Interconnected Dynamic Systems

AN OVERVIEW ON DISTRIBUTED CONTROL

GIANLUCA ANTONELLI

Control problems such as multirobot control, distributed intelligence, swarm intelligence, distributed decision, distributed cognition, congestion control in networks, collective motion in biology, oscillator synchronization in physics, parallelization in optimization theory, distributed estimation, cooperative estimation, equilibria in economics, social interaction modeling, and game theory may be analyzed under the theory of interconnected dynamic systems. Those topics have several overlapping research communities; for that reason they are characterized by different definitions and a variety of approaches ranging from rigorous mathematical analysis to trial-and-error experimental study or emulation by observation of natural phenomena. The areas involved concern robotics, dynamic systems, computer science, signal theory, biology, economics, and mathematics. A shared taxonomy is missing; for example, dynamic systems can be identified in robots, agents, nodes, processors, and entities. An ensemble is called a group, network, platoon, swarm, team, and cluster, and the algorithms are defined as controllers, protocols, and dynamics. In the following, the term *agent* is used to denote the single dynamic system and *network* or *collective* the ensemble.

As an exemplification domain, let us consider the

Digital Object Identifier 10.1109/MCS.2012.2225929
Date of publication: 17 January 2013

1066-033X/13/$31.00©2013IEEE

robotics field; an introduction to interconnected dynamic systems is represented by [1]–[5], and possible specific applications are traffic control, formation control, coordinated environment interaction or manipulation, attitude alignment of satellites, exploring, monitoring, patrolling, coverage, pursuit-evasion, search and rescue, distributed sensor network, deployment, mobile ad-hoc networks, entertainment, and robosoccer. In addition to those topics, distributed estimation, social networks, or games over networks are discussed, for example, in [6].

The goal of this article is to provide a brief overview on control algorithms for interconnected dynamic systems and to focus on the current state of the art and possible future directions. To make it easier for the reader and streamline the presentation of the algorithms while evidencing the different control architectures, a possible classification is also given. Due to the wide area covered and the large number of communities and overlapping topics concerned, this attempt is far from being definite. A basic concept is the neighboring agents, analytically defined in "Graph Theory," here considered as the agents from which each agent can take information from, via either communication or sensing. When a control problem may be formulated as a proper combination of functions expressed in terms of the neighboring states, it is defined as *local*. An example of local control problem is the flocking one [7], [8], where the control objective is the distance among the neighboring agents. On the other hand, the problem is defined *global* when, to be solved, it is necessary to obtain current information, for example the state, from all the network's agents. An example of global problem is given by the control of the average position of the agents, i.e., the centroid control.

The *solution* of the formulated problem may also need global or local information; in this case, it is common the use of the terms *centralized* versus *distributed* control architecture. The control architecture is defined as distributed when the solution is achieved via local interactions only; that is, each agent exchanges information with its neighbors. The term *decentralized* is largely used as a synonym of distributed. The centralized feature arises when at least one single agent needs to sense a global information or communicate with all the other agents at once. Distributed control approaches generally exhibit a stronger fault tolerance in that, for example, they do not experience the presence of a single point of failure. Works [8] and [9] represent an overview of local problems afforded with a distributed architecture. In a centralized control architecture, the algorithm computational load is generally proportional to the network size, i.e., to the number of agents. In a distributed architecture the computational load *for each agent* depends on the specific algorithm implemented; if the load grows with the network size it is said that *it does not scale*. Distributed scalable approaches are thus characterized by generally simpler and cheaper agents. The scalability concept, being shared by several research communities, is characterized by different definitions; it is often used as synonym for distributed.

Control problems dealing with a network of agents are usually approached following two different directions, top down or bottom up. In a top-down approach, a problem is decomposed so that its solution can be distributed among the agents. In a bottom-up approach, a local interaction is devised, and the collective dynamics are analyzed. The latter approach gave raise to what is called *emergent* dynamics or behavior. The terms *emergent*, *aggregate*, or *self-organized* are commonly used to describe local control problems solved with a distributed/decentralized approach.

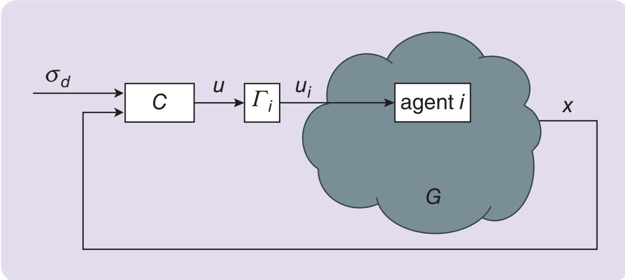

FIGURE 1 Schematic representation of a global control problem afforded with a centralized architecture. All the information flows are global, and the controller is accessing the whole state x.

This article is focused on the control of multiagent systems. The agents are modeled with a continuous-time, single integrator dynamics and their state is generally their physical position; consequently, the agents interconnection is not embedded in the investigated phenomenon but rather imposed by the feedback design. Distributed estimation is not covered in this article.

ARCHITECTURES OF THE POSSIBLE CONTROL SCHEMES

According to the short introduction given, a global control problem afforded with a centralized architecture can be schematically represented as in Figure 1, where the dark cloud graphically represents the network, all the information flows are global, and the controller is accessing the whole state x; see "Definitions and Nomenclature." The matrix $\boldsymbol{\Gamma}_i \in \mathbb{R}^{np \times p}$ is a selection matrix of the form

$$\boldsymbol{\Gamma}_i = [\boldsymbol{O}_p \cdots \boldsymbol{I}_p \cdots \boldsymbol{O}_p].$$

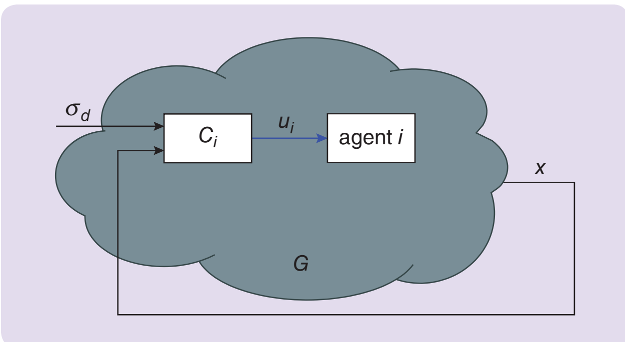

FIGURE 2 Schematic representation of a global control problem afforded with a partially distributed architecture. Notice that the thin blue line denotes a local information flow.

Graph Theory

Graph theory can be useful when it is necessary to model the concept of proximity among the agents. As an example, if the agents are equipped with range-limited communication devices, a graph might be the proper tool to mathematically handle the concept of connectivity among them. Roughly speaking, two agents are neighbors when some kind of connection arises between them. Following [9], a possible list of proximity graphs is given considering the *r*-disk graph, in which the agents are neighbors when the relative distance is smaller than a given value *r*; the Delaunay graph, in which two agents are neighbors when the corresponding Voronoi cells, defined in "Voronoi Partitions," are contiguous; the *r*-limited Delaunay graph, in which two agents are neighbors when the corresponding Voronoi cells are contiguous and when their distance is smaller than *r*; and the visibility graph, in which two agents are neighbors when they are visible one each other. The definitions, and other examples, can be found, among the others, in [9]. In the following, the concepts of graph theory that are useful for the article's comprehension are briefly shown. For a detailed discussion, the interested reader may refer to [71] and [72].

A directed graph, or digraph, is a pair $G = (V, E)$ where V is a nonempty set of nodes, also named vertices in the literature, and $E \subseteq V \times V$ is the set of the corresponding edges. A weighted graph associates a weight to each edge of the graph. The graph G is also defined as topology or information flow [16].

When the edge *(i, j)* belongs to *E,* it means that there is an information flow from the node i, the parent node, to the node j, the child node. The order of a graph is the number of the nodes $|V|$. The set of neighbors of one node v_i is $N_i = \{j \in V : (j, i) \in E\}$, that is all the nodes that the node i can take information from (see Figure S1). A subgraph of (*V, E*) is a graph $(\tilde{V}, \tilde{E})$ such that $\tilde{V} \subseteq V$ and $\tilde{E} \subseteq E$. The output degree of a node v_i: $\text{deg}_{\text{out}}(v_i)$ is the number of outgoing edges e_{ij}, it is $\text{deg}_{\text{out}}(v_i) = |N_i|$. The input degree of a node v_i, $\text{deg}_{\text{in}}(vi)$ is the number of ingoing edges e_{ki}. A graph is defined as being balanced when it has the same number of ingoing and outgoing edges, $\text{deg}_{\text{out}}(v_i) = \text{deg}_{\text{in}}(v_i)$ for all the nodes v_i.

An undirected graph is a particular type of directed balanced graph where $e_{ij} \in E \Rightarrow e_{ji} \in E$.

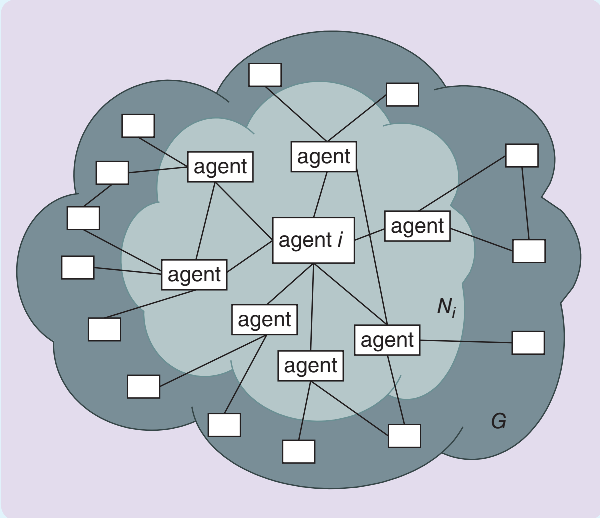

FIGURE S1 The network in an *i*-agent-centric view. Each agent is connected with its neighboring N_i, graphically represented as a light cloud and, in multiple hops, with the whole network G, graphically represented as a darker cloud.

A graph is strongly connected when there exists a directed path connecting every arbitrary pair of distinct nodes. A graph is connected, or weakly connected, when there exists an undirected path connecting every arbitrary pair of distinct nodes. Notice that for undirected graphs the two definitions coincide. The diameter of a connected graph is the maximum length of a path in G. A tree is an undirected graph where all the nodes can be connected by the way of a single undirected path. A rooted directed tree is a directed graph in which every node has one single parent except for one node, defined as the root, that has no parent and has a directed path connecting it to all the remaining nodes. In a rooted directed tree, all the edges are going away from the root. A rooted directed tree is also defined as spanning when it connects all the edges of the graph. It can be demonstrated that this implies that there is at least one root node connected with a simple path to all the other nodes. For undirected graphs, a rooted spanning tree is equivalent to the property of being connected; for directed graphs, however, the property of strong connection implies the

Figure 2 represents the schematic representation of a global control problem afforded with an architecture in which the agents are supposed to have access to global information that is elaborated on board; notice that the thin blue line denotes a local information flow. An example of this architecture, also defined as partially distributed, is given by [10].

A local control problem afforded with a distributed architecture is schematically represented in Figure 3 where the lighter cloud represents the *i*th agent neighbors. Notice that all lines are thin and blue, denoting local information flows only. The block labeled as "obs *i*" generically denotes a possible observer with inputs the state $\boldsymbol{x}_i$, the input $\boldsymbol{u}_i$ of the corresponding agent, and the states of the neighboring agents $\boldsymbol{x}_j$. This architecture is typical of a bottom-up approach, or it is the architecture of local control problems such as a relative formation control, a flocking, or a deployment [9].

A global control problem afforded by means of a distributed architecture is shown in Figure 4. The sole graphical existence of a rooted directed spanning tree but not vice versa; in other words, the property of the existence of a rooted directed spanning tree is weaker with respect to the strong connection's one.

An appealing feature of the graphs is the possibility to catch some of the properties into properly defined matrices. Given a graph G, the adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{|V| \times |V|}$ is defined as

$$\boldsymbol{A} = [a_{ij}], \quad a_{ij}, = \begin{cases} 1, & j \in N_i \\ 0, & \text{otherwise} \end{cases}.$$

The input degree matrix $\boldsymbol{D} \in \mathbb{R}^{|V| \times |V|}$ is a diagonal matrix defined as

$$\boldsymbol{D} = [d_{ij}], \quad d_{ij} = \begin{cases} \deg_{\text{in}}(v_i), & i = j \\ 0, & \text{otherwise} \end{cases}$$

and the graph Laplacian matrix $\boldsymbol{L} \in \mathbb{R}^{|V| \times |V|}$ is defined as

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A} = [l_{ij}],$$

which is equivalent to

$$l_{ij} = \begin{cases} -1, & j \in N_i \\ |N_i|, & j = i \\ 0, & \text{otherwise} \end{cases}.$$

The adjacency and the Laplacian matrices are positive semidefinite by definition; in addition, for undirected graphs they also exhibit a symmetric structure. For directed graphs it is common to use the terms not symmetric Laplacian and directed Laplacian [31]. Notice that in [29] the term Laplacian is used for the matrix defined as normalized Laplacian in [16], [8], and [15]. Notice also that different definitions exist for the adjacency matrix and consequently for the Laplacian. This proliferation of nomenclature is an indirect proof of the variety of communities working with these tools.

The Laplacian matrix exhibits several useful properties related to the graph's topology. It holds $\sum_j l_{ij} = 0$ for all i, that is each row sums to zero. For this reason the Laplacian matrix is also defined as row stochastic. The scalar zero is always an eigenvalue of $\boldsymbol{L}, \lambda_1 = 0$. The vector $\mathbf{1}_{|V|}$ is always the corresponding eigenvector

$$\boldsymbol{L}\mathbf{1}_{|V|} = 0_{|V|},$$

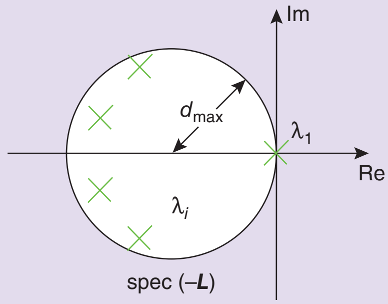

FIGURE S2 The Geršgorin circle theorem allows computing of the spectrum of the Laplacian $\boldsymbol{L}$. In several distributed algorithms the Laplacian appears to be the dynamic matrix of a proper dynamic system.

$\text{Im}(\mathbf{1}_{|V|}) \subseteq \text{Ker}(\boldsymbol{L})$ thus the rank $\rho(\boldsymbol{L}) \leq |V| - 1$. When G is strongly connected, then $\rho(\boldsymbol{L}) = |V| - 1$, the zero eigenvalue is simple; the vice versa is not true unless G is undirected. When the graph contains a directed spanning tree, it can be demonstrated that the zero eigenvalue is simple [15]. When G is undirected then the multiplicity of the zero eigenvalue is equal to the connected components of the graph, or, in other words, the multiplicity is equal to the connected subgraphs in which G can be divided. When G is balanced then also the columns of $\boldsymbol{L}$ sum to zero,

$$\mathbf{1}_{|V|}^{T}\boldsymbol{L} = 0_{|V|}^{T},$$

and $1_{|V|}$ is also left eigenvector of the zero eigenvalue, and the matrix is also defined column stochastic, or row-column doubly stochastic. A side effect of the Geršgorin circle theorem is that the eigenvalues of $\boldsymbol{L}$ are located in the complex plane in a circle with center and radius of value d_{max}, where d_{max} is the maximum graph's input degree, as shown in Figure S2 [16]; for the normalized Laplacian $\bar{\boldsymbol{L}}$ it holds $d_{max} = 1$. For the undirected graphs the matrix $\boldsymbol{L}$ is symmetric and the eigenvalues are real. The second smallest eigenvalue of $\boldsymbol{L}$, the Fiedler eigenvalue, is the algebraic connectivity; it represents a metric on the speed convergence of the consensus algorithm [16]. A short tutorial on graph theory can also be found in [16] and [15].

difference with respect to the previous case is the thicker black line representing the control objective. This small difference totally modifies the dimensionality of the problem, as only a subset of global control objectives is solved with a distributed scheme [11], [12].

The missing matches, a local control problem approached with a centralized or a partially distributed architecture, are not reported here. Such schemes would represent a formalism effort without theoretical nor practical benefit.

NETWORK CONTROL PROBLEMS

This section reviews the main control problems that arise when dealing with networks. The focus is on the control objective, with the corresponding estimation when needed and the corresponding suggested approaches to solve it.

Rendezvous, Consensus

Consensus is the process of reaching an agreement concerning a certain quantity of interest depending on the state of all the agents [13]; a consensus algorithm is an

Definitions and Nomenclature

The ith agent state is denoted as $\boldsymbol{x}_i \in \mathbb{R}^p$ where $p = \{1,2,3\}$ in case of one-dimensional, two-dimensional, and three-dimensional case studies. The vector $\boldsymbol{x} \in \mathbb{R}^{pn}$ collects the states of all the n agents, that is, the collective network's state.

It is common to consider the dynamics of the agents as a simple integrators,

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

with $\boldsymbol{u}_i \in \mathbb{R}^p$, and a corresponding collective dynamics

$$\dot{\boldsymbol{x}} = \boldsymbol{u}, \tag{S1}$$

with $\boldsymbol{u} \in \mathbb{R}^{pn}$.

The identity matrix of dimension $m \times m$ is denoted as $\boldsymbol{I}_m$, with the vector 0_m represents a vector of m zeros and the vector $\boldsymbol{1}_m$ represents the vector of unitary components $\boldsymbol{1}_m = [1 \;\; 1 \;\; \cdots \;\; 1]^{\mathrm{T}}$.

Given two matrices $\boldsymbol{A} \in \mathbb{R}^{n_a \times m_a}$ of entries a_{ij}, and $\boldsymbol{B} \in \mathbb{R}^{n_b \times m_b}$ of entries b_{ij}, their Kronecker product [69]

$$\boldsymbol{C} = \boldsymbol{A} \otimes \boldsymbol{B}$$

is defined as the matrix whose blocks are $a_{ij}\boldsymbol{B}$; the matrix $\boldsymbol{C}$ has dimension $(n_a n_b \times m_a m_b)$. Notice that when the first matrix is the identity matrix, the Kronecker product is duplicating the second matrix on the block diagonal entries, as an example

$$\boldsymbol{C} = \boldsymbol{I}_2 \otimes \boldsymbol{A} = \begin{bmatrix} \boldsymbol{A} & 0 \\ 0 & \boldsymbol{A} \end{bmatrix}.$$

When the second matrix is the identity matrix, on the other hand, it is always possible to find a permutation matrix $\boldsymbol{T}$ that allows to exhibit a block-diagonal structure

$$\boldsymbol{C} = \boldsymbol{A} \otimes \boldsymbol{I}_2 = \boldsymbol{T}(\boldsymbol{I}_2 \otimes \boldsymbol{A}) = \boldsymbol{T} \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A} \end{bmatrix},$$

notice, also, that C and $\boldsymbol{T}^{-1}\boldsymbol{C}\boldsymbol{T}$ share the same eigenvalues.

The variable to be controlled is defined as $\boldsymbol{\sigma} \in \mathbb{R}^m$, usually a function of the configuration

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}(\boldsymbol{x}), \tag{S2}$$

for which a desired value $\boldsymbol{\sigma}_d(t)$ is usually designed. The corresponding tracking error is defined as

$$\tilde{\boldsymbol{\sigma}}(t) = \boldsymbol{\sigma}_d(t) - \boldsymbol{\sigma}(t).$$

The first-order differential relationship of (S2) is

$$\dot{\boldsymbol{\sigma}} = \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{x}} \dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{x})\dot{\boldsymbol{x}}, \tag{S3}$$

where $\boldsymbol{J} \in \mathbb{R}^{m \times np}$ is the function Jacobian, usually configuration dependent. In this article $m \le np$ is assumed. When the ith Jacobian is function of the neighbors only (see "Graph Theory" for the definition of a neighbor), the Jacobian may be further rewritten as

$$\boldsymbol{J}(N_i) = [\boldsymbol{J}_1(N_1) \quad \boldsymbol{J}_2(N_2) \quad \cdots \quad \boldsymbol{J}_n(N_n)], \tag{S4}$$

where $\boldsymbol{J}_i(N_i) \in \mathbb{R}^{m \times p}$ is associated with the ith agent.

interaction rule that specifies the information flow among neighboring agents [14]. The problem of rendezvous is defined as the requirement for all the agents to converge on one common point by resorting to local negotiations [15], [8]. The basic concept of rendezvous, thus, is a consensus problem under a different perspective; often the word rendezvous is used when the state is the physical position of the agents.

Each agent x_i can take information from its neighbors $x_j \in N_i$. One basic consensus algorithm is in the form

$$\dot{x}_i(t) = \sum_{j \in N_i} (x_j(t) - x_i(t)), \tag{1}$$

in which the i dynamics is simply linked to the difference of the neighboring states. After some matrix computation it is possible to derive the collective dynamics, that is the given interaction rule from a network perspective:

$$\dot{x}(t) = -(L \otimes I_p)x(t), \tag{2}$$

where $x \in \mathbb{R}^{np}$ is the network state, and $L \in \mathbb{R}^{n \times n}$ is the graph Laplacian; see "Graph Theory."

For connected graphs the system represented by (1) is Lyapunov stable. Its dynamic matrix exhibits $p(n-1)$ eigenvalues with negative real part and one zero eigenvalue with algebraic and geometric multiplicity equal to p. Its equilibrium points x_{ss} are given by the kernel of the Laplacian

$$x_{ss} = \boldsymbol{\alpha} \otimes \boldsymbol{1}_n,$$

with $\boldsymbol{\alpha} \in \mathbb{R}^p$ vector of arbitrary components. These points are also defined as collective decision [7] or group decision value [16]. If the graph is balanced, as for undirected graphs, the generic coefficient α_j is the average of the initial values

$$\alpha_j = \frac{1}{n} \sum_{i=1}^{n} x_{j,i}(0).$$

In words, for a connected and balanced graph all the agents converge to the average of their initial values. When G is connected but not balanced, the value of α_j needs to be calculated [16]. When G contains a rooted directed spanning tree, it means that there is at least one node without incoming edges, the root. In such a case, it can be demonstrated that the steady state is the initial state of the root for all the

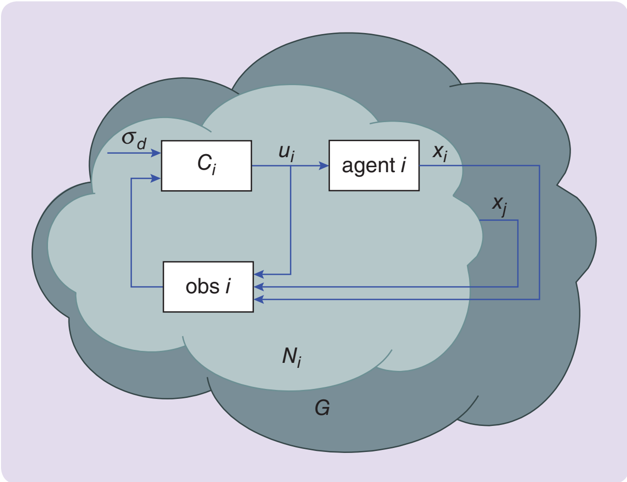

FIGURE 3 A local control problem afforded with a distributed architecture. All lines are thin, blue denoting only local information flows.

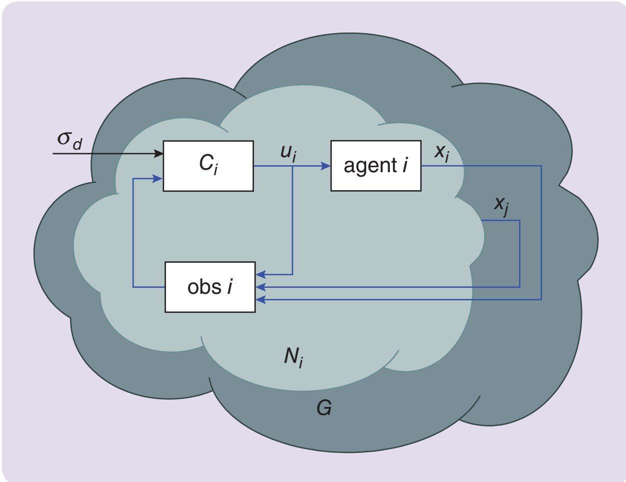

FIGURE 4 Global control problem in a distributed architecture. The thin, blue lines denote local information flows while the black line represents a global control problem.

nodes. A common trick to force the network in reaching the same constant value, is to add a virtual node connected to a proper node of the network as shown in Figure 5.

Note that the convergence time length is related to the algebraic connectivity (λ_2) or to its real part for the directed graphs [16]. Different networks characterized by a different algebraic connectivity exhibit different convergence times; often, the addition of randomly placed links allow for a decrease in the settling time. Note that it is possible to surround this limit by resorting to a weighted graph.

Consensus equations can be interpreted under a gradient-based approach perspective. In fact, assuming the network undirected and connected, the following function $\sigma \in \mathbb{R}$, the Laplacian potential [16]

$$\sigma(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}(\boldsymbol{L} \otimes \boldsymbol{I}_p)\boldsymbol{x} = \frac{1}{2}\sum_{i,j \in E}(\boldsymbol{x}_i - \boldsymbol{x}_j)^2 \tag{3}$$

is null when consensus is reached. The objective function, thus, is local. Its Jacobian $\boldsymbol{J}(N_i) \in \mathbb{R}^{1 \times np}$ is given by

$$\boldsymbol{J}(N_i) = \boldsymbol{x}^{\mathrm{T}}(\boldsymbol{L} \otimes \boldsymbol{I}_p),$$

where the dependency by the neighboring N_i is evidenced. In practice, a gradient-based approach would lead to the basic consensus equations, which is a distributed algorithm.

A further problem is to achieve rendezvous while preserving connectivity as discussed in [17] and [9] or in the recent survey [18] where both centralized an distributed algorithms are presented.

One of the first analytical results is [19]. In the literature, several tutorial articles can be found on consensus algorithms. Short introductions, on both continuous and discrete time, can be found in [15] and [13], which contains also a rich bibliography. It is interesting to study the convergence property in the presence of a time-varying topology; this is the case of moving agents that communicate on networks given by proximity graphs [20], [21], [13], [16], [22]. The same references discuss the properties of consensus algorithms in the case of discrete-time modeling. An alternative approach can be achieved by defining a proper objective function [23]. The case of consensus in networks affected by switching topology and time-delays is considered in [16] and the average consensus on a time-varying signal in [24]. Most of the consensus algorithms can be classified as distributed estimators as, for example, the work [25] that achieves agreement on a class of functions of the initial state. A distributed state observer is presented in [26], the case of noisy measurement is discussed in [27] and [28].

Formation

Equation (1) turns gently in a proper solution for a formation control by adding an offset $\boldsymbol{d}_{ij}^i \in \mathbb{R}^p$ embedding the desired formation information

$$\dot{\boldsymbol{x}}_i = \sum_{j \in N_i}(\boldsymbol{x}_j - \boldsymbol{x}_i - \boldsymbol{d}_{ij}^i). \tag{4}$$

Formation control can thus be considered as a local control problem; in its basic formulation, in fact, the control objective is the relative coordinate among the agents.

It turns out that it is possible to demonstrate stability of the collective dynamics driven by (4). The graph topology enters into the analysis by means of its matrix counterpart, the Laplacian's spectrum. Since, for connected graphs, there is at least one null eigenvalue, the formation can be guaranteed but not its absolute position unless a more complex control scheme is adopted whose details can be found in [29] and [30].

Formation control might be achieved with a leader in charge of following a proper path; the remaining agents are then coherently defined as followers. To implement a leader-follower strategy it is sufficient to move one single agent labeled as l according to

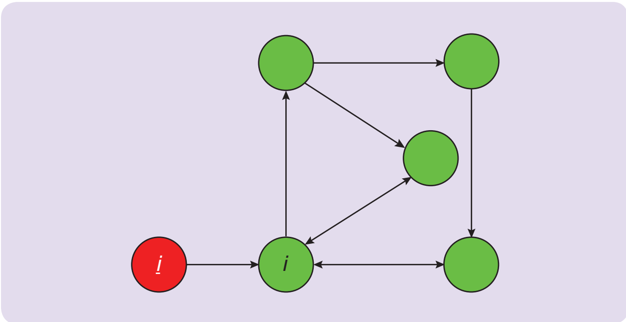

FIGURE 5 An exogenous input is added as a virtual agent, labeled as l in the given example. This trick allows, under mild assumption on the graph's topology, a desired value to all the network's agents to be imposed.

$$\boldsymbol{u}_l = k(\boldsymbol{d}_l - \boldsymbol{x}_l).$$

For convergence purposes the l agent needs to be the root of a rooted directed spanning tree.

In formation control the concept of *virtual agent* is often used around which the single agents are driven. One example is given by [31]

$$\boldsymbol{u}_i = \dot{\boldsymbol{x}}_{i,d} + \alpha_i(\boldsymbol{x}_{i,d} - \boldsymbol{x}_i) + \sum_{j \in N_i} [(\boldsymbol{x}_{j,d} - \boldsymbol{x}_j) - (\boldsymbol{x}_{i,d} - \boldsymbol{x}_i)], \quad (5)$$

where $\boldsymbol{x}_{i,d}$ is the desired position for each of the agents defined by an offset with respect to the virtual reference agent $\boldsymbol{x}_c \in \mathbb{R}^p$. Equation (5) still embeds the problem of the information flow related to the virtual reference agent; continuous-time, first-order dynamics are still an open issue while some solutions arise for second-order dynamics [32], [31].

A bottom-up approach, where each agent is driven according to proper traffic rules based on local information, is proposed in [33]. Scalability, robustness, and convergence issues are also discussed. An alternative approach to formation control is achieved by considering a distributed algorithm that determines a target for each agent without a priori assignment; those are then driven to the corresponding target by common path planning techniques [34], [35], [6].

Various kinds of formation control can be achieved by designing different control objectives related to, e.g., the distance from an obstacle or one or more neighbors. Similarly to the target assignment approach, it is then possible to remove the constraint to assign a priori the agent's position on the graph. Moreover, with a proper formulation of the control objective, it is possible to resort to gradient-based approaches. The distance from a point of one single agent can be controlled by the function $\sigma(\boldsymbol{x}_i) \in \mathbb{R}$

$$\sigma(\boldsymbol{x}_i) = \frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{c})^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{c}),$$

where $\boldsymbol{c} \in \mathbb{R}^p$ represents the obstacle's coordinates. In this case the sole ith agent is concerned by this function, and the structure of the collective Jacobian $\boldsymbol{J} \in \mathbb{R}^{1 \times np}$

$$\boldsymbol{J}(\boldsymbol{x}_i) = [0_p^{\mathrm{T}} \; \cdots \; (\boldsymbol{x}_i - \boldsymbol{c})^{\mathrm{T}} \; \cdots \; 0_p^{\mathrm{T}}],$$

is different from zero only for the components of the ith agent. A slightly different version of this function is achieved when, in the same function, several distances are considered in a implied function $\boldsymbol{\sigma} \in \mathbb{R}^n$

$$\boldsymbol{\sigma}(\boldsymbol{x}) = \frac{1}{2}\begin{bmatrix} \vdots \\ (\boldsymbol{x}_i - \boldsymbol{c})^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{c}) \\ \vdots \end{bmatrix},$$

whose Jacobian exhibits a diagonal structure

$$\boldsymbol{J} = \mathrm{diag}\{(x_1 - c)^T, \ldots, (x_i - c)^T, \ldots, (x_n - c)^T\}.$$

The latter choice couples only apparently the tasks that still can be considered as locals. A local problem still arises when considering the relative distance between two agents in a task function $\sigma \in \mathbb{R}$

$$\sigma = \frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{x}_j)^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{x}_j),$$

whose corresponding Jacobian $\boldsymbol{J} \in \mathbb{R}^{1 \times np}$ is

$$\boldsymbol{J} = [0 \;\; \cdots \;\; (\boldsymbol{x}_i - \boldsymbol{x}_j)^{\mathrm{T}} \;\; \cdots \;\; -(\boldsymbol{x}_i - \boldsymbol{x}_j)^{\mathrm{T}} \;\; \cdots \;\; 0],$$

that couples only the two agents involved. Coupling is obviously achieved if the control objective $\boldsymbol{\sigma} \in \mathbb{R}^n$ is the distance of the agents from their centroid

$$\boldsymbol{\sigma}(\boldsymbol{x}) = \frac{1}{2}\begin{bmatrix} \vdots \\ (\boldsymbol{x}_i - \overline{\boldsymbol{x}})^{\mathrm{T}}(\boldsymbol{x}_i - \overline{\boldsymbol{x}}) \\ \vdots \end{bmatrix},$$

or the sum of all the distances in a one dimensional function $\sigma \in \mathbb{R}$

$$\sigma(\boldsymbol{x}) = \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{x}_i - \overline{\boldsymbol{x}})^{\mathrm{T}}(\boldsymbol{x}_i - \overline{\boldsymbol{x}}).$$

Within this perspective, flocking, the operation to group together the agents, is a specific case of the relative distance task. One possible scalar function σ that, if minimized, represents a flock behavior is given by the deviation energy

$$\sigma(\boldsymbol{x}) = \frac{1}{|E(\boldsymbol{x})| + 1}\sum_{i=1}^{n}\sum_{j \in N_i}(\|\boldsymbol{x}_i - \boldsymbol{x}_j\| - d)^2. \quad (6)$$

Any α-lattice geometry represents a local minimum for such an index [7], depending on the way the set of neighboring agents is computed this value can be at most null. Equation (6) can be rewritten as

$$\sigma(x) = \sum_{i=1}^{n} \sigma_i(N_i),$$

which points out that the flocking problem is structurally decomposed into the sum of nonnegative functions $\sigma_i(N_i)$. Each of those functions $\sigma_i(N_i)$ depends only on the neighbors of the ith agent, and thus the overall problem is naturally distributed. The agents do not need to estimate any global information, and thus they do not have to reach consensus on any variable to flock.

In the literature, it is possible to find the term cohesiveness related to similar functions [8]. The work [7] gives a solution specific to this problem; a stability analysis is also provided. A distributed potential-field-based approach is also presented in [36]. Various distributed bottom-up approaches based on relative distance are discussed in [8], such as the move-away-from-closest-neighbor or the noninterference problem, the worst-case problem, or an alternative solution to the rendezvous.

Coverage

The terms coverage, deployment, optimal placement of resources, and sometimes sampling denote similar problems of distributing the agents over a certain region according to given criteria [6], [9], [37]. Conceptually close to those concepts is also exploration [38] and [39].

A key role is commonly represented by a spatial density function modeling the frequency of random events taking place [40]. Given a convex set S, the agents will be deployed according to the given density function $\Phi \in \mathbb{R}$. On the other hand, the necessity to avoid that they all locate in the most interesting point is achieved by introducing a performance function $f \in \mathbb{R}$. These concepts can be formulated from a mathematical point of view with a scalar function $\sigma(x) \in \mathbb{R}$ defined as

$$\sigma(x) = \int_S \max_i f(\| s - x_i \|) \Phi(s)\, ds.$$

Interesting, this integral index can be rewritten according to local contributions by resorting to the Voronoi partitions. One possible performance function is the norm of the distances leading to a function that, when maximized, represents an optimal deployment [37]

$$\sigma(x) = \frac{1}{2} \sum_{i=1}^{n} \int_{Vor(x_i)} (s - x_i)^{\mathrm{T}} (s - x_i) \Phi(s)\, ds \qquad (7)$$

where $s \in \mathbb{R}^p$ is the generic point of the set S, the set $Vor(x_i)$ is the Voronoi partition associated to the ith agent, and $\Phi(s) \in \mathbb{R}$ is a proper density function. Maximization of (7) is defined as the distortion problem in [8] and [9].

The associated Jacobian $J(x) \in \mathbb{R}^{1 \times np}$ is structurally decoupled, $J(x) = J(N_i)$, and exhibits the structure

$$J(N_i) = \left[\cdots \int_{Vor(x_i)} \Phi(s)(s - x_i)^{\mathrm{T}} ds \ \cdots \right]$$

where the generic term reported is the $(1 \times p)$ element that multiplies $\dot{x}_i$. The partial derivative of (7) needs care [9].

Voronoi Partitions

The Voronoi partitions, also named as tessellations or diagrams, are a subdivisions of a set S characterized by a metric with respect to a finite number of points belonging to the set. Given a set $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ with $\mathbf{x}_i \in \mathbb{R}^p$, the corresponding n Voronoi cells $Vor(x_i)$ are given by

$$Vor(\mathbf{x}_i) = \{\mathbf{s} \in S \ \mid \| \mathbf{s} - \mathbf{x}_i \| \le \| s - x_j \| \text{ for all } j\}.$$

Figure S3 reports the Voronoi decomposition of a rectangular bidimensional set with respect to randomly generated points.

The Voronoi cells are not overlapping. Each point x_i can compute the corresponding cell $Vor(\mathbf{x}_i)$ by knowing its position and the neighboring positions; the Voronoi partition, thus, is computed by a distributed algorithm. Finally, it holds

$$\bigcup_{i=1}^{n} Vor(\mathbf{x}_i) = S \quad \text{and} \quad Vor(\mathbf{x}_i) \bigcap Vor(\mathbf{x}_j) = \emptyset \ \text{ for } i \ne j.$$

The above properties can be exploited to distribute some map-based operation among the agents thanks to the Voronoi partitions. Details on the Voronoi-based theory can be found in [73] or [37]; this field involves applications in several domains of the science from biology to chemistry, geology, hydrodynamics, and medicine.

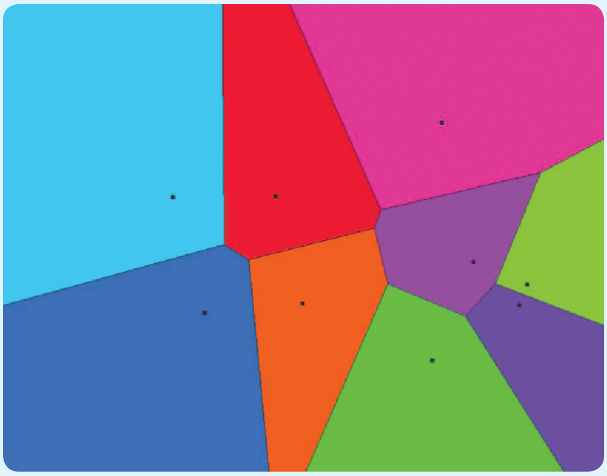

FIGURE S3 Voronoi partitions of a rectangular two-dimensional set. The cells do not overlap, their union returns the original set, and their computation is structurally distributed.

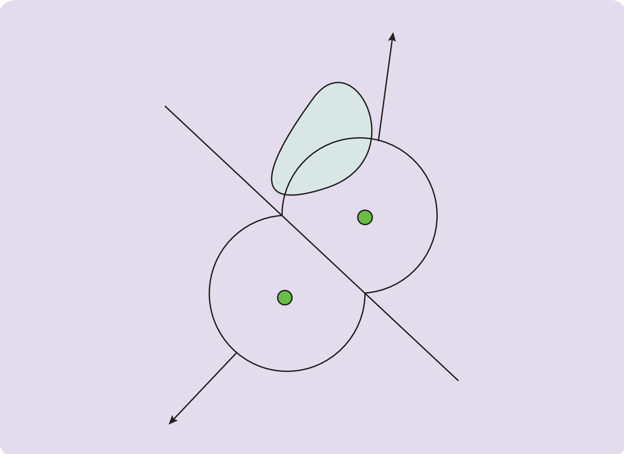

FIGURE 6 Graphical interpretation of the Jacobian for the area problem with two agents. The gray area represents a higher density region that pulls the interest of the motion of the influenced agent toward that region.

To compute the Jacobian, it is necessary to know the absolute agent position, the density function, and the neighboring positions; it is thus possible to implement a distributed gradient-based algorithm. The Jacobian can be rewritten as

$$J(N_i) = [\cdots \quad m(x_i)(c(x_i) - x_i)^{\mathrm{T}} \quad \cdots],$$

where

$$m(x_i) = \int_{Vor(x_i)} \Phi(s)\,ds,$$

$$c(x_i) = \frac{\int_{Vor(x_i)} s\Phi(s)\,ds}{\int_{Vor(x_i)} \Phi(s)\,ds},$$

are the scalar mass and the p-dimensional centroid of the ith Voronoi partition, respectively. A straightforward interpretation is that the stationary points of this function are the centroids of the Voronoi partitions, in such a case the partition is also defined as centroidal Voronoi tessellation [37].

Another deployment problem, known as the area problem, is aimed at maximizing the area covered by the agents eventually weighted by the performance function [8] and [9]. Define

$$1_S(s) = \begin{cases} 1 & \text{if} \quad s \in S \\ 0 & \text{if} \quad s \notin S \end{cases}$$

with a possible implementation of this idea is given by

$$\sigma(x) = \sum_{i=1}^{n} \int_{Vor(x_i)} 1_{B(x_i,r)}((s - x_i)^{\mathrm{T}}(s - x_i))\Phi(s)\,ds. \qquad (8)$$

where $B(x_i, r)$ is the circle centered in x_i with radius r. The corresponding Jacobian, being the function discontinuous, requires a couple of steps [8], [9]; it is given by

$$J(N_i) = \left[\cdots \quad \int_{\mathrm{arc}(Vor(x_i))} n_{B(x_i,r)} \Phi(s)\,ds \quad \cdots\right],$$

where the vector $n_B(x_i, r)$ is the outward unitary vector normal to it and $\mathrm{arc}(Vor(x_i))$ is the union of all arcs in the boundaries of the ith Voronoi partition. This integral represents the weighted normal to the boundary of the ith Voronoi partition. Figure 6 shows a simple planar image with two agents computing their direction of motion: the gray area represents a higher density region that pulls the interest of the motion of the influenced agent toward that region. It is worth remarking that, with (7) and (8) based on an integral measurement, the solution might result in unnatural configurations such as, for example, if the Voronoi cell experiences two symmetric peaks and the agent places itself in the center, that is in a point characterized by limited interest.

Deployment is a largely studied topic [8], [9], [41]. In detail in [41], the unification of different deployment strategies is achieved by designing a proper cost function; by changing a single scalar parameter it is possible to switch among a probabilistic, a Voronoi-based, and a potential field approach. The price to pay is that the gradient-based approach developed is centralized; some hints about a possible distribution are given but their convergence not discussed.

Network's Shape

A class of controlling approaches is designed aimed at controlling the shape of the network. The concept of shape is given by some aggregate task function such as, for example, the centroid or the distribution of the agents around their centroid. In the literature, several algorithms can be found; the work [10] designs a centralized approach aimed at controlling centroid, variance, and orientation of the network, and in [42] and [43] several tasks are defined and controlled in a centralized task-priority framework. Both

Feedback Linearization

The idea of feedback linearization is to find a cancellation term such that the dynamics defined in (S3) appears to be linear in a new control variable [70]. A choice of the kind $\boldsymbol{u}(\mathbf{x}) = \bar{\boldsymbol{J}}(\mathbf{x})\boldsymbol{u}'(\mathbf{x})$ with

$$\boldsymbol{J}(\mathbf{x})\bar{\boldsymbol{J}}(\mathbf{x}) = \boldsymbol{I}_m$$

transforms the nonlinear equation (S3) in

$$\dot{\sigma}(\mathbf{x}) = \boldsymbol{u}'(\mathbf{x}),$$

that, with the choice of $\boldsymbol{u}'(\mathbf{x}) = \dot{\sigma}_d + \Lambda\tilde{\sigma}(\mathbf{x})$, with $\Lambda \in \mathbb{R}^{m\times m}$ positive definite, results in the first-order dynamic system

$$\dot{\tilde{\sigma}}(\mathbf{x}) + \Lambda\tilde{\sigma}(\mathbf{x}) = 0_m$$

where $\tilde{\sigma}$ converges to zero.

works develop a rigorous stability analysis that is flawed by the requirements that the Jacobian does not lose rank during the transient.

In the following, the functions representing the centroid, variance, and orientation are given together with the corresponding Jacobians; those can be used both in gradient-based and feedback linearization approaches (see "Feedback Linearization"). Discussion on the network's moments, or statistics according to a distributed approach [11], [12] follows.

Centroid

The function that represents the average agents' position $\boldsymbol{\sigma}(\boldsymbol{x}) \in \mathbb{R}^p$, in the literature also named average or mean position, is given by

$$\boldsymbol{\sigma}(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i = \bar{\boldsymbol{x}},$$

whose local Jacobian $\boldsymbol{J} \in \mathbb{R}^{p \times pn}$ can be computed as

$$\boldsymbol{J} = \frac{1}{n}[\cdots \quad I_p \quad \cdots].$$

Variance

Another possible variable to control is the variance of all the agents' positions as a synthetic data on their spreading around the centroid. Considering $p=3$, the function for the network variance $\boldsymbol{\sigma} \in \mathbb{R}^p$ is defined as

$$\boldsymbol{\sigma}(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} \begin{bmatrix} (x_{i,1}-\bar{x}_1)^2 \\ (x_{i,2}-\bar{x}_2)^2 \\ (x_{i,3}-\bar{x}_3)^2 \end{bmatrix},$$

whose $(p \times pn)$ Jacobian is

$$\boldsymbol{J}(\boldsymbol{x}) = \frac{2}{n}\begin{bmatrix} \cdots & x_{i,1}-\bar{x}_1 & 0 & 0 & \cdots \\ \cdots & 0 & x_{i,2}-\bar{x}_2 & 0 & \cdots \\ \cdots & 0 & 0 & x_{i,3}-\bar{x}_3 & \cdots \end{bmatrix}.$$

Note that this Jacobian may loose rank when all the agents are aligned along one direction. This situation has a clear kinematic interpretation in the sense that, in such a configuration, it is not possible to arbitrary assign desired velocities in the variance space; see [10] and [42].

Orientation

By defining the vector $\boldsymbol{r}_i \in \mathbb{R}^2$ associated to the ith agent as

$$\boldsymbol{r}_i = \mathcal{R}(\theta)(\boldsymbol{x}_i - \bar{\boldsymbol{x}}),$$

where $\mathcal{R}(\theta) \in \mathbb{R}^{2\times 2}$ is the rotation matrix, which is a function of the orientation angle $\theta \in]-\pi, \pi[$. The value of θ such that

$$\sum_{i=1}^{n} r_{i,1} r_{i,2} = 0$$

is the current network's orientation. In [10], the symbolic expressions of the variable θ, and the corresponding Jacobian, are provided.

Moments

Let us consider a global function $\boldsymbol{\sigma}(\boldsymbol{x}) \in \mathbb{R}^m$ to be controlled that exhibits the structure

$$\boldsymbol{\sigma}(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} g(x_i) = \begin{bmatrix} \sigma_1 \\ \vdots \\ \sigma_m \end{bmatrix}.$$

A centralized approach might consist of a gradient-based controller aimed at minimizing

$$V(\boldsymbol{x}) = (\boldsymbol{\sigma}_d - \boldsymbol{\sigma}(\boldsymbol{x}))^{\mathrm{T}} \boldsymbol{\Gamma} (\boldsymbol{\sigma}_d - \boldsymbol{\sigma}(\boldsymbol{x})),$$

such as

$$\dot{\boldsymbol{x}} = -k\Big[\frac{\partial V}{\partial \boldsymbol{\sigma}}\frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{x}}\Big]^{\mathrm{T}}, \tag{9}$$

where the first partial derivative $\partial V/\partial \boldsymbol{\sigma}$ has dimension $1 \times m$, and it is common to all the agents. Equation (9) can be rewritten as

$$\dot{\boldsymbol{x}} = -k\begin{bmatrix} \boldsymbol{J}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{J}_n^{\mathrm{T}} \end{bmatrix}\begin{bmatrix} \frac{\partial V}{\partial \sigma_1} \\ \vdots \\ \frac{\partial V}{\partial \sigma_m} \end{bmatrix}.$$

In this centralized solution, the single agents are driven according to

$$\dot{\boldsymbol{x}}_i = -k\begin{bmatrix} \frac{\partial \sigma_1}{\partial \boldsymbol{x}_{i,1}} & \cdots & \frac{\partial \sigma_m}{\partial \boldsymbol{x}_{i,1}} \\ \vdots & & \vdots \\ \frac{\partial \sigma_1}{\partial \boldsymbol{x}_{i,p}} & \cdots & \frac{\partial \sigma_m}{\partial \boldsymbol{x}_{i,p}} \end{bmatrix}\begin{bmatrix} \frac{\partial V}{\partial \sigma_1} \\ \vdots \\ \frac{\partial V}{\partial \sigma_m} \end{bmatrix} = -k\boldsymbol{J}_i^{\mathrm{T}}\begin{bmatrix} \frac{\partial V}{\partial \sigma_1} \\ \vdots \\ \frac{\partial V}{\partial \sigma_m} \end{bmatrix},$$

where only the matrix $\boldsymbol{J}_i(\boldsymbol{x}_i) \in \mathbb{R}^{m \times p}$ can be computed locally.

The function $\boldsymbol{\sigma}(\boldsymbol{x})$ is estimated by running n local estimators, one for each agent. Let us define $\hat{\boldsymbol{\sigma}}^i \in \mathbb{R}^m$ as the m-dimensional vector that represents the estimate that the i agent has of the global function $\boldsymbol{\sigma}(\boldsymbol{x})$.The estimator has the form [44]

$$\dot{\boldsymbol{\omega}}_i = -\gamma \boldsymbol{\omega}_i + k_p \sum_{j \in N_i} (\hat{\boldsymbol{\sigma}}^j - \hat{\boldsymbol{\sigma}}^i),$$
$$\hat{\boldsymbol{\sigma}}^i = \boldsymbol{\omega}_i + \boldsymbol{g}(\boldsymbol{x}_i).$$

The convergence properties are discussed in [44], and the estimator converges to the true value if $\gamma = 0$ and the network is connected with fixed topology.

The centralized gradient-based controller may be modified such as each agent feeds back the local estimate of $\boldsymbol{\sigma}$ instead of the, inaccessible, true value, and the gains are modified accordingly [11], [12]

$$\begin{aligned}\dot{\boldsymbol{x}}_i &= -[\boldsymbol{I}_m + \boldsymbol{J}_i^{\mathrm{T}}\boldsymbol{\Lambda}_i\boldsymbol{J}_i]^{-1}\frac{\partial V}{\partial \boldsymbol{x}_i}\Big|_{\boldsymbol{\sigma}=\hat{\boldsymbol{\sigma}}^i} \\ &= -[\boldsymbol{I}_m + \boldsymbol{J}_i^{\mathrm{T}}\boldsymbol{\Lambda}_i\boldsymbol{J}_i]^{-1}\boldsymbol{J}_i^{\mathrm{T}}\boldsymbol{\Gamma}(\boldsymbol{\sigma}_d - \hat{\boldsymbol{\sigma}}_i).\end{aligned}$$

The research field of interconnected dynamic systems is attracting a growing number of researchers from different communities.

The control variable for the planar case study $(p=2)$ is

$$\sigma(x) = \frac{1}{n}\sum_{i=1}^{n}\begin{bmatrix} x_{i,1} \\ x_{i,2} \\ x_{i,1}^2 \\ x_{i,1}x_{i,2} \\ x_{i,2}^2 \end{bmatrix},$$

and each matrix $J_i(x_i) \in \mathbb{R}^{5\times 2}$ assumes the form

$$J_i(x_i) = \frac{1}{n}\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 2x_{i,1} & 0 \\ x_{i,2} & x_{i,1} \\ 0 & 2x_{i,2} \end{bmatrix}.$$

Specific Control Problems

The field of research discussed in this article is not confined within the class of problems reviewed above. In this section, some specific control problems and few corresponding solutions are reported.

In robotics, the patrolling or surveillance application [45]–[48] often embeds some of the analytic problems discussed; several variables need to be taken into account such as the necessity to monitor a perimeter or an area, the possibility of communication, and thus the eventual decentralized architecture, the sensor equipment for the robots, and the battery or fuel capacity. In a sense, the patrolling can be considered as a deployment-like problem with severe implementation constraints. Entrapment, escort, and encirclement are some of the terms used to define the problem of surrounding a target and tracking it. Key aspects are the requirement to stay at a defined distance from the target and eventually minimize its escape window [49]–[51].

Adaptive sampling is the problem of sampling a certain field by selecting the next sampling point on the basis of prior, and eventually shared, information among the agents. In communication theory, the term adaptive is referred to the temporal variable, while for physical agents it has a spatial meaning. This problem too can be considered as a kind of deployment problem; here, however, additional terms in the cost functions might be considered such as, for example, the cost of moving the agents or the time variance of the field to be sampled. A gradient-based strategy is described in [52], and the specific case of marine robotics is handled in [53]–[55].

As an example of a field where networked dynamic systems are gaining importance is given by the smart grid, an electricity network with distributed power generation and absorption; the same agent, here defined as node, can be both an energy supplier and client. The research community is focusing on the possibility to implement real-time operations and distributed decision making for possible monitoring, fault detection, and optimization applications [56], [57].

The work [58] represents one of the first experimental implementations of swarm concepts for a micro-assembly operation achieved by resorting to the collaboration of flagellated bacteria.

A Bioinspired Perspective: A Note on Behavioral Control

The concept of behavioral control has been introduced in [59] and [60], and a detailed discussion is given by [61]. Behavioral approaches can be considered as the most diffused bottom-up approaches, and they are applied to control problems of multiagent systems in, among the others, [62]–[64].

In short, a behavior represents a function connecting the agent sensors to its inputs. This description is to be intended as an abstract meaning, in case of a robot the output of the behavior might be a force/torque signal as well as a velocity reference to a low level controller; for a generic dynamic system it is its input $\boldsymbol{u}_i$. Each behavior can be implemented using the preferred algorithm; for instance, one of the most used approaches is based on the virtual potential concept [65], [36] or by implementing a gradient-based algorithm.

Note that it is of interest to try to handle several behaviors simultaneously, both at the agent or network level, i.e., behaviors concerning a single agent or the collective dynamics. This problem is known as behavioral coordination [61]. A possible solution to solve conflicts among the tasks is provided in [66], where a layered subsumption architecture is adopted. According to the layered control approach [59], the behaviors are seen as competitors, and thus at each time instant only one is selected to be active. A different technique is discussed in [64], based on the behavioral approach introduced in [60]. Following the motor schema control, all the tasks, or behaviors, are summed together through suitable weight coefficients that set the relative priorities among them. Nevertheless, to attenuate a counteracting component of some partial velocity due to a conflicting task, the associated weight has to reduce the overall partial velocity itself. Unproper behavior coordination may lead to undesired side effects [67]. When a behavior concerns the interaction among neighboring agents it is connecting dynamically the agent themselves. Behavior-based approaches, thus, can be properly interpreted as interconnected dynamic systems.

The discussion above leads to the consideration that it is rather complex to design a collective behavior made of agents controlled each by a behavioral approach. Most of the results presented in the literature lying in the framework of bioinspired/biomimetic control are heuristically supported, they exhibit a narrow spectrum, and they are validated in ad-hoc experimental case studies only; within this perspective any generalization is difficult if not impossible. On the other hand, since the applications such as, for example, robotics are progressively moving to unstructured environments or to human interaction all the assessed validation tools became insufficient; the community is currently interrogating itself in finding a shared validation methodology [68].

CONCLUSIONS

The research field of interconnected dynamic systems is attracting a growing number of researchers from different communities. This article addresses control problems for networked agents. In particular, continuous-time, single-integrator dynamics agents have been considered whose interconnection is not part of their model but imposed by the feedback design.

Stability and convergence analysis for such systems are rather complex problems; the research community is called for an effort in removing most of the strong assumptions or debatable conjectures that plague some key results.

The classification between global and local given in the beginning of this article may also be object of discussion. It may happen that a global problem is further reformulated as local and thus implemented more easily in a decentralized architecture. The key point is that it is still an open issue what kind of information is necessary and/or sufficient for a problem to be formulated globally or locally.

Among the additional open challenges, note that the problem of characterizing all the functions whose gradient is spatially distributed with respect to a given graph, the design of a distributed observer-controller scheme and the corresponding classification of the achievable control objectives, the inclusion of realistic communication constraints within the agents and the corresponding effects on the algorithm dynamics, the inclusion of environment-caused constraints, and the extension of the results from undirected to directed graphs.

AUTHOR INFORMATION

Gianluca Antonelli is with the University of Cassino and Southern Lazio in Cassino, Italy. From 2005 to 2009 he was an associate editor of *IEEE Transactions on Robotics*. From 2007 to 2009 he was an editor of *Springer Journal of Intelligent Service Robotics* and, from 2009, regional editor for Europe/Africa of the same journal. From 2008 to 2011 he was chair of the IEEE Robotics and Automation Society (RAS) Technical Committee on Marine Robotics. Since 2011, he is chair of the IEEE RAS Chapter of the IEEE-Italy Section. His research interests include control of underwater robotic systems, force/motion control of robot manipulators, multirobot systems, and identification. He has published 31 international journal papers and more than 80 conference papers. He is author of the book *Underwater Robots* (Springer-Verlag, 2003, 2006) and coauthored the chapter "Underwater Robotics" for the *Handbook of Robotics* (Springer-Verlag, 2008). He was scientifically responsible for the EC-funded STREP Co3AUVs, key person for the IP project ARCAS, and associate editor for the IP project ECHORD. He can be contacted at the Dipartimento di Ingegneria Elettrica e dell'Informazione, Università degli Studi di Cassino e del Lazio Meridionale, via G. Di Biasio 43, 03043 Cassino (FR), Italy.

REFERENCES

[1] V. Kumar, D. Rus, and S. Sukhatme, "Networked robots," in *Springer Handbook Robotics*, B. Siciliano and O. Khatib, Eds. New York: Springer-Verlag, 2008, pp. 943–958.
[2] Y. U. Cao, A. S. Fukunaga, and A. B. Kahng, "Cooperative mobile robotics: Antecedents and directions," in *Robot Colonies Special Issue Autonomous Robots*, vol. 4, R. C. Arkin and G. A. Bekey, Eds. Norwell, MA: Kluwer Academic, 1997, pp. 7–27.
[3] L. E. Parker, "Distributed intelligence: Overview of the field and its application in multi-robot systems," *J. Physical Agents*, vol. 2, no. 1, p. 5, 2008.
[4] L. E. Parker, "Multiple mobile robot systems," in *Springer Handbook Robotics*, B. Siciliano, and O. Khatib, Eds. New York: Springer-Verlag, 2008, pp. 921–941.
[5] V. Gazi and K. Passino, *Swarm Stability and Optimization*. New York: Springer-Verlag, 2010.
[6] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton, NJ: Princeton Univ. Press, 2010.
[7] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. Autom. Control*, vol. 51, no. 3, pp. 401–420, Mar. 2006.
[8] S. Martinez, J. Cortes, and F. Bullo, "Motion coordination with distributed information," *IEEE Control Syst. Mag.*, vol. 27, no. 4, pp. 75–88, Aug. 2007.
[9] F. Bullo, J. Cortes, and S. Martinez, *Distributed Control of Robotic Networks* (Applied Mathematics). Princeton, NJ: Princeton Univ. Press, 2009.
[10] C. Belta and V. K. Kumar, "Abstraction and control of groups of robots," *IEEE Trans. Robot.*, vol. 20, no. 5, pp. 865–875, 2004.
[11] P. Yang, R. A. Freeman, and K. M. Lynch, "A general stability condition for multi-agent coordination by coupled estimation and control," in *Proc. American Control Conf.*, 2007, pp. 723–728.
[12] P. Yang, R. A. Freeman, and K. M. Lynch, "Multi-agent coordination by decentralized estimation and control," *IEEE Trans. Autom. Control*, vol. 53, no. 11, pp. 2480–2496, 2008.
[13] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.
[14] N. A. Lynch, *Distributed Algorithms*. San Rafael, CA: Morgan Kaufmann, 1996.
[15] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Syst. Mag.*, vol. 27, no. 2, pp. 71–82, Apr. 2007.
[16] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, 2004.
[17] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness, *IEEE Trans. Robot.*, vol. 23, no. 4, pp. 693–703, 2007.
[18] M. Zavlanos, M. Egerstedt, and G. Pappas, "Graph theoretic connectivity control of mobile robot networks," *Proc. IEEE*, vol. 99, no. 9, pp. 1525–1540, 2011.
[19] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[20] H. Tanner, A. Jadbabaie, and G. J. Pappas, "Flocking in fixed and switching networks," *IEEE Trans. Autom. Control*, vol. 52, no. 5, pp. 863–868, May 2007.
[21] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Trans Autom. Control*, vol. 50, no. 2, pp. 169–182, 2005.
[22] W. Ren, "Multi-vehicle consensus with a time-varying reference state," *Syst. Control Lett.*, vol. 56, nos. 7–8, pp. 474–483, 2007.
[23] L. Moreau, "Stability of continuous-time distributed consensus algorithms," in *Proc. 43rd IEEE Conf. Decision Control*, 2005, pp. 3998–4003.
[24] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Dynamic consensus on mobile networks," in *Proc. IFAC World Congr.*, 2005.
[25] D. Bauso, L. Giarre, and R. Pesenti, "Non-linear protocols for optimal distributed consensus in networks of dynamic agents," *Syst. Control Lett.*, vol. 55, no. 11, pp. 918–928, 2006.
[26] S. S. Stankovic, M. S. Stankovic, and D. M. Stipanovic, "Consensus based overlapping decentralized estimator," *IEEE Trans. Autom. Control*, vol. 54, no. 2, pp. 410–415, 2009.
[27] D. Bauso, L. Giarre, and R. Pesenti, "Lazy consensus for networks with unknown but bounded disturbances," in *Proc. 46th IEEE Conf. Decision Control*, 2007, pp. 2283–2288.
[28] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, "Distributed kalman filtering based on consensus strategies," *IEEE J. Selected Areas Commun.*, vol. 26, no. 4, pp. 622–633, 2008.
[29] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1465–1476, 2004.
[30] J. A. Fax. (2001). "Optimal and cooperative control of vehicle formations," Ph.D. thesis, California Inst. Technol., Berkeley, CA, [Online]. Available: http://thesis.library.caltech.edu/4230
[31] W. Ren and R. W. Beard, "Distributed consensus in multi-vehicle cooperative control," in *Communication Control Engineering.* Berlin, Germany: Springer-Verlag, 2008.
[32] Y. Hong, G. Chen, and L. Bushnell, "Distributed observers design for leader-following control of multi-agent networks," *Automatica*, vol. 44, no. 3, pp. 846–850, 2008.
[33] L. Pallottino, V. G. Scordio, E. Frazzoli, and A. Bicchi, "Decentralized cooperative policy for conflict resolution in multi-vehicle systems," *IEEE Trans. Robot.*, vol. 23, no. 6, pp. 1170–1183, 2007.
[34] N. Michael, M. M. Zavlanos, V. Kumar, and G. J. Pappas, "Distributed multi-robot task assignment and formation control," in *Proc. IEEE Int. Conf. Robotics Automation*, 2008, pp. 128–133.
[35] S. L. Smith and F. Bullo, "Monotonic target assignment for robotic networks," *IEEE Trans. Autom. Control*, vol. 54, no. 9, pp. 2042–2057, 2009.
[36] N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Proc. 40th IEEE Conf. Decision Control,* Dec. 2001, pp. 2968–2973.
[37] Q. Du, V. Faber, and M. Gunzburger, "Centroidal Voronoi tessellations: Applications and algorithms," *SIAM Review*, vol. 41, no. 4, pp. 637–676, 1999.
[38] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, "Collaborative multi-robot exploration," in *Proc. IEEE Int. Conf. Robotics Automation,* 2000, vol. 1, pp. 476–481.
[39] A. Franchi, L. Freda, G. Oriolo, and M. Vendittelli, "The sensor-based random graph method for cooperative robot exploration," *IEEE/ASME Trans. Mechatron.*, vol. 14, no. 2, pp. 163–175, 2009.
[40] W. Li and C. G. Cassandras, "Distributed cooperative coverage control of sensor networks," in *Proc. 44th IEEE Conf. European Control Decision Control*, 2005, pp. 2542–2547.
[41] M. Schwager, D. Rus, and J. J. Slotine, "Unifying geometric, probabilistic, and potential field approaches to multi-robot deployment," *Int. J. Robot. Res.*, vol. 30, no. 3, pp. 371–383, 2011.
[42] G. Antonelli and S. Chiaverini, "Kinematic control of platoons of autonomous vehicles," *IEEE Trans. Robot.*, vol. 22, no. 61, pp. 285–1292, Dec. 2006.
[43] G. Antonelli, "Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems," *IEEE Trans. Robot.*, vol. 25, no. 5, pp. 985–994, Oct. 2009.
[44] R. A. Freeman, P. Yang, and K. M. Lynch, "Stability and convergence properties of dynamic average consensus estimators," in *Proc. 45th IEEE Conf. Decision Control*, 2006, pp. 338–343.
[45] D. Kingston, R. W. Beard, and R. S. Holt, "Decentralized perimeter surveillance using a team of UAVs," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1394–1404, 2008.
[46] A. Marino, L. Parker, G. Antonelli, and F. Caccavale, "Behavioral control for multi-robot perimeter patrol: A finite state automata approach," in *Proc. 2009 IEEE Int. Conf. Robotics Automation*, May 2009, pp. 831–836.
[47] Y. Guo, L. Parker, and R. Madhavan, "Collaborative robots for infrastructure security applications," *Mobile Robots: Evolutionary Approach*, vol. 50, pp. 185–200, 2007.
[48] H. Nam Chu, A. Glad, O. Simonin, F. Sempe, A. Drogoul, and F. Charpillet, "Swarm approaches for the patrolling problem, information propagation vs. pheromone evaporation," in *Proc. 19th IEEE Int. Conf. Tools Artificial Intelligence*, 2007, vol. 1, pp. 442–449.
[49] G. Antonelli, F. Arrichiello, and S. Chiaverini, "The entrapment/escorting mission: An experimental study using a multirobot system," *IEEE Robot. Autom. Mag. Special Issues Design, Control, Appl. Real-World Multi-Robot Syst.*, vol. 15, no. 1, pp. 22–29, Mar. 2008.
[50] I. Mas, S. Li, J. Acain, and C. Kitts, "Entrapment/escorting and patrolling missions in multi-robot cluster space control," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, 2009, pp. 5855–5861.
[51] A. Franchi, P. Stegagno, M. Di Rocco, and G. Oriolo, "Distributed target localization and encirclement with a multi-robot system," in *Proc. 7th IFAC Symp. Intelligent Autonomous Vehicles*, 2010.
[52] P. Ogren, E. Fiorelli, and N. E. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Trans. Autom. Control*, vol. 49, no. 8, pp. 1292–1302, 2004.
[53] E. Fiorelli, N. E. Leonard, P. Bhatta, D. A. Paley, R. Bachmayer, and D. M. Fratantoni, "Multi-AUV control and adaptive sampling in Monterey Bay," *IEEE J. Oceanic Eng.*, vol. 31, no. 4, pp. 935–948, 2006.
[54] B. Zhang, G. S. Sukhatme, and A. A. Requicha, "Adaptive sampling for marine micro-organism monitoring," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, 2004, vol. 2, pp. 1115–1122.
[55] A. Caiti, A. Munafo, and R. Viviani, "Adaptive on-line planning of environmental sampling missions with a team of cooperating autonomous underwater vehicles," *Int. J. Control*, vol. 80, no. 7, pp. 1151–1168, 2007.
[56] A. Bose, "Smart transmission grid applications and their supporting infrastructure," *IEEE Trans. Smart Grid*, vol. 1, no. 1, pp. 11–19, 2010.
[57] A. A. Aquino-Lugo, R. Klump, and T. J. Overbye, "A control framework for the smart grid for voltage support using agent-based technologies," *IEEE Trans. Smart Grid*, vol. 1, no. 3, pp. 357–363, 2010.
[58] S. Martel and M. Mohammadi, "Using a swarm of self-propelled natural microrobots in the form of flagellated bacteria to perform complex micro-assembly tasks," in *Proc. 2010 IEEE Int. Conf. Robotics Automation*, May 2010, pp. 500–505.
[59] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robot. Autom.*, vol. 2, no. 1, pp. 14–23, 1986.
[60] R. C. Arkin, "Motor schema based mobile robot navigation," *Int. J. Robot. Res.*, vol. 8, no. 4, pp. 92–112, 1989.
[61] R. C. Arkin, *Behavior-Based Robotics*. Cambridge, MA: MIT Press, 1998.
[62] L. E. Parker, "On the design of behavior-based multi-robot teams," *Advanced Robot.*, vol. 10, no. 6, pp. 547–578, 1996.
[63] M. J. Mataric, "Behavior-based control: Examples from navigation, learning, and group behavior," *J. Experim. Theoretical Artificial Intell.*, vol. 9, nos. 2–3, pp. 323–336, 1997.
[64] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Trans. Robot. Autom.*, vol. 14, no. 6, pp. 926–939, 1998.
[65] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Trans. Syst., Man, Cybern.*, vol. 19, no. 5, pp. 1179–1187, 1989.
[66] L. E. Parker, "Designing control laws for cooperative agent teams," in *Proc. 1993 IEEE Int. Conf. Robotics Automation*, vol. 3, pp. 582–587.
[67] G. Antonelli, F. Arrichiello, and S. Chiaverini, "The null-space-based behavioral control for autonomous robotic systems," *J. Intell. Service Robot.*, vol. 1, no. 1, pp. 27–39, Jan. 2008.
[68] R. Madhavan, A. P. del Pobil, and E. Messina, "Performance evaluation and bench-marking of robotic and automation systems [TC Spotlight]," *IEEE Robot. Autom. Mag.*, vol. 17, no. 1, pp. 120–122, 2010.
[69] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ Press, 1990.
[70] H. K. Khalil, *Nonlinear Systems*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1996.
[71] R. Diestel, *Graph Theory*. New York: Springer-Verlag, 1997.
[72] C. Godsil and G. Royle, *Algebraic Graph Theory* (Graduate Texts Mathematics). New York: Springer-Verlag, 2001.
[73] A. Okabe, B. Boots, and K. Sugihara, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Hoboken, NJ: Wiley, 1992.