard, and E.M. Atkins. Information consensus in multivehicle cooperative control. *IEEE Control Systems Magazine*, 27(2):71–82, Apr. 2007.

[15] G. Royle and C. Godsil. *Algebraic Graph Theory*. Springern Graduate Texts in Mathematics, New York, NY, 2001.

[16] R.S. Smith and F.Y. Hadaegh. Closed-loop dynamics oof cooperative vehicle formations with parallel estimators and communication. *IEEE Transactions on Automatic Control*, 52(8):1404–1414, 2007.

[17] D.P. Spanos, R. Olfati-Saber, and R.M. Murray. Dynamic consensus on mobile networks. In *Proceedings of the IFAC World Congress*, 2005.

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