Recent Research in Cooperative Control of Multi-Vehicle Systems

Richard M. Murray
Control and Dynamical Systems
California Institute of Technology

Abstract

This paper presents a survey of recent research in cooperative control of multi-vehicle systems, using a common mathematical framework to allow different methods to be described in a unified way. The survey has three primary parts: an overview of current applications of cooperative control, a summary of some of the key technical approaches that have been explored, and a description of some possible future directions for research. Specific technical areas that are discussed include formation control, cooperative tasking, spatio-temporal planning and consensus.

1 Introduction

Research on control of multi-vehicle systems performing cooperative tasks dates back to the late 1980s, initially beginning in the field of mobile robotics (see [37] for a more detailed history). Aided by the development of inexpensive and reliable wireless communications systems, research in this area increased substantially in the 1990s. California's Partners for Advanced Transit and Highways (PATH) project [2] demonstrated multiple automobiles driving together in "platoons" and this was quickly followed by other highway automation projects [18, 10]. In the late 1990s and early 2000s, cooperative control of multiple aircraft, especially unmanned aerial vehicles (UAVs), became a highly active research area in the United States [3], spurring further advances. Over the last decade this research area has blossomed, with many new systems being proposed in application areas ranging from military battle systems to mobile sensors networks to commercial highway and air transportation systems.

The purpose of this article is to provide a survey of some of the recent research in cooperative control of multi-vehicle systems. We focus on research in the last two decades, with some historical notes on work before this period. To help focus the topics that are surveyed, we focus exclusively on control of multi-vehicle systems that are working together to complete a shared task. Several other surveys of the literature in cooperative control are available that complement the current paper (see, e.g., [37]).

1

It will be helpful in the sequel to have a clear notion of some terms that will define the object of the survey, in particular a concise definition of "cooperative", which has been used in many different ways by the broad research communities interested in this topic. For the purposes of this survey, we will consider a *vehicle* to be a dynamical system whose position is given by its location in three dimensional space. We will consider a collection of N vehicles that are performing a shared task, where the task depends on the relationship between the locations of the individual vehicles. The vehicles are able to communicate with each in carrying out the task, with the individual vehicles able to communicate with some subset of the other vehicles.

We assume that the dynamics of the ith vehicle can be written as

$$\dot{x}^i = f^i(x^i, u^i) \quad x^i \in \mathbb{R}^n, u^i \in \mathbb{R}^m$$
$$\dot{y}^i = h^i(x^i) \qquad y^i \in SE(3),$$

where x^i is the state of the ith vehicle, u^i is the input that controls the vehicle's state and f^i is a smooth vector field representing its dynamics. We assume that the location of the vehicle is given by the output $y^i \in SE(3)$, where $SE(3)$ is the set of rigid body configurations (position and orientation). More general formulations allowing position and velocity as part of the location description are possible as well, but will be omitted for simplicity. We let $x = (x^1, \ldots, x^N)$ represent the complete state for a collection of N vehicles.

In addition to the location of the vehicle, we will also assume that each vehicle has a discrete state, α^i, which we define as the *role* of the vehicle. The role of the vehicle will be represented as an element of a discrete set $\mathcal{A}$ whose definition will depend on the specific cooperative control problem under consideration. As indicated by the terminology, we will generally consider the role variable α^i to represent the portion of the vehicle's overall state that encodes its current actions and its relationship with the overall task being performed. We will assume that the role of a vehicle can change at any time and we will write a change of role as

$$\alpha' = r(x, \alpha),$$

where α' indicates the new value of α. We let $\alpha = (\alpha^1, \ldots, \alpha^N)$ represent the roles of the collection of N vehicles and write $\alpha^i(t)$ for the role of vehicle i at time t.

We assume that the vehicles are able to communicate with some set of other vehicles and we represent the set of possible communication channels by a graph $\mathcal{G}$. The nodes of the graph represent the individual vehicles and a directed edge between two nodes represents the ability of a vehicle to receive information from another vehicle. We write $\mathcal{N}^i(\mathcal{G})$ to represent the neighbors of vehicle i, that is, the set of vehicles that vehicle i is able to obtain information from (either by explicit communication or by sensing the position of the other vehicle). In general, $\mathcal{N}^i$ can depend on the locations and roles of the vehicles, in which case we will write $\mathcal{N}^i(x, \alpha)$. The number of neighbors of the ith vehicle is given by the number of elements of $\mathcal{N}^i$, written $|\mathcal{N}^i|$.

Given a collection of vehicles with state x and roles α, we will define a *task* in terms of a performance function

$$J = \int_0^T L(x, \alpha, u)\, dt + V(x(T), \alpha(T)),$$

where T is the horizon time over which the task should be accomplished, L represents the incremental cost of the task and V represents the terminal cost of the task. As special cases, we can take $T = \infty$ to represent infinite horizon problems or take $L = 0$ to represent tasks in which we are

2

only interested in the final state. We may also have constraints on the states or inputs, although we shall generally consider such constraints to be included in the cost function (eg, via Lagrange multipliers) for ease of presentation.

A *strategy* for a given task is an assignment of the inputs u^i for each vehicle and a selection of the roles of the vehicles. We will assume that the inputs to the vehicles' dynamics are given by control laws of the form

$$u^i = \gamma(x, \alpha)$$

where γ is a smooth function. For the choice of roles, we make use of the notion of a *guarded command language* (see [27]): a program is a set of commands of the form

$$\{g^i_j(x, \alpha) : r^i_j(x, \alpha)\}$$

where g^i_j is a guard that evaluates to either true or false and r^i_j is a rule that defines how the role α^i should be updated if the rule evaluates to true. Thus, the role evolves according to the update law

$$\alpha^{i\,\prime} = \begin{cases} r^i_j(x, \alpha) & g(x, \alpha) = \text{true} \\ \text{unchanged} & \text{otherwise.} \end{cases}$$

This update is allowed to happen asynchronously, although in practice it may be assigned by a central agent in the system, in which case it may evolve in a more regular fashion. We write Σ^i to represent the overall strategy (control law and guarded commands) for the ith vehicle. $\Sigma = (\Sigma^1, \ldots, \Sigma^N)$ is used to represent the complete strategy for the system.

Using these definitions, we can now provide a more formal description of a cooperative control problem. We say that a task can be *additively decoupled* (or just decoupled) if the cost function J can be written as

$$J = \sum_{i=0}^{N} \left(\int_0^T L^i(x^i, \alpha^i, u^i)\, dt + V^i(x^i(T), \alpha^i(T)) \right).$$

If a task is not decoupled, we say that the task is *cooperative*, by which we mean that the task performance depends on the joint locations, roles and inputs of the vehicles. (Note that we are assuming here that all vehicles are trying to solve a common objective and hence not considering adversarial tasks, for which a more careful notation would be required.)

We say that a strategy is *centralized* if Σ^i depends on the location or role of any vehicle that is not a neighbor of i. A strategy is *decentralized* if

$$u^i(x, \alpha) = u^i(x^i, \alpha^i, x^{-i}, \alpha^{-i})$$
$$\{g^i_j(x, \alpha) : r^i_j(x, \alpha)\} = \{g^i_j(x^i, \alpha^i, x^{-i}, \alpha^{-i}) : r^i_j(x^i, \alpha^i, x^{-i}, \alpha^{-i})\},$$

where we use the shorthand x^{-i} and α^{-i} to represent the location and roles of vehicle i's neighbors (hence $x^{-i} = \{x^{j_1}, \ldots, x^{j_{m_i}}$ where $j_k \in \mathcal{N}^{\rangle}$ and $m_i = |\mathcal{N}^i|.\}$. We will mainly be interested in cooperative tasks that can be solved using a decentralized strategy.

We note that the definitions used here are not the most general possible and we have ignored some subtleties regarding the formal definition of the "solution" of a task (i.e., we assume existence and uniqueness of solutions for a given strategy). These details are important and can be found in the various papers referenced in this survey. One alternative set of definitions for cooperative agents can be found in the work of Parker [35], which makes use of the notions of local/global goals and control.

We these definitions in hand, we now proceed to consider some of the primary applications of cooperative control of multi-vehicle systems, followed by some of the key technical results that have been proposed in the last decade. We end the paper with a partial listing of some of the open research directions that are currently under exploration.

2 Applications Overview

In this section we summarize some of the main applications for cooperative control of multi-vehicle systems. This summary is based on those applications of which the author is most aware (including the results of a result survey of future directions in control, dynamics and systems [31]), as well as a survey of the literature (with emphasis on papers that are frequently referenced by others). Although not comprehensive, the applications cited here demonstrate some of the key features that must be addressed in solving cooperative control problems.

2.1 Military Systems

Modern military systems are becoming increasingly sophisticated, with a mixture of manned and unmanned vehicles being used in complex battlefield environments, such as the one depicted in Figure 1. Traditional solutions involve a centralized resource allocation (assignment of planes to targets), followed by decentralized execution (each attack vehicle is responsible for a set of targets). More modern battlespace management systems are considering the use of cooperative operation of large collections of distributed vehicles, with location computation, global communication connections and decentralized control actions [31, 46].

Formation flight   One of the simplest cooperative control problems is that of formation flight: a set of aircraft fly in a formation, specified by the relative locations of nearby aircraft. This area has received considerable attention in the literature. Some of the earliest work in this area is that of Parker [35], who consider the design of control laws that use a combination of local and global knowledge to maintain a formation.

NASA has experimented with formation flight as a method for reducing drag on a collection of aircraft [28]. The key idea is to locate the aircraft such that the tip vortices of one aircraft help reduce the induced drag of the tailing aircraft. This task requires precise alignment of an aircraft with the aircraft in front of it. To date, demonstrations of this concept in engineering systems have been restricted to small numbers of aircraft. Similar formations in nature can involve many more individuals [14].

Cooperative classification and surveillance   Chandler et al. [3] define the cooperative classification problem as "the task of optimally and jointly using multiple vehicles' sightings to maximize the probability of correct target classification". More generally, we can define the cooperative surveillance problem as that of using a collection of vehicles to maintain a centralized or decentralized description of the state of a geographic area. This description might include the current state of features that are spatially fixed (such as the number of people in a given location) or information about entities that are moving in the region of interest (eg, locations of cars and planes in a given region).
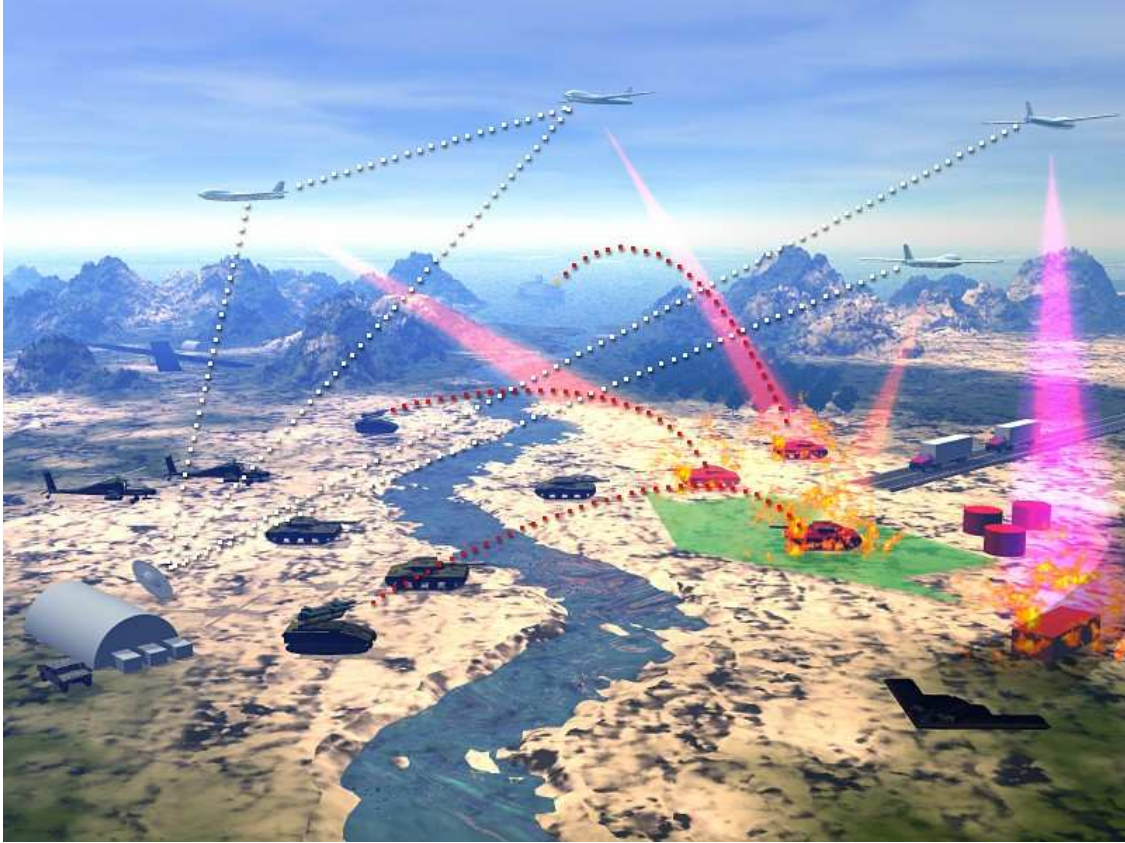
4

Figure 1: Battle space management scenario illustrating distributed command and control between heterogeneous air and ground assets. Figure courtesy of DARPA.

The cooperative classification problem is one in which the performance function involves the collection of maximal amounts of relevant information. One typically assumes that the vehicles can communicate over some range (possibly limited by line of site, especially for ground-based vehicles) and information shared between the vehicles can be used by the vehicles in determining their motion.

Cooperative attack and rendezvous The rendezvous problem involves bringing a collection of vehicles to a common location at a common time. Depending on the application, the rendezvous time may either be fixed ahead of time or determined dynamically, based on when all vehicles reach the same area. Military applications of rendezvous include minimizing exposure to radar by allowing aircraft to fly individual paths that are locally optimized [3].

Mixed initiative systems A variant of the cooperative control problem is the *mixed initiative* cooperative control problem, in which collections of autonomous vehicles and human operators (on the ground or in vehicles) must collectively perform a task or a mission. This class of problems adds the complexity of providing situational awareness to the operators and allow varying levels of control of the autonomous system.
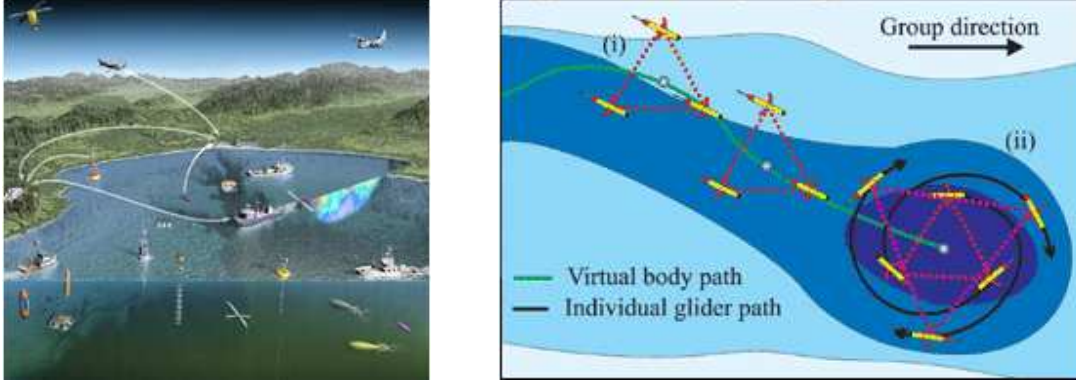
Figure 2: Autonomous ocean sampling network: (a) a depiction of the collection of vehicles that were part of the summer 2003 experiment; (b) an example of using a collection of gliders for sampling a region of interest.

2.2 Mobile Sensor Networks

A second area of application in cooperative control is networks of sensors that can be position so as to maximize the amount of information they are able to gather. In this section we provide some examples of the types of cooperative control applications that are being pursued in this area.

Environmental sampling The Autonomous Ocean Sampling Network (AOSN) [30], pictured in Figure 2 is an example of an environmental sampling network. The network consists of a collection of robotic vehicles that are used for "adaptive sampling", in which the motion of the vehicles is based on the observations taken by the vehicles. This approach allows the sensors to be positioned in the areas in which they can do the most good, as a function of the data already collected. Because of the distributed nature of the measurements being taken, a cooperative control strategy is used to control the motion of the vehicles. In tests done in the summer of 2003, over 20 vehicles were controlled over 4 weeks weeks to collect data [30].

Distributed aperture observing A related application for cooperative control of multi-vehicle systems is distributed aperture (or phased array) imaging, illustrated in Figure 3. The proposed TechSat 21 project was sponsored by the US Air Force Research Laboratory (AFRL) and was to have launched a collection of "microsatellites" that would be used to form a "virtual" satellite with a single, large aperture antenna (the project was canceled in 2003). Another example of a distributed aperture observing system is the terrestrial planet finder (TPF), being proposed by NASA. TPF uses optical interferometry to image distance stars and to detect slight shifts in the stars positions that indicated the presence of planets orbiting the stars [20].

2.3 Transportation Systems

Finally, the use of cooperative control in transportation systems has received considerable attention over the last few decades.

Figure 3: Distributed aperture observing systems: (a) the proposed TechSat 21 concept would use a collection of microsatellites to form the equivalent of a larger aperture imaging system; (b) the terrestrial planet finder uses formation flying to enable optimal interferometry for detecting planets.

(a)   (b)

Figure 4: Transportation systems. (a) A platoon of cars driving down the San Diego freeway as part of the PATH project [2]. (b) The San Francisco Bay area aircraft arrival and departure routes (courtesy of Federal Aviation Authority).

Intelligent highways   Several groups around the world have begun to explore the use of distributed control for problems related to intelligent highway and transportation systems. These problems include increased interaction between individual vehicles to provide safer operations (e.g., collision warning and avoidance), as well interaction between vehicles and the roadway infrastructure. These latter applications are particularly challenging since they begin to link heterogeneous vehicles through communications systems that will experience varying bandwidths and latency (time delays) depending on the local environment. Providing safe, reliable, and comfortable operation for such systems is a major challenge that will have application in a variety of consumer, industrial, and military products and systems.

A representative example of this class of applications is the California Partners for Advanced Transit and Highways (PATH) project [2]. In 1997 the PATH project developed and demonstrated a system for allowing cars to be driven automatically down a freeway at close spacing, as shown in Figure 4a. By decreasing the spacing of cars, the density of traffic on a highway can be increase
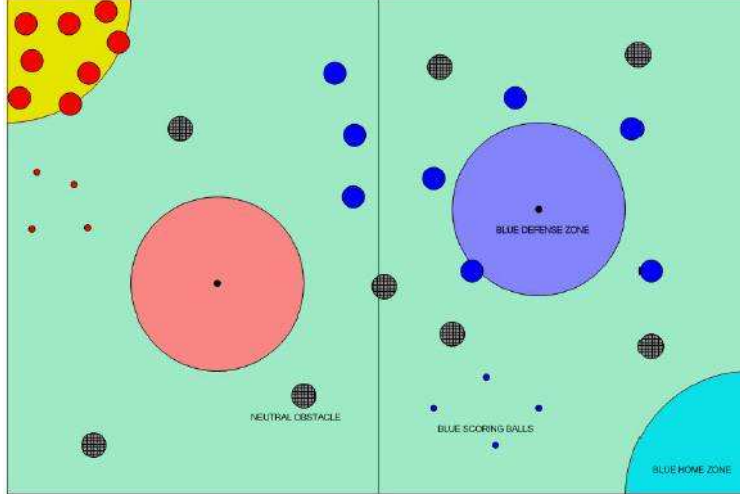
Figure 5: The RoboFlag playing field [7].

without requiring additional lanes. Additional work within the PATH project has looked at a variety of other systems for better managing traffic flow [2].

Air traffic control Air traffic control systems are another area where methods for cooperative control are being explored (see, e.g., [45]). As the density of air traffic continues to increase, congestion at major airports and automated collision warning systems are becoming increasingly common. Figure 4b illustrates some of the complexity of the current air traffic control networks. Next generation air traffic control systems will likely move from a human-controlled, centralized structure within given region to a more distributed system with "free flight" technologies allowing aircraft to travel in direct paths rather than staying in pre-defined air traffic control corridors. Efforts are now being made to improve the current system by developing cockpit "sensors" such as augmented GPS navigation systems and data links for aircraft to aircraft communication citeatc.

2.4 Testbeds

A variety of testbeds have been developed to explore cooperative control problems in laboratory settings. Perhaps the most well known is *RoboCup*, a multi-vehicle game of robot soccer. RoboCup was initially conceived as an attempt to foster research in artificial intelligence, specifically that of multiple vehicles in a highly dynamic environment [25]. The RoboCup competition is now held annually and has competitions involving a variety of different physical and simulation platforms. Most of the RoboCup competitions allow the use of centralized computation, although some teams have made use of decentralized strategies [6].

A related game, dubbed *RoboFlag* has been developed at Cornell [7] and is loosely based on "Capture the Flag" and "Paintball". Two teams play the game, the red team and the blue team, as depicted in Figure 5. The red team's objective is to infiltrate blue's territory, grab the blue flag, and bring it back to the red home zone. At the same time, the blue team's objective is to infiltrate red's territory, grab the red flag, and bring it back to the blue home zone. The game is thus a mix of offense and defense: secure the opponent's flag, while at the same time prevent the opponent

from securing your flag. Sensing and communications are both limited to provide a more realistic distributed computing environment. The game is meant to provide an example of multi-vehicle, semi-autonomous systems operating in dynamic, uncertain, and adversarial environments. Human operators can also be present in the system and can be used either as high level controllers or as low level (remote) "pilots". A centralized control unit may be used coordinate the vehicles, but it must respect the communication constraints (bandwidth and latency) of the system.

Several physical testbeds have also been developed, ranging from wheeled robots such as those sued in RoboCup, to hovercraft that provide some of the dynamics more typical of aircraft [5, 47], to small-scale aircraft [17, 24] and helicoptersSKS03-cep,SV99-cca. These citations are far from complete, but give an example of the range of physical testbeds that have been developed.

3   Technology Overview

In this section we provide a brief survey of some of the techniques that have been developed for designing strategies for cooperative control tasks. We make use of the mathematical notation defined in the introduction wherever possible. We focus primarily on the problem formulation and the approached used in its solution, leaving the details of the proofs of stability, convergence and optimality to the original papers.

3.1   Formation Control

Many of the applications above have as part of their solution the ability to maintain the position of a set of vehicles relative to each other or relative to a reference. This problem is known as *formation control* and has received considerable attention, both as a centralized and as a decentralized problem.

Optimization-based approaches   One way to approach the formation control problem is to formulate it as an optimization problem. If we let $L^i(x^i, x^{-i})$ represent the individual formation error between the ith vehicle and its neighbors, then we can establish a cost function

$$L(x, \alpha, u) = \sum L^i(x^i, x^{-1}) + \|u^i\|_R^2,$$

where the summation over the individual formation errors gives the *cumulative formation error* [35] and the final term is a penalty on the inputs (other forms could be used).

This problem can be solved in either a centralized manner or a distributed manner. One distributed approach is the work of Dunbar *et al.* [11], who considers cooperative control problems using receding horizon optimal control. For a cost function whose coupling reflects the communication constraints of the vehicles, he generates distributed optimal control problems for each subsystem and establishes that the distributed receding horizon implementation is asymptotically stabilizing. The communication requirements between subsystems with coupling in the cost function are that each subsystem obtain the previous optimal control trajectory of those subsystems at each receding horizon update. The key requirements for stability are that each distributed optimal control not deviate too far from the previous optimal control, and that the receding horizon updates happen sufficiently fast.

Figure 6 shows a simulation of Dunbar's results. The vehicles are flying in "fingertip formation", with vehicles 2 and 3 maintaining position relative to vehicle 1 and vehicle 4 maintaining position relative to vehicle 2. The control goal is to maintain formation around the black square, which is
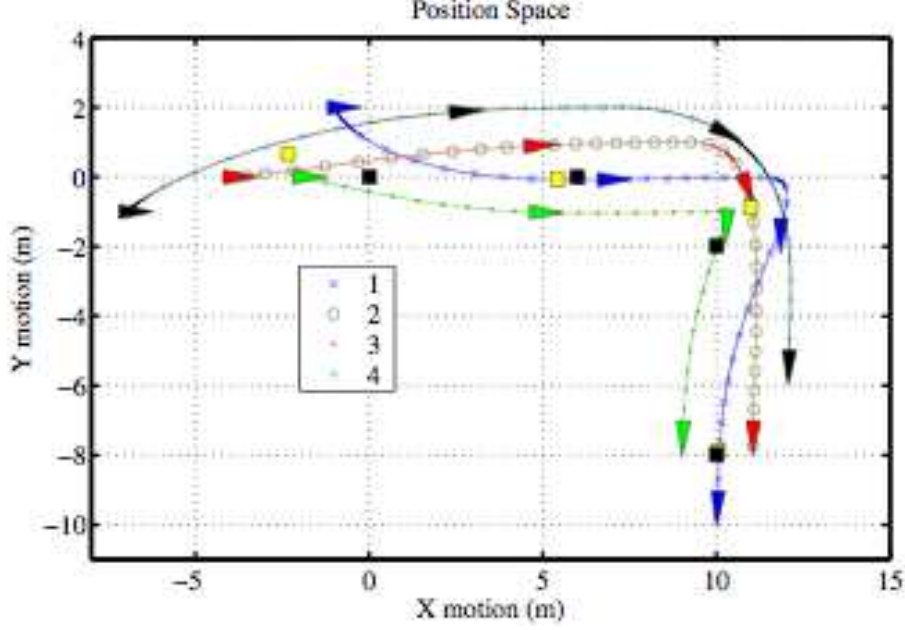
Figure 6: Four vehicle formation using distributed receding horizon control [11].

flying along a trajectory that is not known to the individual aircraft. The localized optimization for each vehicle uses a previous optimal path for its neighbors while constraining its own path to stay near the previous path that it communicated to others.

Potential field solutions Another approach to solving the formation control problem is to consider the mechanical nature of the systems and to shape the dynamics of the formation using potential fields. In this case, the control law for the individual vehicles has the form

$$u^i = \nabla V(\alpha^i, x^i, x^{-1})$$

where V is a potential function that depends on the mode of the vehicle, α^i (typically whether it is a leader or a follower).

A representative body of research in this area is the work of Fiorelli and Leonard, who use the concept of "virtual leaders" that guide the motion of the other vehicles [29, 32]. They consider two types of potential functions: an interaction function V_I and a potential generated by "leaders", V_h. Each function generates a repulsive force if vehicles are very close to each other, an attractive force if the vehicles are within some interaction range but not too close or too far and no force for vehicles beyond a certain radius. Their resulting control law is of the form

$$u^i = -\sum_{j \neq i}^{N} \nabla V_I(\|y^i - y^j\|) - \sum_{k \in \mathcal{L}} \nabla V_h(\|y^i - y^k\|) + f_{v^i},$$

where $\mathcal{L}$ is the set of leaders, f_{v^i} is a dissipative force based on the velocity of the i vehicle, and local coordinates are used for $y^i \in SE(3)$. By appropriate choice of f_{v^i} they are able to show asymptotic stability of various schooling and flocking behaviors.
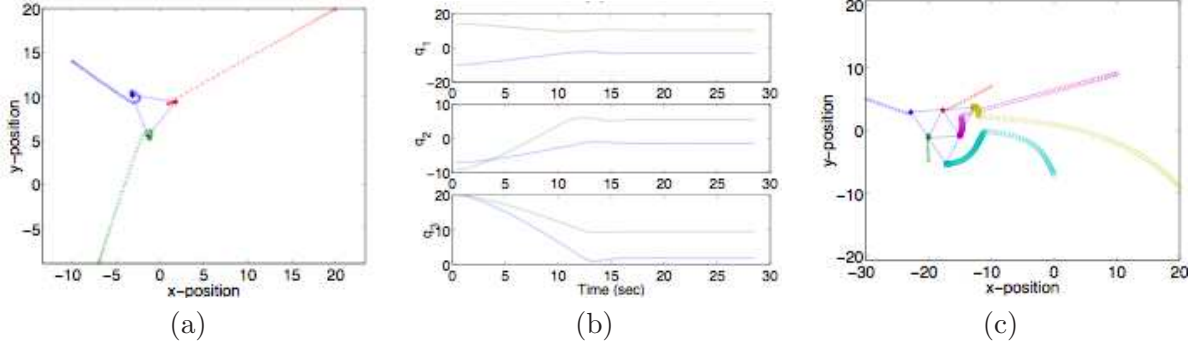
10

Figure 7: Formation stabilization using potential functions [33]. (a) Stabilization of three vehicles in the plane. (b) Time traces for individual positions of the vehicles. (c) Stabilization of a six vehicle formation.

Other work on the use of potential fields includes that of Olfati-Saber [33], who uses potential functions obtained from structural constraints of a desired formation in a way that leads to a collision-free, distributed, and bounded state feedback law for each vehicle. Figure 7 demonstrates some of the results of his algorithm for formation control.

String stability One issue that arises in formation control is that of "string stability," in which disturbances grow as they propagate through a system of vehicles [42]. One of the early sources of research on this problem was in the control of vehicle platoons, in which one wanted to insure that small disturbances at the beginning of a chain of vehicles did not get amplified as one progressed down the chain.

For simplicity, we assume that the disturbances enter through the initial states of the vehicles. String stability is defined in terms of an infinite collection of vehicles and our goal is to find a control law for each of the vehicles so that given $\epsilon > 0$ there exists a $\delta > 0$ such that

$$\sup_i \|x^i(0)\| < \delta \quad \Longleftrightarrow \quad \sup_i \|x^i(\cdot)\|_\infty < \epsilon,$$

where the ∞ norm is taken with respect to time. In particular, this implies that the motion of each vehicle is bounded for all time. More general norms can also be used, as described in [42].

Using this definition, one can show that a system is string stable if the H_∞ gain between any two neighbors is less than one. If this is the case, then disturbances are attenuated as they pass down the chain of vehicles. Conversely, if the dynamics and control laws for each vehicle are identical and if the gain of the transfer function is greater than 1 at some frequency, then disturbances at that frequency can be amplified as they propagate down the chain. These definitions can be generalized to different topologies in which the neighbor sets are more complicated than a single chain.

To help compensate for string instabilities, one can make use of globally transmitted information that allows the vehicles to pre-compensate for disturbances. In essence, one changes the topology of the information flow from one in which each vehicle only sees the vehicle in front of it, to one in which vehicles also have global information about the position of the lead vehicle. Figure 8 shows the responses of a set of vehicles with different topologies and different levels of global information. In this simulation, the lead vehicle responds to a step input at time $t = 15$. The variable α controls
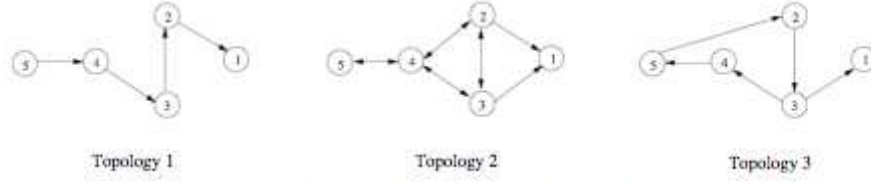
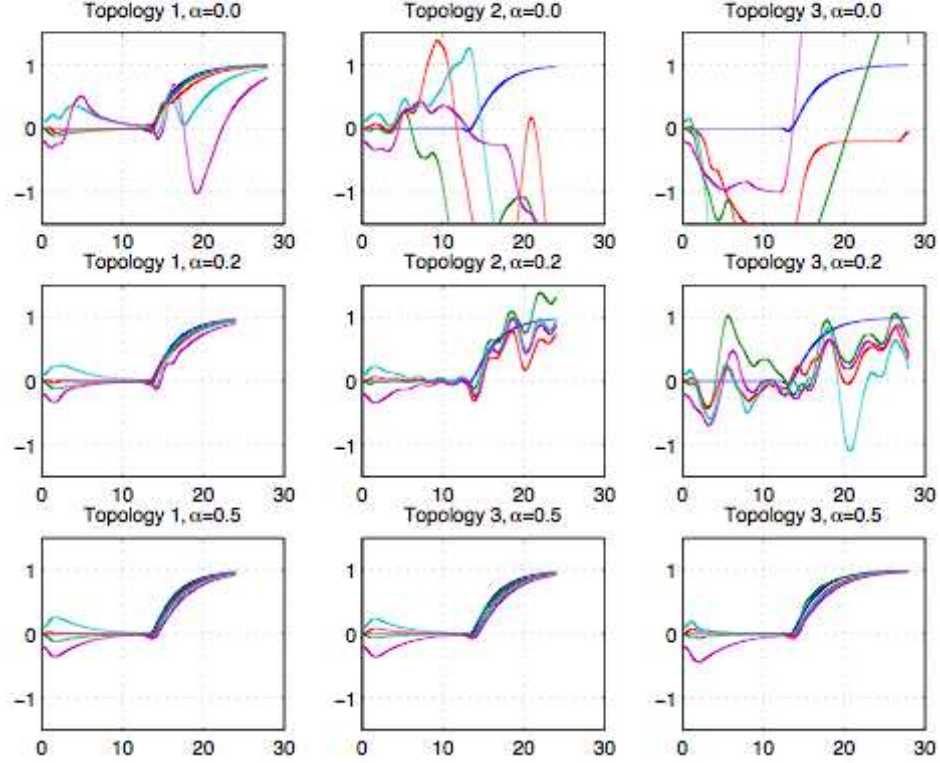Fig. 9. Three topologies of the five-vehicle formation

Figure 8: String stability results for a five vehicle formation [23]. Each column represents a different information topology, as shown in the diagram at the top of the column. The first row of plots corresponds to the use of purely local information, while the second two rows allow increasing amounts of global information.

the amount of mixing between the purely local strategy ($\alpha = 0$) and a purely centralized strategy ($\alpha = 1$).

It is also possible to define the performance in ways that are more structured than string stability, for example asking whether the distances between specified sets of vehicles have certain levels of disturbance attenuation [22, 21].

Swarms Finally, although not strictly a formation control problem, there has been a great deal of interest in so-called "swarms" of vehicles. Roughly speaking, a swarm is a large collection of vehicles that perform in a collective fashion, such as flying together in a given direction. One early work in swarm-like behavior was that of Reynolds, who developed a set of rules that he used to

generate realistic motion of vehicles for animation purposes [38].

An innovative approach to understanding swarm behavior was taken by Jadbabaie, Lin and Morse [19], who described how to achieve coordination fo groups of mobile autonomous agents using nearest neighbor rules. The control law was quite simple, making use of a simple heading model in which each agent updated its heading according to the rule

$$u^i = \frac{1}{1 + |\mathcal{N}^i(t)|} \left(\theta^i(t) + \sum_{j \in \mathcal{N}^i(t)} \theta^j(t) - \theta^i(t) \right)$$

where $\mathcal{N}^i(t)$ is the set of vehicle that are within a radius r of vehicle i at time t. The first term is the average heading of the neighbors of vehicle i and hence this control essentially tells each vehicle to steer in the same direction as its neighbors.

Jadbabaie et al. are able to demonstrate that with this control law, all vehicles will converge to a common heading. They make use of an "eventual connectivity" assumption in which the vehicles are connected together across intervals. In other words, while it may never be the case that at a given instant of time the graph describing the interconnectivity is complete, as long as over a suitable interval all vehicles are able to share information, the solution will converge to a common value.

Control laws for swarms often involve using attractive and repulsive functions between nearby vehicles. In addition to the work of Leonard et al. already described above, another representative work in this regard is that of Olfati-Saber [40], who makes of a control input consisting of three terms

$$u^i = f_g^i + f_d^i + f_\gamma^i.$$

The first term $f_g^i = -\nabla V(y^i, y^{-1})$ is a gradient-based term where V is a potential function. The second term f_d^i is a damping term based on the relative velocities of neighboring vehicles and has the form $\alpha(q)(v^i - v^j)$. The final term f_γ^i is a navigational feedback term that takes into account a group objective, such as moving to a given rendezvous point. Figure 9 shows a sample maneuver in which 150 agents squeeze through an opening without collision.

3.2   Cooperative Tasking

A major element of cooperative control is deciding on the tasks that different vehicles will perform to satisfy the team objective. This essentially amounts to choosing the role of the vehicles, α^i.

MILP formulations   Several groups have formulated this problem as a mixed integer linear program (MILP), in which the integer variables correspond to the role α^i.

The work of Richards et al. [39] considers the problem of designing trajectories for a group of vehicles that collectively visit a set of waypoints within a given set of time constraints. They minimize a cost function of the form

$$J = \bar{t} + \rho_1 \sum_{p=1}^{N} \left(t^p + \rho_2 \sum_{t=0}^{T} \left(|u_1(t)| + |u_2(t)| \right) \right)$$

where t^p is the time at which the pth vehicle completes its task and $\bar{t}$ is the time at which the last vehicle completes its task. This cost function thus trades off the input forces on the vehicles with the time that the overall task is completed as well as the tasks of the individual vehicles.
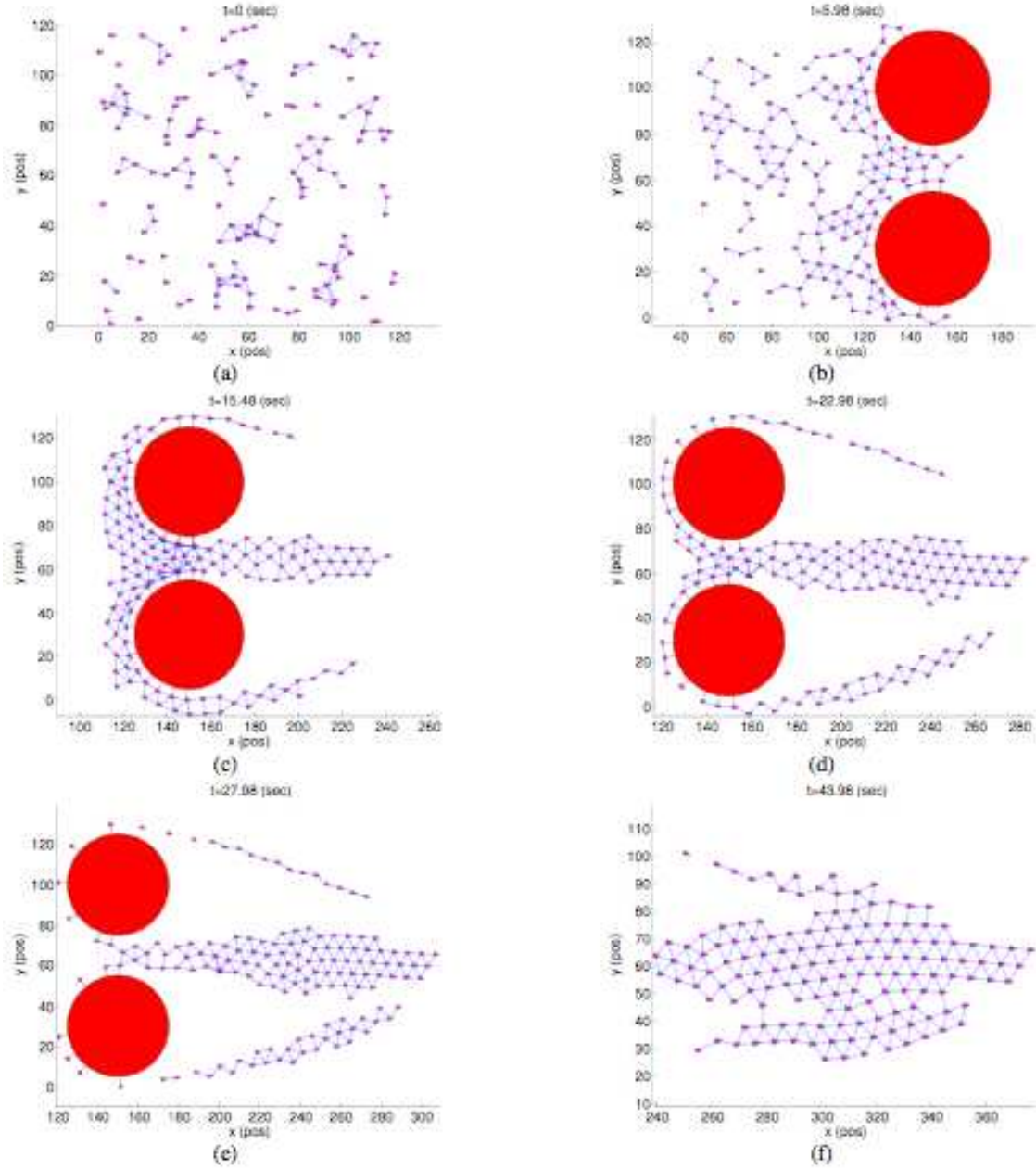
Figure 9: A squeezing maneuver using flocking algorithms of Olfati-Saber [40].

In the MILP formulation used by Richards et al. [39], the individual assignments of waypoints to vehicles is handled by using decision variables to constrain the problem such that each waypoint is visited exactly once by a vehicle. This constraint can be written in the form

$$\sum_{t=0}^{T} \sum_{p=1}^{N} K_{pi} b_{ipt} = 1 \qquad \text{for all waypoints } i$$
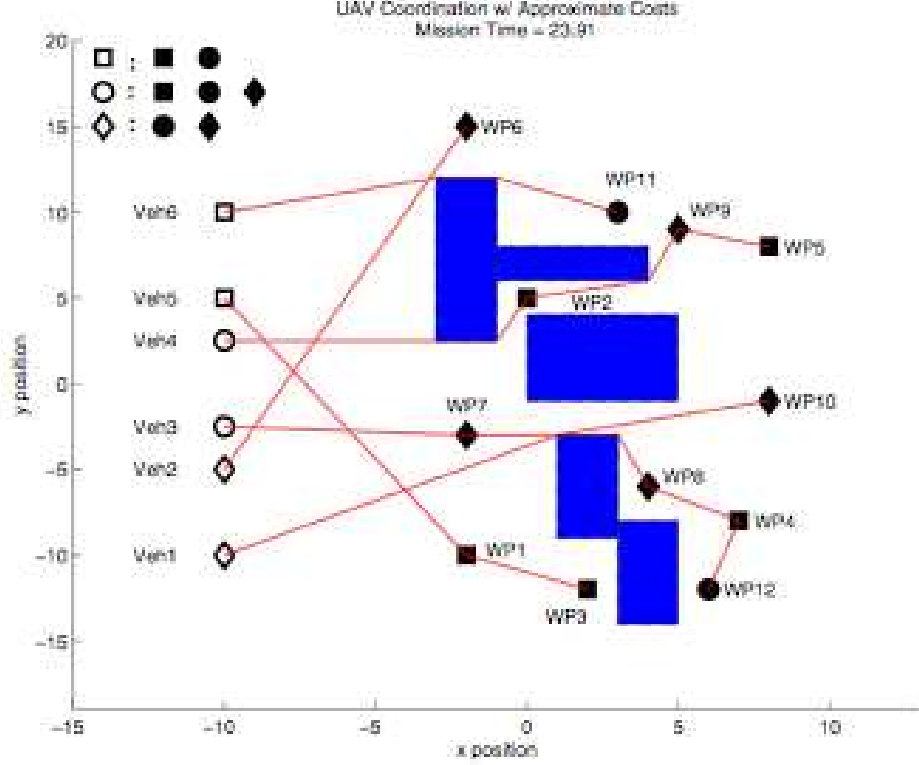
14

Figure 10: Resource allocation using mixed integer linear programming (MILP) [39].

where K_{pi} is the suitability of vehicle p to visit waypoint i and b_{ipt} is 1 if vehicle p visits waypoint i and time t and zero otherwise.

Figure 10 shows an example of the allocation problem applied to set set of 6 vehicles. The scenario includes 12 waypoints that must all be visited, along with a region of no fly zones (obstacles). An approximate method described in [39] is used to solve the problem in 27 seconds on a standard PC.

A similar approach has been developed independently by Earl and D'Andrea [12], in which the MILP formulation is used to solve a subproblem of the RoboFlag example in Section 2.4. Specifically, they solve the problem of guarding a defense zone from attackers that are trying to enter it. They formulate the problem in discrete time to be consistent with the MILP framework; for simplicity we will use a single time discretization here and re-use t as the discrete time.

The objective function is given by

$$J = \sum_{t=0}^{T} \gamma(t) + \rho \sum_{t=0}^{T} |u(t)|$$

where $\gamma(t)$ is a binary variable that takes on the value 1 if and only if one of the attackers is in the defense zone at time t. This function must be minimized while also constraining the position of the defending robots so that they avoid collisions with each other and stay outside of the defense zone.

In addition to the dynamics of the vehicles, a complete description of the problem also requires that we define the dynamics of the attacking robots. We do this using a describe variable β^i for
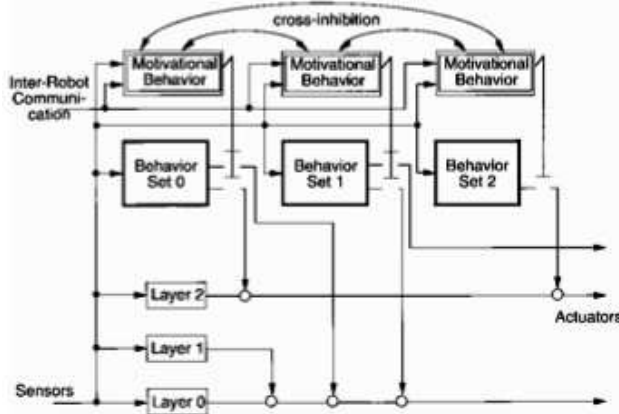
15

Figure 11: The ALLIANCE architecture [36].

each *attacker* that describes whether an attacker is active or inactive. An attacker is active initially and becomes inactive if it either enters the defense region or is "intercepted" by a defending robot (modeled by a defending robot getting within a certain distance of the attacking robot). We assume that if an attacking robot is active, it moves toward the defense zone in a straight line.

Note that in both of these formulations, the assignment is handled implicitly: the problem does not explicitly assign a given defender to specific attacker, but rather relies on the optimization to choose motions of the group of defenders such that no attackers enter the defense region.

Assignment protocols Another approach to the cooperative tasking problem has been to develop protocols that are used to decide on who is assigned to what task. By "protocol" we mean a set of rules that are used to determine the individual roles (assignments) of each vehicle. One seeks to prove that this protocol results in all tasks being assigned to a vehicle, even in the presence of changing environmental conditions or failures.

One of the early approaches to distributed task allocation was the ALLIANCE software architecture developed by Parker [36]. Their approach made us of behavior sets that were activated under certain conditions. Each behavior could itself inhibit other behaviors, so that it was possible for a single behavior set to control the motion of the robot. Figure 11 illustrates this architecture.

The activation of a behavior set is controlled through "motivational behaviors". Each motivation behavior responds to some set of inputs, including external sensors, inter-robot communications, inhibitory feedback from other behaviors, and internal motivations. The two internal motivations, robot impatience and robot acquiescence, allow the robot to progress when other robots fail to complete a task or when the robot itself fails to accomplish a task. These motivational behaviors can be viewed in the context of the guarded command framework discussed in Section 1.

A related approach has been taken by Klavins [26], who constructed a language for describing and verifying protocols for cooperative control. The computation and control language (CCL) uses the guarded command formalism to specify sets of actions for a collection of robots. Figure 12 gives an example of how a distributed area denial task can be solved in CCL. In this example, drawn from the RoboFlag game, 6 defensive robots are trying to protect a defense zone for an incoming set of robots, which descend vertically at a fixed speed. The defending robots must move underneath the incoming robots, but are not allowed to run into each other. The defenders are randomly assigned
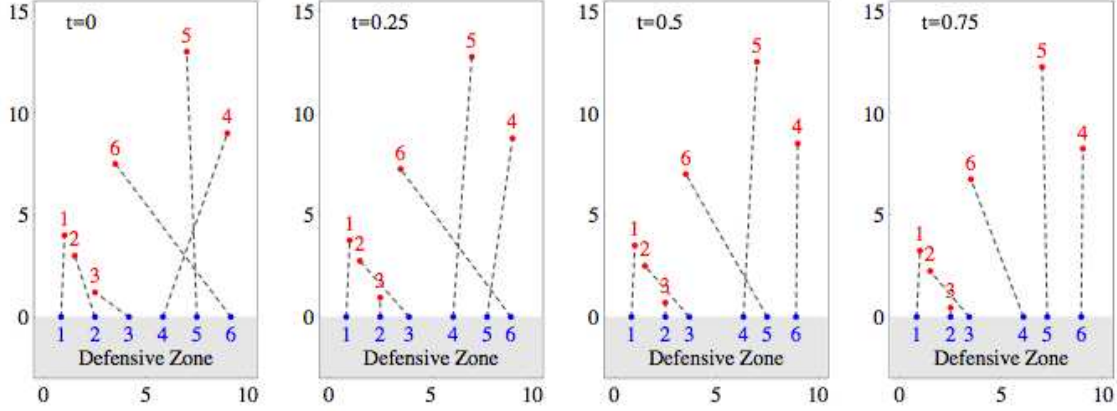
16

Figure 12: The RoboFlag Drill.

incoming robots and are allowed to talk to their neighbors and switch assignments under a given protocol. A protocol was developed in [26] that is able to provably solve this problem, including insuring that no two robots collide and that all defensive robots eventually end up assigned to an incoming robot with now crossing of assignments. Extensions to this approach for observability and controllability have also been developed [9, 8].

3.3 Spatio-Temporal Planning

A broad collection of technological developments can be described under the heading of "spatio-temporal planning", in which the paths of the robots and their locations with respect to time are to be specified and controlled. In this section we consider two typical spatio-temporal planning problems: rendezvous and coverage.

Rendezvous The rendezvous problem is a specific cooperative task in which one wants to have a number of individual vehicles meet at a common point at a common time. The key element in the rendezvous problem is that all agents should arrive at the *same* time, so that if one vehicle is delayed the other vehicles should adjust their trajectories so that they do not arrive early.

Bhattacharya et al. [1, 43] formulated the rendezvous problem by defining a rendezvous region $\mathcal{R}$ around the rendezvous point (take as the origin) and letting ρ be the ratio of the maximum and minimum distances of the vehicles at the time that one of them enters the rendezvous point. Letting δ be the radius of the rendezvous region and t_a the time in which the time at which the first vehicle enters the region, they define ρ as

$$\rho = \frac{\max(\|x_i(t_a)\|)}{\delta}.$$

The goal can then be defined as finding control laws such that from all initial conditions,

$$\rho \leq \rho_{\text{des}} \leq 1.$$

The case of "perfect" rendezvous corresponds to $\rho = 1$, in which case all vehicles must reach the rendezvous region at precisely the same time.
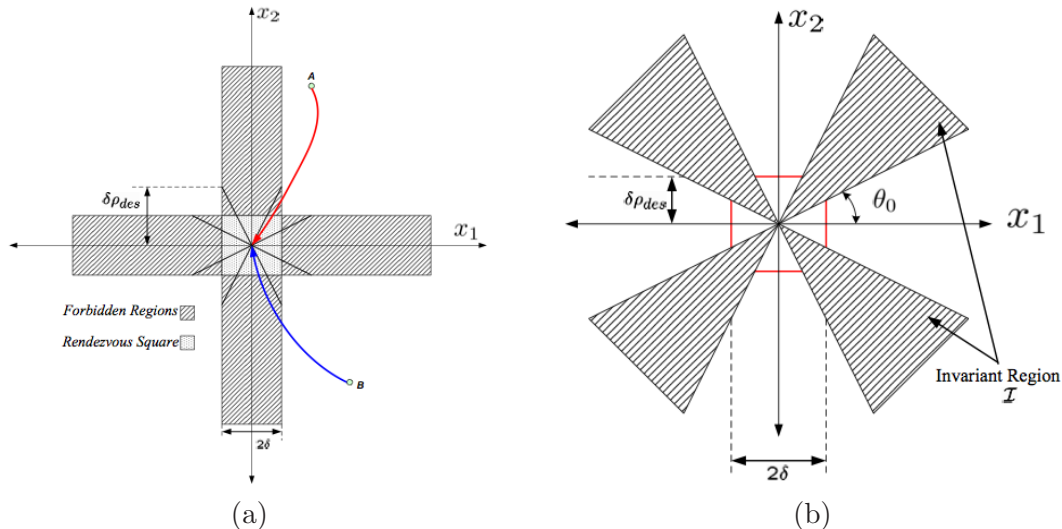
17

Figure 13: (a) Definition of the rendezvous problem for two scalar agents. (b) Solution via construction of invariant cones.

This problem can be solved using a Lyapunov-based approach that uses feedback to create an invariant cones in the phase space [43, 1], as illustrated in Figure 13. The problem definition is shown in Figure 13a, which shows the phase space for two scalar vehicles. To achieve rendezvous, these vehicles must reach $x = 0$ at approximately the same time, without either of the individual vehicles coming near $x = 0$ before that time. This creates a set of forbidden regions in the phase space. By proper choice of control law, it is possible to render certain cones as invariant, as shown in Figure 13b. The resulting trajectories satisfy the rendezvous problem. The feedback in this case is centralized, requiring each vehicle to communicate its position to nearby vehicles.

Coverage The coverage control problem refers to the use of a collection of vehicles to provide sensor coverage for a given geographic area. It is thus one approach to the cooperative surveillance problem. Given a set of N vehicles, we wish to allocate each vehicle to a region in which it is responsible for providing senor information. The centralized version of this problem is referred to as the locational optimization problem and there is a large literature describing different approaches (see [4] for a survey). We focus here on the decentralized solution proposed by Cortes et al. [4].

The approach taken by Cortes et al. is to partition a region Q into a set of polytopes $\mathcal{W} = \{W^1, \ldots, W^N\}$ that cover Q. Each polytope is assigned to a specific vehicle to each region and we let $f^i : \mathbb{R}_+ \to \mathbb{R}_+$ represent the sensing performance of a vehicle based on its distance from a given point, with f small representing good performance. We then form the coverage control problem as choosing the locations of each vehicle such that we minimize

$$L = \sum_{i=1}^{n} \int_{W^i} f(\|q - y^i\|)\phi(q)dq, \tag{1}$$

where $\phi(q)$ is a distribution density function that represents the importance of a given area.

It can be shown that if the location of the vehicles are fixed, the optimal decomposition of the
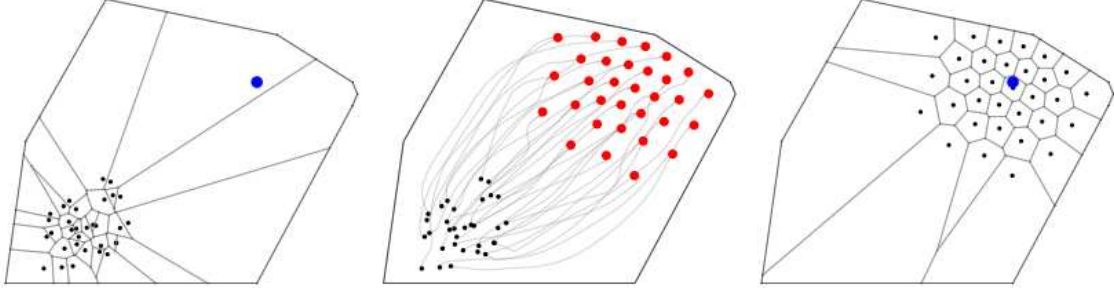
18

Figure 14: Coverage control applied to a polygonal region with Gaussian density function around the point in the upper right [4].

space Q is a Voronoi decomposition where

$$W^i = \{q \in Q | \|q - y^i\| \leq \|q - y^j\|, \forall j \neq i\}.$$

This decomposition corresponds to each vehicle being responsible for the points that are closest to it. This decomposition also introduces a natural graph of neighbors, with two vehicles being neighbors if their Voronoi partitions share an edge.

If we let C_{V^i} represent the centroids of the Voronoi partition, then it turns out that the control law

$$u^i = -k(y^i - C_{V^i})$$

converges asymptotically to a set of critical points for the cost function, and hence provides (locally) optimal coverage. A key element of this approach is that the only communication required is with the nearest neighbors of the vehicle (since this is what is needed to determine the Voronoi decomposition). Figure 14 illustrates the coverage algorithm applied to a region with $\phi(q)$ being a Gaussian around the point in the upper right portion of the region.

The above formulation assumes that the collection of vehicles that is available is sufficient to cover the entire region of interest. A slightly different problem occurs when there is not enough sensor range to simultaneous view all portions of the environment that are of interest. In this case, one must selectively cover different regions of space and change those regions over time (so that no region goes unviewed forever). Several groups have considered this problem [15, 34, 44]

3.4 Consensus algorithms

As a final technology in cooperative control, we briefly describe the problem of "consensus". The consensus problem is to have a group of vehicles (or more general agents) reach a common assessment or decision based on distributed information and a communications protocols. Many of the decentralized problems listed above, especially those involving assignment, can be thought of as special cases of consensus.

The consensus problem has been formulated as a coordinated control problem by Fax [13] and Olfati-Saber [41]. A particularly simple solution to the consensus problem is to let the behavior of each agent be governed by the first order differential equation

$$\dot{x}^i = -\frac{1}{|\mathcal{N}^i|} \sum_{j=1}^{|\mathcal{N}^i|} (x^i - x^j),$$

where $x^i \in \mathbb{R}$ is the internal state of the agent. For this system, one can show that if the information flow is bidirectional (if agent i is a neighbor of agent j, then j is a neighbor of i), the states of the individual vehicles asymptotically converage to the average of the initial state values for any connected graph $\mathcal{G}$.

If $\mathcal{G}$ is not bidirectional (so that there are asymmetries in the information available to each agent), then the interaction above does not necessarily lead to average consensus. We define a graph to be *balanced* if the in-degree and out-degree of all nodes are equal. In the case of balanced graphs, one can once again show that any connected graph solves the average consensus problem using the interaction rules above [41]. Furthermore, even if the connections are changing as a function of time, it can be shown that the average consensus is still reached.

When the behavior of the individual agents is more complicated, we can still pose the problem in a similar manner. Supposed that each agent's dynamics are governed by

$$\dot{x}^i = Ax^i + Bu^i$$
$$y^i = Cx^i \tag{2}$$

Fax [13] considers a control law in which each system attempts to stabilize itself relative to its neighbors. This is accomplished by constructing an error for each system that is a weighted combination of the relative outputs of the neighbors:

$$e^i = \sum_{j \in \mathcal{N}^i} \alpha_{ij}(y^j - y^i) \tag{3}$$

where α_{ij} is the relative weight. For simplicity, we consider uniform weighting here, so that $\alpha_{ij} = 1/|\mathcal{N}^i|$ where $\mathcal{N}^i$ is the number of neighbors of node i. The results are easily extended to the more general case.

Given the error (3), we apply a compensator that attempts to stabilize the overall system. For simplicity, we assume here that the stabilizer is given by a constant gain

$$u^i = Ke^i, \tag{4}$$

with $K \in \mathbb{R}^{m \times m}$ representing the compensation (gain) matrix. In practice, one can use a dynamic compensator to improve performance, but for analysis purposes we can just assume these dynamics are included in the system dynamics (2).

The interconnectedness of the system, represented by the neighbor sets $\mathcal{N}_\rangle$ can be studied using tools from graph theory. In particular, for the the case of uniform weighting of the errors, it turns out that the combined error vector $e \in \mathbb{R}^{N \cdot m}$ can be written as

$$e = (\bar{L} \otimes I)x \tag{5}$$

where $\otimes$ represents the Kronecker product and $\bar{L}$ is the *weighted Laplacian* associated with the (directed) graph that models the neighbors of each node. The weighted Laplacian is a standard object in graph theory and can be defined as

$$\bar{L} = D^{-1}(D - A)$$

where D is a diagonal matrix whose entries are the out-degree of each node and A is the adjacency matrix for the graph (see [13] for more detail). Using this framework, Fax showed the following:
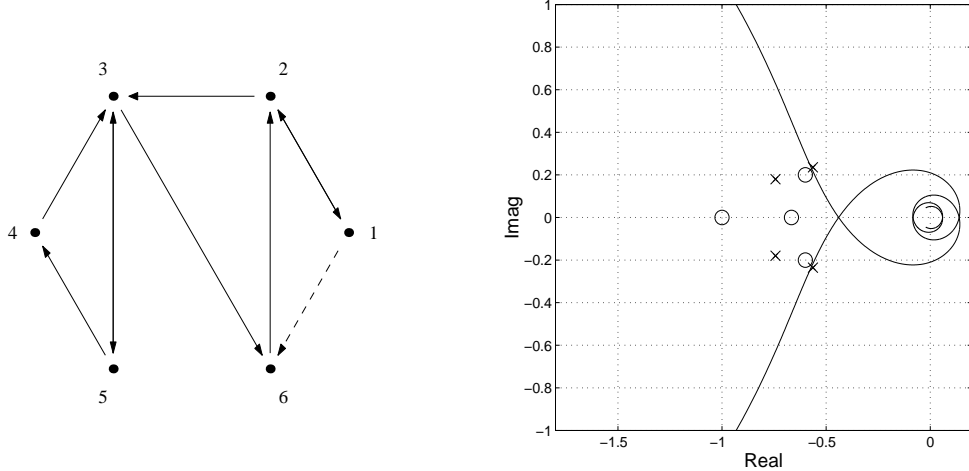
Figure 15: Interpretation of Theorem 1. The left figure shows the graph representation of the interconnected system and the right figure shows the corresponding Nyquist test. The addition of the dashed line to the graph moves the negative, inverse eigenvalues of $\bar{L}$ from the positions marked by circles to those marked by crosses.

Theorem 1. *A local controller K stabilizes the formation dynamics in equation (2) with error (5) and gain K if and only if it stabilizes the set of N systems given by*

$$\begin{aligned}
\dot{x} &= Ax + B \cdot \lambda_i \cdot (Ky) \\
y &= Cx
\end{aligned} \tag{6}$$

where λ_i are the eigenvalues of the weighted graph Laplacian $\bar{L}$.

This theorem has a very natural interpretation in in terms of the Nyquist plot of dynamical system. In the standard Nyquist criterion, one checks for stability of a feedback system by plotting the open loop frequency response of the system in the complex plane and checking for net encirclements of the -1 point. The conditions in Theorem 1 correspond to replacing the -1 point with $-1/\lambda_i$ for each eigenvalue λ_i of $\bar{L}$. This interpretation is illustrated in Figure 15. The results can easily be extended to consider weightings that are nonuniform.

Theorem 1 illustrates how the *dynamics* of the system, as represented by equation (2), interacts with the *information flow* of the system, as represented by the graph Laplacian. In particular, we see that it is the eigenvalues of the Laplacian that are critical for determining stability of the overall system. Additional results in this framework allow tuning of the information flow (considered as both sensed and communicated signals) to improve the transient response of the system [13]. Extensions in a stochastic setting [16, 41] allow analysis of interconnected systems whose dynamics are not identical and where the graph topology changes over time.

4 Future Directions

While there has been substantial work in cooperative control over the past decade, there are still many open problems that remain to be solved. In this section we provide a brief review of some

of the future opportunities in cooperative control. The topics listed here are not intended to be exhaustive, but rather to be indicative of the classes of problems which remain open. Many of these are drawn from the recent report on future directions in control, dynamics and systems [31].

4.1 Integrated control, communications and computer science

By its very nature, cooperative control involves the integration of communications and (distributed) computing systems with feedback control. In many applications, the traditional separation of computing, communications and control is no longer valid and new methods that integrate advances from the different disciplines are needed. Research research in hybrid systems, in which continuous and logical domains are integrated, are a step in the right direction but these techniques often ignore issues associated with distributed computing and communication channels. Theories that define fundamental limits such as real-time computational complexity and performance limits of feedback systems with rate limited channels are needed.

4.2 Verification and validation

Prescribed safety and reliability is a significant challenge for current mission-critical systems. Requirements, design, and test coverage and their quantification all significantly impact overall system quality, but software test coverage is especially significant to development costs. For certain current systems, verification and validation (V&V) can comprise over 50% of total development costs. This percentage will be even higher using current V&V strategies on emerging autonomous systems. Although traditional certification practices have historically produced sufficiently safe and reliable systems, they will not be cost effective for next-generation autonomous systems due to inherent size and complexity increases from added functionality.

New methods in high confidence software combined with advances in systems engineering and the use of feedback for active management of uncertainty provide new possibilities for fundamental research aimed at addressing these issues. These methods move beyond formal methods in computer science to incorporate dynamics and feedback as part of the system specification.

4.3 Higher levels of decision making

The research surveyed in this paper has focused on cooperative control problems that can be formalated as optimization problems over some cost function. Many autonomous systems must make decisions for which an underlying set of continuous and discrete variables may not provide an appropriate level of abstract for decision making. Cooperative systems that must reason about the complex interactions between the group's dynamics and the environment in which they operate may require different levels of representation of their task and their dynamics. Techniques from artificial intelligence that allow identification of strategies and tactics that can be coded as lower-level optimization-based problems are needed.

4.4 Networked control systems

Modern control theory is largely based on the abstraction that information ("signals") are transmitted along perfect communication channels and that computation is either instantaneous (continuous time) or periodic (discrete time). This abstraction has served the field well for 50 years and has led to many success stories in a wide variety of applications.
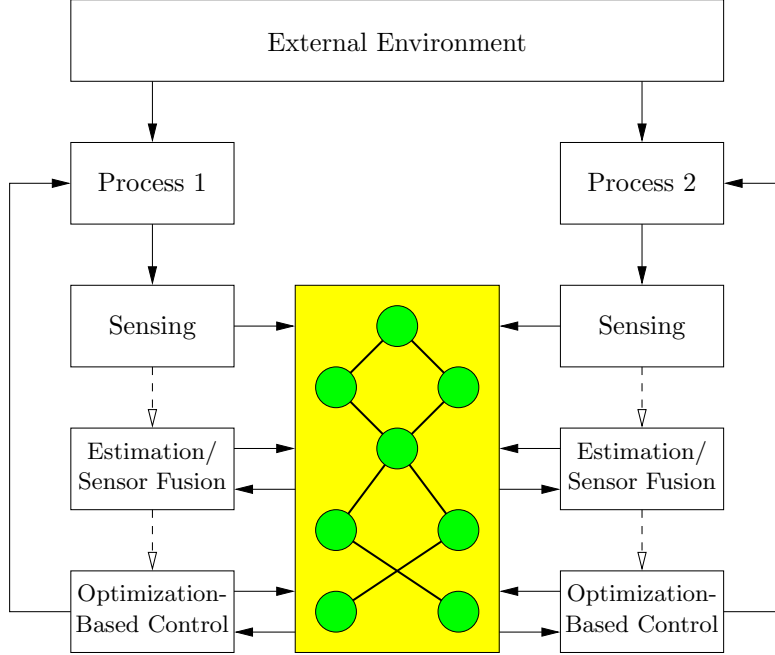
Figure 16: Control architecture for a networked control system.

Future applications of control will be much more information-rich than those of the past and will involve networked communications, distributed computing, and higher levels of logic and decision-making, as described above. New theory, algorithms, and demonstrations must be developed in which the basic input/output signals are data packets that may arrive at variable times, not necessarily in order, and sometimes not at all. Networks between sensors, actuation, and computation must be taken into account, and algorithms must address the tradeoff between accuracy and computation time. Progress will require significantly more interaction between information theory, computer science, and control than ever before.

An emerging architecture for networked control systems is shown in Figure 16. This architecture separates the traditional elements of sensing, estimation, control, and actuation for a given system across a network and also allows sharing of information between systems. Careful decisions need to be made on how the individual components in this architecture are implemented and how the communications across the networked elements is managed. This architecture can be used to model either a single system (using either half of the diagram) or multiple systems that interact through the network.

The opportunity for networked control systems is the ability to quickly add functionality to systems by incorporating new sensors and algorithms into an existing system. For cooperative control systems, the ability to make use of a "virtual sensor" located on another vehicle without having to redesign the control system from scartch is an example of the types of functionality one would like to achieve. Similarly, new software modules that add functionality should provide "plug and play" compatibility so that they can be integrated into systems quickly and reliably. This "network centric" approach to control will require substantially better frameworks for implementing complex, distributed control systems than currently exist today.

5 Conclusions

In this survey we have described some of the driving applications of cooperative control, surveyed some of the relevant technology that has been developed over the past decade and provided some possible directions for future study. Given the large and growing literature in this area, many interesting results have not been included in an attempt to capture some of the key areas are interest.

What is clear is that many of the basic problems of cooperative control have been explored and a wealth of results are available demonstrating the potential of such systems. To transition these research results to applciations will require additional effort in the integration of control, communications and computer science; decision-making at higher levels of abstraction; verification and validation of distributed embedded systems; and an extensible architecture for networked control systems implementation.

References

[1] R. Bhattacharya, A. Tiwari, J. Fung, and R. M. Murray. Cone invariance and rendezvous of multiple agents. In *IEEE Transactions on Automatic Control*, 2006. Submitted.

[2] California Partners for Advanced Transit and Highways. `http://www.path.berkeley.edu`, 2006.

[3] P. R. Chandler, M. Pachter, and S. Rasmussen. Uav cooperative control. In *Proc. American Control Conference*, pages 50–55, 2001.

[4] J. Cortes, S. Martnez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. In *Proc. IEEE International Conference on Robotics and Automation*, pages 1327–1332, 2002.

[5] L. Cremean, W. Dunbar, D. van Gogh, J. Hickey, E. Klavins, J. Meltzer, and R. M. Murray. The caltech multi-vehicle wireless testbed. In *Proc. IEEE Control and Decision Conference*, 2002.

[6] R. D'Andrea. Cornell RoboCup. `http://robocup.mae.cornell.edu`, 2005.

[7] R. D'Andrea and R. M. Murray. The RoboFlag competition. In *Proc. American Control Conference*, 2003.

[8] D. Del Vecchio. Discrete dynamic feedback for a class of hybrid systems on a lattice. In *IEEE International Symposium on Computer-Aided Control Systems Design*, 2006.

[9] D. Del Vecchio, R. M. Murray, and Erik Klavins. Discrete state estimators for systems on a lattice. *Automatica*, 2006.

[10] Department of Transportation. National transportation library—intelligent transporatation systems. `http://ntl.bts.gov/ref/biblio/its/its_references.html`.

[11] W. B. Dunbar and R. M. Murray. Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, 42(4):549–558, 2006.

[12] M. G. Earl and R. D'Andrea. Multi-vehicle cooperative control using mixed integer linear programming. Technical Report arXiv:cs.RO/0501092, `http://arXiv.org`, 2005.

[13] J. A. Fax and R. M. Murray. Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 2004.

[14] F. B. Gill. *Ornithology*. W. H. Freeman, second edition, 1994.

[15] V. Gupta, T. H. Chung, B. Hassibi, and R. M. Murray. On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage. *Automatica*, 2004. In revision.

[16] V. Gupta, R. M. Murray, and B. Hassibi. Stability analysis of stochastically varying formations of dynamic agents. In *Proc. IEEE Control and Decision Conference*, 2003.

[17] G. Hoffmann, D. G. Rajnarayan, S. L. Waslander, D. Dostal, J. S. Jang, and C. J. Tomlin. The Stanford testbed of autonomous rotorcraft for multi-agent control (STARMAC). In *AIAA Digital Avionics Systems Conference*, 2004.

[18] Intelligent Transportation Society of America. `http://www.itsa.org`.

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

[20] Jet Propulsion Laboratory. Terrestrial planet finder. `http://tpf.jpl.nasa.gov`.

[21] Z. Jin. *Coordinated Control of Networked Multi-Agent Systems*. PhD thesis, California Institute of Technology, Electrical Engineering, 2006.

[22] Z. Jin and R. M. Murray. Stability and performance analysis with double-graph model of vehicle formations. In *Proc. American Control Conference*, 2003.

[23] Z. Jin and R. M. Murray. Double-graph control strategy of multi-vehicle formations. In *Proc. IEEE Control and Decision Conference*, 2004.

[24] E. King, Y. Kuwata, M. Alighanbari, and J. How. Coordination and control experiments for uav teams. *Advances in the Astronautical Sciences*, 118:145–155, 2004.

[25] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa. Robocup: The robot world cup initiative. In *Autonomous Agents 97*, pages 340–347, 1997.

[26] E. Klavins. A formal model of a multi-robot control and communication task. *Proc. IEEE Control and Decision Conference*, 2003.

[27] E. Klavins and R. M. Murray. Distributed computation for cooperative control. *IEEE Pervasive Computing*, 2003. In revision.

[28] E. Lavretsky. F/a-18 autonomous formation flight control systems design. In *aiaaGNC*, pages AIAA 2002–4757, 2002.

[29] N. E. Leonard and E. Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. In *Proc. IEEE Control and Decision Conference*, pages 2968–2973, 2001.

[30] Monterey Bay Aquarium Research Institute. Autonomous ocean sampling network. `http://www.mbari.org/aosn`, 2006.

[31] R. M. Murray, editor. *Control in an Information Rich World: Report of the Panel on Future Direcitons in Control, Dynamics and Systems*. SIAM, 2003. Available at `http://www.cds.caltech.edu/~murray/cdspanel`.

[32] P. Ogren, E. Fiorelli, and N. E. Leonard. Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *IEEE Transactions on Automatic Control*, 49(8):1292–1302, 2004.

[33] R. Olfati-Saber and R. M. Murray. Distributed cooperative control of multiple vehicle formations using structural potential functions. In *Proc. IFAC World Congress*, 2002.

[34] J. Ousingsawat and M. E. Campbell. Estabilishing trajectories for multi-vehicle reconnaissance. In *AIAA Conference on Guidance, Navigation, and Control*, pages AIAA 2004–5224, 2004.

[35] L. E. Parker. Designing control laws for cooperative agent teams. In *Proc. IEEE International Conference on Robotics and Automation*, pages 582–587, 1993.

[36] L. E. Parker. ALLIANCE: An architecture for fault tolerant multirobot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2):220–240, 1998.

[37] L. E. Parker. Current state of the art in distributed autonomous mobile robotics. In *International Symposium on Distributed Autonomous Robotic Systems (DARS)*, 2000.

[38] C. W. Reynolds. Herds, and schools: A distributed behavioral model. *Computer Graphics (SIGGRAPH '87 Conference Proceedings)*, 21(4):25–34, 1987.

[39] A. Richards, J. Bellingham, M. Tillerson, and J. How. Co-ordination and control of multiple UAVs. In *AIAA Conference on Guidance, Navigation, and Control*, 2002.

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

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

[42] D. Swaroop and J. K. Hedrick. String stability of interconnected systems. *IEEE Transactions on Automatic Control*, 41(3):349–357, 1996.

[43] A. Tiwari, J. Fung, J. M. Carson III, R. Bhattacharya, and R. M. Murray. A framework for Lyapunov certificates for multi-vehicle rendezvous problems. In *Proc. American Control Conference*, 2004.

[44] A. Tiwari, M. Jun, D. E. Jeffcoat, and R. M. Murray. The dynamic sensor coverage problem. In *Proc. IFAC World Congress*, 2005.

[45] C. Tomlin, G. J. Pappas, and S. Sastry. Conflict resolution for air traffic management: A study in multiagent hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):509–521, 1998.

[46] United States Department of Defense. Network centric warfare. Report to Congress, July 2001. Available at `http://www.dod.mil/nii/NCW`.

[47] V. Vladimerou, A. Stubbs, J. Rubel, A. Fulford, and G.E. Dullerud. A hovercraft testbed for decentralized and cooperative control. In *Proc. American Control Conference*, 2004.