A Survey on Aerial Swarm Robotics

Soon-Jo Chung, *Senior Member, IEEE,* Aditya Paranjape, Philip Dames, *Member, IEEE,* Shaojie Shen, *Member, IEEE,* and Vijay Kumar, *Fellow, IEEE*

***Abstract*—The use of aerial swarms to solve real-world problems has been increasing steadily, accompanied by falling prices and improving performance of communication, sensing, and processing hardware. The commoditization of hardware has reduced unit costs, thereby lowering the barriers to entry to the field of aerial swarm robotics. A key enabling technology for swarms is the family of algorithms that allow the individual members of the swarm to communicate and allocate tasks amongst themselves, plan their trajectories, and coordinate their flight in such a way that the overall objectives of the swarm are achieved efficiently. These algorithms, often organized in a hierarchical fashion, endow the swarm with autonomy at every level, and the role of a human operator can be reduced, in principle, to interactions at a higher level without direct intervention. This technology depends on the clever and innovative application of theoretical tools from control and estimation. This paper reviews the state of the art of these theoretical tools, specifically focusing on how they have been developed for, and applied to, aerial swarms. Aerial swarms differ from swarms of ground-based vehicles in two respects: they operate in a three-dimensional (3-D) space, and the dynamics of individual vehicles adds an extra layer of complexity. We review dynamic modeling and conditions for stability and controllability that are essential in order to achieve cooperative flight and distributed sensing. The main sections of the paper focus on major results covering trajectory generation, task allocation, adversarial control, distributed sensing, monitoring, and mapping. Wherever possible, we indicate how the physics and subsystem technologies of aerial robots are brought to bear on these individual areas.**

***Index Terms*—Aerial robotics, distributed robot systems, networked robots.**

I. Introduction

Aerial robotics has become an area of intense research within the robotics and control community. Autonomous aerial robots can capitalize on the three-dimensional (3-D) airspace with aplomb, oftentimes equipped with vertical take-off and landing capabilities using zero-emission distributed electric fans. Swarms of such aerial robots or autonomous Unmanned Aerial Vehicles (UAVs) are emerging as a disruptive technology to enable highly-reconfigurable, on-demand, distributed intelligent autonomous systems with high impact on many areas of science, technology, and society, including tracking, inspection, and transporting systems. In any application, autonomous aerial swarms are expected to be more capable than a single large vehicle, offering significantly enhanced flexibility (adaptability, scalability, and maintainability) and robustness (reliability, survivability, and fault-tolerance) [1].

This survey article reflects on advances in aerial swarm robotics and recognizes that a number of technological gaps need to be bridged in order to achieve the aforementioned benefits of swarms of aerial robots through autonomous and safe operation. The papers included in this survey article represent the most important and promising approaches to modeling, control, planning, sensing, design, and implementation of aerial swarms, with an emphasis on enhanced flexibility, robustness, and autonomy.

Swarming aerial robots must autonomously operate in a complex 3-D world including urban canyons and an airspace that is getting increasingly crowded with drones and commercial airplanes. The success of aerial swarms flying in a 3-D world is predicated on the distributed and synergistic capabilities of controlling individual and collective motions of aerial robots with limited resources for on-board computation, power, communication, sensing, and actuation (the so-called size, weight and power, or SWaP, tradeoff). The goal is to provide a unified framework within which to analyze the three-way trade-off among computational efficiency for large-scale swarms, stability and robustness of control and estimation algorithms, and optimal system performance.

Compared to prior survey articles focused on robotic swarms [2], we emphasize swarms of aerial robots flying in a 3-D world. Other related survey papers on swarm robotics include [3], which focused on problems such as formation control, cooperative tasking, spatiotemporal planning, and consensus for generic multi-robot system. Our survey paper addresses the challenges associated with transitioning from 2-D to 3-D with limited SWaP with applications to swarm coordination or collaboration and distributed tracking and estimation. Our survey paper also addresses the challenges of integrating autonomous aerial swarm systems with other types of robots, such as ground vehicles. From a technological standpoint, the broader impacts of research in aerial swarm robotics include scalability and down-compatibility with 2-D robotic networks (e.g., ground robots) and other 3-D unmanned systems such as spacecraft swarms [4], [5] and underwater swarms [6]. The distinguishing characteristics of aerial swarm robotics are summarized as follows:

3-D Flow and Swarm Autonomy: Motion planning and control methods for aerial swarms rely on autonomously-generated 3-D traffic flows that do not have fixed edges or roads. Real-time flight control and swarm operation must

S.-J. Chung is with the Graduate Aerospace Laboratories of California Institute of Technology (GALCIT), Pasadena, CA 91125, USA (Email: sjchung@caltech.edu).

A. A. Paranjape is with the Department of Aeronautics, Imperial College London, South Kensington, London, United Kingdom (Email: a.paranjape@imperial.ac.uk).

P. Dames is with the Department of Mechanical Engineering, Temple University, Philadelphia, PA 19122, USA (Email: pdames@temple.edu).

S. Shen is with the Department of Electronic & Computer Engineering, Hong Kong University of Science and Technology, Hong Kong (Email: eeshaojie@ust.hk).

V. Kumar is with the School of Engineering & Applied Science, University of Pennsylvania, Philadelphia, PA 19104, USA (Email: kumar@seas.upenn.edu).

also take into account high-fidelity six-degree-of-freedom (6-DOF) flight dynamic models, traffic variations, weather, and other time-varying operational conditions found in crowded urban environments. These aspects stand in stark contrast to those focused on 3-D air traffic flow control with much longer time horizon [7]–[9] as well as 2-D road traffic flow theory, bipartite matching, and transport operation theory that assume fixed flight pathways and road/route topologies [10]. Furthermore, existing air traffic control systems require human operators to perform real-time control of airport congestion and prevention of mid-air collision [7]. We will describe methods of simultaneous 6-DOF trajectory generation and optimal swarm routing or control techniques for autonomous aerial swarm system that require a minimal level of human intervention.

Scalability Through Hierarchy and Multi-Modality: Enabling large-scale swarm autonomy in complex environments will require theoretically well-founded, computationally-efficient, and scalable algorithms. This can be realized through the use of hierarchical architectures for decentralized planning, reasoning, learning, and perception that address scalability and information management in the presence of uncertainties. Hierarchical approaches are pervasive in both the machine learning and control fields for dealing with complexity and high dimensionality (e.g., hierarchical task networks (HTNs) [11], hierarchical tree or lattice networks employed in Sequential Game Theory [12], and singular perturbation theory [13], [14]). They are also especially well-suited for aerial robots due to the inherent diversity of time scales in the system. The inner-loop flight control, and especially the attitude dynamics, must run faster than the timescales of the rigid body dynamics of the aerial robot as well as the structural dynamics of the wings or propellers to ensure stable flight. Onboard perception algorithms must also run at a time scale that is appropriately small to enable robots to avoid collisions with dynamic, unexpected obstacles. Transient maneuvers of aerials swarms are controlled at the same time scale as the rigid-body flight dynamics, while outer-loop control (i.e., motion planning of swarms) and the cooperative estimation and planning algorithms run an order of magnitude slower than the flight dynamics. These outer-loop components must be integrated closely with perception and reasoning of other vehicles, environmental conditions, and scientific or customer needs. This complexity in aerial swarms can be reduced by exploiting hierarchical connections in spatial and temporal scales of large-scale aerial swarm networks. In this survey paper, we expand on the algorithms and technologies for aerial robotics that depend on hierarchical architectures.

The organization of the present paper is shown in Fig. 1. In each section, we attempt to provide elementary working solutions, taken from the literature, for each subproblem. We will then present refinements of these solutions, which constitute the state of the art in the respective subject areas. In Sec. 2, we review modeling the dynamics of a swarm and nonlinear stability tools, in particular for hierarchical decomposition, as well as issues of controllability for aerial swarms. In Sec. 3, we review optimal control, motion planning, task assignment, and other control algorithms. In Sec. 4, we discuss distributed sensing and estimation using aerial swarms, specifically addressing the problems of (multi-)target tracking, distributed surveillance, and cooperative mapping. In Sec. 5, we review essential system-level and component technologies for aerial swarms. Finally, we conclude the paper in Sec. 6 with a discussion of open problems in the area of aerial swarms.

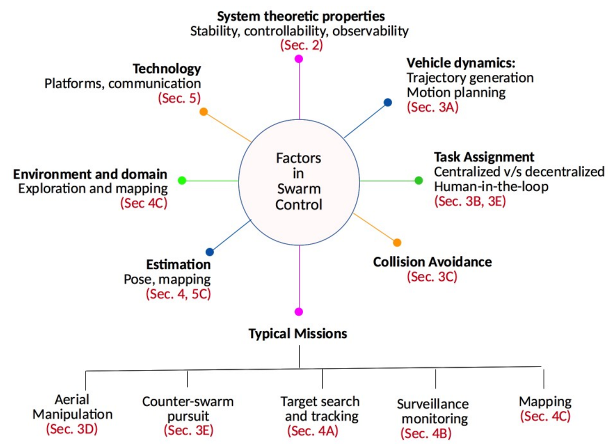

Fig. 1. Major themes in swarm control and the organization of the paper.

II. Models, Stability and Controllability of Swarms

A. Types of Multiagent Systems

Table I presents a classification of multiagent systems based on the number of agents and their interaction. It has a direct bearing on how the systems are modeled: the choice of the governing equations, the assumptions made about the underlying connectivity, and the nature of the control inputs and information exchange.

In a *team*, the behavior and strategies of each individual agent seek to explicitly maximize a local objective. In some cases, this may cause the agents to compete against each other, while in other cases the locally optimal behavior may also (approximately) maximize the global reward. The latter is the premise of game-theoretic methods and auction algorithms [15], [16]. In auctions, for instance, maximizing the local benefit also maximizes the net global benefit (defined as the sum of individual benefits) and concurrently solves the dual pricing problem [15]. In contrast to a team, a *formation* almost always consists of cooperative interactions, and the relationship between the states of the agents is well-defined for objectives such as energy efficiency (e.g., flocks of birds in an aerodynamically optimum V-formation [17]). A *swarm* generally refers to a group of similar agents that displays emergent behavior arising from local interactions among the agents. The local interaction can be competitive or cooperative. Although a swarm typically implies a large group of agents (10s to 100s or more), this survey article uses “swarm” to also include smaller groups as well (see Table I).

TABLE I
CLASSIFICATION OF MULTI-AGENT SYSTEMS

Type	Scope	Size
Team	Typically small groups; each agent optimizes individual objectives in a cooperative or competitive manner	typically ≤ 10
Formation	Each agent is typically assigned a specific sub-task, role, or placement	typically ≤ 10s
Swarm	Typically large groups of dispensable agents; global capability arises from emergent behavior	large

B. Models for Swarm Dynamical Systems

One of the earliest engineering models for flocking is from Reynolds [18], who used it to generate a realistic visualization of flocks for computer graphics. Reynolds rules cover basic neighbor-to-neighbor interaction: a nonlinear function which governs the steady state separation between the agents, and a velocity feedback term which seeks to ensure that the velocity of each agent tracks the average of its neighbors. Reynolds' model is given as:

$$\ddot{\boldsymbol{x}}_i = \dot{\boldsymbol{v}}_i = \sum_{j \in \mathcal{N}_i} (k_s \nabla W(\boldsymbol{x}_j - \boldsymbol{x}_i) + k_a(\boldsymbol{v}_j - \boldsymbol{v}_i)) + \boldsymbol{f}_i \quad (1)$$

where $\boldsymbol{x}_i$ and $\boldsymbol{v}_i$ denote the position and the velocity of the i^{th} agent; $W(\boldsymbol{x}_j - \boldsymbol{x}_i)$ is a coupling function; $\mathcal{N}_i$ is the neighborhood of i^{th} agent; and $\boldsymbol{f}_i$ denotes an external influence on the agent, such as that of the leader or an intruder.

Another early work [19] studied a flock moving in two-dimensional space and discrete time using the following equations:

$$\begin{aligned} \boldsymbol{x}_i(t+1) &= \boldsymbol{x}_i(t) + \boldsymbol{v}_i(t)\Delta t \\ \theta_i(t+1) &= \frac{1}{\text{card}(\mathcal{N}_i)} \sum_{j \in \mathcal{N}_i} \theta_j(t) + \Delta\theta_i(t) \end{aligned} \quad (2)$$

where the noise $\Delta\theta_i(t)$ is normally distributed in the set $[-\eta, \eta]$. Importantly, the velocity $\boldsymbol{v}_i$ is assumed to have a constant magnitude for all i and t, with its heading given by $\theta_i(t)$. Despite the apparent simplicity of the model, it is able to capture the possibility of long-range order, as explained later in this section.

A generalized representation of the models in [18] and [19] can be obtained by using partial *difference* equations (PdEs) [20], [21]. The rules for obtaining PdEs permit a natural association with continuum PDEs, and consequently, ways for deriving flocking laws based on PDEs other than the wave equation used in [20].

A unified, *nonlinear* continuum model, as against models based on discretely defined agents on a graph, was proposed in [22]:

$$\begin{aligned} &\frac{\partial \boldsymbol{v}}{\partial t} + \underbrace{(\boldsymbol{v}\cdot\nabla)\boldsymbol{v}}_{\text{convection}} = \alpha\boldsymbol{v} - \beta\|\boldsymbol{v}\|^2\boldsymbol{v} - \nabla P(\rho) \\ &+ \underbrace{D_L\nabla(\nabla\cdot\boldsymbol{v}) + D_1\nabla^2\boldsymbol{v} + D_2(\boldsymbol{v}\cdot\nabla)^2\boldsymbol{v}}_{\text{diffusion}} + \boldsymbol{f} \\ &\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\boldsymbol{v}) = 0 \end{aligned} \quad (3)$$

This model was claimed to resemble that of bird flocks for two spatial dimensions, although the model itself is not constrained to any particular number of dimensions and could be applicable to three-dimensional flocks as well. The constants β, $D_{\{\cdot\}}$ are all positive; the term $\alpha > 0$ corresponds to an ordered velocity state (steady flight speed $\|\boldsymbol{v}\| = \sqrt{\alpha/\beta}$), while $\alpha < 0$ gives rise to a disordered phase (e.g., a flock loitering around a fixed point). The pressure term $P = \sum_k \sigma_k(\rho - \rho_0)^k$, where σ_k's are constants and ρ_0 is the mean local density, replaces the potential-like term in Reynolds' model. Finally, $\boldsymbol{f}$ denotes disturbances, modeled as Gaussian noise.

Increasing the value of the noise (i.e., η) in (2) causes the flock to spontaneously choose an ordered state [19], where the critical value of the noise is correlated with the number of agents in the flock. This is conjectured to be due to the diffusive flow of information in the flock; i.e., agents interacting with a time-varying set of neighbors and, in the long run, this causes diffusion of information throughout the flock. This conjecture was borne out in [22] for a two-dimensional flock, wherein the nonlinear convection terms in (3) were found to be responsible for stabilizing the ordered state across large length scales.

In the context of swarms, one is interested in the questions of stability and convergence of the states of the individual agents. For such analysis, it is common to use a system of linear(ized) equations, the simplest of which is the system

$$\dot{\boldsymbol{x}}_i = \sum_{j \in \mathcal{N}_i} w_{ij}(\boldsymbol{x}_j - \boldsymbol{x}_i), \quad i = 1, \dots, n \quad (4)$$

$$\Leftrightarrow \qquad \dot{\boldsymbol{x}} = -(\mathcal{L} \otimes \mathbf{I}_p)\boldsymbol{x}, \ \mathcal{L}_{ij} = \begin{cases} w_{ij} & \exists \text{ edge from node j} \\ & \text{to node i} \\ 0 & \text{otherwise} \end{cases}$$

The matrix $\mathcal{L}$ or $(\mathcal{L} \otimes \mathbf{I}_p)$ is called a Laplacian matrix and satisfies $\mathcal{L}\mathbf{1}_n = 0$, where $\mathbf{1}_n \in \mathbb{R}^n$ is a vector of ones. It is evident that a constant $\mathcal{L}$ corresponds to a fixed communication topology; when the communication topology evolves with time, a time-varying $\mathcal{L}(t)$ is used. This is identical to the diffusive coupling term one would find from (2).

For problems involving assignment or routing, it helps to model the environment as a collection of "functional bins," together with accessibility conditions which restrict the agents' transition between the bins. The end objective is to assign n agents to a set of m bins, where each bin can accommodate up to $p_i \geq 1$ ($m < n$; $i = \{1, 2, \dots, m\}$) agents. For each agent i and a bin j, the accessible set E_{ij} implicitly accounts for the dynamics of the agent as well as the geometric constraints imposed by the environment. Such models have been used to control swarm shape with probabilistic transition maps between the bins [23] and quadrotor formation control with deterministic transition laws [24], [25].

C. Physics-Based Models for Robotic Agents

General linear systems similar to (4) can be constructed readily in a double integrator setting (e.g., attitude dynamics on SO(3) or rigid body motions on SE(3)), or by replacing the dynamics with a nonlinear version. Of particular interest here are swarm systems comprising Euler-Lagrange equations:

$$M_i(\boldsymbol{q}_i)\ddot{\boldsymbol{q}}_i+C_i(\boldsymbol{q}_i,\dot{\boldsymbol{q}}_i)\dot{\boldsymbol{q}}_i+\boldsymbol{g}_i(\boldsymbol{q}_i)=\boldsymbol{\tau}_i(\boldsymbol{q}_i,\dot{\boldsymbol{q}}_i,\boldsymbol{q}_d,\boldsymbol{q}_{j\in\mathcal{N}_i},\dot{\boldsymbol{q}}_{j\in\mathcal{N}_i}) \tag{5}$$

where $\boldsymbol{q}_i \in \mathbb{R}^p$ are the generalized states of the i^{th} agent; $\boldsymbol{q}_d(t)$ is the desired trajectory or a virtual leader for a target collective motion; and $\boldsymbol{\tau}_i$ are the external forces/torques, which are the source of coupling between agents. If a linear diffusive coupling is used, $\boldsymbol{\tau}_i$ would produce $\mathcal{L}$ similar to (4). The Euler-Lagrange equations appear routinely robotics in the study of rigid body motions of manipulators [26], [27] and spacecraft or aircraft (SE(3)), which have attitude dynamics on SO(3) [28]–[30] and often times have articulated wings [14], [31], [32], appendages, or manipulators attached:

In [33], the full 6-DOF aircraft model is used with actuator time delays to compute optimal motion primitives and 3-D path planning for fast flight through a forest. It shows that a conventional 2-D Dubin's vehicle model, often times used for 2-D aircraft motion planning and swarm control, is not appropriate for aerial robots moving in 3-D. For the purpose of studying swarms of fixed- or flapping-wing aerial robots, it may suffice to model the aerial robots as point-masses (mass m) with velocity dynamics (speed V, climb angle γ, and heading χ) described by

$$\begin{aligned}
[\dot{x},\,\dot{y},\,\dot{h}] &= V[\cos\gamma\,\cos\chi,\,\cos\gamma\sin\chi,\,\sin\gamma]\\
m\dot{V} &= T\cos\alpha - D(V,\alpha) - mg\sin\gamma\\
mV\dot{\gamma} &= \frac{1}{mV}(L(V,\alpha)+T\sin\alpha)\cos\mu - W\cos\gamma\\
mV\dot{\chi} &= (L+T\sin\alpha)\frac{\sin\mu}{\cos\gamma}
\end{aligned} \tag{6}$$

where L, D, and T are the lift, drag, and thrust, respectively. In flapping-wing aerial robots, T is additionally a function of V and α. This model is accurate under the assumption that the rotational dynamics (α and μ) are stable and converge rapidly to the commanded value. The 3-D aerial robot model can be used effectively to reduce the computational burden on a motion planning system and generate trajectories that are optimal, stable, and safe (i.e., with collision avoidance) [14], [33]. In [34], model-based control laws were derived, together with a collision-avoiding system, for a swarm of parafoil-payload systems. A model similar to (6) was employed, and feedback about the position of the neighboring agents was used to command the desired value of the turn rate ($\dot{\chi}$ in (6)) of each agent.

Although the terms L, D, and T have been presented in the spirit of control inputs in (6), it is important to note that their values could be affected significantly in a swarm of aerial robots by flow induced by neighboring aircraft. When an aerial robot experience failures and is unable to hold its position accurately, it could have a detrimental effect on the efficiency of the formation due to the adverse disruption in the flow field experienced by the faulty aircraft's neighbors. This sort of physics-based interaction is unique to atmospheric flight vehicles.

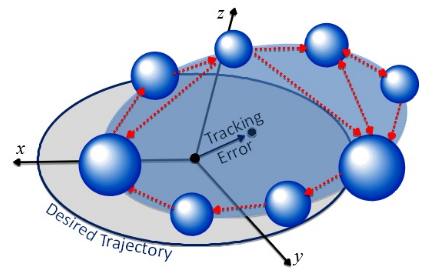

Fig. 2. A swarm of heterogeneous rigid bodies converging to the desired shape (ellipsoid), whose center can be viewed as a virtual leader for the swarm [28]. The angular separation between the vehicles is synchronized actively and shows smaller synchronization error than tracking errors. Dotted lines show diffusive couplings via communication or relative sensing.

D. Synchronization with Leader Following

To control a swarm, it is useful at times to define a physical or virtual leader that the rest of swarm agents then follow (see Fig. 2). The motion of the leader can be given a priori or controlled directly by separate dynamics. Alternatively, a desired trajectory (i.e., the path of a virtual leader) can be computed using optimal control or motion planning algorithms (see Sec. 3). The remaining agents are controlled indirectly through interaction between neighbors [35] or through interaction with the leader [36], [37]. The problem of tracking the trajectory of the virtual leader or the desired collective behavior for agents with highly nonlinear dynamics (e.g., swarms rigid bodies with dynamics on SE(3) or agents with multi-DOF manipulators) can be addressed simultaneously with the problem of synchronization with neighboring agents [27]. Based on time-scale separation, this unified framework integrates trajectory tracking with an exponentially-stabilizing consensus controller that synchronizes the relative motions of swarms faster than following a common leader or a desired trajectory. This yields a smaller synchronization error than an uncoupled tracking control law in the presence of bounded disturbances and modeling errors [28] (see Fig. 2). This time-scale separation can be interpreted as a hierarchical connection of faster and slower dynamics as discussed in Sec. II-F. Other works follow the same problem formulation of synchronizing coupled nonlinear dynamical systems concurrently with trajectory tracking for various multi-robot/multi-vehicle applications. We can leverage concurrent synchronization of mixing multiple virtual leaders with many synchronized groups to create a complex time-varying swarm comprised of numerous heterogeneous systems [27], [28], [30], [38]. One needs to determine *how many* (virtual) leaders need to be chosen, and *which* agents to nominate as leaders. This question is analogous to that of controllability, while the dual observability problem corresponds to sensor placement for distributed estimation.

E. Leader Selection and Sensor Placement

When the dynamics of the aerial swarm agents are identical, controllability from a given set of leader nodes (equivalently, observability from a given set of sensors) depends on the topology of the graph *as well as* the individual edge weights. A system defined on a graph is said to be structurally controllable when it is controllable for *almost all* edge weights, and strongly structural controllable when it is controllable for *all* edge weights. The existence of a rooted tree is necessary and sufficient for structural controllability with a single leader node [39]–[42]. In [43]–[45], conditions and algorithms are derived to determine whether a set of input nodes permit strong structural controllability. Formulas linking the number of driver nodes needed for a large network and its aggregate properties (number of nodes, mean degree and the degree exponent) are presented in [46]. It was observed that driver nodes with the highest degree of controllability tend *not* to be nodes of the largest degree.

In practical problems, we are generally interested in controllability for a *given* set of edge weights, and especially for identical edge weights (i.e., the system is described by the Laplacian matrix $-\mathcal{L}$ in (4)). For this problem, there exist *necessary* conditions based on symmetry and equipartition [47]–[49]. While these conditions are not sufficient, a set of sufficient conditions have been presented for path and cycle graphs [50], and for a class of weakly-connected digraphs [51].

It must be noted that the selection of a leader (equivalently, sensor placement) need not be optimal by the virtue of its controllability (respectively, observability) properties alone, and it is therefore necessary to measure the influence of the candidate driver nodes on the actual control/estimation objectives [52]. If the objective is to optimize an objective function, as shall be seen in (7), one can solve the problem of determining the leader/sensor nodes computationally using techniques from sub-modular optimization [53]–[55]. Maximization of sub-modular functions is NP-hard; however, greedy algorithms can yield approximate solutions with guaranteed sub-optimality using at most $O(n^2)$ computations of the objective function. Sub-modular optimization can also be viewed from the hierarchical organization standpoint emphasized in this paper.

F. Synchronization and Hierarchical Stability for Swarms

Consider (4) with diffusive couplings. It is well-known that the matrix $\mathcal{L}$ gives rise to a stable system under the following conditions on the underlying graph:

1) Undirected time-invariant graph: the graph is connected [56].
2) Directed time-invariant graph: consensus to the average value if and only if the graph is balanced and weakly connected [57]. Existence of a rooted tree guarantees consensus, though not necessarily to the average value [58].
3) Time-varying undirected/directed graph: satisfies a generalized strong connectivity condition [58, Propositions 1 and 2], [59].

The Laplacian matrix ($\mathcal{L}$) captures the effect of diffusive coupling terms on swarm or synchronization stability. The spectral characteristics of Laplacian matrices have been used to prove the stability of flocks obeying Reynolds' rules [19], [59]–[61], the stability under a distance-based communication topology [62], and the exponential stabilization of networked, nonlinear Euler-Lagrange systems [27], [28], [31], [63]. [64] illustrate the effect of nonlinearities on the stability of networked systems through bifurcations. Alternate methods for stability analysis include tools from renormalization groups [22] and the theory of normally hyperbolic invariant manifolds [65]. The Laplacian matrix defined above can be replaced by its variant, the *edge Laplacian matrix*, to solve for stability as well as robustness and optimality [66].

The aforementioned conditions are conclusive in the absence of other dynamical terms like (4). The passivity of the input-output dynamics [67]–[69] is commonly used to analyze the stability of networked nonlinear systems that have both the Laplacian matrix ($\mathcal{L}$ in (4)) and the nonlinear dynamical terms (e.g., convection terms of (3) or the Lagrangian form in (5)). Input-to-State Stability (ISS) is used to study stability of swarm systems with bounded uncertainties [70], [71]. Contraction analysis [72] is used to study global exponential stability of multiple solution trajectories, and hence forms a basis of incremental stability analysis. Contraction-based incremental stability analysis represents an important departure from traditional passivity-based methods using Lyapunov functions, which are concerned primarily with stability of equilibrium points.

Such an *exponentially-safe* and *robust* synchronization framework can also be used to study the synchronization stability and robustness of networked nonlinear dynamics connected by a synchronization controller or by diffusive communication couplings [27], [73]. One major advantage of incremental stability in a synchronization framework [27], [28], [73] over the passivity formalism is that a hierarchically-combined structure of dynamic systems, emphasized in this paper, can be handled more easily because of differential contraction analysis without using some implicit motion integrals.

Further, it can be shown that contraction-based exponential incremental stability using a Riemannian metric possesses superior robustness related to input-to-state stability (ISS), output passivity, and finite-gain L_p stability [28]. Many types of model uncertainty can be cast into a bounded perturbation term, including constant unknown time delays [27], [72] and errors arising from heterogeneous dynamics [27], [63]. Recently, incremental stability has been extended to synchronization stability of multiple Itō stochastic nonlinear differential equations [38], [74] with unbounded stochastic disturbances.

An extension of some of the aforementioned results arises in the form of event-triggered information exchange. Instead of exchanging communication continuously or over finite intervals of time, as in the previous cases, it is sufficient for stability to exchange information between neighboring agents at discrete *instants* of time. Conditions for stability in such cases have been found for single integrator dynamics on undirected graphs [75], consensus on balanced digraphs [76], convergence to a trajectory on time-varying graphs [77], and synchronization of general nonlinear dynamics on balanced graphs [28], [73], [78]. These conditions typically depend on the underlying dynamics and also help determine the conditions under which communication must be triggered.

III. Control of Swarms in 3-D Worlds

Typical tasks for which swarms are suitable include distributed sensing, search and rescue [79], and imaging using sparse aperture techniques [1], [5]. These problems can be split into two distinct classes: one where the environment is to be explored (e.g., coverage, map building), and one where the environment is only to be traversed or exploited (e.g., crossing a field of obstacles) with a prescribed goal state or a desired formation. In order to effectively complete any of these tasks a swarm must be capable of planning paths for all team members to safely and reliably reach their final destinations. Not only does each individual robot need to avoid collisions with static and dynamic obstacles in the environment, but the individuals in the swarm must also avoid collisions with one another. Furthermore, in complex, obstacle-filled environments, the robots need to sequence their motions to avoid having one robot block the paths for others. For example, if the swarm needs to pass through a small bottleneck and the end goal for one of the agents is just through that bottleneck, then it must be the last one to pass through in order to not block the rest of the team [80].

A. Trajectory Generation and Motion Planning for Swarms

Approaches to trajectory generation may be classified on the basis of whether or not the trajectories are generated in conjunction with the task allocation problem discussed in Sec. III-B. Trajectories generated independently of the task assignment algorithm can be thought of in the same light as traditional optimal motion planning or boundary value problems. Popular randomized algorithms, such as PRM [81], RRT [82], and RRT* [83], may not be effective for obtaining optimal and safe flight of multiple 6-DOF aerial robots; not only can they not effectively handle 6-DOF nonlinear dynamics, but they also use a finite set of primitives predicated on asymptotic optimality without using higher-fidelity dynamic models, which could preclude a large set of otherwise flyable trajectories in a high-dimensional space. The rapid advancement in computing capacity combined with algorithmic improvements has enabled the development of tools that are capable of solving constrained optimization problems in real-time, which can better provide explicit or approximate solutions to an optimal control problem of the form

$$\sum_{j=1}^{N}\left(h\big(t_f^j, x^j(t_f^j)\big) \;+\int_{t_0^j}^{t_f^j} L\big(\gamma^j(t), u^j(t), \alpha^j(t), t\big)\,dt\right) \tag{7}$$

Subject to:

Valid goal and task assignment, including terminal states

Robot dynamics, capabilities, and input constraints

State constraints (collision-free region, sensing restrictions)

where $\gamma^j(t)$ denotes the trajectory for robot j, $h(\cdot)$ denotes a terminal cost, $\alpha^j(\cdot)$ denotes a set of parameters of a mode of operation, and $L(\cdot)$ is the cost-to-go functional. The first constraint ensures that robots are assigned to valid goals or end at desired terminal states ($x^j(t_f^i)$) while the second constraint ensures that the trajectories obey both the kinematic and dynamic constraints of the robots and the input ($u^j(\cdot)$) constraints. The third constraint ensures that the optimal trajectories begin at the actual initial states while ensuring safety and other state-dependent constraints. Since the cost function is optimized in real-time over a finite-time horizon, often times recomputed using the current states of the robots as the initial conditions, (7) can be viewed as model predictive control (MPC) [25], [84]–[87]. Another approach to multi-agent planning under uncertainty over a discretized state domain is to employ a decentralized partially observable Markov decision process (POMDP) [88], [89].

Optimality in the multi-robot path planning problem (7) may be with respect to any number of different objectives, including integrated control effort, maximum single-robot travel distance, last arrival time, and total distance or time [90]. Although solving for the exact optimal solutions is NP-hard, approximate sub-optimal solutions can be computed efficiently using well-chosen heuristics [90]. One must ensure that the resulting paths are kinematically or dynamically feasible for the robots to follow [91], [92]. Direct optimal control approaches [25], [85]–[87] cast the dynamics into equality constraints between the states in successive time steps for optimization (e.g., iterative linearization of dynamics in sequential convex programming [25], [87]). Alternatively, one can find a geometric path for each robot to reach its goal and then use these paths as inputs to a trajectory optimization step to make the paths dynamically feasible [92].

Another objective of trajectory design and motion planning is to enable the design of control laws for the robotic agents. One direct way to obtain control input values is to re-solve the trajectory generation problem in the MPC setting (7) and apply the new optimal control input value frequently. But the process can be computationally expensive and stability guarantees are challenging. Alternately, the control design can be separated from optimal trajectory design by treating the optimized state trajectory for each robot, obtained from (7), as a desired trajectory for the tracking controller [25], [28], [87], [93], [94]. This approach has the benefit of setting up the control design problem in the traditional input-tracking or model reference setting with guaranteed closed-loop stability. It is particularly suitable for robotic systems, such as aerial robots, whose physical models are complex but well-understood from the point of view of control design. Alternately, control laws designed without virtual leaders typically consist of a sum of terms that represent the multiple objectives: trajectory-following, coordination with neighbors, and collision-avoidance. As explained above, trajectory-following laws can be derived readily using a physical model of the robots. Terms for coordination and collision-avoidance require sensing and communication with other agents in the formation. Controllers capable of accommodating time-varying communication topologies have been derived and demonstrated for quadrotors using modified temporal coordinates [95], for Dubin's vehicles using local potential functions [96], and for spacecraft [97].

Trajectory generation occasionally requires a hierarchical "model-based" approach when motion requirements stem from specific tasks that the swarm needs to perform, or from needing to deal with exigencies. For instance, cooperative construction [98], [99] requires formation-like motion as well as specialized low-level motions for stabilizing and manipulating objects. In such cases, physics-based models for manipulation can be solved to find a relative motion plan for the robots, while a global path plan can be constructed using any of the well-known path planning algorithms [98].

Specialized controllers must be designed to allow aerial robots flying in an energy-efficient formation to deal with actuator failures in individual aircraft and enable them to hold their formation [100]. These controllers benefit from aerodynamic models which help estimate the influence of neighboring aircraft on the controllability of a given aircraft. As pointed out in Sec. II-B, aerial swarms differ from ground-based robots in that the individual aircraft are coupled aerodynamically, due to the flow induced by one vehicle affecting its neighbors. The team may also generate trajectories that account for these aerodynamic effects and plan trajectories that minimize disturbance [101]. Alternately, the team can reconstruct the wake profile, as demonstrated in [102], although it requires that the aircraft perform cross-track motion to ensure the stability of the estimator.

Looking at constraints beyond collision avoidance and dynamic feasibility, one key factor with UAVs is their limited battery life. To extend the mission life, a cooperative team of ground vehicles may be used as mobile recharging stations [103]. The UAVs then plan paths to ensure that they are able to accomplish their missions while maintaining power.

B. Simultaneous Planning with Distributed Assignment

In a homogeneous swarm of robots, it does not matter which agent completes a given task. This fact may be exploited to do simultaneous task assignment and trajectory planning for teams of 100's of agents in a centralized or decentralized formulation [80]. This decentralized formulation is sub-optimal compared to the centralized solution, but it is still complete. For example, simultaneous optimal assignment and trajectory planning computes an optimal terminal point constraint of (7) for shape reconfiguration control [25], [87].

A special case of assignment is cooperative pursuit, wherein multiple pursuers seek a single target. A pursuit strategy and conditions for a successful pursuit in a bounded domain were determined in [104]. The conditions for a successful pursuit link the relative speeds of the pursuers and the evader, the turning radius of the pursuer (assuming an arbitrarily agile evader), and the total number of pursuers. More realistic, physics-based scenarios have been investigated in the context of missile interception, wherein multiple defensive missiles are used to intercept one or more incoming (target) missiles, which are assumed to use a standard optimal guidance and evasion law. Estimating the states and guidance laws of the target missiles is a significant challenge, compounded by the fact that the time delay involved in estimating the states can have a severely adverse effect on the pursuit [105]. In [106], cooperative estimation of the target states (compared to each missile using solely its own estimates) was shown to improve the likelihood of success significantly. Information sharing between the missiles can also be used to directly tune their navigation law, as demonstrated in [107], to achieve a synchronized hit on the target.

A scenario related to cooperative pursuit is that of multiple UAVs tracking a single target. From the point of view of trajectory generation, it is of interest to consider scenarios wherein the environment is populated with no-go areas and with terrain features that may sporadically occlude the pursuers' view of the target, such as a typical urban neighborhood. In order to facilitate the generation of trajectories which minimize occlusion, it is beneficial to develop adequate models of the sensors, such as gimbaled cameras, that are used to track the target. The constraints of the tracking system can then be added to the dynamic limitations of each UAV to generate guidance laws for the complete team of UAVs [108].

The simplest task assignment problem is the following static, symmetric problem: given a set of n agents, n bins, and a matrix of rewards $P \in \mathbb{R}^{n\times n}$ (or, equivalently, a matrix of costs $C \in \mathbb{R}^{n\times n}$) , where $P_{i,j}$ (resp. $C_{i,j}$) denotes the reward derived (resp. cost incurred) by agent i from being assigned to bin j and $P_{i,j} = -\infty$ (resp. $C_{i,j} = \infty$) denotes an infeasible assignment, determine the map $A : i \mapsto j = A(i)$ which assigns to each agent *a unique* bin while maximizing the collective reward $\sum_i P_{i,\,A(i)}$ (resp. minimizing the equivalent collective cost). Parallel or distributed algorithms to solve target assignment include many variants of distributed auction algorithms [16], [25], [109]–[111] and decentralized hierarchical strategies [112] that approximate true optimality of Kuhn's centralized Hungarian method. As an illustration of the computational complexity of auction algorithms, the number of computations required for the distributed algorithm from [16] to converge is $O(\Delta n^2)$, where Δ is the diameter of the communication graph underlying the network of agents participating in the auction.

An elementary auction algorithm is illustrated in Algorithm 1. This algorithm is centralized, and requires a central register where information about the bids and assignments is maintained. In contrast, distributed algorithms distribute computation as well as communication among the agents. For instance, the algorithm in [25] adjusts the number of targets based on the number of agents available at a specific stage. This is accomplished through bidding, rather than a consensus-like process, which is useful in large swarms with agents that may drop out spontaneously. This distributed target assignment can be solved simultaneously to provide goal states of real-time optimal trajectory generation, thereby effectively solving (7) [25], [80], [113]

An equivalent geometric problem involves partitioning a physical volume into portions that are then assigned to each agent inside the volume. A well-known result is that the optimal partition corresponds to the generation of Voronoi cells using a suitable metric function [114]. This approach was introduced in [114] for sensor coverage, and generalized in [115], [116] to cover learning (of the task distribution) and decentralized information sharing.

Assignment can be obtained as the solution to an optimal transport problem [117] when the transition between bins is modeled in a probabilistic framework through homogeneous

Algorithm 1 Symmetric Auction-based Assignment

```
Given: Terminal bins to be filled; for each (agent, bin), the cost of occupancy; an initial price for each bin
while There is an unoccupied bin do
    for Each unassigned agent do
        Identify bins with lowest and second lowest value of (cost + price)
        Bid = difference in the prices + random number
        Cost of the least expensive bin increased by the bid
        Assign the least expensive bin to the agent, and the previous occupant (if any) is unassigned
    end for
end while
End: Each agent has a unique bin assigned to it
```

Markov matrices. An improved approach has been proposed in [23] using time-inhomogeneous Markov chains which allow for the inclusion of feedback terms, thereby solving both bin-to-bin swarm shape control and stochastic target assignment.

C. *Collision Avoidance and Collision-Free Motions*

The problem of collision avoidance becomes particularly challenging in swarms because the obstacles encountered by a robot include other members of its swarm, and collision avoidance has to factor in the need to maximize the performance of the swarm (e.g., avoid increasing the time to complete an assignment). The most intuitive techniques for avoiding collisions are speed adjustment [118] and sequentially re-planning the trajectories [87] without changing the assignment in an optimal control framework (7). In particular, mixed-integer linear programming (MILP) has been successfully derived for optimal collision-free motions and applied to mobile robots, spacecraft, and UAVs [85], [86], [119]. More recently, sequential convex programming (SCP) has been used to approximate collision-free regions by incrementally drawing hyperplanes and has been demonstrated in simulation and experiments on swarms [25], [87]. The conservatism of hyperplane-based convexification of collision-free regions has been relaxed by expanding convex spherical regions along graph-based primitive paths in [120]. An alternate approach to replanning just the trajectories involves reassigning the goals as shown in [80]. The reassignment is purely local and need not affect the criteria used for the assignment in the first place.

A more direct approach to collision avoidance in swarms involves the use of artificial potential fields [121]–[124] or barrier functions [125], [126]. It must be noted that Reynolds' model (1) also includes the gradient of a potential function. Potential fields are computationally easy to implement for the purpose of collision avoidance, but not necessarily for path planning. Furthermore, artificial potential fields directly couple the dynamics of the individual robots and this can adversely affect the stability of the swarm if the communication topology is not selected properly. Connectivity is not enough to guarantee stability in directed graphs (see Sec. II-F). An approach similar to potential fields involves using the gradient of a Lyapunov function, which implicitly takes into account the possibility of collisions. Such control laws have been constructed using a differential game approach [127], [128] and simultaneously solve a greedy optimization problem. The difficulty lies in solving the optimal control problem in the presence of nonlinearities and local communication.

D. *Aerial Manipulation*

Aerial robotic swarms have the ability to transport objects in two ways, where each individual robot is capable of carrying an object or where multiple robots are required to lift a single object. In either scenario, the object may be suspended via cables attached to the robots [129]–[133] or may be rigidly attached to the robots [134]–[138]. UAVs that are rigidly attached to the objects use a variety of grippers, including friction-based [134], penetration-based [135], or magnetic [137].

When each individual robot is capable of grasping an object, having a swarm of robots allows a large number of objects to be moved more quickly. This can be used for tasks such as package delivery [133], [138] and construction [134].

When multiple robots are required to move a single object, small teams of robots may be used to cooperatively transport a single object [129]–[132], [135]–[137]. This task requires some type of communication between the robots. This is typically done in an explicit manner, but can also be done implicitly by sensing the internal forces of the robots acting on the transported object [132]. The swarm also seeks to minimize these internal forces, as these represent wasted energy usage [136].

E. *External Control of Aerial Swarms*

External control of swarms refers to one of two situations:

1) The swarm is assigned objectives in real time by an external user, especially a human operator.
2) Some or all members of the swarm interact with an adversary or a hostile agent which, in turn, is within a human user's control.

At the simplest level, a human teleoperator sends motion commands to the swarm. In order to reduce the cognitive load on the operator, it is desirable to minimize the number of inputs that the operator must provide and manage. To this end, it is possible to control the bulk motion of the swarm by guiding a single virtual leader and controlling the size and shape of the swarm with respect to this virtual leader [139], [140]. An alternative to using a virtual leader is to use the virtual rigid body framework, developed and demonstrated in [141], [142]. The human could also issue a command in a *language* that the swarm is designed to understand. This is no different conceptually from the usual setting of an autonomous swarm, since it involves the human acting essentially outside the algorithmic loop. It has been argued that humans are able to guide a swarm better using a dynamic set of leaders [143], as compared to manipulating a fixed leader. There is also evidence which suggests that human operators can adapt their handling of (virtual) leaders to guide large swarms through obstacle-rich environments in a better manner than built-in, standard flocking rules [144]. The next level of sophistication involves the human issuing commands using natural language, while still staying outside the algorithmic loop that controls the swarm. Here, the challenge is one of inferring a specific command from the operator's verbiage [145]. The highest explored level of sophistication is using computer to infer human intent. Here, the human is very a much a part of the algorithmic loop: the algorithm that controls the swarm actively seeks input from the human about its performance. [146] proposes a framework to extend this idea to a team of robotic agents (including UAVs).

The concept of adversarial control addresses the case where there is no direct way of tapping into a swarm's command and control algorithm. An example of adversarial control is the family containment and herding strategies modeled after dolphins [21], sheep-dogs [147]–[152] and birds of prey used to herd a flock of birds [153]. In [153], the authors examined the use of a robotic UAV, possibly one built to resemble a bird of prey like a falcon, to herd flocks of birds away from sensitive areas like airports and solar farms. The UAV interacts with the flock by engaging birds located on the boundary of the flock. The herding algorithm make use of the flock's inherent tendency to maintain a cohesive structure to reduce the number of robotic agents required to achieve herding. The perturbation in the velocity of the birds on the boundary of the flock diffuses through the swarm and causes the flock to alter its heading and speed. It has been shown in [153] that a single robotic agent suffices to herd a flock of birds, while related work [150] suggests that the quality of the herding can be improved substantially by using multiple UAVs.

One particular problem of interest in the context of adversarial control is inferring the model underlying the swarm's motion. If the model is known, together with the response of the swarm to an adversary, it would be possible to not just design optimal strategies to divert or control the swarm, but also derive guarantees on the performance of such strategies. In [153], experimental data was used to identify a model, based on [154], [155], for the response of a flock of birds to a UAV located within a certain range of the flock. The approach adopted in [153] works for flocks whose response to perturbations is based on a static, deterministic law. When the response takes a more strategic, dynamic form, it is necessary to use learning-based techniques which expressly account for this behavior [156], [157]. A filter-based technique lies midway between the two sets of aforementioned approaches. Consider the case of missiles where it is known that a target missile follows one of a well-defined set of navigation laws at all times. The exact law and its parameters are unknown. Such problems can be solved efficiently using a bank of filters to determine the most likely model, as demonstrated in [158].

IV. Aerial Distributed Sensing, Monitoring, and Cooperative Mapping

Distributed sensing is one of the main application areas of aerial robotic swarms. Swarms of aerial robots have the ability to simultaneously gather information from disjoint locations. They are also more robust to failures in sensing and actuation since there is some redundancy in the system. Distributed sensing tasks can have three main foci: targets, space, and maps. With any focus, the robots need to have information about the area of interest and the objects within it to safely and successfully complete the task. However, the goal in each sub-task is different. In the first, the goal is to search for and/or track targets inside of an area of interest. In the second, the goal is to maximize some measure of sensor coverage or to ensure that all areas are eventually covered, possibly at a desired frequency. In the last, the goal is to build a map of the unknown or partially-known environment.

A. Target Search and Tracking

Target search and tracking is a canonical distributed sensing task. From an aerial robotics perspective, several key variants of this problem have been studied. The divisions between the variants occur along two main categories: static vs. dynamic targets and single- vs. multi-target. In the multi-target case, there are two sub-cases of a known vs. unknown number of targets. Note that the latter problem can be significantly more difficult: when the number of targets is known, then a detection (or lack thereof) not only gives the team information about what is in the field of view of the sensors, but *also* what is outside of the field of view. For example, if the team knows that there are 8 targets and that 4 of them are currently visible, then they know that there are 4 left to be found. In the case where the number of targets is unknown, then seeing 4 targets only tells the team that there are at least 4 targets.

1) Single, Dynamic Target Estimation: A team of robots has the ability to simultaneously view disjoint regions of an area of interest or to simultaneously view the same region from different perspectives. The former allows the team to more quickly gain global information while the latter allows the team to more quickly decrease uncertainty and to be robust to sensor errors. This problem can be written as a distributed estimation task [159], [160]. In a general discrete-time representation, the target's dynamics are given by:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{w}_k, \Delta)\,,\ \forall k \in \mathbb{N}\,, \tag{8}$$

where $\boldsymbol{f}_k$ is a nonlinear, time-varying function of the target state $\boldsymbol{x}_k$, the independent and identically distributed (i.i.d.) process noise $\boldsymbol{w}_k$, and the discretization time step size Δ. A network of N heterogeneous sensing agents are simultaneously tracking (8). Let $\boldsymbol{y}_k^i$ denote the measurement taken by the i^{th} agent at the k^{th} time instant:

$$\boldsymbol{y}_k^i = \boldsymbol{h}_k^i(\boldsymbol{x}_k, \boldsymbol{v}_k^i),\ \forall i \in \mathcal{V} = \{1, \ldots, N\}\,,\ \forall k \in \mathbb{N}\,, \tag{9}$$

where $\boldsymbol{h}_k^i$ is a nonlinear time-varying function of the state $\boldsymbol{x}_k$ and the i.i.d. measurement noise $\boldsymbol{v}_k^i$. Then, agents are able to use the distributed Bayesian filtering proposed in [159], to track the state of the target (see Algorithm 2). A similar cooperative estimation algorithm can be used to cooperatively map a target region as well to obtain pose estimates of UAVs (see Sec. IV-C).

This type of problem is found in a variety of settings, including tracking a radio-tagged animal with a team of UAVs [161] or seeking, tracking, and capturing a hostile UAV [162]. Additionally, these UAV teams may collaborate with a team

Algorithm 2 Distributed Bayesian Filtering Algorithm

1: Compute the local prior of the target(s)
2: Obtain local measurements and communicate with neighbors
3: Fuse probability distributions using Logarithmic Opinion Pool
4: Compute posteriors and iterate

of ground robots and/or fixed sensors [161]. The problem of target tracking is further complicated when not only the state of the target unknown, but also the state of each UAV [163].

2) Estimation of Multiple Targets of Known Number: The simplest form of multi-target tracking is when the number of targets is known and the targets are stationary [161]. However, tracking dynamic targets has been of greater interest to the research community as it represents a much greater share of practical applications. One common situation for a small team of UAVs is where the number of targets is larger than the number of robots. In this situation the team must decide between focusing on tracking the largest number of targets and tracking individual targets with a high quality of tracking. This tradeoff typically results in a decision about the elevation of the robot, where a high elevation results in a large sensor field-of-view but higher sensor noise [113], [164]. Furthermore, simultaneously planning trajectories for large teams can be computationally expensive and slow. This problem is typically mitigated through the use of approximation algorithms [113], [164] or anytime planning algorithms [165]. In order to cooperatively plan, the robots must be able to share information across the team. In the case where robots have limited communication range, line-of-sight visibility, and operate in a cluttered environment, it can be difficult to maintain connectivity across the team [166].

3) Multiple Targets of Unknown Number: As mentioned above, when the number of targets is unknown the search problem becomes much more difficult and the team must always explore the entire environment in order to complete the task. The standard method used to solve this task is to utilize a quadtree representation to adaptively refine the environment in areas that are likely to contain targets [167]–[169]. The main distinction between these three works is that [167], [168] assume that each robot sees one and only one cell, which implicitly connects the elevation of the robots to the quadtree resolution (and the sensing quality) while [169] allows the robots to see multiple cells and utilize the theory of random finite sets [170] to estimate the set of targets.

Tracking an unknown number of dynamic targets is even more difficult since, barring being able to see the entire environment at one time, there is no way for the team to be sure that they have seen every target. [171] considers the situation where the number of targets is unknown but constant. This focuses on creating an efficient, camera-based tracking for collision avoidance within a large swarm of UAVs, which is run on board UAVs in real time. In a single-team situations, this allows the system to be robust to delays or failures in the communication, and it is also useful in situations where there are multiple, non-communicating teams in the same airspace.

Perhaps the most challenging target search and tracking problem is when the number of targets is unknown and dynamically changes over time, e.g., due to targets entering and leaving the area of interest. The tool most commonly used in this scenario is the PHD filter [170], which allows the team to simultaneously estimate the number of targets and the dynamic state of each target. This has been used by a small team of fixed-wing UAVs to track vehicles on roadways using an information-based technique [172] and by a large team of multi-rotor UAVs to track ground robots using a Voronoi-based controller [173].

B. Surveillance and Monitoring

Target tracking, as the name implies, takes a target-centric approach. The alternative is to take an area-centered approach, where the team of robots focuses on covering an area of interest. This is often called surveillance. If all areas of interest must be visited at some desired, or maximum, frequency, the problem is called persistent monitoring. Surveillance and monitoring have been the subject of a large body of literature, including many aerial swarm-specific approaches.

A surveillance or monitoring task is a tuple $(\mathcal{R}, \gamma, Q)$, where $\mathcal{R}$ is the robot model, γ are the curves followed by the robots, and Q is the set of points of interest [174]. Let $\phi(q,t)$ be the field at point q and time t, which often represents the time elapsed since the point q was last seen by some robot or some local measure of uncertainty about the environment. Then the goal is to find a set of trajectories γ that minimizes the cost

$$\begin{aligned} \gamma^* = \arg\min_{\gamma} & \left(\max_{q\in Q} \left(\limsup_{t\to\infty} \phi(q,t) \right) \right) \qquad (10) \\ \text{subject to} \quad & \limsup_{t\to\infty} \phi(q,t) \text{ is finite } \forall q \in Q \\ & \text{Robot capabilities}, \mathcal{R} \end{aligned}$$

In general, the value of the field $\phi(q,t)$ increases over time and decreases only when a robot observes it. Due to the finite time horizon considered in this problem, it is computationally expensive to solve for robot trajectories, especially in the multi-UAV case.

1) Persistent Monitoring: Early works in multi-UAV persistent monitoring used a heuristic approach to extend traditional single-UAV solutions to multiple UAVs [175], [176]. Other work focused on enabling real-time computation on-board the UAVs by using parameterized B-spline curves to define the set of feasible trajectories [177]. The authors later extended this work to account for the fact that the sensor field of view and the turning radius of fixed-wing UAVs are typically of a comparable length scale, making it difficult to see all parts of the environment. [178].

Another way to think about a monitoring problem is as a vehicle routing problem, where the UAVs must visit a desired set of locations [179].

2) Surveillance: The primary distinction between persistent monitoring and surveillance is that in surveillance there is no hard requirement that each area be visited with a certain frequency. Instead, the goal is often to maximize some measure of coverage or information [180], [181]. The robots in the team communicate over a multi-hop network and solve the surveillance task in a distributed fashion.

In addition to maximizing coverage or information, the swarm can be tasked to monitor a spatio-temporal field over the environment. Such spatio-temporal fields are often found in environmental monitoring and precision agriculture tasks, where the field could be something like water temperature or nutrient concentration. One recent approach to this is Rapidly-exploring Random Cycles (RRCs) [182], which is better suited to surveillance and monitoring tasks where areas must be consistently revisited than its cousin, Rapidly-exploring Random Trees (RRTs) [82], which focus on single-use trajectories. Monitoring spatio-temporal fields is a challenging task and may often be better accomplished by using a heterogeneous team [183].

C. Cooperative Aerial Mapping

In contrast to surveillance and monitoring tasks, where the goal is to only to observe the environment, mapping is the process of acquiring a globally-consistent representation of an environment. Such representations can be sparse [184], semi-dense [185], or fully dense [186]. While dense representations can be directly used for autonomous navigation [186] or geographical reference, sparse representations are often only used for state estimation [187] or collaborative control of robotic agents. Due to the fact that the environment is often only partially known or even totally unknown, mapping tasks are often coupled with localization (pose estimation) issues, which turn them into the classic simultaneous localization and mapping (SLAM) problem. Admittedly, SLAM and its extension to distributed multi-robot SLAM are extensively studied areas. Solving SLAM using multiple distributed sensors (e.g., multiple cameras carried by different aerial robots) is hence related to target tracking and estimation discussed in Sec. IV-A and Algorithm 2. In this paper, we limit our discussion to only those that are most relevant to aerial robot swarms.

The technical contributions of collaborative mapping experiments are limited when they are conducted in simulation or in a lab setting. However, due to the high technical barrier of deploying multiple aerial robots in a real-world setting, a very small number of collaborative mapping systems have so far been tested in realistic settings. Even for successful applications, the scale has been limited to a few (fewer than ten) robots. Further discussion of these technical challenges follows in Sec. V.

Problems and current solutions for multi-robot mapping are reviewed in [188]. In the following, we categorize mapping solutions based on their sensing modalities and representation of the environment as either visual sparse, visual dense, or lidar-based solutions.

1) Visual Sparse Mapping: Visual sparse representation consists of points and lines, which are extracted and tracked from images. Points are usually augmented with descriptors for feature matching purposes. By matching 3-D points and lines, robots can estimate their relative poses and fuse their local maps to maintain geometric consistency and achieve effective cooperation in large-scale environments [189]. Robots may also maintain the position uncertainty of each point in the map for handling of dynamic objects [190]. Early work on vision-based, collaborative SLAM for aerial robots was introduced in [184], in which a centralized ground station was used to collect data from multiple aerial robots. The data was used to perform sparse feature matching for robot localization, and to detect overlap in the sensor field of view of different robots. Recent results utilizing similar mapping frameworks were presented in [191], [192].

Visual-inertial SLAM frameworks are often more suitable for aerial robot systems than other robotic platforms thanks to the guaranteed availability of onboard IMUs. State-of-the-art visual-inertial SLAM frameworks are often able to process multi-session maps [193], [194], making them ideal for merging maps acquired by multiple robots into globally consistent representations. The global localization capability of these frameworks also enables drift-free pose estimation of multiple aerial robots in the same sparse visual map.

Real-world swarm systems typically have very strict constraints in communication bandwidth. To this end, researchers have been focusing on minimizing or limiting the amount of data required to perform decentralized mapping [195], [196]. Specifically, [196] proposes a decentralized SLAM framework based on decentralized place recognition and optimization algorithms. These algorithms scale linearly with respect to the size of the team and build highly compact representations of the environment, resulting in very low bandwidth usage. This enables robots to navigate in environments where absolute positioning is not available, and where there is no central base station. Another approach to decrease bandwidth usage is for the robots to utilize object-based models rather than exchanging raw sensor measurements (e.g., point clouds or RGB-D data) [197].

2) Visual Dense Mapping: Dense mapping systems describe the environment using a dense collection of points or planes. Dense representations are very powerful for autonomous navigation in cluttered environments, but they also pose much higher requirements in terms of processing power and data storage. RGB-D cameras that provide both depth and color information for each image are often used for cooperative visual dense mapping. Due to the high computational load required to process dense maps in real time, robots with limited computational power may choose to send local maps to a cloud server to perform map merging and batch optimization [198], [199]. Recent work demonstrated real-time pose estimation for autonomous flight and cooperative dense mapping using onboard computation with two quadrotors equipped with RGB-D cameras [200], and with a heterogeneous team of a quadrotor and a ground robot [201].

3) Lidar-based Mapping: Lidar is another commonly used sensor for mapping applications. In [202], a small heterogeneous team of a quadrotor and a ground robot is used for cooperative mapping, where the actuation advantages of each agent can be utilized to ensure that the entire space is explored. Scan matching is used for merging maps from the two robots. An expectation maximization (EM) algorithm that utilizes lidar scan information was proposed in [203] for efficient identification of inliers in multi-robot loop closure. This significantly improves the trajectory accuracy over long-term navigation.

V. Technology for Swarming

In this section, we discuss the practicalities of operating a swarm of aerial robots, focusing on the main hardware and software components. Aerial robotic swarms have been studied extensively in simulations, but have not been used in full-scale experimental tests in real-world scenarios until recently. This is due to a confluence of factors. On-board sensing and computation has significantly improved to the point where it is possible to do real-time state estimation. At the same time, drone hardware has significantly improved in recent years with the explosive growth of the commercial drone market. In the US alone, the commercial market grew from \$40M in 2012 to nearly \$1B in 2017 [204] and the global UAV market is expected to surpass \$12B by 2021 [205]. This growth has lowered hardware costs enough to make large-scale swarms possible.

A. Platforms

Some of the first work on aerial robotic swarm hardware focused on providing an open-source hardware and software stack that did not require any external infrastructure [206]. This was meant as an educational tool to teach young scholars to work with hardware and to give them a testbed to implement their ideas, but the swarm only had a handful of robots in it. Other indoor swarms utilize motion capture systems for localization [207], [208]. More recently, the field has focused on expanding the size of the swarm, with one of the largest indoor swarms consisting of 49 CrazyFlie quadrotors simultaneously flying in a motion capture system [209]. The palm-sized CrazyFlie platform does not have sufficient on-board computation or sensing for state estimation, but it is ideal for large-scale, indoor swarms.

Other researchers have focused on getting the swarm outside of the lab, including a swarm of 12 quadrotors working both indoors and outdoors without the need for any external infrastructure [210] and outdoor formation flight of 10 aerial robots [211]. The robots use Visual-Inertial Odometry (VIO) for state estimation, which allows them to navigate in challenging outdoor conditions, including at night and with wind. An even larger outdoor swarm of 50 fixed-wing UAVs was also recently demonstrated [212], with the goal of becoming a testbed to study adversarial swarm systems. Note that popular drones shows performed by Intel[1] or EHang[2] use GPS-based navigation with pre-defined trajectories.

B. Vehicle Power Management

With any swarms, one of the key challenges is power management. For example, in [212] the full 50 robots in the swarm were all simultaneously airborne for only 10 minutes out of the 60 minutes it took for all of the vehicles to be launched and land safely. Unlike fixed-wing UAVs, vertical take-off and landing UAVs, such as multi-rotors, are able to simultaneously take off and land but have a significantly shorter flight time.

Recharging or refueling robots can be done from static charging pads [213], [214] or on mobile charging pads (i.e., on top of ground vehicles) [103]. Health monitoring beyond fuel or battery levels is also important. For example, the operator of the team may also be interested in malfunctions, degradation, or failure of sensors, actuators, and other components [215].

C. Pose and State Estimation

Due to the inherently unstable dynamics of most aerial robot configurations, robust state estimation is essential for almost all aerial robot applications. This is the fundamental building block that enables transition from simulation or lab settings (with external motion capture systems) to real-world deployment. In the following, we categorize state estimation solutions as either based on external sensors or self-contained with on-board sensors.

1) Pose Estimation using External Sensors: External sensing options, such as real-time kinematic (RTK) GPS, optical motion capture systems, and ultra-wideband (UWB) solutions, have enabled impressive cooperative missions on aerial robots. It is well known that GPS, which provides absolute longitude and latitude information, is suitable for large scale outdoor environments. RTK GPS further achieves centimeter-level accuracy with the help of additional base stations. These GPS-based solutions have powered various commercial aerial swarm shows by Intel, EHang, and others. In indoor GPS-denied environments, optical motion capture systems enable millimeter-level position tracking utilizing multiple infrared cameras [216], [217]. Alternatively, UWB-based solutions offer a less expensive and more flexible, but less accurate, state estimate for large-scale indoor aerial swarms [218]. The major drawback of any of these systems is that they require the installation of fixed infrastructure, limiting the swarm to operate in a fixed airspace.

2) Pose Estimation using On-board Sensors: To enable swarms to operate in any environment, one must eliminate the need for external sensors for state estimation. Instead, the robots must rely on on-board sensors, such as cameras, lidars, and inertial measurement units (IMUs). Cameras and lidars are exteroceptive sensors, relying on external features to provide incremental pose estimates [219]. On the other hand, IMUs are interoceptive sensors, providing high-frequency velocity and attitude feedback for the purpose of real-time control. A recent major breakthrough in this area is the use of visual-inertial odometry (VIO) [210] for real-time state estimation and feedback control. On-board camera sensors can also be used to localize other members of the swarm [220]–[225]. This can be used to enable distributed formation control without the need for any explicit communication between agents. However, cooperative estimation and multi-agent SLAM techniques discussed in Sec. IV-C can also be used to provide state and pose estimates of each aerial robot.

D. Communication Infrastructure

The communication infrastructure is another essential building block for real-world deployment of aerial swarm systems, as it enables exchange of state information, motion plans, and

[1]https://www.intel.com/content/www/us/en/technology-innovation/aerial-technology-light-show.html

[2]http://www.ehang.com/news/249.html

high-level swarm behaviors. Researchers often select short-range but low power consumption communication protocols, such as Bluetooth, ultra-wideband (UWB), or standard Wi-Fi, for building up the communication infrastructure. A detailed discussion of these protocols was presented in [226]. However, due to limited bandwidths, these protocols may not satisfy the communication requirement for large-scale swarms. Researchers are looking into possible alternatives that demonstrate low latency, high reliability, and high bandwidth, such as URLLC [227].

Due to a limited selection of physical communication infrastructure components, current swarm realizations are limited to using one of a small number of communication topologies. Centralized topologies with one ground station and multiple agents are used for most cases [209], [216], [217]. Decentralized communication topologies are still mostly used in the realm of theoretical research [57], [59]. Very limited experimental results are presented in the literature [228], [229].

VI. Conclusion and Future Work

In the near future, our airspace will be populated by swarms of aerial robots, performing complex tasks that would be impossible for a single vehicle. This papers reviews work that could provide the fundamental algorithmic, analytic, perceptive, and technological building blocks necessary to realize this future. The research issues discussed in this survey paper span hierarchical integration of swarm synchronization control with safe trajectory optimization and assignment, and cooperative estimation and control with perception in the loop, offering the readers a broad perspective on aerial swarm robotics.

In addition, we emphasize the importance of the three-way tradeoff between computational efficiency, stability and robustness, and optimal system performance. To truly address this tradeoff, we argue that it is imperative to advance beyond methods that are currently being used in autonomous drones and general swarm robotics in order to realize long-term autonomy of aerial swarm systems.

One important area of further study is to develop learning and decision-making architectures that will endow swarms of aerial robots with high levels of autonomy and flexibility. We argue that such architectures will ultimately lead to reduced risk and cost as well as long-term autonomous operations. To be successful, any such architecture must provide the framework for reasoning about the wide-ranging nature of uncertainties and modeling errors, ranging from *known* unknowns (e.g., sensor and actuator noise) to *unknown* unknowns (e.g., wind disturbance, hardware failures). All of these impact the safety and robustness of algorithms and system-level functions of swarm behaviors. Furthermore, computation and communication within a swarm must be fast enough to ensure stability under model changes and mission specifications at the various timescales and bandwidths within the system.

For aerial swarm systems with highly uncertain environmental models, the role of high-level planning, decision making, and classification in flight in conjunction with low-level swarm control and estimation systems can be characterized mathematically through the properties of stability, convergence, and robustness. Various aspects of the swarm decision-making, control, and estimation should come in different timescales and hierarchical levels to exploit scalability and computational efficiency. An example of such characterization on stability would be a mathematical theorem correlating desired models and parameters to be updated on-line as well as their update or learning rates, to functions of various system features, such as sampling rate, swarm control law update rate, bandwidth of dynamics and communication, dimensions of dynamic systems, and properties of environmental uncertainties. This should also provide a guideline as to gauge how efficient and robust a particular swarm algorithm or system-level architecture is at achieving autonomy in aerial swarms. For example, distributed optimal planning (e.g., [25], [87]) requires robots to share their optimal solutions with their neighbors, up to a certain time horizon. Adding simultaneous target or task allocation to this problem further increases the required size of communicated information. It would be beneficial to combine such methods with on-line adaptation methods that can forecast the neighbors' future behavior and would, in turn, effectively reduce communication requirements. The key idea is again combining formal mathematical analysis with the hierarchical and multi-modal decomposition discussed earlier. Another important area is to establish rigorous methodologies for fault detection, isolation, and recovery to handle various potential faults occurring at sub-system levels, individual system levels, and swarm levels.

As swarms are deployed to a greater extent for aggressive or agile autonomous missions, it will become necessary to create the means to exert some form of adversarial control on swarms. Such counter-swarm techniques can also be used for civilian purposes, such as maintaining law and order and herding birds and animals away from environmental hazards such as floods or wildfires. The work reported in Sec. III-E is a good starting point for these techniques. Key open questions include the type of maneuvers that need to be performed to rapidly estimate a swarm's location and intent; assess an aerial swarm's internal dynamics; identify the task and role assignment within a given swarm; and identify the primary leader and sensing nodes. The next level of questions pertain to identifying ways of defeating these types of probing maneuvers from an adversarial swarm, which is a direct analogue of the usual minimax paradigm for games. It is interesting to note the similarities exhibited in the case of social networks, which suggests that an adoption of the tools from that literature may provide early breakthroughs for counter-swarm development. Yet, even by adopting well-established tools from the theory of social networks and games, an important and significant challenge at both levels is identifying the role of the aerial vehicle dynamics in enabling, and defeating, the probing operations. A cleverly executed set of maneuvers could help identify, and equally provide deceptive leads about, a swarm's intent, organization and capabilities.

In summary, many open problems and research issues in aerial swarm robotics involve the characterization of the interdependencies between the properties of swarm vehicle dynamics, the properties of uncertainties, and different swarm learning/control methods employed. Only by understanding these interdependencies, either through careful system identification or integrated system design, can fully-autonomous aerial swarms be proven to operate in complex, real-world environments.

Acknowledgment

The authors very much appreciate Frank Park's guidance and encouragement. The authors thank Salar Rahili for his comments and Lily Kass for her editorial assistance.

References

[1] F. Y. Hadaegh, S.-J. Chung, and H. M. Manohara, "On development of 100-gram-class spacecraft for swarm applications," *IEEE Sys. J.*, vol. 10, no. 2, pp. 673–684, 2016.

[2] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, 2013.

[3] R. M. Murray, "Recent research in cooperative control of multivehicle systems," *J. Dynamic Syst. Meas. Control*, vol. 129, no. 5, pp. 571–583, 2007.

[4] S. Bandyopadhyay, R. Foust, G. P. Subramanian, S.-J. Chung, and F. Y. Hadaegh, "Review of formation flying and constellation missions using nanosatellites," *J. Spacecraft and Rockets*, vol. 53, no. 3, pp. 567–578, 2016.

[5] D. P. Scharf, F. Y. Hadaegh, and S. R. Ploen, "A survey of spacecraft formation flying guidance and control. part ii: control," in *Proc. Amer. Control Conf.*, vol. 4, 2004, pp. 2976–2985.

[6] M. A. Joordens and M. Jamshidi, "Consensus control for a system of underwater swarm robots," *IEEE Syst. J.*, vol. 4, no. 1, pp. 65–73, 2010.

[7] H. Balakrishnan, "Control and optimization algorithms for air transportation systems," *Annual Rev. Control*, vol. 41, pp. 39–46, 2016.

[8] A. M. Bayen, R. L. Raffard, and C. J. Tomlin, "Adjoint-based control of a new eulerian network model of air traffic flow," *IEEE Trans. Control Sys. Tech.*, vol. 14, no. 5, pp. 804–818, 2006.

[9] P. K. Menon, G. D. Sweriduk, and K. D. Bilimoria, "New approach for modeling, analysis, and control of air traffic flow," *J. Guid. Control Dyn.*, vol. 27, no. 5, pp. 737–744, 2004.

[10] C. Tomlin, G. J. Pappas, and S. Sastry, "Conflict resolution for air traffic management: A study in multiagent hybrid systems," *IEEE Trans. Autom. Control*, vol. 43, no. 4, pp. 509–521, 1998.

[11] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artificial Intell. Res.*, vol. 4, pp. 237–285, 1996.

[12] S. M. LaValle, *Planning algorithms*. Cambridge University Press, 2006.

[13] H. Khalil, *Nonlinear Systems*, ser. Pearson Education. Prentice Hall, 2002.

[14] A. A. Paranjape, S.-J. Chung, and J. Kim, "Novel dihedral-based control of flapping-wing aircraft with application to perching," *IEEE Trans. Robot.*, vol. 29, no. 5, pp. 1071–1084, 2013.

[15] D. P. Bertsekas, "Auction algorithms for network flow problems: A tutorial introduction," *Computational Optimization and Applications*, vol. 1, no. 1, pp. 7–66, 1992.

[16] M. M. Zavlanos, L. Spesivtsev, and G. J. Pappas, "A distributed auction algorithm for the assignment problem," in *Proc. IEEE Conf. Decis. Control*, 2008, pp. 1212–1217.

[17] P. Seiler, A. Pant, and J. Hedrick, "A systems interpretation for observations of bird V-formations," *J. Theoretical Biology*, vol. 221, no. 2, pp. 279–287, 2003.

[18] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4, pp. 25–34, 1987.

[19] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Phys. Rev. Lett.*, vol. 75, pp. 1226–1229, 1995.

[20] G. Ferrari-Trecate, A. Buffa, and M. Gati, "Analysis of coordination in multi-agent systems through partial difference equations," *IEEE Trans. Autom. Control*, vol. 51, no. 6, pp. 1058–1063, 2006.

[21] M. Ji, G. Ferrari-Trecate, M. Egerstedt, and A. Buffa, "Containment control in mobile networks," *IEEE Trans. Autom. Control*, vol. 53, no. 8, pp. 1972–1975, 2008.

[22] J. Toner and Y. Tu, "Long-range order in a two-dimensional dynamical xy model: How birds fly together," *Phys. Rev. Lett.*, vol. 75, no. 3, pp. 4326–4329, 1995.

[23] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, "Probabilistic and distributed control of a large-scale swarm of autonomous agents," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1103–1123, 2017.

[24] M. Turpin, N. Michael, and V. Kumar, "An approximation algorithm for time optimal multi-robot routing," in *Algorithmic Foundations of Robotics XI*, ser. Springer Tracts in Advanced Robotics, vol. 107. Springer, Cham, 2015, pp. 627–640.

[25] D. Morgan, G. P. Subramanian, S.-J. Chung, and F. Y. Hadaegh, "Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and sequential convex programming," *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1261–1285, 2016.

[26] N. Chopra, M. W. Spong, and R. Lozano, "Synchronization of bilateral teleoperators with time delay," *Automatica*, vol. 44, no. 8, pp. 2142–2148, 2008.

[27] S.-J. Chung and J.-J. E. Slotine, "Cooperative robot control and concurrent synchronization of Lagrangian systems," *IEEE Trans. Robot.*, vol. 25, no. 3, pp. 686–700, 2009.

[28] S.-J. Chung, S. Bandyopadhyay, I. Chang, and F. Y. Hadaegh, "Phase synchronization control of complex networks of Lagrangian systems on adaptive digraphs," *Automatica*, vol. 49, no. 5, pp. 1148–1161, 2013.

[29] W. Ren, "Formation keeping and attitude alignment for multiple spacecraft through local interactions," *J. Guid. Control Dyn.*, vol. 30, no. 2, pp. 633–638, 2007.

[30] S.-J. Chung, U. Ahsun, and J.-J. E. Slotine, "Application of synchronization to formation flying spacecraft: Lagrangian approach," *J. Guid. Control Dyn.*, vol. 32, no. 2, pp. 512–526, 2009.

[31] S.-J. Chung and M. Dorothy, "Neurobiologically inspired control of engineered flapping flight," *J. Guid. Control Dyn.*, vol. 33, no. 2, pp. 440–453, 2010.

[32] A. Ramezani, S.-J. Chung, and S. Hutchinson, "A biomimetic robotic platform to study flight specializations of bats," *Science Robotics*, vol. 2, no. 3, p. eaal2505, 2017.

[33] A. A. Paranjape, K. C. Meier, X. Shi, S.-J. Chung, and S. Hutchinson, "Motion primitives and 3D path planning for fast flight through a forest," *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 357–377, March 2015.

[34] A. J. Calise and D. Preston, "Swarming/flocking and collision avoidance for mass airdrop of autonomous guided parafoils," *J. Guid. Control Dyn.*, vol. 31, no. 4, pp. 1123–1132, 2008.

[35] J. 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. 88–1001, 2003.

[36] H. Su, X. Wang, and Z. Lin, "Flocking of multi-agents with a virtual leader," *IEEE Trans. Autom. Control*, vol. 54, no. 2, pp. 293–307, 2009.

[37] S. Ghapani, J. Mei, W. Ren, and Y. Song, "Fully distributed flocking with a moving leader for Lagrange networks with parametric uncertainties," *Automatica*, vol. 67, pp. 67–76, 2016.

[38] Q.-C. Pham, N. Tabareau, and J.-J. E. Slotine, "A contraction theory approach to stochastic incremental stability," *IEEE Trans. Autom. Control*, vol. 54, no. 4, pp. 816–820, 2009.

[39] C.-T. Lin, "Structural controllability," *IEEE Trans. Autom. Control*, vol. 19, no. 3, pp. 201–208, June 1974.

[40] E. J. Davison, "Connectability and structural controllability of composite systems," *Automatica*, vol. 13, no. 2, pp. 109–123, 1977.

[41] S. Hosoe, "Determination of generic dimensions of controllable subspaces and its application," *IEEE Trans. Autom. Control*, vol. 25, no. 6, pp. 1192–1196, 1980.

[42] K. Murota and S. Poljak, "Note on a graph-theoretic criterion for structural output controllability," *IEEE Trans. Autom. Control*, vol. 35, no. 8, pp. 939–942, 1990.

[43] H. Mayeda and T. Yamada, "Strong structural controllability," *SIAM J. Control Optim.*, vol. 17, no. 1, pp. 123–138, 1979.

[44] G. Reissig, C. Hartung, and F. Svaricek, "Strong structural controllability and observability of linear time-varying systems," *IEEE Trans. Autom. Control*, vol. 59, no. 11, pp. 3087–3092, 2014.

[45] N. Monshizadeh, S. Zhang, and M. K. Camlibel, "Zero forcing sets and controllability of dynamical systems defined on graphs," *IEEE Trans. Autom. Control*, vol. 59, no. 9, pp. 2562–2567, 2014.

[46] Y.-Y. Liu, J.-J. E. Slotine, and A.-L. Barabasi, "Controllability of complex networks," *Nature*, vol. 473, no. 7346, pp. 167–173, 2011.

[47] A. Rahmani, M. Ji, M. Mesbahi, and M. Egerstedt, "Controllability of multi-agent systems from a graph-theoretic perspective," *SIAM J. Control Optim.*, vol. 48, no. 1, pp. 162–186, 2009.

[48] C. O. Aguilar and B. Gharesifard, “Graph controllability classes for the laplacian leader-follower dynamics,” *IEEE Trans. Autom. Control*, vol. 60, no. 6, pp. 1611–1623, 2015.

[49] ——, “Almost equitable partitions and new necessary conditions for network controllability,” *Automatica*, vol. 80, pp. 25–31, 2017.

[50] G. Parlangeli and G. Notarstefano, “On the reachability and observability of path and cycle graphs,” *IEEE Trans. Autom. Control*, vol. 57, no. 3, pp. 743–748, 2012.

[51] X. Chen, M.-A. Belabbas, and T. Basar, “Controllability of formations over directed time-varying graphs,” *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 3, pp. 407–416, 2017.

[52] N. J. Cowan, E. J. Chastain, D. A. Vilhena, J. S. Freudenberg, and C. T. Bergstrom, “Nodal dynamics, not degree distributions, determine the structural controllability of complex networks,” *PLOS One*, vol. 7, no. 6, p. e38398, 2012.

[53] S. L. Tawaid and S. L. Smith, “Submodularity and greedy algorithms in sensor scheduling for linear dynamical systems,” *Automatica*, vol. 61, pp. 282–288, 2015.

[54] T. H. Summers, F. L. Cortesi, and J. Lygeros, “On submodularity and controllability in complex dynamical systems,” *IEEE Trans. Netw. Syst.*, vol. 3, no. 1, pp. 91–101, 2016.

[55] A. Clark, B. Alomair, L. Bushnell, and R. Poovendran, “Submodularity in input node selection for networked linear systems,” *IEEE Control Syst. Mag.*, vol. 37, no. 6, pp. 52 – 74, 2017.

[56] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, Princeton, 2010.

[57] 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.

[58] L. Moreau, “Stability of multiagent systems with time-dependent communication links,” *IEEE Trans. Autom. Control*, vol. 50, no. 2, pp. 169 – 182, 2005.

[59] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, “Flocking in fixed and switching networks,” *IEEE Trans. Autom. Control*, vol. 52, no. 5, pp. 863–868, 2007.

[60] F. Cucker and S. Smale, “Emergent behavior in flocks,” *IEEE Trans. Autom. Control*, vol. 52, no. 5, pp. 852–862, 2007.

[61] R. Olfati-Saber, “Flocking for multi-agent dynamic systems: Algorithms and theory,” *IEEE Trans. Autom. Control*, vol. 51, no. 3, pp. 401–420, 2006.

[62] D. V. Dimarogonas and K. H. Johansson, “On the stability of distance-based formation control,” in *Proc. IEEE Conf. Decis. Control*, 2008, pp. 1200–1205.

[63] E. Nuno, R. Ortega, L. Basanez, and D. Hill, “Synchronization of networks of nonidentical Euler-Lagrange systems with uncertain parameters and communication delays,” *IEEE Trans. Autom. Control*, vol. 56, no. 4, pp. 935–941, 2011.

[64] M. Belabbas, “On global stability of planar formations,” *IEEE Trans. Autom. Control*, vol. 58, no. 8, pp. 2148–2153, 2013.

[65] J. A. Carrillo, Y. Huang, and S. Martin, “Nonlinear stability of flock solutions in second-order swarming models,” *Nonlinear Analysis: Real World Applications*, vol. 17, pp. 332 – 343, 2014.

[66] D. Zelazo, A. Rahmani, and M. Mesbahi, “Agreement via the edge Laplacian,” in *Proc. IEEE Conf. Decis. Control*, 2007, pp. 2309–2314.

[67] M. Arcak, “Passivity as a design tool for group coordination,” *IEEE Trans. Autom. Control*, vol. 52, no. 8, pp. 1380–1390, 2007.

[68] T. Hatanaka, Y. Igarashi, M. Fujita, and M. W. Spong, “Passivity-based pose synchronization in three dimensions,” *IEEE Trans. Autom. Control*, vol. 57, no. 2, pp. 360–375, 2012.

[69] I.-A. F. Ihle, M. Arcak, and T. I. Fossen, “Passivity-based designs for synchronized path-following,” *Automatica*, vol. 43, no. 9, pp. 1508–1518, 2007.

[70] D. Nesic and A. R. Teel, “Input-to-state stability of networked control systems,” *Automatica*, vol. 40, no. 12, pp. 2121–2128, 2004.

[71] B. S. Rüffer, C. M. Kellett, and S. R. Weller, “Connection between cooperative positive systems and integral input-to-state stability of large-scale systems,” *Automatica*, vol. 46, no. 6, pp. 1019–1027, 2010.

[72] W. Lohmiller and J.-J. E. Slotine, “On contraction analysis for non-linear systems,” *Automatica*, vol. 34, no. 6, pp. 683–696, 1998.

[73] Q.-C. Pham and J.-J. E. Slotine, “Stable concurrent synchronization in dynamic system networks,” *Neural Networks*, vol. 20, no. 1, pp. 62–77, 2007.

[74] A. P. Dani, S.-J. Chung, and S. Hutchinson, “Observer design for stochastic nonlinear systems via contraction-based incremental stability,” *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 700–714, 2015.

[75] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson, “Distributed event-triggered control for multi-agent systems,” *IEEE Trans. Autom. Control*, vol. 57, no. 5, pp. 1291–1297, 2012.

[76] C. Nowzari and J. Cortés, “Distributed event-triggered coordination for average consensus on weight-balanced digraphs,” *Automatica*, vol. 68, pp. 237–244, 2016.

[77] S. Kia, J. Cortés, and S. Martínez, “Distributed event-triggered communication for dynamic average consensus in networked systems,” *Automatica*, vol. 59, pp. 112–119, 2015.

[78] D. Liuzza, D. Dimarogonas, M. D. Bernardo, and K. Johansson, “Distributed model based event-triggered control for synchronization of multi-agent systems,” *Automatica*, vol. 73, pp. 1–7, 2016.

[79] L. Marconi *et al*, “The SHERPA project: Smart collaboration between humans and ground-aerial robots for improving rescuing activities in alpine environments,” in *IEEE Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2012, pp. 1–4.

[80] M. Turpin, N. Michael, and V. Kumar, “CAPT: Concurrent assignment and planning of trajectories for multiple robots,” *Int. J. Robot. Res.*, vol. 33, no. 1, pp. 98–112, 2014.

[81] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, 1996.

[82] S. M. LaValle and J. J. Kuffner Jr, “Randomized kinodynamic planning,” *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, 2001.

[83] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.

[84] W. B. Dunbar and R. M. Murray, “Distributed receding horizon control for multi-vehicle formation stabilization,” *Automatica*, vol. 42, no. 4, pp. 549–558, 2006.

[85] M. G. Earl and R. D’andrea, “Iterative MILP methods for vehicle-control problems,” *IEEE Trans. Robot.*, vol. 21, no. 6, pp. 1158–1167, 2005.

[86] A. Richards, T. Schouwenaars, J. P. How, and E. Feron, “Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming,” *J. Guid. Control Dyn.*, vol. 25, no. 4, pp. 755–764, 2002.

[87] D. Morgan, S.-J. Chung, and F. Y. Hadaegh, “Model predictive control of swarms of spacecraft using sequential convex programming,” *J. Guid. Control Dyn.*, vol. 37, no. 6, pp. 1725–1740, 2014.

[88] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein, “The complexity of decentralized control of markov decision processes,” *Mathematics of Operations Research*, vol. 27, no. 4, pp. 819–840, 2002.

[89] S. Omidshafiei, A.-A. Agha-Mohammadi, C. Amato, S.-Y. Liu, J. P. How, and J. Vian, “Decentralized control of multi-robot partially observable markov decision processes using belief space macro-actions,” *Int. J. Robot. Res.*, vol. 36, no. 2, pp. 231–258, 2017.

[90] J. Yu and S. M. LaValle, “Optimal multirobot path planning on graphs: Complete algorithms and effective heuristics,” *IEEE Trans. Robot.*, vol. 32, no. 5, pp. 1163–1177, 2016.

[91] W. Hönig, T. S. Kumar, L. Cohen, H. Ma, H. Xu, N. Ayanian, and S. Koenig, “Multi-agent path finding with kinematic constraints,” in *Proc. Int. Conf. Autom. Planning Scheduling*, 2016, pp. 477–485.

[92] S. Tang, J. Thomas, and V. Kumar, “Hold or take optimal plan (HOOP): A quadratic programming approach to multi-robot trajectory generation,” *Int. J. Robot. Res.*, p. 0278364917741532, 2018.

[93] M. Egerstedt and X. Hu, “Formation constrained multi-agent control,” *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 947–951, 2001.

[94] E. Xargay, I. Kaminer, A. Pascoal, N. Hovakimyan, V. D. V. Cichella, A. P. Aguiar, and R. Ghabcheloo, “Time-critical cooperative path following of multiple unmanned aerial vehicles over time-varying networks,” *J. Guid. Control Dyn.*, vol. 36, no. 2, pp. 499–516, 2013.

[95] V. Cichella, I. Kaminer, V. Dobrokhodov, E. Xargay, R. Choe, N. Hovakimyan, A. P. Aguiar, and A. M. Pascoal, “Cooperative path following of multiple multirotors over time-varying networks,” *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 3, pp. 945–957, 2015.

[96] S. Mastellone, D. M. Stipanovic, C. R. Graunke, K. A. Intlekofer, and M. W. Spong, “Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments,” *Int. J. Robot. Res.*, vol. 27, no. 1, pp. 107–126, 2008.

[97] R. S. Smith and F. Y. Hadaegh, “Control of deep-space formation-flying spacecraft; relative sensing and switched information,” *J. Guid. Control Dyn.*, vol. 28, no. 1, pp. 106–114, 2005.

[98] A. Yamashita, T. Arai, J. Ota, and H. Asama, “Motion planning of multiple mobile robots for cooperative manipulation and transportation,” *IEEE Trans. Robot. Autom.*, vol. 19, no. 2, pp. 223–237, 2003.

[99] A. Braithwaite, T. Alhinai, M. Haas-Heger, E. McFarlane, and M. Kovac, "Tensile web construction and perching with nano aerial vehicles," in *Robotics Research*. Springer, 2018, pp. 71–88.

[100] Z. Yu, Y. Qu, and Y. Zhang, "Safe control of trailing uav in close formation flight against actuator fault and wake vortex effect," *Aerospace Science and Technology*, vol. 77, pp. 189–205, 2018.

[101] J. A. Preiss, W. Hönig, N. Ayanian, and G. S. Sukhatme, "Downwash-aware trajectory planning for large quadcopter teams," *arXiv preprint arXiv:1704.04852*, 2017.

[102] M. S. Hemati, J. D. Elredge, and J. L. Speyer, "Wake sensing for aircraft formation flight," *J. Guid. Control Dyn.*, vol. 37, no. 2, pp. 513–524, 2014.

[103] K. Yu, A. K. Budhiraja, and P. Tokekar, "Algorithms for routing of unmanned aerial vehicles with mobile recharging stations," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018. [Online]. Available: https://arxiv.org/abs/1704.00079

[104] S. D. Bopardikar, F. Bullo, and J. P. Hespanha, "A cooperative homicidal chauffeur game," *Automatica*, vol. 45, no. 7, pp. 1771–1777, 2009.

[105] V. Shaferman, "Near optimal evasion from acceleration estimating pursuer," in *Proc. AIAA Guid., Navigat. Control Conf.*, 2017, aIAA Paper 2017-1015.

[106] V. Shaferman and Y. Oshman, "Cooperative interception in a multi-missile engagement," in *Proc. AIAA Guid., Navigat. Control Conf.*, 2009, aIAA Paper 2009-5783.

[107] I.-S. Jeon, J.-I. Lee, and M.-J. Tahk, "Homing guidance law for cooperative attack of multiple missiles," *J. Guid. Control Dyn.*, vol. 33, no. 1, pp. 275–280, 2010.

[108] V. Shaferman and T. Shima, "Unmanned aerial vehicles cooperative tracking of moving ground target in urban environments," vol. 31, no. 5, pp. 1360–1371, 2008.

[109] D. P. Bertsekas, "The auction algorithm: A distributed relaxation method for the assignment problem," *Annals Operations Res.*, vol. 14, no. 1, pp. 105–123, 1988.

[110] D. P. Bertsekas and D. A. Castañon, "Parallel synchronous and asynchronous implementations of the auction algorithm," *Parallel Computing*, vol. 17, no. 6-7, pp. 707–732, 1991.

[111] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE Trans. Robot.*, vol. 25, no. 4, pp. 912–926, 2009.

[112] J. Yu, S.-J. Chung, and P. G. Voulgaris, "Target assignment in robotic networks: Distance optimality guarantees and hierarchical strategies," *IEEE Trans. Autom. Control*, vol. 60, no. 2, pp. 327–341, 2015.

[113] Y. Sung, A. K. Budhiraja, R. Williams, and P. Tokekar, "Distributed simultaneous action and target assignment for multi-robot multi-target tracking," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018. [Online]. Available: https://arxiv.org/abs/1706.02245

[114] J. Cortes, S. Martinez, T. Karatas, , and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robot. Autom.*, vol. 20, no. 2, pp. 243–255, 2004.

[115] M. Schwager, J.-J. E. Slotine, and D. Rus, "Consensus learning for distributed coverage control," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2008, pp. 1042–1048.

[116] M. Schwager, D. Rus, and J.-J. E. Slotine, "Decentralized, adaptive coverage control for networked robots," *Int. J. Robot. Res.*, vol. 28, no. 3, pp. 357–375, 2009.

[117] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, "Probabilistic swarm guidance using optimal transport," in *IEEE Conf. Control Applications*, 2014, pp. 498–505.

[118] S. B. Mehdi, V. Cichella, T. Marinho, and N. Hovakimyan, "Collision avoidance in multi-vehicle cooperative missions using speed adjustment," in *Proc. IEEE Conf. Decis. Control*, 2017, pp. 2152–2157.

[119] S. Akella and S. Hutchinson, "Coordinating the motions of multiple robots with specified trajectories," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, 2002, pp. 624–631.

[120] S. Bandyopadhyay, F. Baldini, R. Foust, A. Rahmani, J.-P. de la Croix, S.-J. Chung, and F. Y. Hadaegh, "Distributed spatiotemporal motion planning for spacecraft swarms in cluttered environments," in *AIAA SPACE and Astronautics Forum and Exposition*, 2017, p. 5323.

[121] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, no. 1, pp. 90–98, 1986.

[122] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Trans. Robot. Autom.*, vol. 8, no. 5, pp. 501–518, 1992.

[123] N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Proc. IEEE Conf. Decis. Control*, vol. 3, 2001, pp. 2968–2973.

[124] M. T. Wolf and J. W. Burdick, "Artificial potential functions for highway driving with collision avoidance," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2008, pp. 3731–3736.

[125] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Trans. Robot.*, vol. 33, no. 3, pp. 661–674, 2017.

[126] D. Panagou, D. M. Stipanović, and P. G. Voulgaris, "Distributed coordination control for multi-robot networks using lyapunov-like barrier functions," *IEEE Trans. Autom. Control*, vol. 61, no. 3, pp. 617–632, 2016.

[127] T. Mylvaganam, M. Sassano, and A. Astolfi, "A differential game approach to multi-agent collision avoidance," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 4229–4235, 2017.

[128] T. Mylvaganam and M. Sassano, "Autonomous collision avoidance for wheeled mobile robots using a differential game approach," *European J. Control*, vol. 40, pp. 53–61, 2018.

[129] N. Michael, J. Fink, and V. Kumar, "Cooperative manipulation and transportation with aerial robots," *Autonomous Robots*, vol. 30, no. 1, pp. 73–86, 2011.

[130] T. Lee, K. Sreenath, and V. Kumar, "Geometric control of cooperating multiple quadrotor UAVs with a suspended payload," in *Proc. IEEE Conf. Decis. Control*. IEEE, 2013, pp. 5510–5515.

[131] S. Dai, T. Lee, and D. S. Bernstein, "Adaptive control of a quadrotor UAV transporting a cable-suspended load with unknown mass," in *Proc. IEEE Conf. Decis. Control*. IEEE, 2014, pp. 6149–6154.

[132] M. Tognon, C. Gabellieri, L. Pallottino, and A. Franchi, "Aerial co-manipulation with cables: The role of internal force for equilibria, stability, and passivity," *IEEE Robot. Autom. Lett.*, 2018.

[133] "Project wing project wing," 2017. [Online]. Available: https://x.company/wing/

[134] Q. Lindsey, D. Mellinger, and V. Kumar, "Construction of cubic structures with quadrotor teams," *Proceedings of Robotics: Science & Systems VII*, 2011.

[135] D. Mellinger, M. Shomin, N. Michael, and V. Kumar, "Cooperative grasping and transport using multiple quadrotors," in *Distributed Autonomous Robotic Systems*. Springer, 2013, pp. 545–558.

[136] G. Loianno and V. Kumar, "Cooperative transportation using small quadrotors using monocular vision and inertial sensing," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 680–687, 2018.

[137] G. Loianno, V. Spurny, J. Thomas, T. Baca, D. Thakur, D. Hert, R. Penicka, T. Krajnik, A. Zhou, A. Cho *et al.*, "Localization, grasping, and transportation of magnetic objects by a team of mavs in challenging desert-like environments," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1576–1583, 2018.

[138] "Amazon prime air." [Online]. Available: https://www.amazon.com/Amazon-Prime-Air/b?ie=UTF8&node=8037720011

[139] N. Ayanian, A. Spielberg, M. Arbesfeld, J. Strauss, and D. Rus, "Controlling a team of robots from a single input," in *Proc. IEEE Int. Conf. Robot. Autom.*, Hong Kong, Jun 2014.

[140] D. Lee, A. Franchi, H. I. Son, C. Ha, H. H. Bülthoff, and P. R. Giordano, "Semiautonomous haptic teleoperation control architecture of multiple unmanned aerial vehicles," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 4, pp. 1334–1345, 2013.

[141] D. Zhou and M. Schwager, "Virtual rigid bodies for coordinated agile maneuvering of teams of micro aerial vehicles," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 1737–1742.

[142] ——, "Assistive collision avoidance for quadrotor swarm teleoperation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 1249–1254.

[143] P. Walker, S. A. Amraii, N. Chakraborty, M. Lewis, and K. Sycara, "Human control of robot swarms with dynamic leaders," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 1108–1113.

[144] A. Kolling, K. Sycara, S. Nunnally, and M. Lewis, "Human-swarm interaction: An experimental study of two types of interaction with foraging swarms," *J. Hum.-Robot Interact.*, vol. 2, no. 2, pp. 103–129, 2013.

[145] A. C. Trujillo, J. Puig-Navarro, S. B. Mehdi, and A. K. McQuarry, "Using natural language to enable mission managers to control multiple heterogeneous UAVs," in *Advances in Human Factors in Robots and Unmanned Systems*, ser. Advances in Intelligent Systems and Computing, P. Savage-Knepshield and J. Chen, Eds., vol. 499, 2017.

[146] S. Kim and T. P. Coleman, in 5^{th} *International IEEE/EMBS Conference on Neural Engineering (NER)*, 2011, pp. 605–608.

[147] M. Haque, A. Rahmani, and M. Egerstedt, "A hybrid, multi-agent model for foraging bottlenose dolphins," in *Analysis and Design of Hybrid Systems, University of Zaragoza, Spain*, 2009, pp. 262 – 267.

[148] R. Vaughan, N. Sumpter, J. Henderson, A. Frost, and S. Cameron, "Experiments in automatic flock control," *Robot. Auton. Syst.*, vol. 31, no. 1, pp. 109 – 117, 2000.

[149] J.-M. Lien, B. Bayazit, R. T. Sowell, S. Rodriguez, and N. M. Amato, "Shepherding behaviors," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 4, 2004, pp. 4159–4164.

[150] J.-M. Lien, S. Rodriguez, J.-P. Malric, and N. M. Amato, "Shepherding behaviors with multiple shepherds," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2005, pp. 3402–3407.

[151] D. Strömbom, R. P. Mann, A. M. Wilson, S. Hailes, A. J. Morton, D. J. Sumpter, and A. J. King, "Solving the shepherding problem: heuristics for herding autonomous, interacting agents," *Journal of The Royal Society Interface*, vol. 11, no. 100, p. 20140719, 2014.

[152] A. Pierson and M. Schwager, "Bio-inspired non-cooperative multi-robot herding," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2015, pp. 1843–1849.

[153] A. A. Paranjape, S.-J. Chung, K. Kim, and D. H. Shim, "Robotic herding of a flock of birds using an unmanned aerial vehicle," *IEEE Trans. Robot.*, 2018, in press.

[154] M. Zheng, Y. Kashimori, O. Hoshino, K. Fujita, and T. Kambara, "Behavior pattern (innate action) of individuals in fish schools generating efficient collective evasion from predation," *J. Theoretical Biology*, vol. 235, no. 2, pp. 153–167, 2005.

[155] S.-H. Lee, H. Pak, and T.-S. Chon, "Dynamics of prey-flock escaping behavior in response to predator's attack," *J. Theor. Biology*, vol. 240, no. 2, pp. 250–259, 2006.

[156] D. Carmel and S. Markovitch, "Opponent modeling in multi-agent systems," in *Int. Joint Conf. Artificial Intelligence*. Springer, 1995, pp. 40–52.

[157] ——, "Exploration strategies for model-based learning in multi-agent systems: Exploration strategies," *Autonomous Agents and Multi-Agent Systems*, vol. 2, no. 2, pp. 141–172, 1999.

[158] V. Shaferman and T. Shima, "Cooperative multiple-model adaptive guidance for an aircraft defending missile," *J. Guid. Control Dyn.*, vol. 33, no. 6, pp. 1801–1813, 2010.

[159] S. Bandyopadhyay and S.-J. Chung, "Distributed Bayesian filtering using logarithmic opinion pool for dynamic sensor networks," *Automatica*, 2018, to appear.

[160] M. F. Ridley, B. Upcroft, and S. Sukkarieh, "Data fusion and tracking with multiple UAVs," in *Handbook of Unmanned Aerial Vehicles*. Springer, 2015, pp. 461–490.

[161] H. Bayram, N. Stefas, K. S. Engin, and V. Isler, "Tracking wildlife with multiple UAVs: System design, safety and field experiments," in *2017 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, 2017, pp. 97–103.

[162] S. Buerger, J. R. Salton, D. K. Novick, R. Fierro, A. Vinod, B. Hom-Chaudhuri, and M. Oishi, "Reachable set computation and tracking with multiple pursuers for the asap counter-uas capability." Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), Tech. Rep., 2016.

[163] K. Hausman, J. Mueller, A. Hariharan, N. Ayanian, and G. Sukhatme, "Cooperative control for target tracking with onboard sensing," *Int. J. Robot. Res.*, vol. 34, no. 13, pp. 1660–1677, November 2015.

[164] P. Tokekar, V. Isler, and A. Franchi, "Multi-target visual tracking with aerial robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.* IEEE, 2014, pp. 3067–3072.

[165] B. Schlotfeldt, D. Thakur, N. Atanasov, V. Kumar, and G. J. Pappas, "Anytime planning for decentralized multirobot active information gathering," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1025–1032, 2018.

[166] T. Nestmeyer, P. R. Giordano, H. H. Bülthoff, and A. Franchi, "Decentralized simultaneous multi-target exploration using a connected network of multiple robots," *Autonomous Robots*, vol. 41, no. 4, pp. 989–1011, 2017.

[167] S. Carpin, D. Burch, N. Basilico, T. H. Chung, and M. Kölsch, "Variable resolution search with quadrotors: Theory and practice," *J. Field Robot.*, vol. 30, no. 5, pp. 685–701, 2013.

[168] S. Carpin, D. Burch, and T. H. Chung, "Searching for multiple targets using probabilistic quadtrees," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.* IEEE, 2011, pp. 4536–4543.

[169] P. M. Dames, M. Schwager, D. Rus, and V. Kumar, "Active magnetic anomaly detection using multiple micro aerial vehicles," *IEEE Robot. Autom. Lett.*, vol. 1, no. 1, pp. 153–160, 2016.

[170] R. P. Mahler, *Statistical multisource-multitarget information fusion*. Artech House, Inc., 2007.

[171] J. Li, D. H. Ye, T. Chung, M. Kolsch, J. Wachs, and C. Bouman, "Multi-target detection and tracking from a single camera in unmanned aerial vehicles (UAVs)," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.* IEEE, 2016, pp. 4992–4997.

[172] P. Dames, P. Tokekar, and V. Kumar, "Detecting, localizing, and tracking an unknown number of moving targets using a team of mobile robots," *Int. J. Robot. Res.*, vol. 36, no. 13-14, pp. 1540–1553, 2017.

[173] P. Dames, "Distributed multi-target search and tracking using the phd filter," in *2017 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, 2017, pp. 1–8.

[174] S. L. Smith, M. Schwager, and D. Rus, "Persistent robotic tasks: Monitoring and sweeping in changing environments," *IEEE Trans. Robot.*, vol. 28, no. 2, pp. 410–426, 2012.

[175] N. Nigam and I. Kroo, "Persistent surveillance using multiple unmanned air vehicles," in *IEEE Aerospace Conference*. IEEE, 2008, pp. 1–14.

[176] N. Nigam, S. Bieniawski, I. Kroo, and J. Vian, "Control of multiple UAVs for persistent surveillance: algorithm and flight test results," *IEEE Trans. Control Syst. Technol.*, vol. 20, no. 5, pp. 1236–1251, 2012.

[177] J. Keller, D. Thakur, V. Dobrokhodov, K. Jones, M. Pivtoraiko, J. Gallier, I. Kaminer, and V. Kumar, "A computationally efficient approach to trajectory management for coordinated aerial surveillance," *Unmanned Systems*, vol. 1, no. 01, pp. 59–74, 2013.

[178] J. Keller, D. Thakur, M. Likhachev, J. Gallier, and V. Kumar, "Coordinated path planning for fixed-wing uas conducting persistent surveillance missions," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 17–24, 2017.

[179] N. Michael, E. Stump, and K. Mohta, "Persistent surveillance with a team of MAVs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.* IEEE, 2011, pp. 2708–2714.

[180] M. Schwager, B. Julian, M. Angermann, and D. Rus, "Eyes in the sky: Decentralized control for the deployment of robotic camera networks," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1541–1561, September 2011.

[181] B. J. Julian, M. Angermann, M. Schwager, and D. Rus, "Distributed robotic sensor networks: An information-theoretic approach," *Int. J. Robot. Res.*, vol. 31, no. 10, pp. 1134–1154, September 2012.

[182] X. Lan and M. Schwager, "Rapidly-exploring random cycles: Persistent estimation of spatio-temoral fields with multiple sensing robots," *IEEE Trans. Robot.*, vol. 32, no. 5, pp. 1230–1244, 2016.

[183] P. Tokekar, J. Vander Hook, D. Mulla, and V. Isler, "Sensor planning for a symbiotic UAV and UGV system for precision agriculture," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1498–1511, 2016.

[184] C. Forster, S. Lynen, L. Kneip, and D. Scaramuzza, "Collaborative monocular slam with multiple micro aerial vehicles," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013.

[185] L. von Stumberg, V. Usenko, J. Engel, J. Stckler, and D. Cremers, "From monocular SLAM to autonomous drone exploration," in *Proc. Euro. Conf. Mobile Robot.*, 2017.

[186] Y. Lin, F. Gao, T. Qin, W. Gao, T. Liu, W. Wu, Z. Yang, and S. Shen, "Autonomous aerial navigation using monocular visual-inertial fusion," *J. Field Robot.*, vol. 00, pp. 1–29, 2017.

[187] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *arXiv preprint arXiv:1708.03852*, 2017.

[188] S. Saeedi, M. Trentini, M. Seto, and H. Li, "Multiple-robot simultaneous localization and mapping: A review," *J. Field Robot.*, vol. 33, no. 1, pp. 3–46, 2016.

[189] T. A. Vidal-Calleja, C. Berger, J. Solà, and S. Lacroix, "Large scale multiple robot visual mapping with heterogeneous landmarks in semi-structured terrain," *Robotics and Autonomous Systems*, vol. 59, no. 9, pp. 654–674, 2011.

[190] D. Zou and P. Tan, "Coslam: Collaborative visual slam in dynamic environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 354–366, 2013.

[191] P. Schmuck and M. Chli, "Multi-uav collaborative monocular slam," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017.

[192] M. Karrer and M. Chli, "Towards globally consistent visual-inertial collaborative SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018.

[193] T. Qin, P. Li, and S. Shen, "Relocalization, global optimization and map merging for monocular visual-inertial slam." in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018.

[194] T. Schneider, M. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, and R. Siegwart, "maplab: An open framework for research in visual-inertial mapping and localization," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1418–1425, 2018.

[195] A. Cunningham, V. Indelman, and F. Dellaert, "DDF-SAM 2.0: Consistent distributed smoothing and mapping," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013.

[196] T. Cieslewski, S. Choudhary, and D. Scaramuzza, "Data-efficient decentralized visual SLAM," *Proc. IEEE Int. Conf. Robot. Autom.*, 2018.

[197] S. Choudhary, L. Carlone, C. Nieto, J. Rogers, H. I. Christensen, and F. Dellaert, "Distributed mapping with privacy and communication constraints: Lightweight algorithms and object-based models," *Int. J. Robot. Res.*, vol. 36, no. 12, pp. 1286–1311, 2017.

[198] L. Riazuelo, J. Civera, and J. Montiel, "C2tam: A cloud framework for cooperative tracking and mapping," *Robotics and Autonomous Systems*, vol. 62, no. 4, pp. 401–413, 2014.

[199] G. Mohanarajah, V. Usenko, M. Singh, R. D'Andrea, and M. Waibel, "Cloud-based collaborative 3D mapping in real-time with low-cost robots," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 423–431, 2015.

[200] G. Loianno, J. Thomas, and V. Kumar, "Cooperative localization and mapping of MAVs using RGB-D sensors," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015.

[201] B. Charrow, S. Liu, V. Kumar, and N. Michael, "Information-theoretic mapping using cauchy-schwarz quadratic mutual information," in *Proc. IEEE Int. Conf. Robot. Autom.* IEEE, 2015, pp. 4791–4798.

[202] B. Kim, M. Kaess, L. Fletcher, J. Leonard, A. Bachrach, N. Roy, and S. Teller, "Multiple relative pose graphs for robust cooperative mapping," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010.

[203] J. Dong, E. Nelson, V. Indelman, N. Michael, and F. Dellaert, "Distributed real-time cooperative localization and mapping using an uncertainty-aware expectation maximization approach," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015.

[204] P. Cohn, A. Green, M. Langstaff, and M. Roller, "Commercial drones are here: The future of unmanned aerial systems," Dec 2017. [Online]. Available: https://www.mckinsey.com/industries/capital-projects-and-infrastructure/our-insights/commercial-drones-are-here-the-future-of-unmanned-aerial-systems

[205] D. Joshi, "Commercial unmanned aerial vehicle (UAV) market analysis industry trends, companies and what you should know," Aug 2017. [Online]. Available: http://www.businessinsider.com/commercial-uav-market-analysis-2017-8

[206] R. Spica, P. Robuffo Giordano, M. Ryll, H. Blthoff, and A. Franchi, "An open-source hardware/software architecture for quadrotor UAVs," in *2nd Workshop on Research, Education and Development of Unmanned Aerial System*, Compigne, France, November 2013.

[207] N. Michael and V. Kumar, "Control of ensembles of aerial robots," *Proc. the IEEE*, vol. 99, no. 9, pp. 1587–1602, 2011.

[208] J. P. How, B. Behihke, A. Frank, D. Dale, and J. Vian, "Real-time indoor autonomous vehicle test environment," *IEEE Control Syst.*, vol. 28, no. 2, pp. 51–64, 2008.

[209] J. A. Preiss, W. Honig, G. S. Sukhatme, and N. Ayanian, "Crazyswarm: A large nano-quadcopter swarm," in *Proc. IEEE Int. Conf. Robot. Autom.* IEEE, 2017, pp. 3299–3304.

[210] A. Weinstein, A. Cho, G. Loianno, and V. Kumar, "Visual inertial odometry swarm: An autonomous swarm of vision-based quadrotors," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1801–1807, 2018.

[211] G. Vásárhelyi, C. Virágh, G. Somorjai, N. Tarcai, T. Szörényi, T. Nepusz, and T. Vicsek, "Outdoor flocking and formation flight with autonomous aerial robots," in *iros*, 2014, pp. 3866–3873.

[212] T. H. Chung, M. R. Clement, M. A. Day, K. D. Jones, D. Davis, and M. Jones, "Live-fly, large-scale field experimentation for large numbers of fixed-wing UAVs," in *Proc. IEEE Int. Conf. Robot. Autom.* IEEE, 2016, pp. 1255–1262.

[213] K. Leahy, D. Zhou, C.-I. Vasile, K. Oikonomopoulos, M. Schwager, and C. Belta, "Persistent surveillance for unmanned aerial vehicles subject to charging and temporal logic constraints," *Autonomous Robots*, vol. 40, pp. 1363–1378, 2016.

[214] Y. Mulgaonkar and V. Kumar, "Autonomous charging to enable long-endurance missions for small aerial robots," in *Micro-and Nanotechnology Sensors, Systems, and Applications VI*, vol. 9083. International Society for Optics and Photonics, 2014, p. 90831S.

[215] B. Bethke, J. P. How, and J. Vian, "Group health management of UAV teams with applications to persistent surveillance," in *American Control Conference*. IEEE, 2008, pp. 3145–3150.

[216] A. Kushleyev, D. Mellinger, C. Powers, and V. Kumar, "Towards a swarm of agile micro quadrotors," *Autonomous Robots*, vol. 35, no. 4, pp. 287–300, 2013.

[217] R. Ritz, M. W. Müller, M. Hehn, and R. D'Andrea, "Cooperative quadrocopter ball throwing and catching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 4972–4978.

[218] A. Ledergerber, M. Hamer, and R. D'Andrea, "A robot self-localization system using one-way ultra-wideband communication," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 3131–3137.

[219] J. Yang, A. Dani, S.-J. Chung, and S. Hutchinson, "Vision-based localization and robot-centric mapping in riverine environments," *J. Field Robot.*, vol. 34, no. 3, pp. 429–450, 2017.

[220] E. Montijano, E. Cristofalo, D. Zhou, M. Schwager, and C. Sagues, "Vision-based distributed formation control without an external positioning system," *IEEE Trans. Robot.*, vol. 32, no. 2, pp. 339–351, 2016.

[221] R. Tron, J. Thomas, G. Loianno, K. Daniilidis, and V. Kumar, "A distributed optimization framework for localization and formation control: applications to vision-based measurements," *IEEE Control Syst.*, vol. 36, no. 4, pp. 22–44, 2016.

[222] M. Saska, T. Baca, J. Thomas, J. Chudoba, L. Preucil, T. Krajnik, J. Faigl, G. Loianno, and V. Kumar, "System for deployment of groups of unmanned micro aerial vehicles in GPS-denied environments using onboard visual relative localization," *Autonomous Robots*, vol. 41, no. 4, pp. 919–944, 2017.

[223] F. Schiano, A. Franchi, D. Zelazo, and P. R. Giordano, "A rigidity-based decentralized bearing formation controller for groups of quadrotor UAVs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.* IEEE, 2016, pp. 5099–5106.

[224] F. Schiano and P. Robuffo Giordano, "Bearing rigidity maintenance for formations of quadrotor UAVs," in *Proc. IEEE Int. Conf. Robot. Autom.*, Singapore, May 2017.

[225] C. Cigla, R. Thakker, and L. Matthies, "Onboard stereo vision for drone pursuit or sense and avoid," in *IEEE Conf. Comput. Vis. Pattern Recognition Workshops*, 2018, pp. 625–633.

[226] J.-S. Lee, Y.-W. Su, and C.-C. Shen, "A comparative study of wireless protocols: Bluetooth, uwb, zigbee, and wi-fi," in *IEEE Conf. Industrial Electronics*, 2007, pp. 46–51.

[227] S. Eldessoki, D. Wieruch, and B. Holfeld, "Impact of waveforms on coexistence of mixed numerologies in 5g urllc networks," in *Proc. Int. ITG Workshop on Smart Antennas*. VDE, 2017, pp. 1–6.

[228] X. Dong, Y. Zhou, Z. Ren, and Y. Zhong, "Time-varying formation control for unmanned aerial vehicles with switching interaction topologies," *Control Eng. Practice*, vol. 46, pp. 26–36, 2016.

[229] A. Franchi, C. Secchi, M. Ryll, H. H. Bulthoff, and P. R. Giordano, "Shared control: Balancing autonomy and human assistance with a group of quadrotor uavs," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 57–68, 2012.

Soon-Jo Chung (M'06–SM'12) received the B.S. degree (summa cum laude) from Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 1998; the S.M. degree in aeronautics and astronautics; and the Sc.D. degree in estimation and control from Massachusetts Institute of Technology, Cambridge, MA, USA, in 2002 and 2007, respectively. He is an Associate Professor of Aerospace and Bren Scholar and Jet Propulsion Laboratory Research Scientist in the California Institute of Technology. Dr. Chung was on the faculty of the University of Illinois at Urbana-Champaign (UIUC) during 2009-2016. His research focuses on spacecraft and aerial swarms and autonomous aerospace systems, and in particular, on the theory and application of complex nonlinear dynamics, control, estimation, guidance, and navigation of autonomous space and air vehicles. Dr. Chung is received the UIUC Engineering Deans Award for Excellence in Research, the Beckman Faculty Fellowship of the UIUC Center for Advanced Study, the U.S. Air Force Office of Scientific Research Young Investigator Award, the National Science Foundation Faculty Early Career Development Award, and three Best Conference Paper Awards from the IEEE, and the American Institute of Aeronautics and Astronautics. He is an Associate Editor of IEEE Transactions on Robotics and Journal of Guidance, Control, and Dynamics.

Aditya A. Paranjape received the B.Tech and M.Tech degrees in Aerospace Engineering from the Indian Institute of Technology (IIT) Bombay, Mumbai, India in 2007; and Ph.D. in Aerospace Engineering from the University of Illinois at Urbana-Champaign, Urbana, IL, USA, in 2011. He is currently Lecturer in Aerial Robotics at Imperial College London, London, United Kingdom. His research focuses on the dynamics and control of aircraft, spacecraft, infinite dimensional systems, and multi-agent systems.

Philip Dames received both his B.S. (summa cum laude) and M.S. degrees in Mechanical Engineering from Northwestern University in 2010 and his Ph.D. degree in Mechanical Engineering and Applied Mechanics from the University of Pennsylvania in 2015. From 2015–2016 he was a Postdoctoral Researcher in Electrical and Systems Engineering at the University of Pennsylvania. Since 2016, Philip has been an Assistant Professor of Mechanical Engineering at Temple University. His current research interests include the intersection of estimation, control, and communication to enable autonomy in large-scale, multi-agent systems.

Shaojie Shen received his B.Eng. degree in Electronic Engineering from the Hong Kong University of Science and Technology in 2009. He received his M.S. in Robotics and Ph.D. in Electrical and Systems Engineering in 2011 and 2014, respectively, all from the University of Pennsylvania. He joined the Department of Electronic and Computer Engineering at the Hong Kong University of Science and Technology in September 2014 as an Assistant Professor. His research interests are in the areas of robotics and unmanned aerial vehicles, with focus on state estimation, sensor fusion, computer vision, localization and mapping, and autonomous navigation in complex environments.

Vijay Kumar received the B.Tech. degree from the Indian Institute of Technology, Kanpur, India, and the Ph.D. degree from The Ohio State University, Columbus, OH, USA, in 1987. He is the Nemirovsky Family Dean of Penn Engineering with appointments in the Departments of Mechanical Engineering and Applied Mechanics, Computer and Information Science, and Electrical and Systems Engineering at the University of Pennsylvania. He has been on the Faculty in the Department of Mechanical Engineering and Applied Mechanics with a secondary appointment in the Department of Computer and Information Science at the University of Pennsylvania since 1987. He was the Deputy Dean for Research in the School of Engineering and Applied Science from 2000 to 2004. He was the Director of the General Robotics, Automation, Sensing and Perception Laboratory, a multidisciplinary robotics and perception laboratory, from 1998 to 2004. He was the Chairman of the Department of Mechanical Engineering and Applied Mechanics from 2005 to 2008. He was the Deputy Dean for Education in the School of Engineering and Applied Science from 2008 to 2012. He then served as the Assistant Director of Robotics and Cyber Physical Systems, White House Office of Science and Technology Policy (20122013). His research interests include robotics, specifically multirobot systems, and micro aerial vehicles. Dr. Kumar is a Fellow of the American Society of Mechanical Engineers (2003), a Fellow of the Institute of Electrical and Electronic Engineers (2005), and a member of the National Academy of Engineering (2013). He has been on the editorial boards of IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, ASME Journal of Mechanical Design, ASME Journal of Mechanisms and Robotics, and Springer Tract in Advanced Robotics. He currently serves as the Editor of ASME Journal of Mechanisms and Robotics and as an Advisory Board Member of AAAS Science Robotics Journal. He received the 1991 National Science Foundation Presidential Young Investigator award, the 1996 Lindback Award for Distinguished Teaching (University of Pennsylvania), the 1997 Freudenstein Award for significant accomplishments in mechanisms and robotics, the 2012 ASME Mechanisms and Robotics Award, the 2012 IEEE Robotics and Automation Society Distinguished Service Award, the 2012 World Technology Network Award, and the 2014 Engelberger Robotics Award. He has won best paper awards at DARS 2002, ICRA 2004, ICRA 2011, RSS 2011, and RSS 2013, and has advised doctoral students who have won Best Student Paper Awards at ICRA 2008, RSS 2009, and DARS 2010.