SICE Journal of Control, Measurement, and System Integration, Vol. 10, No. 6, pp. 001–009, November 2017

Coordinated Control of Multi-Robot Systems: A Survey

Jorge CORTÉS * and Magnus EGERSTEDT **

Abstract: Recently, significant gains have been made in our understanding of multi-robot systems, and such systems have been deployed in domains as diverse as precision agriculture, flexible manufacturing, environmental monitoring, search-and-rescue operations, and even swarming robotic toys. What has enabled these developments is a combination of technological advances in performance, price, and scale of the platforms themselves, and a new understanding of how the robots should be organized algorithmically. In this paper, we focus on the latter of these advances, with particular emphasis on decentralized control and coordination strategies as they pertain to multi-robot systems. The paper discusses a class of problems related to the assembly of preferable geometric shapes in a decentralized manner through the formulation of descent-based algorithms defined with respect to team-level performance costs.

Key Words: multi-robot systems, decentralized control, mobile sensor networks.

1. Introduction

During the last decade, multi-robot systems have gone from isolated and anecdotal laboratory systems to robustly deployed across a number of domains, such as warehousing [14],[18],[54], precision agriculture [3],[30], search-and-rescue [4],[29], and environmental monitoring and exploration [17],[19],[53]. The fundamental reason why more robots are preferable in these types of domains is that there is strength in numbers. By using a large number of robots, redundancy is automatically built into the system – if one robot fails, there are still a number of operational robots left to continue the mission. A wider spatial area can also be covered more efficiently if more robots are deployed, and heterogeneous capabilities can be distributed across the team without having to dramatically change the payload (and thus price) of individual robots, e.g.,[32].

In response to these technology and application drivers, a number of different control and coordination strategies have been proposed for organizing the robots in order to enable them to come together and solve team-level, global tasks using local interaction rules. This paper discusses some of these developments and identifies some of the salient features common to a number of proposed coordination strategies. Broadly speaking, a distributed multi-robot coordination algorithm has to satisfy four different constraints for it to be useful, namely it must be (i) *local* in the sense that individual robots can only act on information it has available to it, i.e., through sensing or active communications – this is sometimes referred to as "distributed"; (ii) *scalable* in that the algorithms executed by the individual robots cannot depend on the size of the entire team – sometimes referred to as "decentralized"; (iii) *safe* – as robots are physical agents deployed in the real world, they must be safe both relative to collisions with each other and relative to the environment; and (iv) *emergent* in the sense that global properties should emerge from the local interaction rules – preferably in a provable manner [6],[40],[46].

A number of algorithms that satisfy these four constraints have been proposed and they have been used successfully for achieving and maintaining formations [28],[49],[51],[67], for covering areas [9],[11],[42], for securing and tracking boundary curves [47],[61],[69], or for mimicking biological social behaviors such as flocking and swarming [12],[13],[24],[59]. In this overview, we illustrate some of the common features and unifying assumptions behind this body of work.

To make matters more concrete, consider a collection of N robots, with positions $x_i \in \mathbb{R}^p$, $i = 1, \dots, N$, with $p = 2$ in the case of planar robots and $p = 3$ for aerial robots. These robots could for instance be equipped with omni-directional range-sensors, which enable them to measure the positions of near-by robots relative to their own positions, i.e., Robot i can measure the value $x_j - x_i$ if Robot j is within range of Robot i's sensors. The flow of information through sensor measurements or over communication channels can be encoded abstractly as an adjacency relation between robots, which allows us to define the graph structure induced by the multi-robot team as $G = (V, E)$, where $V = \{1, \dots, N\}$ is the vertex set associated with the individual robots, and $E \subset V \times V$ is the edge-set that encodes the adjacency relationships, i.e., $(j, i) \in E$ if Robot j is within range of Robot i's sensors. For the purpose of this paper, we assume that the team is homogeneous, i.e., they have the same kind of sensors and/or communication modalities, which means that $(i, j) \in E$ if and only if $(j, i) \in E$, and we say that the resulting graph is undirected. Note, however, that as the robots move around, adjacency relationships will come and go, i.e., we cannot assume that the edge set E is static over time.

In this paper, we describe a number of control and coordination strategies that adheres to this basic interaction model and that have a common starting point in that the algorithms can be viewed as descent algorithms defined relative to performance costs. The overarching tasks are encoded through these costs – may they be locational costs for describing how well a team of mobile robots are covering an area of interest or energy-like functions describing the pairwise mismatch between robots as

* Department of Mechanical and Aerospace Engineering, University of California, San Diego, USA
** Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, USA
E-mail: cortes@ucsd.edu,magnus@gatech.edu
(Received July 22, 2017)

compared to their desired geometric configuration. In particular, Section 2 addresses the problem of making teams of robots assemble desired geometric formations, Section 3 discusses the coverage control problem, while Section 4 describes how to go from abstract algorithms defined over simplified robot models and interaction modalities to deployment on real robotic systems on the ground as well as in the air. Section 5 concludes the paper with a general description of the optimization-based approach employed in our discussion for the design of coordination control laws.

2. Formation Control

2.1 Reaching Decentralized Agreement

To ensure that the algorithms are indeed local, we insist on the robots acting solely based on the measurable information, e.g., the relative displacements, in such a way that appropriate global objectives are achieved. One such objective could be for the robots to meet at a common location – the robots do not necessarily know where they are so no *a priori* agreed upon meeting location can be used. This is known as the *rendezvous* problem [1],[10],[39], and one way of evaluating how well the robots are doing towards this aim is to evaluate the total error

$$\mathcal{E}(x) = \frac{1}{2}\sum_{i=1}^{N}\sum_{(j,i)\in E}\|x_i - x_j\|^2.$$

We note that the gradient of the total error with respect to the individual robot positions is given by

$$\frac{\partial \mathcal{E}(x)}{\partial x_i} = \sum_{(j,i)\in E}(x_i - x_j), \quad i = 1,\ldots,N.$$

One direct way to minimize such an error function is to use a gradient descent flow, i.e., we could simply let the robots move in the direction of the negative gradient of the total error,

$$\dot{x}_i = -\sum_{(j,i)\in E}(x_i - x_j).$$

The resulting equation is referred to as the node-level dynamics of the system since it describes the movements of the individual robots. However, in order to analyze the behavior of the global system, we need the ensemble-level dynamics. To this end, we first note that the node-level dynamics is linear. Moreover, it only involves the relative position differences between adjacent robots, i.e., it encodes the graph structure itself. To make this explicit, let $x = (x_1^T,\ldots,x_N^T)^T \in \mathbb{R}^{pN}$. If the robots are scalars, i.e., $p = 1$, then the previous node-level dynamics can be written in ensemble form as

$$\dot{x} = -Lx,$$

where L is the (possibly time-varying) $N\times N$-dimensional graph Laplacian [23] associated with the multi-robot network, given by

$$L = [\ell_{ij}]_{i,j=1}^{N}, \quad \ell_{ij} = \begin{cases} \deg(i) & \text{if } i = j, \\ -1 & \text{if } (i,j)\in E, \\ 0 & \text{otherwise,} \end{cases}$$

where the degree, $\deg(i)$, is the number of vertices that are adjacent to vertex i. Moreover, if $p > 1$, the resulting ensemble-level dynamics becomes

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

where I_p is the $p\times p$ identity matrix and $\otimes$ denotes the Kronecker product.

The reason for starting the discussion with the rendezvous problem is three-fold. First, it hints at a systematic way of obtaining decentralized multi-robot control laws by starting with an error function and then producing robot motions that explicitly reduce this error. This will be generalized in subsequent sections together with an introduction of some of the key tools needed to analyze such systems. Second, it calls out the sometimes intricate coupling between robot motions and the evolution of the underlying network structure. In other words, what makes distributed multi-robot control tricky is that it is not enough to consider the individual motions. Instead the motions must be understood in conjunction with their effects on the underlying network structure. In fact, this is a pervasive feature in the literature, see e.g.[6],[45],[46],[57]. Third, the equation $\dot{x} = -Lx$ is, by itself, one of the most important equations in the multi-agent literature. In fact, it plays such a prominent role that it even has its own name – the *consensus equation* [5],[21],[52],[58]. The reason for this is that by moving around, the robots are agreeing (or reaching consensus) on where to meet, as shown in Figure 1(a).

Numerous variations to the consensus equation have been proposed, and we here discuss two such direct extensions. For example, let $p = 2$, i.e., the robots are planar, and assume that they are arranged in a directed cycle topology. Then, one can, instead of letting the robots "aim" towards their neighbors, they can instead move with a slight offset, as

$$\begin{aligned} \dot{x}_i &= R(-\psi)(x_{i+1} - x_i), \quad i = 1,\ldots,N-1 \\ \dot{x}_N &= R(-\psi)(x_1 - x_N), \end{aligned}$$

where $R(-\psi)$ is the rotation matrix of angle $-\psi$. Now, if the offset angle is $\psi = \pi/N$, a perfect circular motion is asymptotically achieved – so-called *cyclic pursuit* – while if $\psi < \pi/N$ the robots will spiral inwards towards a consensus point, and if $\psi > \pi/N$, they will spiral outwards, away from each other [41]. This is illustrated in Figure 1(b).

Finally, if instead of reaching agreements over the positions, the robots agree on what direction they should move in, i.e., the consensus equation operates on the robot headings instead of their positions, $\dot{\phi} = -L\phi$, where ϕ_i is the heading of Robot i, then the so-called *flocking* behavior emerges [27],[51],[62], as seen in Figure 1(c).

In the context of the four constraints imposed on multi-robot coordination algorithms, the consensus equation is both local (only involves the measurably information $x_i - x_j$) and scalable (only involves the neighborhood sets N_i as opposed to the full set of robots). It is moreover emergent, as it has been shown that it will indeed drive all robots to a common position as long as the underlying information exchange network is "rich enough". In the static and undirected case, the necessary and sufficient condition is that the graph is connected, i.e., that there exists a path through the graph from every pair of vertices. In the static and directed case, consensus is achieved if and only if the graph contains, as a subgraph, a spanning (all vertices are present) out-branching tree (a tree graph where the edges all point in the same direction – away from the root node). In the dynamic case, the conditions are slightly more involved but a

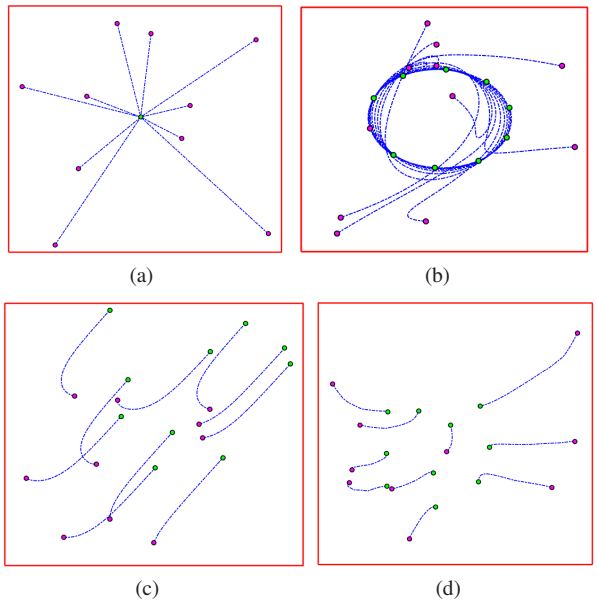


Fig. 1 Variations on the consensus equation: (a) Rendezvous, (b) Cyclic Pursuit, (c) Flocking, and (d) Formation Control. In these figures, the robots start at the red locations and arrive at the green locations after a a certain amount of time.

sufficient condition is that the necessary and sufficient conditions for the static case holds at each point in time, regardless of whether the graph is undirected or directed. (The purpose of this survey is not to cover all of the intricacies of the stability analysis of the consensus equation – for derivations and full characterizations of these results, see for example [46] and the references therein.) But, the consensus equation is not safe! In fact, rendezvous is by design achieving a massive collision among all the robots. To remedy this and turn the consensus equation into a truly useful multi-robot coordination law, we need to augment it to ensure that the robots do not get too close to each other.

2.2 Weighted Protocols

The construction in Section 2.1 can be generalized by defining a symmetric, pairwise performance cost between robots i and j as $\mathcal{E}_{ij}(\|x_i - x_j\|) = \mathcal{E}_{ji}(\|x_j - x_i\|)$, with the global performance cost being defined by

$$\mathcal{E}(x) = \sum_{i=1}^{N} \sum_{(j,i)\in E} \mathcal{E}_{ij}(\|x_i - x_j\|).$$

The Chain Rule tells us that

$$\begin{aligned}\frac{\partial \mathcal{E}_{ij}(\|x_i - x_j\|)}{\partial x_i} &= \frac{\partial \mathcal{E}_{ij}(\|x_i - x_j\|)}{\partial \|x_i - x_j\|} \frac{(x_i - x_j)}{\|x_i - x_j\|} \\ &= w_{ij}(\|x_i - x_j\|)(x_i - x_j),\end{aligned}$$

i.e., the partial derivative is a scalar function of the inter-robot distance times the relative displacement. As such, the gradient descent rule is given by a weighted consensus protocol,

$$\dot{x}_i = -\frac{\partial \mathcal{E}}{\partial x_i} = -\sum_{(j,i)\in E} w_{ij}(\|x_i - x_j\|)(x_i - x_j).$$

The reason why this construction is systematic and theoretically justified is that if we restrict $\mathcal{E}$ to positive semi-definite functions that are 0 only at the desired, global configuration, we note that

$$\frac{d\mathcal{E}}{dt} = \frac{\partial \mathcal{E}}{\partial x}\dot{x} = \sum_{i=1}^{N} \frac{\partial \mathcal{E}}{\partial x_i}\dot{x}_i = -\left\|\frac{\partial \mathcal{E}}{\partial x}\right\|^2.$$

In other words, $\mathcal{E}$ is a Lyapunov function and, with bounded trajectories, one can resort to the LaSalle Invariance Principle [31] to ensure that the desired configuration is at least a locally asymptotically stable equilibrium point as long as the edge set E does not change. If E does change, i.e., edges come and go, then $\mathcal{E}$ will experience discontinuities, and either a hybrid version of the LaSalle Invariance Principle must be used, or arguments must be employed that establish that sooner or later, the edge set becomes static, see e.g.,[16],[26],[48].

A number of examples of this construction have been discussed in the literature. First, the standard consensus equation covered above can be derived from

$$\mathcal{E}_{ij}(\|x_i - x_j\|) = \frac{1}{2}\|x_i - x_j\|^2 \;\Rightarrow\; w_{ij} = 1.$$

If the error is just the norm, as opposed to the square of the norm, then

$$\mathcal{E}_{ij}(\|x_i - x_j\|) = \|x_i - x_j\| \;\Rightarrow\; w_{ij} = \frac{1}{\|x_i - x_j\|},$$

which is a form that has been used in [12] to describe coordinated behaviors among schooling fish. The interpretation here is that, as fish pay more attention to near-by fish, the square norm counter-acts this by penalizing far-away fish in an overly aggressive manner.

If the robots are supposed to arrange themselves at a prescribed inter-robot distance δ, we obtain a *formation control* protocol,[15],[20],[36],[38],[49],[69], as opposed to a rendezvous protocol. An example of this found in [46] is given by

$$\mathcal{E}_{ij}(\|x_i - x_j\|) = \frac{1}{2}(\|x_i - x_j\| - \delta)^2 \;\Rightarrow\; w_{ij} = \frac{\|x_i - x_j\| - \delta}{\|x_i - x_j\|}.$$

The interpretation here is that the weight is negative if the robots are closer than δ apart, thereby repelling away from each other, while agents that are further than δ apart are attracted through the corresponding positive weight.

An additional complication associated with multi-robot networks is that, throughout the maneuvers, the robot network should stay connected,[28],[60],[66],[68]. One way of ensuring this *connectivity maintenance* property is to ensure that the weights become sufficiently large as the inter-robot distance approaches Δ, which is the distance where the robots are no longer able to sense each other. In [28], the following choices were shown to guarantee connectivity maintenance

$$\mathcal{E}_{ij}(\|x_i - x_j\|) = \frac{\|x_i - x_j\|^2}{\Delta - \|x_i - x_j\|} \;\Rightarrow\; w_{ij} = \frac{2\Delta - \|x_i - x_j\|}{(\Delta - \|x_i - x_j\|)^2}.$$

A combined formation control and connectivity maintenance protocol could thus become

$$\begin{aligned}\mathcal{E}_{ij}(\|x_i - x_j\|) &= \frac{1}{2(\Delta - \delta)}\left(\frac{\|x_i - x_j\| - \delta}{\Delta - \|x_i - x_j\|}\right)^2 \\ &\Rightarrow\; w_{ij} = \frac{1 - \frac{\delta}{\|x_i - x_j\|}}{(\Delta - \|x_i - x_j\|)^3},\end{aligned}$$

as seen in Figure 1(d).

What all of these constructions show is that it is possible to achieve rich and diverse multi-robot responses through a systematic selection of scalar weights in the consensus equation. But, if the objective is not to assemble a particular shape, but rather to spread the robots out to cover an area, modifications to this construction are needed.

3. Coverage Control

Another example of the theme of formulating an error function and then flowing in a negative gradient direction involves the problem of having the robots cover a planar area in an optimal way [6],[11]. To this end, let the agents be tasked with covering an area $\mathcal{D}$, and let each robot be in charge of all the points in $\mathcal{D}$ that are closest to it. This corresponds to partitioning $\mathcal{D}$ into N Voronoi cells, with the robot locations x_i, $i = 1, \ldots, N$ as seeds,

$$\mathcal{V}_i(x) = \{p \in \mathcal{D} \mid \|x_i - p\| \leq \|x_j - p\|, \ \forall j \neq i\}.$$

Note that here we use the Euclidean distance to define what it means to be "closest". The concept of Voronoi partition is flexible enough to allow for other notions of distance that can be used to capture robot capabilities such as limited energy [33], different sensing ranges or footprints [35], or motion constraints [2],[34].

3.1 Lloyd's Algorithm and Locational Costs

If we moreover assume that points closer to Robot i are covered more effectively than point further away, we can write down a so-called locational cost associated with the robot positions as

$$\mathcal{E}(x) = \sum_{i=1}^{N} \int_{\mathcal{V}_i(x)} \|x_i - p\|^2 \varphi(p) dp.$$

Here, the function $\varphi : D \to \mathbb{R}$ measures the relative importance of points in the environment, i.e., if $\varphi(p) > \varphi(q)$, then the point p is more important than the point q for the robot ensemble. As before, taking the partial derivative of this locational cost gives

$$\frac{\partial \mathcal{E}}{\partial x_i} = 2 \int_{\mathcal{V}_i(x)} (x_i - p) \varphi(p) dp.$$

The reason why the application of Leibniz rule at the area over which the integral is evaluated does not seem to matter is because whatever area is moved into $\mathcal{V}_i$ by the infinitesimal movement of x_i, exactly the same area is lost in some other cell, i.e., the effects cancel out.

Following the program laid out in the previous paragraphs of using a gradient descent flow as a way of enabling LaSalle Invariance Principle to be applicable, gives us

$$\dot{x}_i = 2 \int_{\mathcal{V}_i(x)} (p - x_i) \varphi(p) dp = 2 m_i(x)(x_i - \rho_i(x)),$$

where $m_i(x) = \int_{\mathcal{V}_i(x)} \varphi(p) dp$ and $\rho_i(x)$ are, respectively, the mass and center of mass of the i:th Voronoi cell.

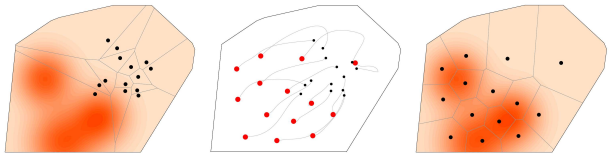
One can also scale the control action by a positive gain, and, as such, consider a scaled descent flow. Using a particular choice of gain, the new flow is given by

$$\frac{1}{2m_i(x)} \frac{\partial \mathcal{E}}{\partial x_i} = x_i - \rho_i(x).$$

The resulting control law is a continuous-time version of Lloyd's Algorithm for *coverage control*,

$$\dot{x}_i = \rho_i(x) - x_i.$$

This law reaches [11] asymptotically the set of so-called Centroidal Voronoi Tessellations, whereby $x_i = \rho_i(x)$, $i = 1, \ldots, N$. An example of this algorithm in action is shown in Figure 2.



(a) (b) (c)

Fig. 2 Gradient-based coverage control: (a) Initial configuration, (b) Evolution of the ensemble, and (c) Final Configuration. Points in the environment are colored according to their importance.

The simplicity of this coordination law makes it especially appealing and, in fact, numerous extensions have been investigated. We discuss below two of these extensions to time-varying scenarios and problems that involve specifications in terms of equitable partitions of the overall load among the robots. The common denominator of these extensions is that their synthesis follows the optimization-based approach to design of coordination laws that we advocate in this survey.

3.2 Time-Varying Locational Optimization and Generalized Voronoi Partitions

Consider the scenario where the function φ measuring the relative importance of points in the environment changes with time [11],[37], e.g., according to the preferences specified by a human operator. Formally, we have $\varphi : D \times \mathbb{R} \to \mathbb{R}$, $(p, t) \mapsto \varphi(p, t)$. Based on the discussion above, robots should seek to achieve a centroidal Voronoi configuration. One can formalize this by writing the error function

$$\mathcal{E}(x, t) = \frac{1}{2} \sum_{i=1}^{N} \|x_i - \rho_i(x, t)\|^2$$

Note that the evolution of this function along the robot trajectories can be written as

$$\frac{d}{dt}(\mathcal{E}(x,t)) =$$
$$\sum_{i=1}^{N} (x_i - \rho_i(x,t))(\dot{x}_i - \frac{\partial \rho_i}{\partial t}(x,t) - \sum_{j=1}^{N} \frac{\partial \rho_i}{\partial x_j}(x,t)\dot{x}_j) =$$
$$(x - \rho(x,t))\Big((I_N - \frac{\partial \rho}{\partial x}(x,t))\dot{x} - \frac{\partial \rho}{\partial t}(x,t)\Big),$$

where, for simplicity, we use the short-hand notation $\rho(x, t) = (\rho_1(x, t), \ldots, \rho_N(x, t))$. This computation reveals how the design of the coordination strategy above should be modified to deal with time-varying functions φ. Specifically, if one sets

$$\Big((I_N - \frac{\partial \rho}{\partial x}(x,t))\dot{x} - \frac{\partial \rho}{\partial t}(x,t)\Big) = k(\rho(x) - x)$$

then the evolution of the error function $\mathcal{E}$ takes the form

$$\frac{d}{dt}(\mathcal{E}(x(t),t)) = -(x - \rho(x,t))k(x - \rho(x,t)) = -2k\mathcal{E}(x(t),t),$$

and hence $\mathcal{E}(x(t),t) = \mathcal{E}(x(0),0)\exp(-2kt)$, guaranteeing exponential convergence. The implementation of this design, however, is challenging from a distributed viewpoint, because it requires the inversion of the matrix $(I_N - \frac{\partial \rho}{\partial x}(x,t))$ to compute the robots' motion. The matrix is sparse, but its inversion is not. One can tackle this, for instance, by approximating the inverse matrix with the Taylor series expansion which, as the matrix sparse, is amenable to distributed implementation [37]. An example of this approach is shown in Figure 3 (bottom row), where a team of robots execute the dynamic coverage control algorithm.

A different take on the locational problem discussed in Section 3.1 is to consider the optimization by the robots of the locational cost subject to fair partitioning of the areas of the environment tasked to each of them [8],[55],[56]. For instance, in case of heterogeneous robots, some with more mobility than others, the fast robots may be tasked with larger regions than the slow ones. For the case of homogeneous teams, it makes sense to prescribe an equitable partition among the robots, where the mass of each region is the same for all. Interestingly enough, this is not guaranteed by the centroidal Voronoi configurations achieved by the Lloyd's algorithm. To illustrate how to deal with this, we consider the latter case. Formally, the robots seek to solve a constrained optimization problem with objective

$$\mathcal{E}(x) = \sum_{i=1}^{N} \int_{\mathcal{V}_i(x)} \|x_i - p\|^2 \varphi(p) dp$$

and constraints

$$m_i(x) = m_j(x), \quad \forall i,j \in \{1,\dots,N\}$$

Remarkably, one can show that the optimal way of partitioning the environment to solve this problem also corresponds to a Voronoi partition, albeit the metric employed to construct is different from the Euclidean one. Specifically, consider partitioning $\mathcal{D}$ into N cells, with the robot locations $x_i,\ i = 1,\dots,N$ and some weights $w_i,\ i = 1,\dots,N$ as seeds according to

$$\mathcal{V}_i(x,w) = \{p \in \mathcal{D} \mid \|x_i - p\|^2 - w_i \le \|x_j - p\|^2 - w_j,\ \forall j \neq i\}.$$

The interpretation of the role of the weights is clear. The larger the weight w_i is with respect to the other weights, the bigger that the region of Robot i gets. This Voronoi partition is called the power diagram. More exotic examples of generalized Voronoi partitions exist, e.g.,[50], and these notions play an important role when considering more general versions of the locational cost defined above. The key observation here is that, given arbitrary robot positions, one can always select the weights $w_i,\ i = 1,\dots,N$ so that the equitable constraints $m_i(x) = m_j(x)$, for all $i,j \in \{1,\dots,N\}$ are satisfied. Therefore, the design of the coordination law to solve the constrained locational optimization problem described above with equitable partitioning consists of two interconnected algorithms: one strategy prescribing the physical motion of the robots by having them chase the centroid of their (power diagram) Voronoi cell and another strategy prescribing how the weights are selected to partition the environment equitably given the robot positions.

4. Deployment on Real Robots

A crucial assumption underlying the discussion in the previous sections was the assumption that we could directly control the velocities of the robots, i.e., that

$$\dot{x}_i = u_i,\ i = 1,\dots,N$$

regardless of the dimension of the state of the system. This is obviously not true for real robots, and, to this end, we need to be able to go from integrators to full-blown robot kinematics in order to actually deploy these control laws. The standard manner in which this is done is to use the velocities resulting from the coordinated control algorithms as "plans" and then wrap nonlinear controllers around these plans in order to deploy them on real robotic systems. Rather than characterizing all the different ways in which this has been done on a large number of different types of platforms, we illustrate here this on two standard classes of robots, namely wheeled, differential drive robots and aerial quadcopters. The common denominator for both robots is the use of the concept of differential flatness [22],[63] to carry out the control design.

Before we start discussing the different platforms, a few words must be made about the notation used. First, as we will focus exclusively on the kinematics of an individual robot in this section, we will suppress the subscript i for the sake of notational convenience. Second, if we let the position of the robot be in $\mathbb{R}^p$, with $p = 2$ in the case of ground robots and $p = 3$ in the case of aerial robots, we will let $u \in \mathbb{R}^p$ denote the desired velocity obtained from the single integrator model, based (perhaps) on one of the previously discussed coordinated controllers. We will subsequently use (x,y) or (x,y,z) to denote the position of the robot, and let ϕ be the heading (or yaw in the aerial case) of the robot, θ be the pitch and ψ be the roll in the aerial case.

4.1 Ground Robots

One of the most commonly used robotic platforms is the wheeled, differential-drive ground robot. It is equipped with two independently controlled wheels of radius R, where the control inputs are the angular velocities of the right ω_r and left ω_l wheels. If the wheel axis has length L, then the kinematics of the differential drive robot is

$$\begin{aligned} \dot{x} &= \tfrac{R}{2}(\omega_r + \omega_l)\cos\phi \\ \dot{y} &= \tfrac{R}{2}(\omega_r + \omega_l)\sin\phi \\ \dot{\phi} &= \tfrac{R}{L}(\omega_r - \omega_l). \end{aligned}$$

Now, as it is not particularly natural to define motions in terms of wheel velocities, a standard transformation is to map this to a unicycle model, where the control inputs are instead given by the translational v and angular ω velocities of the robot. As the unicycle dynamics is given by

$$\begin{aligned} \dot{x} &= v\cos\phi \\ \dot{y} &= v\sin\phi \\ \dot{\phi} &= \omega \end{aligned}$$

we somehow have to map (v,ω) onto (ω_r,ω_l). By equating $(\dot{x},\dot{y},\dot{\phi})$ in the two expressions above, we get

$$\begin{aligned} v &= \tfrac{R}{2}(\omega_r + \omega_l) \\ \omega &= \tfrac{R}{L}(\omega_r - \omega_l) \end{aligned}$$

which inverts to

$$\begin{aligned} \omega_r &= \tfrac{2v+\omega L}{2R} \\ \omega_l &= \tfrac{2v-\omega L}{2R}. \end{aligned}$$

As a result, we can design controllers for the unicycle model and then simply generate (ω_r, ω_l) from the velocities (v, ω). But we still have to go from $u \in \mathbb{R}^2$ to (v, ω). And, if we are willing to ignore the orientation of the vehicle, this can be achieved by considering a point off the wheel axis of the robot by a distance $0 \neq \ell \in \mathbb{R}$,

$$\tilde{x} = x + \ell \cos\phi$$
$$\tilde{y} = y + \ell \sin\phi.$$

In fact, these variables work as flat outputs. Their derivative is given by

$$\dot{\tilde{x}} = v\cos\phi - \ell\omega\sin\phi$$
$$\dot{\tilde{y}} = v\sin\phi + \ell\omega\cos\phi.$$

If we now postulate that $(\dot{\tilde{x}}, \dot{\tilde{y}})^T = u$ we can invert this expression to get (v, ω) in terms of u as

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{\ell} \end{bmatrix} R(-\phi)u,$$

where R is the rotation matrix.

A series of different examples of employing this method for going from $u \in \mathbb{R}^2$ via (v, ω) to (ω_r, ω_l) are shown in Figure 3 for a number of the coordinated controllers discussed in this paper.

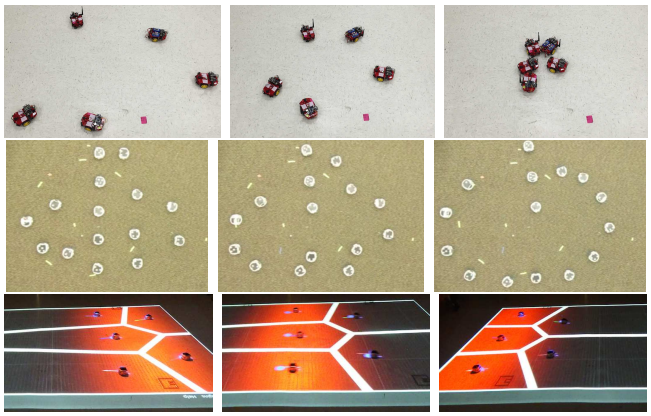


Fig. 3 Top: Five robots are solving the rendezvous problem. Middle: 15 robots are forming a "G" by executing an energy-based formation control strategy resulting in a weighted consensus equation. Bottom: The robots move to cover a time-varying density function by minimizing the locational cost.

4.2 Aerial Robots

For quadcopters, the situation gets slightly more involved due to the complexity of the dynamics. In fact, under the standard assumption that the body rotational rates of the quadcopter are directly controllable through the fast response of an onboard controller, the state of the system becomes $q = [x, y, z, \dot{x}, \dot{y}, \dot{z}, \phi, \theta, \psi]^T$. Moreover, by controlling the angular velocities of the four rotors (analogous to the wheel velocities in the differential-drive robot case), these input velocities can be mapped bijectively to the body rotational rate $\omega = [\omega_x, \omega_y, \omega_z]^T$ and the thrust τ, resulting in the four-dimensional input signal $w = [\omega^T, \tau]^T$. Finally, the dynamics of the quadcopters, as discussed for example in [25],[44], takes on the form

$$\dot{q} = F(q, w),$$

which is a highly nonlinear dynamical system.

Luckily, this system is differentially flat too, meaning that the state and input can be algebraically recovered from a subset of states and their derivatives. In fact, as shown in [43],[64],[70], the flat output of the system is $\eta = [x, y, z, \phi]^T$, i.e., the position and the yaw of the quadcopter together describe the system in the sense that

$$\begin{bmatrix} q \\ w \end{bmatrix} = G(\eta, \dot{\eta}, \ddot{\eta}, \dddot{\eta})$$

for certain mapping G. This construction is in part why quadcopters are reasonably easy to control – one can almost think of them as unicycles with an added, freely controlled z-dimension.

The way we now can connect the output from the coordination protocols $u \in \mathbb{R}^3$ is as follows: when the quadcopter is in location (x, y, z), we simply generate a waypoint a short while into the future through $[x, y, z]^T + \delta_t u$ for some look-ahead horizon δ_t. Setting the new yaw equal to the old yaw, and producing a thrice continuously differentiable interpolating curve η_d, with

$$\eta_d(t) = [x, y, z, \phi]^T, \quad \eta_d(t + \delta_t) = [x + \delta_t u_x, y + \delta_t u_y, z + \delta_t u_z, \phi]^T$$

can thus be used to generate the feedforward part of the quadcopter motion. (A feedback law is typically added as well to ensure increased robustness to modeling errors.)

An example of combining the described flatness-based controller with a coordinated control signal for a team of quadcopters is shown in Figure 4.

Fig. 4 Five quadcopters execute a distributed formation control strategy based on planned trajectories for the flat outputs.

5. Conclusions and Optimization-Based Coordinated Control

What all the previous discussion have in common is the reliance on the formulation of the desired robot behavior through a cost to be minimized. If we again let x_i be the state of Robot i and use x_{N_i} to denote the states of all robots adjacent to Robot i (however the adjacency relationship happens to be defined), then one can realize that all the costs already discussed were of the following form:

$$\mathcal{E}(x) = \sum_{i=1}^{N} F_i(x_i, x_{N_i}).$$

For example, in the weighted formation control costs, we had

$$F_i(x_i, x_{N_i}) = \sum_{(i,j)\in E} \mathcal{E}_{ij}(\|x_i - x_j\|),$$

while the locational cost for coverage control was

$$F_i(x_i, x_{N_i}) = \int_{\mathcal{V}_i(x_i, x_{N_i})} \|x_i - p\|^2 \varphi(p) dp.$$

The reason why a gradient descent algorithm is particularly appropriate for coordinated control is that the adjacency relationship implied in the cost is made explicit by the descent algorithm in that

$$\frac{\partial \mathcal{E}(x)}{\partial x_i} = \frac{\partial F_i(x_i, x_{N_i})}{\partial x_i} + \sum_{(i,j)\in E} \frac{\partial F_j(x_j, x_{N_j})}{\partial x_i},$$

i.e., Robot i can evaluate this expression solely by having access to the states of adjacent agents. This preservation of adjacency is in general lost if higher-order derivatives are taken. Directed topologies also complicate the design of provably correct coordination laws through this path. This is because the evaluation of the gradient of the cost function requires an undirected flow of information and is therefore not implementable over directed graphs. Some works [7],[52],[65], however, are able to still draw inspiration from the undirected case to design implementable coordination strategies, but the directed information flow makes in general the convergence analysis described next harder.

The second reason why the gradient descent flow is useful is that even through the introduction of a strictly positive (possibly state-dependent) gain

$$\dot{x}_i = -\gamma_i(x_i, x_{N_i}) \frac{\partial \mathcal{E}(x)}{\partial x_i}$$

leads to

$$\frac{d\mathcal{E}(x(t))}{dt} = -\frac{\partial \mathcal{E}(x(t))}{\partial x}^T \Gamma(x) \frac{\partial \mathcal{E}(x(t))}{\partial x} = -\left\| \frac{\partial \mathcal{E}(x(t))}{\partial x} \right\|^2_{\Gamma(x)} \leq 0,$$

where $\Gamma(x) \succ 0$ is a positive definite, diagonal matrix with the individual gains on the diagonal.

As a consequence, through LaSalle Invariance Principle, we can (subject to sufficient regularity assumptions on the cost) draw the conclusion that the state converges to the set of stationary points, i.e., points where the gradient of the cost is zero. In this paper, we have illustrated this fact through a number of different examples – from rendezvous and formation control to coverage control. This distributed optimization-based paradigm in fact transcends multi-robot systems and has found its way into numerous domains involving network systems.

Acknowledgments

This work of the first author was partially supported by NSF Award ECCS-1307176 and AFOSR Award FA9550-15-1-0108 and the work of the second author was partially supported by NSF Awards 1531195 and 1544332.

References

[1] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita. Distributed memoryless point convergence algorithm for mobile robots with limited visibility. *IEEE Transactions on Robotics and Automation*, 15(5):818–828, 1999.

[2] E. Bakolas and P. Tsiotras. The Zermelo-Voronoi diagram: a dynamic partition problem. *Automatica*, 46:2059–2067, 2010.

[3] A. Barrientos, J. Colorado, J. Cerro, A. Martinez, C. Rossi, D. Sanz, and J. Valente. Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots. *Journal of Field Robotics*, 28:667–689, 2011.

[4] J. L. Baxter, E. K. Burke, J. M. Garibaldi, and M. Norman. Multi-robot search and rescue: A potential field based approach. *Autonomous Robots and Agents* (Studies in Computational Intelligence book series), 26:9–16, 2007.

[5] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation*. Prentice-Hall, 1989.

[6] F. Bullo, J. Cortés, and S. Martínez. Distributed Control of Robotic Networks. *Princeton University Press*, 2009.

[7] J. Cortés. Global and robust formation-shape stabilization of relative sensing networks. *Automatica*, 45(12):2754–2762, 2009.

[8] J. Cortés. Coverage optimization and spatial load balancing by robotic sensor networks. *IEEE Transactions on Automatic Control*, 55(3):749–754, 2010.

[9] J. Cortés, S. Martínez, and F. Bullo. Spatially-distributed coverage optimization and control with limited-range interactions. *Control, Optimisation & Calculus of Variations*, 11(4):691–719, 2005.

[10] J. Cortés, S. Martínez, and F. Bullo. Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. *IEEE Transactions on Automatic Control*, 51(8):1289–1298, 2006.

[11] J. Cortés, S. Martínez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.

[12] I. Couzin, J. Krause, N. R. Franks, and S. A. Levin. Effective leadership and decision-making in animal groups on the move. *Nature*, 433:513–516, 2005.

[13] I. Couzin, J. Krause, R. James, G. Ruxton, and N. Franks. Collective memory and spatial sorting in animal groups. *Journal of Theoretical Biology*, 218:1–11, 2002.

[14] R. D'Andrea. A revolution in the warehouse: A retrospective on kiva systems and the grand challenges ahead. *IEEE Transactions on Automation Science and Engineering*, 9(4):638–639, Oct. 2012.

[15] R. D'Andrea and G. E. Dullerud. Distributed control design for spatially interconnected systems. *IEEE Transactions on Automatic Control*, 49(9):1478–1495, 2003.

[16] P. Dayawansa and C. F. Martin. A converse Lyapunov theorem for a class of dynamical systems which undergo switching. *IEEE Transactions on Automatic Control*, 44(4):751–760, 1999.

[17] A. Dhariwal, G.S. Sukhatme, and A.A.G. Requicha. Bacterium-inspired robots for environmental monitoring. *IEEE International Conference on Robotics and Automation*, May 2004.

[18] V. Digani, L. Sabattini, C. Secchi, and C. Fantuzzi. Towards decentralized coordination of multi robot systems in industrial environments: A hierarchical traffic control strategy. *IEEE International Conference on Intelligent Computer Communication and Processing*, 2013.

[19] M. Dunbabin and L. Marques. Robots for environmental monitoring: Significant advancements and applications. *IEEE Robotics and Automation Magazine*, 19(1):24–39, March 2012.

[20] T. Eren, P.N. Belhumeur, B. D. O. Anderson, and A.S. Morse. A framework for maintaining formations based on rigidity. *IFAC World Congress*, 2002.

[21] 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.

[22] M. Fliess, J. Lévine, P. Martin, and P. Rouchon. Flatness and defect of non-linear systems: Introductory theory and examples. *International Journal of Control*, 61(6):1327–1361, 1995.

[23] C. Godsil and G. Royle. *Algebraic Graph Theory*. Springer, 2001.

[24] D. Grunbaum, S. V. Viscido, and J. K. Parrish. Extracting interactive control algorithms from group dynamics of schooling fish. *Cooperative Control Lecture Notes in Control and Information Sciences*, 309:103–117, 2005.

[25] M. Hehn and R. D'Andrea. Real-time trajectory generation for quadrocopters. *IEEE Transactions on Robotics*, 31(4):877–892, April 2015.

[26] J. Hespanha. Uniform stability of switched linear systems: extensions of LaSalles invariance principle. *IEEE Transactions on Automatic Control*, 49(4):470–482, 2004.

[27] 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, 2003.

[28] M. Ji and M. Egerstedt. Distributed coordination control of multi-agent systems while preserving connectedness. *IEEE Transactions on Robotics*, 23(4):693–703, 2007.

[29] G. Kantor, S. Singh, R. Peterson, D. Rus, A. Das, V. Kumar, G. Pereira, and J. Spletzer. Distributed search and rescue with robot and sensor teams. *Field and Service Robotics*, 24:529–538, 2003.

[30] W. Kazmi, M. Bisgaard, F. Garcia-Ruiz, K. D. Hansen, and A. la Cour-Harbo. Adaptive surveying and early treatment of crops with a team of autonomous vehicles. *European Conference on Mobile Robots*, 253–258, Jan. 2011.

[31] H. K. Khalil. *Nonlinear Systems*. Prentice Hall, 3 Edition, 2002.

[32] C. Kitts and M. Egerstedt (Guest Editors). Design, control, and applications of real-world multi-robot systems. *IEEE Robotics and Automation Magazine*, 15(1), March 2008.

[33] A. Kwok and S. Martínez. Deployment algorithms for a power-constrained mobile sensor network. *International Journal on Robust and Nonlinear Control*, 20(7):725–842, 2010.

[34] A. Kwok and S. Martínez. Coverage maximization with autonomous agents in fast flow environments. *Journal of Optimization Theory & Applications*, 155(2):986–1007, 2012.

[35] K. Laventall and J. Cortés. Coverage control by multirobot networks with limited-range anisotropic sensory. *International Journal of Control*, 82(6):1113–1121, 2009.

[36] J. Lawton, R. Beard, and B. Young. A decentralized approach to formation maneuvers. *IEEE Transactions on Robotics and Automation*, 19(6):933–941, 2003.

[37] S. Lee, Y. Diaz-Mercado, and M. Egerstedt. Multi-robot control using time-varying density functions. *IEEE Transactions on Robotics*, 31(2):489–493, April 2015.

[38] N. E. Leonard and E. Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. In *IEEE Conference on Decision and Control*, 2001.

[39] J. Lin, A. S. Morse, and B. D. O. Anderson. The multiagent rendezvous problem. Part 1: The synchronous case. *SIAM Journal on Control and Optimization*, 46(6):2096–2119, 2007.

[40] N. Lynch. *Distributed Algorithms*. Elsevier, 1996.

[41] J. A. Marshall, M. E. Broucke, and B. A. Francis. Formations of vehicles in cyclic pursuit. *IEEE Transactions on Automatic Control*, 49(11):1963–1974, 2004.

[42] J. M. McNew, E. Klavins, and M. Egerstedt. Solving coverage problems with embedded graph grammars. *Hybrid Systems: Computation and Control*, pp. 413–427, 2007.

[43] D. Mellinger and V. Kumar. Minimum snap trajectory generation and control for quadrotors. *IEEE Inter- national Conference on Robotics and Automation*, May 2011.

[44] D. Mellinger, A. Kushleyev, and V. Kumar. Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams. *IEEE International Conference on Robotics and Automation*, May 2012.

[45] M. Mesbahi. State-dependent graphs. *IEEE Conference on Decision and Control*, pp. 3058–3063, Maui, HI, 2003.

[46] M. Mesbahi and M. Egerstedt. *Graph Theoretic Methods for Multi-agent Networks.* Princeton University Press, Princeton, NJ, 2010.

[47] N. Michael and V. Kumar. Controlling shapes of ensembles of robots of finite size with nonholonomic constraints. *Robotics: Science and Systems*, June 2008.

[48] L. Moreau. Stability of multi-agent systems with time-dependent communication links. *IEEE Transactions on Automatic Control*, 50(2):169–182, 2005.

[49] P. Ögren, M. Egerstedt, and X.Hu. A control Lyapunov function approach to multi-agent coordination. *IEEE Transactions on Robotics and Automation*, 18(5):847–851, 2002.

[50] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley Series in Probability and Statistics. Wiley, 2 Edition, 2000.

[51] R. Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, 2006.

[52] 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, 2007.

[53] M. Ouimet and J. Cortés. Collective estimation of ocean nonlinear internal waves using robotic underwater drifters. *IEEE Access*, 1:418–427, 2013.

[54] L. E. Parker. Distributed algorithms for multi-robot observation of multiple moving targets. *Autonomous Robots*, 12(3):231–255, May 2002.

[55] R. Patel, P. Frasca, and F. Bullo. Centroidal area-constrained partitioning for robotic networks. *ASME Journal on Dynamic Systems, Measurement, and Control*, 136(3):31024–31028, 2014.

[56] M. Pavone, A. Arsie, E. Frazzoli, and F. Bullo. Distributed algorithms for environment partitioning in mobile robotic networks. *IEEE Transactions on Automatic Con- trol*, 56(8):1834–1848, 2011.

[57] A. Rahmani, M. Ji, M. Mesbahi, and M. Egerstedt. Controllability of multi-agent systems from a graph-theoretic perspective. *SIAM Journal on Control and Optimization*, 48(1):162–186, 2009.

[58] W. Ren and R. Beard. *Distributed Consensus in Multi-vehicle Cooperative Control*. Springer, 2008.

[59] C. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *ACM SIGGRAPH*, 21:25–34, 1987.

[60] M. D. Schuresko and J. Cortés. Distributed tree rearrangements for reachability and robust connectivity. *SIAM Journal on Control and Optimization*, 50(5):2588–2620, 2012.

[61] S. Susca, S. Martínez, and F. Bullo. Monitoring environmental boundaries with a robotic sensor network. *IEEE Transactions on Control Systems Technology*, 16(2):288–296, 2008.

[62] H. G. Tanner, A. Jadbabaie, and G. J. Pappas. Flocking in fixed and switching networks. *IEEE Transactions on Automatic Control*, 52(5):863–868, 2007.

[63] M. J. van Nieuwstadt, M. Rathinam, and R. M. Murray. Differential flatness and absolute equivalence of nonlinear control systems. *SIAM Journal on Control and Optimization*, 36(4):1225–1239, 1998.

[64] L. Wang and M. Egerstedt. Decentralized barrier certificates for safe and dynamical maneuvers in teams of quadrotors. *IEEE International Conference on Robotics and Automation*, May 2017.

[65] R. K. Williams, A. Gasparri, A. Priolo, and G.S. Sukhatme. Evaluating network rigidity in realistic systems: Decentralization, asynchronicity, and parallelization. *IEEE Transactions on Robotics*, 30(4):950–965, 2014.

[66] P. Yang, R. A. Freeman, G. J. Gordon, K. M. Lynch, S. S Srinivasa, and R. Sukthankar. Decentralized estimation and control of graph connectivity for mobile sensor networks. *Automatica*, 46(2):390–396, 2010.

[67] C. Yu, B. D. O. Anderson, S. Dasgupta, and B. Fidan. Control of minimally persistent formations in the plane. *SIAM Journal on Control and Optimization*, 48(1):206–233, 2009.

[68] M. Zavlanos, M. Egerstedt, and G. Pappas. Graph theoretic connectivity control of mobile robot networks. *Proceedings of the IEEE*, 99(9):1525–1540, 2011.

[69] F. Zhang and N.E. Leonard. Coordinated patterns of unit speed particles on a closed curve. *Systems and Control Letters*,

56(6):397–407, 2007.
[70] D. Zhou and M. Schwager. Vector field following for quadrotors using differential flatness. *IEEE Inter- national Conference on Robotics and Automation*, May 2014.

Jorge Cortés

He is a Professor with the Department of Mechanical and Aerospace Engineering at the University of California, San Diego. He received the Licenciatura degree in mathematics from the Universidadde Zaragoza, Spain, in 1997, and the Ph.D. degree in engineering mathematics from the Universidad Carlos III de Madrid, Spain, in 2001.He held postdoctoral positions at the University of Twente, The Netherlands, and at the University of Illinois at Urbana-Champaign, USA. He was an Assistant Professor with the Department of Applied Mathematics and Statistics at the University of California, Santa Cruz from 2004 to 2007. He is the author of "Geometric, Control and Numerical Aspects of Nonholonomic Systems" (New York: Springer-Verlag, 2002) and co-author of "Distributed Control of Robotic Networks"(Princeton: Princeton University Press, 2009). He received a NSFCAREER award in 2006 and was the recipient of the 2006 Spanish Society of Applied Mathematics Young Researcher Prize. He has co-authored papers that have won the 2008 IEEE Control Systems Outstanding Paper Award, the 2009 SIAM Review SIGEST selection from the SIAM Journal on Control and Optimization, and the 2012 O. Hugo Schuck Best Paper Award in the Theory category. He has been an IEEE Control Systems Society Distinguished Lecturer (2010-2014) and is an IEEE Fellow.

Magnus Egerstedt

He is the Executive Director for the Institute for Robotics and Intelligent Machines at the Georgia Institute of Technology. He is a Professor and the Julian T. Hightower Chair in Systems and Controls in the School of Electrical and Computer Engineering, with secondary appointments in the Woodruff School of Mechanical Engineering, the School of Interactive Computing, and the Guggenheim School of Aerospace Engineering. He received the M.S. degree in Engineering Physics and the Ph.D. degree in Applied Mathematics from the Royal Institute of Technology, Stockholm, Sweden, the B.A. degree in Philosophy from Stockholm University, and was a Postdoctoral Scholar at Harvard University. Dr. Egerstedt conducts research in the areas of control theory and robotics, with particular focus on control and coordination of complex networks, such as multi-robot systems, mobile sensor networks, and cyber-physical systems. Magnus Egerstedt is a Fellow of the IEEE, and has received a number of teaching and research awards, including the Ragazzini Award from the American Automatic Control Council, the Outstanding Doctoral Advisor Award and the HKN Outstanding Teacher Award from Georgia Tech, and the Alumni of the Year Award from the Royal Institute of Technology..