1

Overview of Consensus Algorithms in Cooperative Control

This chapter overviews consensus algorithms in cooperative control. The motivation for information consensus in cooperative control is given. A literature review on consensus algorithms is provided. Theoretical results regarding consensus-seeking under both time-invariant and dynamically changing communication topologies are summarized. A few specific applications of consensus algorithms to multivehicle cooperative control are described. The organization of the monograph is also introduced.

1.1 Introduction

The abundance of embedded computational resources in autonomous vehicles enables enhanced operational effectiveness through cooperative teamwork in civilian and military applications. Compared to autonomous vehicles that perform solo missions, greater efficiency and operational capability can be realized from teams of autonomous vehicles operating in a coordinated fashion. Potential applications for multivehicle systems include space-based interferometers; combat, surveillance, and reconnaissance systems; hazardous material handling; and distributed reconfigurable sensor networks. To enable these applications, various cooperative control capabilities need to be developed, including formation control, rendezvous, attitude alignment, flocking, foraging, task and role assignment, payload transport, air traffic control, and cooperative search.

Cooperative control of multiple autonomous vehicles poses significant theoretical and practical challenges. First, the research objective is to develop a system of subsystems rather than a single system. Second, the communication bandwidth and connectivity of the team are often limited, and the information exchange among vehicles may be unreliable. It is also difficult to decide what to communicate and when and with whom the communication takes place. Third, arbitration between team goals and individual goals needs to be negotiated. Fourth, the computational resources of each individual vehicle will always be limited.

Recent years have seen significant interest and research activity in the area of coordinated and cooperative control of multiple autonomous vehicles (*e.g.*, [11, 15, 22, 24, 25, 28, 37, 49, 51, 62–65, 70, 72, 80, 82, 95, 100–102, 121, 123, 133, 142, 151, 153, 189, 200, 203, 225–228, 234, 244]). Much of this work assumes the availability of global team knowledge, the ability to plan group actions in a centralized manner, and/or perfect and unlimited communication among the vehicles.

A centralized coordination scheme relies on the assumption that each member of the team has the ability to communicate to a central location or share information *via* a fully connected network. As a result, the centralized scheme does not scale well with the number of vehicles. The centralized scheme may result in a catastrophic failure of the overall system due to its single point of failure. Also, real-world communication topologies are usually not fully connected. In many cases, they depend on the relative positions of the vehicles and on other environmental factors and are therefore dynamically changing in time. In addition, wireless communication channels are subject to multipath, fading and drop-out. Therefore, cooperative control in the presence of real-world communication constraints becomes a significant challenge.

To understand the fundamental issues inherent in all cooperative control problems, we offer the following, intuitively appealing, fundamental axiom:

Axiom 1.1 *Shared information is a necessary condition for cooperation.*

Information necessary for cooperation may be shared in various ways. For example, relative position sensors may enable vehicles to construct state information for other vehicles [37], knowledge may be communicated among vehicles using a wireless network [69], or joint knowledge might be preprogrammed into the vehicles before a mission begins [16]. Under this axiom, information exchange becomes a central issue in cooperative control. In the following, we will refer to the information that is necessary for cooperation as *coordination information* or *coordination variable* [139, 193]. Suppose that a particular cooperation strategy has been devised and shown to work if the team has global access to the coordination information. Cooperation will occur if each member in the team has access to consistent, accurate, and complete coordination information. However, in the presence of an unreliable, dynamically changing communication topology and dynamically changing local situational awareness of each vehicle, it is not possible for all of the vehicles to have access to consistent, accurate, or complete coordination information, that is, the vehicles may have different instantiations of the coordination variable. Therefore, distributed algorithms need to be developed to ensure that the team is converging to a consistent view of the coordination information.

As an example, consider a meet-for-dinner problem. In this problem, a group of friends decides to meet for dinner at a particular restaurant but fail to specify a precise time to meet. On the afternoon of the dinner appointment, all of the individuals realize that they are uncertain about the time that the group will meet. A centralized solution to this problem is for the group to have a conference call, to poll all individuals regarding their preferred time for dinner, and to average the answers to arrive at a time when the group will meet. However, this centralized solution requires that a conference line be available and that the time of the conference call be known to the group. Because whatever algorithm was used to convey the time of the conference call to the group could also have been used to convey the time to meet for dinner, the central problem remains.

The coordination variable in this example is the time when the group will meet for dinner. The particular time is not what is important, but rather that each individual in the group has a consistent understanding of that information. A distributed solution to the problem would be for each individual to call, one at a time, a subset of the group. Given his or her current estimate of the meeting time, *i.e.*, his or her instantiation of the coordination variable, the individual might update his or her estimate of the meeting time to be a weighted average of his or her current meeting time and that of the person with whom he or she is conversing. The question is to determine under what conditions this strategy will enable the entire team to converge to a consistent meeting time.

To illustrate the meet-for-dinner example, suppose that there are 10 agents who communicate with exactly one other individual, chosen randomly from the group, for a random length of time. After the communication has expired, the process is repeated. Figure 1.1 shows the evolution of the dinner times with the above mentioned distributed approach, where the initial state is uniformly assigned. Note that the entire team converges to a consistent meeting time under switching communication topologies.

For cooperative control strategies to be effective, a team of vehicles must be able to respond to unanticipated situations or changes in the environment that are sensed as a cooperative task is carried out. For some applications (*e.g.*, cooperative observation on the same phenomenon or target), as the environment changes, the vehicles in the team must agree as to what changes took place. For some other applications (*e.g.*, accurate formation geometry maintenance), the vehicles need to maintain relative states between each other or achieve different group behaviors. A direct consequence of Axiom 1.1 is that cooperation requires that the group of vehicles agrees on the instantiations of the coordination variable or that differences between the instantiations of the coordination variable converge to prespecified values. A critical problem for cooperative control is to determine algorithms so that a team of vehicles can agree on the instantiations of the coordination variable or that differences between the instantiations of the coordination variable converge to prespecified values in the presence of (i) imperfect sensors, (ii) communication dropout, (iii) sparse communication topologies, and (iv) noisy and unreliable communication links.

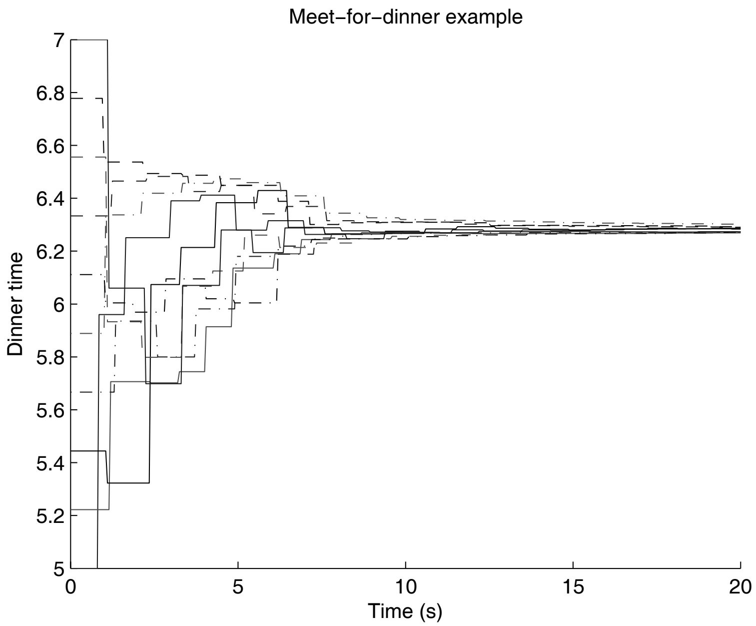

Fig. 1.1. Discrete-time meet-for-dinner simulation

1.2 Literature Review: Consensus Algorithms

When multiple vehicles agree on the value of a variable of interest, they are said to have reached *consensus*. Information consensus guarantees that vehicles sharing information over a network topology have a consistent view of information that is critical to the coordination task. To achieve consensus, there must be a shared variable of interest, called the *information state*, as well as appropriate algorithmic methods for negotiating to reach consensus on the value of that variable, called the *consensus algorithms*. The information state represents an instantiation of the coordination variable for the team. Examples include a local representation of the center and shape of a formation, the rendezvous time, the length of a perimeter being monitored, the direction of motion for a multivehicle swarm, and the probability that a military target has been destroyed. By necessity, consensus algorithms are designed to be distributed, assuming only neighbor-to-neighbor interaction between vehicles. Vehicles update the value of their information states based on the information states of their neighbors. The goal is to design an update law so that the information states of all of the vehicles in the network converge to a common value.

Consensus algorithms have a historical perspective in [32, 41, 55, 81, 131, 236], to name a few, and have recently been studied extensively in the context of cooperative control of multiple autonomous vehicles [69, 97, 126, 145, 158, 190]. Some results in consensus algorithms can be understood in the context of connective stability [215]. Consensus algorithms have applications in rendezvous [26, 58, 124, 125, 135, 216, 220, 221], formation control [69, 115, 118, 127, 134, 165, 174], flocking [50, 59, 120, 147, 155, 169, 232, 238], attitude alignment [117, 176, 179, 188], perimeter monitoring [38], decentralized task assignment [6, 143], and sensor networks [76, 154, 159, 223, 260].

1.2.1 Fundamental Consensus Algorithms

The basic idea of a consensus algorithm is to impose similar dynamics on the information states of each vehicle. If the communication network among vehicles allows continuous communication or if the communication bandwidth is sufficiently large, then the information state update of each vehicle is modeled using a differential equation. On the other hand, if the communication data arrive in discrete packets, then the information state update is modeled using a difference equation. This section overviews fundamental consensus algorithms in which a scalar information state is updated by each vehicle using, respectively, a first-order differential equation and a first-order difference equation.

Suppose that there are n vehicles in the team. The team's communication topology can be represented by directed graph $\mathcal{G}_n \stackrel{\triangle}{=} (\mathcal{V}_n, \mathcal{E}_n)$, where $\mathcal{V}_n = \{1, \ldots, n\}$ is the node set and $\mathcal{E}_n \subseteq \mathcal{V}_n \times \mathcal{V}_n$ is the edge set (see Appendix B for graph theory notations). For example, Fig. 1.2 shows three different communication topologies for three vehicles. The communication topology may be time varying due to vehicle motion or communication dropouts. For example, communication dropouts might occur when an unmanned air vehicle (UAV) banks away from its neighbor or flies through an urban canyon. The most common continuous-time consensus algorithm [69, 97, 126, 158, 190] is given by

$$\dot{x}_i(t) = -\sum_{j=1}^{n} a_{ij}(t)[x_i(t) - x_j(t)], \quad i = 1, \ldots, n, \tag{1.1}$$

where $a_{ij}(t)$ is the (i, j) entry of adjacency matrix $\mathcal{A}_n \in \mathbb{R}^{n\times n}$ associated with $\mathcal{G}_n$ at time t and x_i is the information state of the ith vehicle. Setting $a_{ij} = 0$ denotes the fact that vehicle i cannot receive information from vehicle j. A consequence of (1.1) is that the information state $x_i(t)$ of vehicle i is driven toward the information states of its neighbors. The critical convergence question is, when do the information states of all of the vehicles converge to a common value?

Although (1.1) ensures that the information states of the team agree, it does not dictate a specified common value. For example, consider a cooperative rendezvous problem where a team of vehicles is tasked to arrive simultaneously

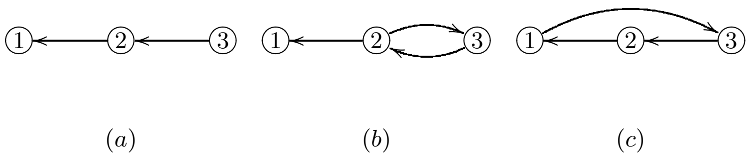

Fig. 1.2. Three different communication topologies for three vehicles. Subplot (c) is strongly connected because there is a directed path between every pair of nodes. However, (a) and (b) are not strongly connected.

at a specified location known to all vehicles. Because the rendezvous time is not given and may need to be adjusted in response to pop-up threats or other environmental disturbances, the team needs to reach consensus on the rendezvous time. To do this, each vehicle first creates an information state x_i that represents the ith vehicle's understanding of the rendezvous time. To initialize its information state, each vehicle determines a time at which it is able to rendezvous with the team and sets $x_i(0)$ to this value. Each team member then communicates with its neighbors and negotiates a team arrival time using consensus algorithm (1.1). Onboard controllers then maneuver each vehicle to rendezvous at the negotiated arrival time. When environmental conditions change, individual vehicles may reset their information state and thus cause the negotiation process to resume.

Note that (1.1) does not permit specifying a desired information state. We show in Section 1.2.2 that if the communication topology is fixed and the gains a_{ij} are time invariant, then the common asymptotic value is a linear combination of the initial information states. In general, it is possible to guarantee only that the common value is a convex combination of the initial information states.

Consensus algorithm (1.1) is written in matrix form as

$$\dot{x}(t) = -\mathcal{L}_n(t)x(t),$$

where $x = [x_1, \dots, x_n]^T$ is the information state and $\mathcal{L}_n(t) = [\ell_{ij}(t)] \in \mathbb{R}^{n\times n}$ is the nonsymmetrical Laplacian matrix associated with $\mathcal{G}_n$ (see Appendix B). Consensus is *achieved* or *reached* by a team of vehicles if, for all $x_i(0)$ and all $i, j = 1, \dots, n$, $|x_i(t) - x_j(t)| \to 0$, as $t \to \infty$.

When communication among vehicles occurs at discrete instants, the information state is updated using a difference equation. The most common discrete-time consensus algorithm has the form [97, 145, 190, 236]

$$x_i[k+1] = \sum_{j=1}^{n} d_{ij}[k]x_j[k], \quad i = 1, \dots, n, \tag{1.2}$$

where k denotes a communication event; $d_{ij}[k]$ is the (i, j) entry of a row-stochastic matrix $\mathcal{D} = [d_{ij}] \in \mathbb{R}^{n\times n}$ (see Appendix C for matrix theory notations) at the discrete-time index k with the additional assumption that

$d_{ii}[k] > 0$ for all $i = 1, \dots, n$ and $d_{ij}[k] > 0$ for all $i \neq j$ if information flows from vehicle j to vehicle i and $d_{ij}[k] = 0$ otherwise. Intuitively, the information state of each vehicle is updated as the weighted average of its current state and the current states of its neighbors. Note that a vehicle maintains its current information state if it does not exchange information with other vehicles at that instant. Discrete-time consensus algorithm (1.2) is written in matrix form as

$$x[k+1] = \mathcal{D}[k]x[k].$$

Similar to the continuous-time case, consensus is *achieved* or *reached* if, for all $x_i[0]$ and for all $i, j = 1, \dots, n$, $|x_i[k] - x_j[k]| \to 0$, as $k \to \infty$.

1.2.2 Convergence Analysis of Consensus Algorithms

Convergence Analysis for Time-invariant Communication Topologies

In this section, we investigate conditions under which the information states of consensus algorithm (1.1) converge when the communication topology is time invariant and the gains a_{ij} are constant, that is, the nonsymmetrical Laplacian matrix $\mathcal{L}_n$ is constant. As noted in Appendix B, zero is always an eigenvalue of $-\mathcal{L}_n$, and all nonzero eigenvalues of $-\mathcal{L}_n$ have negative real parts. As also noted in Appendix B, the column vector $\mathbf{1}_n$ of ones is an eigenvector associated with the zero eigenvalue, which implies that span$\{\mathbf{1}_n\}$ is contained in the kernel of $\mathcal{L}_n$. It follows that if zero is a simple eigenvalue of $\mathcal{L}_n$, then $x(t) \to \bar{x}\mathbf{1}_n$, where $\bar{x}$ is a scalar constant, which implies that $|x_i(t) - x_j(t)| \to 0$, as $t \to \infty$, for all $i, j = 1, \dots, n$. Convergence analysis therefore focuses on conditions to ensure that zero is a simple eigenvalue of $\mathcal{L}_n$. Otherwise the kernel of $\mathcal{L}_n$ includes elements that are not in span$\{\mathbf{1}_n\}$, in which case consensus is not guaranteed.

If the directed graph of $\mathcal{L}_n$ is strongly connected (see Appendix B), then zero is a simple eigenvalue of $\mathcal{L}_n$ [69, Proposition 3]. However, this condition is not necessary. For example, consider the nonsymmetrical Laplacian matrices

$$\mathcal{L}_{3(1)} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1.5 & -1.5 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathcal{L}_{3(2)} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1.5 & -1.5 \\ 0 & -2 & 2 \end{bmatrix}, \quad \mathcal{L}_{3(3)} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1.5 & -1.5 \\ -2 & 0 & 2 \end{bmatrix}, \tag{1.3}$$

of the directed graphs shown in Fig. 1.2. Although all of the nonsymmetrical Laplacian matrices in (1.3) have simple zero eigenvalues, Figs. 1.2a and 1.2b are not strongly connected. The common feature is that the directed graphs of $\mathcal{L}_{3(1)}$, $\mathcal{L}_{3(2)}$, and $\mathcal{L}_{3(3)}$ all have directed spanning trees. As shown in [115, 127, 193, 255], zero is a simple eigenvalue of $\mathcal{L}_n$ if and only if the associated directed graph has a directed spanning tree. This result implies that (1.1) achieves consensus if and only if the directed communication topology has a directed spanning tree or the undirected communication topology is connected.

For discrete-time consensus algorithm (1.2), Theorem C.1 implies that all eigenvalues of $\mathcal{D}$ are either in the open unit disk or at 1. As shown in [190], if 1 is a simple eigenvalue of $\mathcal{D}$, then $\lim_{k\to\infty}\mathcal{D}^k \to \mathbf{1}_n\nu^T$, as $k\to\infty$, where ν is an $n\times 1$ nonnegative left eigenvector of $\mathcal{D}$ associated with the eigenvalue 1 and satisfies $\nu^T\mathbf{1}_n = 1$. As a result, $x[k] = \mathcal{D}^k x[0] \to \mathbf{1}_n\nu^T x[0]$, as $k\to\infty$, which implies that, for all i, $x_i[k]\to\nu^T x[0]$, as $k\to\infty$, and thus $|x_i[k]-x_j[k]|\to 0$, as $k\to\infty$.

Theorem C.5 implies that $1=\rho(A)$ is a simple eigenvalue of row-stochastic matrix A if directed graph $\Gamma(A)$ is strongly connected, or equivalently, if A is irreducible. As in the continuous-time case, this condition is sufficient but not necessary. Furthermore, for row-stochastic matrix $\mathcal{D}$, $\Gamma(\mathcal{D})$ has a directed spanning tree if and only if $\lambda=1$ is a simple eigenvalue of $\mathcal{D}$ and is the only eigenvalue of modulus one [190]. As a result, under a time-invariant communication topology with constant gains a_{ij}, (1.2) achieves consensus if and only if either the directed communication topology has a directed spanning tree or the undirected communication topology is connected [190].

Equilibrium State under a Time-invariant Communication Topology

We now investigate the consensus equilibrium for the special case in which the communication topology is time invariant and the gains a_{ij} are constant (*i.e.*, constant $\mathcal{L}_n$). When the directed communication topology has a directed spanning tree, it follows from [193] that $\lim_{t\to\infty} e^{-\mathcal{L}_n t}\to\mathbf{1}_n\nu^T$, where $\nu=[\nu_1,\ldots,\nu_n]^T$ is an $n\times 1$ nonnegative left eigenvector of $\mathcal{L}_n$ associated with the simple zero eigenvalue and satisfies $\sum_{j=1}^n \nu_j = 1$. As a result, for each $i=1,\ldots,n$, $x_i(t)\to\sum_{j=1}^n \nu_j x_j(0)$, as $t\to\infty$, that is, the equilibrium state is a weighted average of the initial information states in the network. However, some of the components of ν may be zero, implying that the information states of some vehicles do not contribute to the equilibrium.

To illustrate this phenomenon, consider the nonsymmetrical Laplacian matrices given in (1.3). It can be verified that, for $\mathcal{L}_{3(1)}$, $x(t)\to x_3(0)\mathbf{1}_3$, for $\mathcal{L}_{3(2)}$, $x(t)\to[0.5714x_2(0)+0.4286x_3(0)]\mathbf{1}_3$, and, for $\mathcal{L}_{3(3)}$, $x(t)\to[0.4615x_1(0)+0.3077x_2(0)+0.2308x_3(0)]\mathbf{1}_3$. Note that with $\mathcal{L}_{3(1)}$, the initial information states of vehicles 1 and 2 do not affect the equilibrium. With $\mathcal{L}_{3(2)}$, the initial information state of vehicle 1 does not affect the equilibrium. However, with $\mathcal{L}_{3(3)}$, all of the vehicle's initial information states affect the equilibrium. Observing the directed graphs shown in Fig. 1.2, we can see that, for $\mathcal{L}_{3(1)}$, vehicle 3 is the only vehicle that can pass information to all of the other vehicles in the team, either directly or indirectly. Similarly, for $\mathcal{L}_{3(2)}$, both vehicles 2 and 3 can pass information to the entire team, whereas, for $\mathcal{L}_{3(3)}$, all vehicles can pass information to the entire team.

Next, define the nonnegative matrix $M=\max_i \ell_{ii}I_n-\mathcal{L}_n$. Because ν is the nonnegative left eigenvector of $\mathcal{L}_n$ corresponding to the zero eigenvalue, ν is also the nonnegative left eigenvector of M corresponding to the eigenvalue

$\max_i \ell_{ii}$ of M. From Theorem C.1, it follows that $\rho(M) = \max_i \ell_{ii}$. If the directed communication graph is strongly connected, so is the directed graph of M, which also implies that M is irreducible (see Appendix C). By Theorem C.5, if M is irreducible, then ν is positive. Therefore, when the directed communication topology is strongly connected, all of the initial information states contribute to the consensus equilibrium because $\nu_i \neq 0$ for all i. Furthermore, if $\nu_i = 1/n$ for all i, then the consensus equilibrium is the average of the initial information states, a condition called *average consensus* [158]. If the directed communication topology is both strongly connected and balanced, then $\mathbf{1}_n$ is a left eigenvector of $\mathcal{L}_n$ associated with the simple zero eigenvalue. Therefore, as shown in [158], average consensus is achieved if and only if the directed communication topology is both strongly connected and balanced. It can be shown that, in the case of undirected communication, average consensus is achieved if and only if the topology is connected [158].

To illustrate these ideas, Fig. 1.3 shows time histories of the information states for two different updates strategies. Figure 1.3a shows the information states for $\dot{x} = -\mathcal{L}_{3(3)}x$, where $\mathcal{L}_{3(3)}$ is given in (1.3). Because the directed graph of $\mathcal{L}_{3(3)}$ is strongly connected, all of the vehicle's initial conditions contribute to the equilibrium state. However, the equilibrium is not an average consensus because the directed graph is not balanced. In contrast, Fig. 1.3b shows the time histories of the information states for $\dot{x} = -\text{diag}(w)\mathcal{L}_{3(3)}x$, where w is the positive column left eigenvector of $\mathcal{L}_{3(3)}$ corresponding to the zero eigenvalue satisfying $w^T\mathbf{1}_3 = 1$ and $\text{diag}(w)$ is the diagonal matrix whose diagonal entries are given by w. It can be shown that directed graph $\Gamma\left[\text{diag}(w)\mathcal{L}_{3(3)}\right]$ is strongly connected and balanced, resulting in average consensus.

In contrast, when the directed communication topology has a directed spanning tree, the consensus equilibrium is equal to the weighted average of the initial conditions of those vehicles that have a directed path to all other vehicles [193]. Requiring a directed spanning tree is less stringent than requiring a strongly connected and balanced graph. However, as shown above, the consensus equilibrium is a function only of the initial information states of those vehicles that have a directed path to all other vehicles.

Convergence Analysis for Dynamic Communication Topologies

Communication topologies are often dynamic. For example, communication links among vehicles might be unreliable due to multipath effects and other disturbances. Alternatively, if information is exchanged by means of line-of-sight sensors, the neighbors visible to a vehicle might change over time, *e.g.*, when a UAV banks away from its neighbor. Therefore, in this section, we investigate conditions under which consensus algorithms converge under random switching of the communication topologies.

One approach to analyzing switching topologies is to use algebraic graph theory, which associates each graph topology with an algebraic structure of

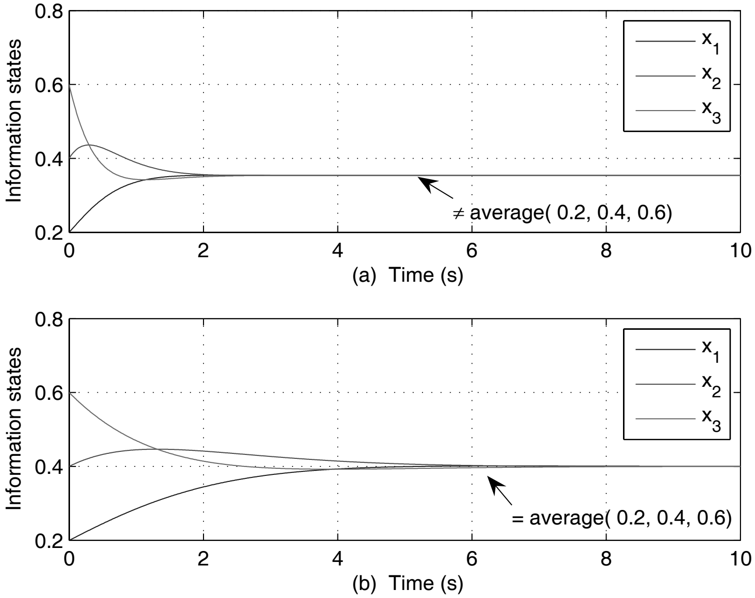

Fig. 1.3. Consensus for three vehicles. Subplots (a) and (b) correspond to $\dot{x} = -\mathcal{L}_{3(3)}x$ and $\dot{x} = -\mathrm{diag}(w)\mathcal{L}_{3(3)}x$, respectively. Because 0.4 is the average of the initial states $(0.2, 0.4, 0.6)$, average consensus is achieved in (b), where the directed graph is strongly connected and balanced, but not in (a), where the directed graph is only strongly connected.

corresponding matrices. Because (1.1) is linear, its solution can be written as $x(t) = \Phi(t,0)x(0)$, where $\Phi(t,0)$ is the transition matrix corresponding to $-\mathcal{L}_n(t)$. $\Phi(t,0)$ is a row-stochastic matrix with positive diagonal entries for all $t \geq 0$ [192]. Consensus is achieved if $\lim_{t\to\infty} \Phi(t,0) \to \mathbf{1}_n\mu^T$, where μ is a column vector. It is typical to assume that the communication topology is piecewise constant over finite lengths of time, called *dwell times*, and that dwell times are bounded below by a positive constant [97]. In this case, $\mathcal{A}_n(t)$ and hence $\mathcal{L}_n(t)$ are piecewise constant with dwell times $\tau_j = t_{j+1} - t_j$, where $t_1, t_2, \ldots$ are the switching instants, and thus consensus is achieved if $\lim_{j\to\infty} e^{-\mathcal{L}_n(t_j)\tau_j} e^{-\mathcal{L}_n(t_{j-1})\tau_{j-1}} \cdots e^{-\mathcal{L}_n(t_0)\tau_0} = \mathbf{1}_n\mu^T$. Because $e^{-\mathcal{L}_n(t_j)(t-t_j)}$ is a row-stochastic matrix, convergence analysis involves the study of infinite products of stochastic matrices.

A classical result given in [253] (see also [192]) demonstrates the convergence property of infinite products of SIA matrices (see Appendix C). Specifically, let $\mathcal{S} = \{S_1, S_2, \ldots, S_k\}$ be a finite set of SIA matrices with the property that every finite product $S_{i_j}S_{i_{j-1}}\cdots S_{i_1}$ is SIA. Then, for each infinite sequence $S_{i_1}, S_{i_2}, \ldots$ there exists a column vector ν such that

$\lim_{j\to\infty} S_{i_j} S_{i_{j-1}} \cdots S_{i_1} = \mathbf{1}\nu^T$. Because the number of potential communication topologies is finite, the set of matrices $\{S_j \triangleq e^{-\mathcal{L}_n(t_j)(t_{j+1}-t_j)}\}_{j=1}^{\infty}$ is finite if the allowable dwell times $\tau_j = t_{j+1} - t_j$ are drawn from a finite set. Reference [97] shows that these matrices are SIA and uses this result to show that the heading angles of a swarm of vehicles achieve consensus using the nearest neighbor rules of [239]. This is a special case of discrete-time consensus algorithm (1.2), if there exists an infinite sequence of contiguous, uniformly bounded time intervals, having one of a finite number of different lengths, with the property that across each interval, the union (see Appendix B) of the undirected communication topologies is connected. See [78, 126, 158, 208] for extensions.

Consider, on the other hand, the more realistic assumption that the dwell times are drawn from an infinite but bounded set or $\mathcal{A}_n(t)$ is piecewise continuous[1] and its nonzero and hence positive entries are uniformly lower and upper bounded. In this case, let $\mathcal{S} = \{S_1, S_2, \dots\}$ be an infinite set of $n \times n$ SIA matrices, let N_t be the number of different types (see Appendix B) of all of the $n \times n$ SIA matrices, and define the matrix function $\chi(P) = 1 - \min_{i_1,i_2} \sum_j \min(p_{i_1 j}, p_{i_2 j})$. Then, $\lim_{j\to\infty} S_{i_j} S_{i_{j-1}} \cdots S_{i_1} = \mathbf{1}\nu^T$ if there exists a constant $d \in [0, 1)$ such that, for every $W \triangleq S_{k_1} S_{k_2} \cdots S_{k_{N_t+1}}$, it follows that $\chi(W) \leq d$ [253]. It can be shown that this condition is satisfied if there exists an infinite sequence of contiguous, uniformly bounded time intervals, with the property that across each interval, the union of the directed communication topologies has a directed spanning tree [190, 192]. Reference [168] considers a similar problem by studying the products of row-stochastic matrices with a lower triangular structure. In addition, a lower bound on the convergence rate of consensus algorithm (1.2) under directed switching communication topologies is derived in [36].

Lyapunov Analysis of Consensus Algorithms

Nonlinear analysis can also be used to study consensus algorithms [71, 128, 145]. For discrete-time consensus algorithm (1.2), a set-valued Lyapunov function V is defined as $V(x_1, \dots, x_n) = (\mathrm{conv}\{x_1, \dots, x_n\})^n$, where $\mathrm{conv}\{x_1, \dots, x_n\}$ denotes the convex hull of $\{x_1, \dots, x_n\}$, and $X^n \triangleq X \times \cdots \times X$. It is shown in [145] that $V(t_2) \subseteq V(t_1)$ for all $t_2 \geq t_1$, and that $x(t)$ approaches an element of the set $\mathrm{span}\{\mathbf{1}_n\}$, which implies that consensus is reached. Using set-valued Lyapunov theory, [145] shows that discrete-time consensus algorithm (1.2) is uniformly globally attractive with respect to the collection of equilibrium solutions $\mathrm{span}\{\mathbf{1}_n\}$ if and only if there exists $K \geq 0$ such that the union of the directed communication topologies has a directed spanning tree across each interval of length Kh, where h is the sample time.

[1] Accordingly, $\mathcal{L}_n(t)$ is piecewise continuous.

For continuous-time consensus algorithm (1.1), [144] considers the Lyapunov candidate $V(x) = \max\{x_1, \ldots, x_n\} - \min\{x_1, \ldots, x_n\}$. It is shown in [144] that the equilibrium set $\mathrm{span}\{\mathbf{1}_n\}$ is uniformly exponentially stable if there is an interval length $T > 0$ such that, for all t, the directed graph of $-\int_t^{t+T} \mathcal{L}_n(s)ds$ has a directed spanning tree.

As an alternative analytic method, [219,246,247] applies nonlinear contraction theory to synchronization and schooling applications, which are related to information consensus. In particular, (1.1) is analyzed under undirected switching communication topologies, and a convergence result identical to the result given in [97] is derived. In addition, [13] uses passivity as a design tool for consensus algorithms over an undirected communication topology.

Information consensus is also studied from a stochastic point of view in [87,88,256], which consider a random network, where the existence of an information channel between a pair of vehicles at each time is probabilistic and independent of other channels, resulting in a time-varying undirected communication topology. For example, adjacency matrix $\mathcal{A}_n = [a_{ij}] \in \mathbb{R}^{n\times n}$ for an undirected random graph is defined as $a_{ii}(p) = 0$, $a_{ij}(p) = 1$ with probability p, and $a_{ij} = 0$ with probability $1 - p$ for all $i \neq j$. In [88], consensus over an undirected random network is addressed by notions from stochastic stability.

Communication Delays and Asynchronous Consensus

When information is exchanged among vehicles through communication, time delays from both message transmission and processing after receipt must be considered. Let σ_{ij} denote the time delay for information communicated from vehicle j to reach vehicle i. In this case, (1.1) is modified as

$$\dot{x}_i = -\sum_{j=1}^{n} a_{ij}(t)[x_i(t - \sigma_{ij}) - x_j(t - \sigma_{ij})].$$

In the simplest case, where $\sigma_{ij} = \sigma$ and the communication topology is time-invariant, undirected, and connected, average consensus is achieved if and only if $0 \leq \sigma < \frac{\pi}{2\lambda_{max}(\mathcal{L}_n)}$ [158], where $\mathcal{L}_n$ is the Laplacian matrix of the undirected communication topology and $\lambda_{max}(\cdot)$ denotes the largest eigenvalue of a matrix. See [30, 119] for extensions.

Alternatively, consider the case in which the time delay affects only the information state that is being transmitted so that (1.1) is modified as

$$\dot{x}_i = -\sum_{j=1}^{n} a_{ij}(t)[x_i(t) - x_j(t - \sigma_{ij})].$$

When $\sigma_{ij} = \sigma$ and the communication topology is directed and switching, the consensus result for switching topologies remains valid for an arbitrary time delay σ [144].

For discrete-time consensus algorithm (1.2), it is shown in [230] that if consensus is reached under a time-invariant undirected communication topology, then the presence of communication delays does not affect consensus. In addition, the result in [145] is extended to take into account bounded time delays in [12]. Furthermore, [258] shows sufficient conditions for consensus under dynamically changing communication topologies and bounded time-varying communication delays.

More generally, in an asynchronous consensus framework [31,35,67,68,140], each vehicle exchanges information asynchronously and updates its state with possibly outdated information from its local neighbors. As a result, heterogenous vehicles, time-varying communication delays, and packet dropout must be taken into account in the same asynchronous consensus framework. Reference [68] categorizes several consensus results in the literature according to synchronism, connectivity, and direction of information flow.

1.2.3 Synthesis and Extensions of Consensus Algorithms

Consensus Synthesis

In some applications, consensus algorithms must satisfy given requirements or optimize performance criteria. For example, when a UAV or micro air vehicle (MAV) swarm consists of hundreds or thousands of vehicles, it might be desirable to solve the fastest distributed linear averaging (FDLA) problem, which is defined as follows [259]. Let $W = [w_{ij}] \in \mathbb{R}^{n\times n}$ be such that $w_{ij} = 0$ if information is not exchanged between vehicle i and vehicle j. Given $x[k+1] = Wx[k]$, find W to minimize

$$r_{asym}(W) = \sup_{x[0]\neq \bar{x}} \lim_{k\to\infty} \left(\frac{\|x[k]-\bar{x}\|}{\|x[0]-\bar{x}\|} \right)^{1/k},$$

subject to the condition that $\lim_{t\to\infty} W^k = \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$, where $\bar{x} = \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T x[0]$. In other words, the FDLA problem is to find the weight matrix W that guarantees the fastest convergence to the average consensus value. In contrast to discrete-time consensus algorithm (1.2), weights w_{ij} can be negative [259]. With the additional constraint $w_{ij} = w_{ji}$, the FDLA problem reduces to a numerically solvable semidefinite program [259]. A related problem is considered in [106], where an iterative, semidefinite-programming-based approach is developed to maximize the algebraic connectivity of the Laplacian matrix of undirected graphs (see Appendix B) with the motivation that the algebraic connectivity of the Laplacian matrix characterizes the convergence rate of the consensus algorithm.

Another problem is considered in [202], which focuses on designing consensus algorithms in which the information state is updated according to $\dot{x}_i = u_i$, and the information available to the ith vehicle is given by $y_i = G_i x$, where $x = [x_1, \ldots, x_n]^T$, $y_i \in \mathbb{R}^{m_i}$, and $G_i \in \mathbb{R}^{m_i\times n}$. The information control input is designed in the form of $u_i = k_i y_i + z_i$, where k_i is a row vector with m_i components and z_i is a scalar.

More generally, consider an interconnected network of n vehicles whose information states are updated according to $\dot{x}_i = \sum_{j=1}^{n} A_{ij} x_j + B_{1i} w_i + B_{2i} u_i$, $i = 1, \ldots, n$, where $x_i \in \mathbb{R}^n$ denotes the information state, $w_i \in \mathbb{R}^m$ denotes disturbances, and $u_i \in \mathbb{R}^r$ denotes the information control input with $i = 1, \ldots, n$. Letting x, w, and u be column vectors with components x_i, w_i, and u_i, respectively, the dynamics of x are denoted by $\dot{x} = Ax + B_1 w + B_2 u$. Reference [52] focuses on synthesizing a decentralized state feedback control law that guarantees consensus for the closed-loop system without disturbances as well as synthesizing a state-feedback controller that achieves not only consensus but optimal $\mathcal{H}_2$ performance for disturbance attenuation.

Extensions of Consensus Algorithms

Consensus algorithm (1.1) is extended in various ways in the literature. For example, [17, 47] generalize the consensus equilibrium to a weighted power mean or arbitrary functions of the initial information states. In [104], quantized consensus problems are studied, where the information state at each node is an integer. In [224], an external input is incorporated in (1.1) so that the information state tracks a time-varying input. Consensus with a constant reference state is addressed in [99, 143], and consensus is addressed with a time-varying reference state in [90,180]. In [229], necessary and sufficient conditions are derived so that a collection of systems is controlled by a team leader. An approach based on nonsmooth gradient flows is developed in [48] to guarantee that average consensus is reached in finite time.

The single-integrator consensus algorithm given by (1.1) is also extended to double-integrator dynamics in [89, 186, 261] to model more naturally the evolution of physical phenomena, such as a coaxial rotorcraft MAV that can be controlled through gentle maneuvers with a decoupled double-integrator model. For double-integrator dynamics, the consensus algorithm is given by

$$\ddot{x}_i = -\sum_{j=1}^{n} a_{ij}(t)[(x_i - x_j) + \gamma(\dot{x}_i - \dot{x}_j)],$$

where $\gamma > 0$ denotes the coupling strength between the information state derivatives and both x_i and $\dot{x}_i$ are transmitted between team members. It is shown in [186] that both the communication topology and coupling strength γ affect consensus-seeking in the general case of directed information exchange. To achieve consensus, the directed communication topology must have a directed spanning tree and γ must be sufficiently large. See [196] for extensions to higher-order dynamics.

Related to consensus algorithms are synchronization phenomena arising in systems of coupled nonlinear oscillators. The classical Kuramoto model [114] consists of n coupled oscillators with dynamics given by

$$\dot{\theta}_i = \omega_i + \frac{k}{n} \sum_{j=1}^{n} \sin(\theta_j - \theta_i), \tag{1.4}$$

where θ_i and ω_i are, respectively, the phase and natural frequency of the ith oscillator; and k is the coupling strength. Note that model (1.4) assumes full connectivity of the network. Model (1.4) is generalized in [98] to nearest neighbor information exchange as

$$\dot{\theta}_i = \omega_i + \frac{k}{n} \sum_{j=1}^{n} a_{ij}(t) \sin(\theta_j - \theta_i).$$

Connections between phase models of coupled oscillators and kinematic models of self-propelled particle groups are studied in [213]. Analysis and design tools are developed to stabilize the collective motions. The stability of the generalized Kuramoto coupled nonlinear oscillator model is studied in [98], where it is proven that, for couplings above a critical value, all oscillators synchronize given identical and uncertain natural frequencies. Extensions of [98] to a tighter lower bound on the coupling strength are given in [46] for the traditional Kuramoto model with full connectivity. The result in [98] is also extended to account for heterogenous time delays and switching topologies in [162].

Synchronization of coupled oscillators with other nonlinear dynamics is also studied in the literature. As an example, consider a network of n vehicles with information dynamics given by

$$\dot{x}_i = f(x_i, t) + \sum_{j=1}^{n} a_{ij}(t)(x_j - x_i), \tag{1.5}$$

where $x = [x_1, \ldots, x_n]^T$. In [219], partial contraction theory is applied to derive conditions under which consensus is reached for vehicles with dynamics (1.5). As another example, [166] studies a dynamic network of n nonlinear oscillators, where the state equation for each oscillator is given by

$$\dot{x}_i = f(x_i) + \gamma \sum_{j=1}^{n} a_{ij}(t)(x_j - x_i),$$

where $x_i \in \mathbb{R}^m$ and $\gamma > 0$ denotes the global coupling strength parameter. It is shown in [166] that the algebraic connectivity of the network Laplacian matrix plays a central role in synchronization.

1.2.4 Design of Coordination Strategies *via* Consensus Algorithms

In this section, we briefly describe a few applications of consensus algorithms to multivehicle coordination problems.

Rendezvous Problem

The rendezvous problem requires that a group of vehicles in a network rendezvous at a time or a location determined through team negotiation. Consensus algorithms can be used to perform the negotiation in a way that is robust to environmental disturbances such as nonuniform wind for a team of UAVs. The rendezvous problem for a group of mobile autonomous vehicles is studied in [124, 125], where synchronous and asynchronous cases are considered. In [124, 125], vehicles execute a sequence of stop-and-go maneuvers to rendezvous in a distributed manner without communication between neighbors. A stop-and-go maneuver takes place within a time interval consisting of a sensing period during which neighbors' positions are determined, as well as a maneuvering period during which vehicles move in response to the positions of their neighbors.

Figure 1.4 shows a simple coordination framework for multivehicle rendezvous, where a consensus manager applies distributed consensus algorithms to guarantee that all vehicles reach consensus on a rendezvous objective such as a rendezvous time or rendezvous location. Based on the output of the consensus manager, each vehicle uses a local control law to drive itself to achieve the rendezvous time and/or location. An application of Fig. 1.4 is described in [110], where multiple UAVs are controlled to converge simultaneously on the boundary of a radar detection area to maximize the element of surprise. Teamwide consensus is reached on time-over-target, requiring each vehicle to adjust its velocity to ensure synchronous arrival.

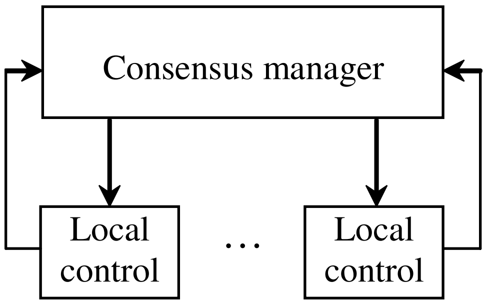

Fig. 1.4. A simple coordination framework for multivehicle rendezvous. The consensus manager applies distributed consensus algorithms to guarantee that the team reaches consensus on a rendezvous objective. Based on the output of the consensus manager, each vehicle applies local control laws to achieve the rendezvous objective.

Formation Stabilization

The formation stabilization problem requires that vehicles collectively maintain a prescribed geometric shape. This problem is relatively straightforward in the centralized case, where all team members know the desired shape and location of the formation. On the other hand, in the decentralized formation stabilization problem, each vehicle knows the desired formation shape, but the location of the formation needs to be negotiated among team members. The information state for this problem includes the center of the formation. Each vehicle initializes its information state by proposing a formation center that does not require it to maneuver into formation. The consensus algorithm is then employed by the team of vehicles to negotiate a formation center known to all members of the team.

In [69], an information flow filter is used to improve stability margins and formation accuracy through propagation of the formation center to all vehicles. Formation stabilization for multiple unicycles is studied in [127] using a consensus algorithm to achieve point, line, and general formation patterns. In addition, the simplified pursuit strategy for wheeled-vehicle formations in [133] can be considered a special case of continuous-time consensus algorithm (1.1), where the communication topology is a unidirectional ring. Furthermore, feedback control laws are derived in [115] using relative information between neighboring vehicles to stabilize vehicle formations.

Formation Maneuvering and Flocking

Consensus algorithms can be applied to execute decentralized formation maneuvers. For example, in [118], a class of formation maneuvers is studied where the desired position of each robot, $h_i^d(t)$, is either communicated to the team by a centralized entity or is preprogrammed on each robot. The robots are to maintain a prespecified formation shape even during transients and in response to environmental disturbances. In other words, when one robot slows down or maneuvers to avoid an obstacle, the other robots must maneuver to maintain the formation shape. The intervehicle communication network is limited and requires a decentralized approach to maintain the formation. The mobile robot dynamic model is feedback linearized as the double-integrator system $\ddot{h}_i = u_i$, where h_i denotes the location of a point on the ith robot that is not on the wheel axis and u_i denotes the control input. The decentralized formation control law is given in [118] as

$$u_i = -K_g\tilde{h}_i - D_g\dot{h}_i - K_f\sum_{j=1}^{n} a_{ij}(\tilde{h}_i - \tilde{h}_j) - D_f\sum_{j=1}^{n} a_{ij}(\dot{h}_i - \dot{h}_j), \tag{1.6}$$

where K_g and K_f are symmetrical positive-definite matrices, D_g and D_f are symmetrical positive-semidefinite matrices, and $\tilde{h}_i \stackrel{\triangle}{=} h_i - h_i^d$. In control law (1.6), the first two terms guarantee that h_i approaches h_i^d, whereas the second two terms guarantee that the pairs $\tilde{h}_i$, $\tilde{h}_j$ and $\dot{h}_i$, $\dot{h}_j$ reach consensus. If consensus can be reached for each $\tilde{h}_j$, the preservation of the desired formation shape is guaranteed during maneuvers.

A similar approach can be applied to the rigid body attitude dynamics

$$\begin{aligned} \dot{\hat{q}}_i &= -\frac{1}{2}\omega_i \times \hat{q}_i + \frac{1}{2}\bar{q}_i\omega_i, \quad \dot{\bar{q}}_i = -\frac{1}{2}\omega_i \cdot \hat{q}_i, \\ J_i\dot{\omega}_i &= -\omega_i \times (J_i\omega_i) + T_i, \end{aligned}$$

where, for the ith rigid body, $\hat{q}_i \in \mathbb{R}^3$, $\bar{q}_i \in \mathbb{R}$, and $q_i = [\hat{q}_i^T, \bar{q}_i]^T \in \mathbb{R}^4$ is the unit quaternion, that is, the Euler parameters (see Appendix D), $\omega_i \in \mathbb{R}^3$ is the angular velocity, and $J_i \in \mathbb{R}^{3\times 3}$ and $T_i \in \mathbb{R}^3$ are, respectively, the inertia tensor and the control torque. Defining $\mathrm{vec}\left([\hat{q}, \bar{q}]^T\right) = \hat{q}$ as the operator that extracts the vector part of a quaternion, the control torque is given by [117, 176, 188]

$$T_i = -k_G \mathrm{vec}\left(q_i^{d*} q_i\right) - D_G\omega_i - k_S \sum_{j=1}^{n} a_{ij} \mathrm{vec}\left(q_j^* q_i\right) - D_S \sum_{j=1}^{n} a_{ij}(\omega_i - \omega_j), \quad (1.7)$$

where $k_G > 0$ and $k_S \geq 0$ are scalars, D_G is a symmetrical positive-definite matrix, D_S is a symmetrical positive-semidefinite matrix, q^* is the quaternion conjugate, and q^d is the centrally commanded quaternion. The first two terms in (1.7) align the rigid body with the prespecified desired attitude q_i^d. The second two terms in (1.7) are consensus terms that cause the team to maintain attitude alignment during the transients and in response to environmental disturbances [176].

Using biologically observed motions of flocks of birds, [198] defines three rules of flocking and applies them to generate realistic computer animations. The three rules of flocking are collision avoidance, velocity matching, and flock centering. Together these rules maintain the flock in close proximity without collision. Reference [198] motivates the use of similar rules for multivehicle robotic systems [155, 232]. As an example, consider the vehicle dynamics

$$\dot{r}_i = v_i, \qquad \dot{v}_i = u_i,$$

where r_i and v_i are the position and velocity of vehicle i, respectively, and u_i denotes its input. In [155], the control input u_i is defined as

$$u_i = -\frac{\partial V(r)}{\partial r_i} + \sum_{j=1}^{n} a_{ij}(r)(v_j - v_i) + f_i^{\gamma}, \quad (1.8)$$

where the first term is the gradient of a collective potential function $V(r)$, the second term drives the system toward velocity consensus, and the third term incorporates navigational feedback. In (1.8), the first term guarantees flock centering and collision avoidance among the vehicles, the second term guarantees velocity matching among the vehicles, and the third term achieves a group objective. Equation (1.8) has been validated for flocking with undirected communication topologies.

1.3 Monograph Overview

The subject of this monograph is distributed coordination of multiple autonomous vehicles. The objective of distributed coordination is to have multiple autonomous vehicles work together efficiently to achieve collective group behavior *via* local interaction. This monograph introduces distributed consensus algorithms and their applications in cooperative control of multiple autonomous vehicles. The consensus algorithms require only neighbor-to-neighbor information exchange, which minimizes the power consumption, increases the stealth, and improves the scalability and robustness of the team. In addition, the consensus algorithms allow the interaction topologies among the vehicles to be dynamically changing, sparse, or intermittent. This feature is particularly useful for real-world application scenarios where communication topologies are usually not fully connected, communication links are often noisy and unreliable, and vehicles have only limited communication range and bandwidth. This monograph includes both theoretical and experimental results in distributed coordination of multiple ground robots, spacecraft, and UAVs. The theoretical results address distributed consensus algorithms and their extensions for single-integrator, double-integrator, and rigid body attitude dynamics and show convergence analysis results in the presence of directed, limited, and unreliable information exchange among vehicles. Those results extend many existing results in the area of cooperative control. In the application chapters of the book, we apply the distributed consensus algorithms to several multivehicle cooperative control applications, including formation keeping for wheeled mobile robots and spacecraft and cooperative perimeter tracking and timing for a team of UAVs. The application results demonstrate issues and challenges in multivehicle cooperative control.

This monograph consists of six parts and six appendices. The first part overviews consensus algorithms in cooperative control of multiple autonomous vehicles (Chapter 1). The second part introduces consensus algorithms for single-integrator dynamics (Chapter 2) and consensus algorithms with a reference state (Chapter 3). The third part introduces consensus algorithms for double-integrator dynamics (Chapter 4) and their extensions to a reference model (Chapter 5). The fourth part focuses on attitude consensus for rigid body attitude dynamics (Chapter 6) and relative attitude maintenance and reference attitude tracking (Chapter 7). The fifth part introduces consensus-based design methodologies for distributed multivehicle cooperative control (Chapter 8). The sixth part applies consensus algorithms to several multivehicle cooperative control applications: rendezvous and axial alignment with multiple wheeled mobile robots (Chapter 9), distributed formation control of multiple wheeled mobile robots with a virtual leader (Chapter 10), a decentralized behavioral approach to multiple robot formation maneuvers (Chapter 11), deep space spacecraft formation flying (Chapter 12), cooperative fire monitoring with multiple UAVs (Chapter 13), and cooperative surveillance with multiple UAVs (Chapter 14). In addition, Appendices A–F review, respectively, selected notations and abbreviations, graph theory notations, matrix theory notations, rigid body attitude dynamics, linear system theory background, and nonlinear stability theory background.

1.4 Notes

The results in this chapter are based mainly on [191]. For further literature review on consensus algorithms, see [156].

Acknowledgment is given to

©2007 IEEE. Reprinted, with permission, from Wei Ren, Randal W. Beard, and Ella M. Atkins, "Information consensus in multivehicle cooperative control: Collective group behavior through local interaction," *IEEE Control Systems Magazine*, vol. 27, no. 2, pp. 71–82, April 2007.

Springer

http://www.springer.com/978-1-84800-014-8

Distributed Consensus in Multi-vehicle Cooperative Control
Theory and Applications
Ren, W.; Beard, R.
2008, XV, 319 p. With online files/update., Hardcover
ISBN: 978-1-84800-014-8