Cooperative Control Design Strategies with Local Interactions

Wei Ren, *Member, IEEE*

***Abstract*— In this paper, we study cooperative control design strategies that only require local information exchange between vehicles. In particular, we focus on how consensus algorithms or their extensions can be applied in cooperative control of multi-vehicle systems. Three types of coupling between vehicles are distinguished and corresponding design strategies are proposed. These design strategies are then applied to several cooperative control applications including cooperative timing, formation maintenance, rendezvous, altitude alignment, and synchronized rotations as a proof of concept.**

I. Introduction

Autonomous vehicle systems are expected to find potential applications in military operations, search and rescue, environment monitoring, commercial cleaning, material handling, and homeland security. While single vehicles performing solo missions will yield some benefits, greater benefits will come from the cooperation of teams of vehicles. One motivation for multiple vehicle systems is to achieve the same gains for mechanically controlled systems as has been gained in distributed computation. Rather than having a single monolithic (and therefore expensive and complicated) machine do everything, the hope is that many inexpensive, simple machines, can achieve the same or enhanced functionality, through coordination.

Cooperative control enables team objective satisfaction by sharing specific information among team members. In some cooperative control problems, there exists an identifiable common reference state for each vehicle in the team, which is called the coordination variable in [1]. If each vehicle has access to the common reference state and responds accordingly, cooperative objectives can be guaranteed for the team. For example, if a team of vehicles are required to form a rigid formation, the formation centroid trajectory can be the common reference state for the whole team. In some other cooperative control problems, there may not exist an obvious common reference state for the team and each vehicle may only have access to the state information of its local neighbors. As a result, emergent group behaviors occur through local intervehicle interactions. There are also some cooperative control problems where the team has an identifiable group goal but under certain circumstances some vehicles may need to sacrifice the group goal during some short time period to react to their environment or local neighbors.

In cooperative control strategy design, decentralized schemes are superior to centralized schemes in terms of robustness and scalability. Information consensus focuses on guaranteeing that each vehicle in a team converges to a consistent view of their information states through local interactions. As an inherently distributed strategy, information consensus has received significant attention in the cooperative control community, see e.g., [2], [3], [4], [5], [6], to name a few. A survey of information consensus in cooperative control is given in [7].

In this paper we distinguish three types of coupling between vehicles and provide a preliminary effort toward general cooperative control design strategies for a class of cooperative control problems. These design strategies have the advantage that only local information exchange between vehicles is required. In particular, we focus on how consensus algorithms or their extensions can be applied in cooperative control strategy design with local interactions.

II. Background and Preliminaries

A. Graph Theory

It is natural to model information exchange between vehicles by directed/undirected graphs. A digraph (directed graph) consists of a pair $(\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is a finite nonempty set of nodes and $\mathcal{E} \in \mathcal{N}^2$ is a set of ordered pairs of nodes, called edges. As a comparison, the pairs of nodes in an undirected graph are unordered. If there is a directed edge from node v_i to node v_j, then v_i is defined as the parent node and v_j is defined as the child node. A directed path is a sequence of ordered edges of the form $(v_{i_1}, v_{i_2}), (v_{i_2}, v_{i_3}), \cdots$, where $v_{i_j} \in \mathcal{N}$, in a digraph. An undirected path in an undirected graph is defined accordingly. A digraph is called strongly connected if there is a directed path from every node to every other node. An undirected graph is called connected if there is a path between any distinct pair of nodes. A directed tree is a digraph, where every node, except the root, has exactly one parent. A (directed) spanning tree of a digraph is a directed tree formed by graph edges that connect all the nodes of the graph. We say that a graph has (or contains) a (directed) spanning tree if there exists a (directed) spanning tree being a subset of the graph. Note that the condition that a digraph has a (directed) spanning tree is equivalent to the case that there exists a node having a directed path to all the other nodes. In the case of undirected graphs, having an undirected spanning tree is equivalent to being connected. However, in the case of directed graphs, having a directed spanning is not equivalent to being strongly connected. The union of a group of digraphs is a digraph with nodes given by the union of the node sets and edges given by the union of the edge sets of those digraphs. Fig. 1 shows a directed graph with more than one possible spanning trees. The double arrows denote one

Wei Ren is with the Department of Electrical and Computer Engineering, Utah State University, Logan, UT 84322-4120, USA (phone: 435-797-2831; fax: 435-797-5982; email: wren@engineering.usu.edu).

1-4244-0065-1/06/$20.00 ©2006 IEEE

possible spanning tree with A_5 as the parent. Spanning trees with A_1 and A_4 as the parent, are also possible.

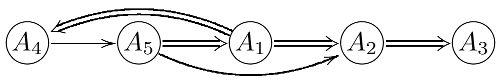

Fig. 1. A directed graph that has more than one possible spanning trees, but is not strongly connected. One possible spanning tree is denoted with double arrows.

The adjacency matrix $A = [a_{ij}]$ of a weighted digraph is defined as $a_{ii} = 0$ and $a_{ij} > 0$ if $(j, i) \in \mathcal{E}$ where $i \neq j$. The Laplacian matrix of the weighted digraph is defined as $L = [\ell_{ij}]$, where $\ell_{ii} = \sum_{j \neq i} a_{ij}$ and $\ell_{ij} = -a_{ij}$ where $i \neq j$. For an undirected graph, the Laplacian matrix is symmetric positive semi-definite.

B. Consensus Algorithms

In this section we review first-order and second-order consensus algorithms.

Let $\xi_i \in I\!R^m$ and $\zeta_i \in I\!R^m$ be the information states of the i^{th} vehicle. For example, ξ_i may take the role of position, altitude, or heading angle while ζ_i may take the role of velocity, climb rate, or angular velocity of the i^{th} vehicle.

For information states with first-order dynamics, we apply the following first-order consensus algorithm:

$$\dot{\xi}_i = -\sum_{j=1}^{n} g_{ij} k_{ij} (\xi_i - \xi_j), \quad i \in \{1, \cdots, n\} \tag{1}$$

where $k_{ij} > 0$, $g_{ii} \stackrel{\triangle}{=} 0$, and g_{ij} is 1 if information flows from vehicle j to vehicle i and 0 otherwise, $\forall i \neq j$.

For first-order consensus algorithm (1), consensus is said to be reached asymptotically among multiple vehicles if $\|\xi_i - \xi_j\| \to 0$, $\forall i \neq j$, as $t \to \infty$ for any $\xi_i(0)$.

Under a fixed information exchange topology, algorithm (1) achieves consensus asymptotically if and only if the information exchange topology has a (directed) spanning tree [8]. Under time-varying information exchange topologies, algorithm (1) reaches consensus asymptotically if there exist infinitely many consecutive uniformly bounded time intervals such that the union of the information exchange graph across each interval has a (directed) spanning tree [6].

For information states with second-order dynamics, we apply the following second-order consensus algorithm:

$$\begin{aligned} \dot{\xi}_i &= \zeta_i \\ \dot{\zeta}_i &= -\sum_{j=1}^{n} g_{ij} k_{ij} [(\xi_i - \xi_j) + \gamma(\zeta_i - \zeta_j)], \quad i \in \{1, \cdots, n\} \end{aligned} \tag{2}$$

where $k_{ij} > 0$, $\gamma > 0$, $g_{ii} \stackrel{\triangle}{=} 0$, and g_{ij} is 1 if information flows from vehicle j to vehicle i and 0 otherwise.

For second-order consensus algorithm (2), consensus is said to be reached asymptotically among multiple vehicles if $\|\xi_i(t) - \xi_j(t)\| \to 0$ and $\|\zeta_i(t) - \zeta_j(t)\| \to 0$, $\forall i \neq j$, as $t \to \infty$ for any $\xi_i(0)$ and $\zeta_i(0)$.

Let $L = [\ell_{ij}]$ be the Laplacian matrix corresponding to the information exchange topology for the team of vehicles, where $\ell_{ii} = \sum_{j \neq i} g_{ij} k_{ij}$ and $\ell_{ij} = -g_{ij} k_{ij}$, $\forall i \neq j$. Under a fixed information exchange topology, algorithm (2) achieves consensus asymptotically if the information exchange topology has a (directed) spanning tree and

$$\gamma > \max_{i=2,\cdots,n} \sqrt{\frac{2}{|\mu_i| \cos(\frac{\pi}{2} - \tan^{-1} \frac{-\mathrm{Re}(\mu_i)}{\mathrm{Im}(\mu_i)})}}, \tag{3}$$

where μ_i, $i = 2, \cdots, n$, are the non-zero eigenvalues of $-L$, and $\mathrm{Re}(\cdot)$ and $\mathrm{Im}(\cdot)$ represent the real and imaginary parts of a number respectively [9].

Let $t_0, t_1, t_2, \cdots$ be the times when the information exchange topology switches. Also let τ be the dwell time such that $t_{i+1} - t_i > \tau$, $\forall i = 0, 1, \cdots$. Algorithm (2) achieves consensus asymptotically if the information exchange topology has a (directed) spanning tree and γ satisfies inequality (3) for each t_i, and the dwell time is sufficiently large [9].

Note that Eqs. (1) and (2) represent the fundamental forms of consensus algorithms. These algorithms can be extended to achieve different convergence results. For example, consensus algorithm (1) can be extended to guarantee that (i) $\xi_i \to \xi_j \to \xi^d$ and (ii) $\xi_i - \xi_j \to \Delta_{ij}$, where Δ_{ij} represent the desired separation between ξ_i and ξ_j. Consensus algorithm (2) can be extended to guarantee that (i) $\xi_i \to \xi_j$ and $\zeta_i \to 0$, (ii) $\xi_i - \xi_j \to \Delta_{ij}$ and $\zeta_i \to \zeta_j$, and (iii) $\xi_i \to \xi_j \to \xi^d$ and $\zeta_i \to \zeta_j \to \zeta^d$, where ξ^d and ζ^d represent the desired values for ξ_i and ζ_i respectively.

III. Cooperative Control Design Strategies via Information Consensus

In this section, we apply consensus algorithms to design cooperative control strategies for multi-vehicle systems. We argue that consensus based schemes are feasible for a class of cooperative control problems. In particular, we consider three types of cooperative control problems. In the first type, there exists an identifiable common reference state for each individual vehicle in the team. In [1], [8] the common reference state is called the "*coordination variable*". The common reference state serves as a basis for each vehicle to derive local control strategies. As a result, having the knowledge of the common reference state facilitates cooperation for the whole team. For example, in formation control problems, the state of an actual or virtual team leader or formation center serves as a reference for each vehicle. In cooperative timing missions, the estimated team arrival time at specified destinations serves as a reference for each vehicle. In the second type, each vehicle adjusts its own state according to the states of its local neighbors. For example, in order to achieve a cooperative observation, altitude or attitude alignment may be required for a team of vehicles with only local interactions. The third type is a combination of the previous two types. For example, each vehicle in the team may have access to the state of the formation center to achieve certain group behaviors as well as the states of its local neighbors to achieve collision avoidance. In the following, the terms "*group-level coupling*", "*vehicle-level coupling*", and "*mixed coupling*" are used to represent the coupling between vehicles in the first, second, and third type respectively.

For group-level coupling, we let ξ denote the common reference state for the whole team. Also let x_i, y_i, and u_i represent the local state, output, and control input of the i^{th} vehicle respectively. A centralized cooperative control strategy can be designed as follows:

$$\begin{aligned}\dot{\xi} &= f(t, \xi, y_1, \cdots, y_n)\\ \dot{x}_i &= g_i(t, x_i, u_i)\\ y_i &= h_i(t, x_i)\\ u_i &= k_i(t, y_i, \xi), \quad i \in \{1, \cdots, n\},\end{aligned}$$

where the i^{th} control input depends on the common reference state ξ and its own output y_i.

In this centralized scheme, the common reference state is implemented at a central location and broadcast to every vehicle in the team. However, this implementation results in a single point of failure and is not scalable well to a large number of vehicles.

A natural remedy to these drawbacks is to instantiate a local copy of the common reference state on each vehicle. If each vehicle implements the same cooperation algorithm, we expect that the decentralized scheme achieves the same cooperation as the centralized scheme. However, due to different local situation awareness of each vehicle, there exist discrepancies among each instantiation of the common reference state. For example, in multi-vehicle simultaneous arrival missions, each vehicle's time-over-target may be dynamically changing as the vehicle encounters pop-up threats. As a result, consensus algorithms need to be applied to guarantee that each instantiation of the common reference state converges to a sufficiently common value.

Fig. 2 shows a decentralized design scheme for multi-vehicle systems with group-level coupling. The hierarchical architecture consists of four layers: mission planner, consensus module, cooperation module, and physical vehicle. Each vehicle instantiates a local copy of the mission planner, consensus module, and cooperation module. Each mission planner instantiation outputs a sequence of desired mission goals $\xi^{(p)}$, $p = 1, \cdots, P$, to the consensus module instantiation. Each consensus module instantiation drives each instantiation of the common reference state ξ_i to a consistent value and also toward the desired mission goal through communication with (possibly time-varying) local neighbors. Based on each instantiation of the common reference state, each cooperation module instantiation specifies local control laws u_i for each vehicle. Note that each lower layer has feedback information describing the performance of that layer to the upper layer as denoted by ϑ_i, z_i, and y_i.

We assume that the dynamics of each instantiation of the common reference state are given by

$$\dot{\xi}_i = f_i(t, \xi_i, \{j \in \mathcal{N}_i(t) | \xi_j\}, y_i),$$

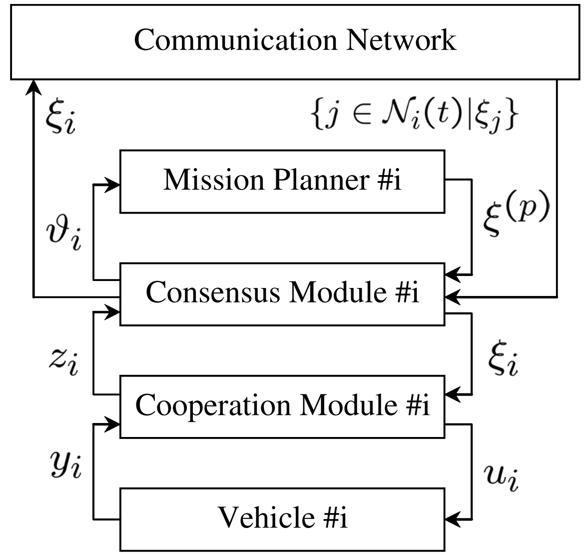

Fig. 2. Decentralized design scheme with group-level coupling.

where $\mathcal{N}_i(t)$ denotes the set of vehicles whose instantiations of the common reference state are available to vehicle i, and y_i introduces group feedback from vehicle i to its instantiation of the common reference state. The goal is to guarantee that $\|\xi_i - \xi_j\| \to 0$, $\forall i \neq j$, and $\xi_i \to \xi^{(p)}$, $\forall i$.

The dynamics of each vehicle are given by

$$\begin{aligned}\dot{x}_i &= g_i(t, x_i, u_i)\\ y_i &= h_i(t, x_i)\\ u_i &= k_i(t, y_i, \xi_i),\end{aligned}$$

where the i^{th} control input depends on its own output and the i^{th} instantiation of the common reference state.

For vehicle-level coupling, the control input to each vehicle depends on information from its local neighbors. Fig. 3 shows a design scheme for multi-vehicle systems with vehicle-level coupling, where each vehicle instantiates different modules such as goal seeking, consensus building, formation keeping, and collision avoidance to characterize different group behaviors. Through local communication or sensing, each vehicle specifies its local control law based on these combined modules. The functionality of the consensus module is to guarantee the alignment of velocities, attitude, altitude, and so on among multiple vehicles.

The dynamics of each vehicle are given by

$$\begin{aligned}\dot{x}_i &= g_i(t, x_i, u_i)\\ y_i &= h_i(t, x_i)\\ z_i &= \eta_i(t, y_i, \{\ell \in \mathcal{J}_i(t) | y_\ell\})\\ u_i &= k_i(t, z_i),\end{aligned}$$

where $\mathcal{J}_i(t)$ denotes the set of vehicles whose information is available to vehicle i through either communication or sensing, and z_i denotes the information sensed by and/or communicated to vehicle i. One example is that z_i is composed of a set of vehicle positions y_j, where $j \in \mathcal{J}_i(t) \bigcup \{i\}$.

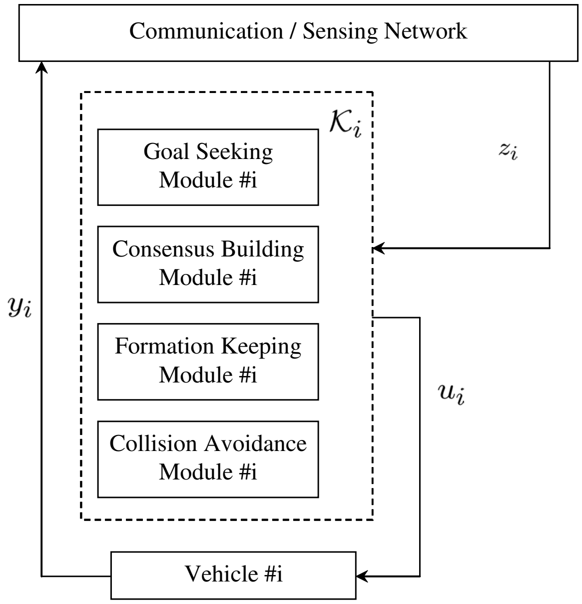

Fig. 3. Decentralized design scheme with vehicle-level coupling.

Another is that z_i is composed of a set of relative position measurements $y_i - y_j$, where $j \in \mathcal{J}_i(t)$.

The case of mixed coupling is a combination of those cases of group-level coupling and vehicle-level coupling.

In the centralized scheme, the dynamics of the common reference state are given by

$$\dot{\xi} = f(t, \xi, z_i).$$

The dynamics of each vehicle are given by

$$\begin{aligned}\dot{x}_i &= g_i(t, x_i, u_i)\\ y_i &= h_i(t, x_i)\\ z_i &= \eta_i(t, y_i, \{\ell \in \mathcal{J}_i(t)|y_\ell\})\\ u_i &= k_i(t, y_i, z_i, \xi).\end{aligned}$$

In the decentralized scheme, the dynamics of each instantiation of the common reference state are given by

$$\dot{\xi}_i = f_i(t, \xi_i, \{j \in \mathcal{N}_i(t)|\xi_j\}, z_i).$$

The dynamics of each vehicle are given by

$$\begin{aligned}\dot{x}_i &= g_i(t, x_i, u_i)\\ y_i &= h_i(t, x_i)\\ z_i &= \eta_i(t, y_i, \{\ell \in \mathcal{J}_i(t)|y_\ell\})\\ u_i &= k_i(t, y_i, z_i, \xi_i, \{j \in \mathcal{N}_i(t)|\xi_j\}).\end{aligned}$$

IV. Application Examples

In this section, we apply the decentralized design schemes with group-level coupling and vehicle-level coupling to several cooperative control problems.

In the following we assume that the cooperative team consists of six vehicles. Without loss of generality, we assume that the information flow topology for the six vehicles is given by Fig. 4. Note that Fig. 4 has a (directed) spanning tree. It is worthwhile to mention that the design schemes are applicable to a class of problems although we only focus on several applications here.

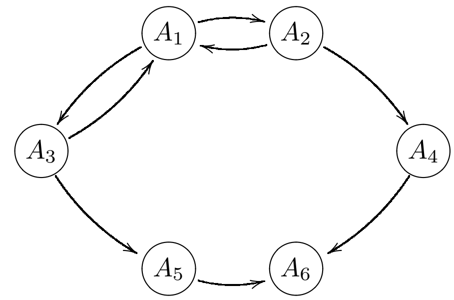

Fig. 4. Information flow topology for six vehicles.

A. Cooperative Timing

In the first example, we apply the design scheme denoted by Fig. 2 to a cooperative timing problem. The cooperative timing problem requires that a team of UAVs arrive at their destinations simultaneously (simultaneous arrival) or in a specified sequence with the time increments between arrival times specified exactly (tight sequencing) [1].

Let ξ_i be the estimated arrival time of the i^{th} UAV. Also let δ_i be constants. We update ξ_i according to the following strategy:

$$\dot{\xi}_i = -\sum_{j=1}^{n} g_{ij}k_{ij}[(\xi_i - \delta_i) - (\xi_j - \delta_j)].$$

As a result, $\xi_i - \delta_i \to \xi_j - \delta_j$, that is, $\xi_i - \xi_j \to \delta_i - \delta_j$. By appropriately choosing δ_ℓ, $\ell \in \{1, \cdots, n\}$, we can guarantee either simultaneous arrival or tight sequencing. At the vehicle level, each UAV adjusts its velocity and path to guarantee that it arrives at the target according to its estimated arrival time.

Let $\delta_j = 3 * (j-1)$, $j = 1, \cdots, 6$. As a result, $\xi_{j+1} \to \xi_j + 3$, $j = 1, \cdots, 5$. Fig. 5 shows the estimated arrival times of six UAVs. Note that the time increment between the estimated arrival times approaches three minutes as desired even if each vehicle has arbitrary initial estimates.

B. Mobile Robot Formation Maneuvering

In the second example, we apply the design scheme denoted by Fig. 2 to guarantee accurate formation maintenance of multiple mobile robots during their maneuvers.

Consider the mobile robot kinematic equation given by

$$\dot{q}_i = u_i, \tag{4}$$

where q_i is the position of the i^{th} robot and u_i is the control input.

Let $\xi = [x_0(s), y_0(s), \theta_0(s)]^T$ represent the state of the formation center, where $(x_0(s), y_0(s))$ and $\theta_0(s)$ are the position and orientation of a coordinate frame located at the formation center parameterized by a parameter s. We

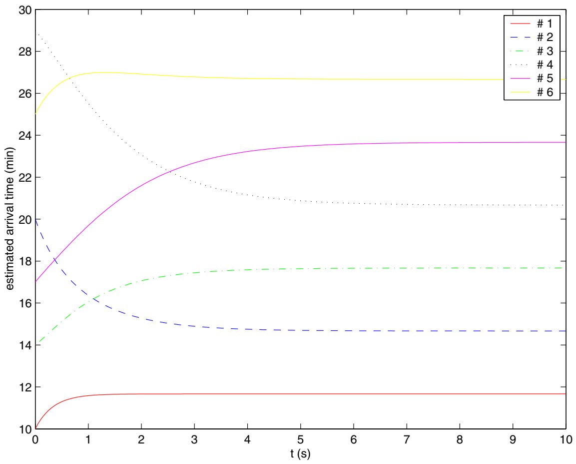

Fig. 5. Tight sequencing of six UAVs.

instantiate a local copy of s_i on each robot and apply the following consensus algorithm to drive $s_i \to s_j$ and $\dot{s}_i \to \nu$:

$$\dot{s}_i = -\sum_{j=1}^{n} g_{ij} k_{ij} (s_i - s_j) + \nu,$$

where ν is a feedforward signal.

We simulate a scenario where the formation center follows a trajectory of a circle while the whole group preserves the desired hexagon formation shape during the maneuver. Fig. 6 shows the formation maneuvers of the six robots at $t = 0$, 25, 50, 75, and 100 (s) respectively. The green circle represents the desired trajectory of the formation center, the actual formation at each time is represented by polygons with square vertices denoting the actual location of each robot, and the desired formation at each time is represented by polygons with star vertices denoting the desired location of each robot.

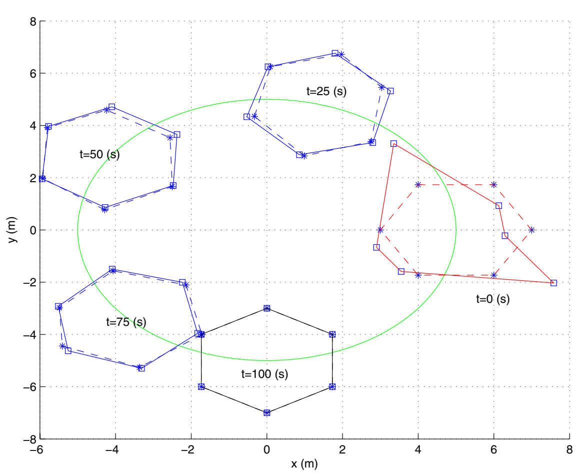

Fig. 6. Formation maneuvering of six mobile robots.

C. Rendezvous

In the third example, we apply the design scheme denoted by Fig. 3 to a rendezvous problem where multiple vehicles are required to arrive at a target location simultaneously. Note that the rendezvous problem can be thought of as a special case of cooperative timing where each vehicle has a common destination.

Consider vehicle dynamics given by

$$\ddot{r}_i = u_i,$$

where r_i is the position and u_i is the control input. Let

$$u_i = -\alpha \dot{r}_i - \sum_{j=1}^{n} g_{ij} k_{ij} [(r_i - r_j) + \gamma(\dot{r}_i - \dot{r}_j)],$$

where $\alpha > 0$ and γ satisfies inequality (3). As a result, $r_i \to r_j$ and $v_i \to 0$.

Fig. 7 shows a scenario where six vehicles start from different locations denoted by circles but arrive at a rendezvous destination denoted by a square through only local information exchange.

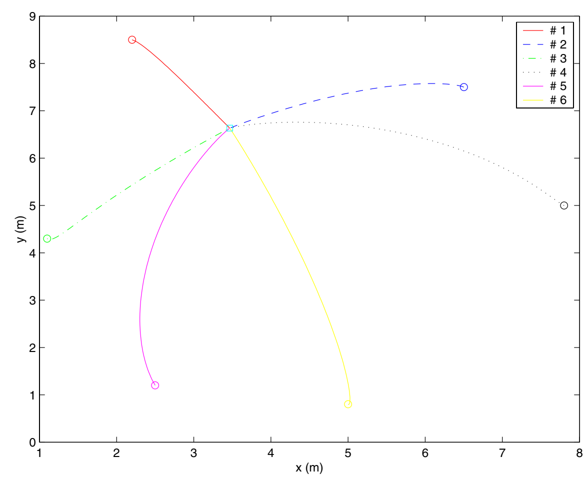

Fig. 7. Six vehicle rendezvous.

D. UAV Altitude Alignment

In the fourth example, we apply the design scheme denoted by Fig. 3 to align the altitudes of multiple UAVs.

Consider the UAV altitude dynamics given by

$$\ddot{h}_i = -\alpha_{\dot{h}_i} \dot{h}_i + \alpha_{hi}(h_i^c - h_i),$$

where we assume that each UAV is equipped with efficient low level autopilots with $\alpha_{\dot{h}i} > 0$ and $\alpha_{hi} > 0$.

We propose the following control law for h_i^c as

$$h_i^c = \frac{1}{\alpha_{hi}} (\mu_{hi} + \alpha_{\dot{h}_i} \dot{h}_i) + h_i,$$

where

$$\mu_{hi} = -\alpha \dot{h}_i - \sum_{j=1}^{n} g_{ij} k_{ij} [(h_i - h_j) + \gamma(\dot{h}_i - \dot{h}_j)], \quad (5)$$

where $\alpha > 0$, $k_{ij} > 0$, and γ satisfied inequality (3). In Eq. (5) the first term is used to guarantee that $\dot{h}_i \to 0$, and the second term is used to guarantee that $h_i \to h_j$ and $\dot{h}_i \to \dot{h}_j$.

Fig. 8 shows the altitudes and climb rates of six UAVs. Note that each UAV aligns its altitude with its local neighbors in a distributed manner.

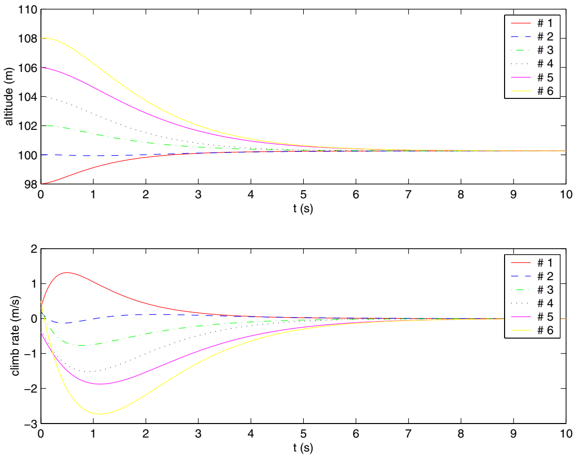

Fig. 8. Altitude alignment among six UAVs.

E. Synchronized Spacecraft Rotations

In the last example, we apply the design scheme denoted by Fig. 3 to synchronize rotations of multiple spacecraft.

Consider spacecraft dynamics given by

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

where $q_i = [\hat{q}_i^T, \bar{q}_i]^T$ is the unit quaternion of the i^{th} spacecraft, ω_i is the angular velocity, and J_i and τ_i are inertia tensor and control torque.

The control torque τ_i can then be designed as

$$\tau_i = -k_G\widehat{q^{d*}q_i} - d_G\omega_i - \sum_{j=1}^{n} g_{ij}[a_{ij}\widehat{q_j^*q_i} + b_{ij}(\omega_i - \omega_j)], \quad (6)$$

where k_G, d_G, a_{ij}, and b_{ij} are positive scalars, and $\hat{p}$ represents the vector part of quaternion p. In Eq. (6), the first term is used to guarantee that $q_i \to q^d$, the second term is used to guarantee that $\omega_i \to 0$, and the last term is used to guarantee that $q_i \to q_j$ and $\omega_i \to \omega_j$. Note that Eq. (6) is more general than the control law in [10].

With the above control law, a group of spacecraft can reach their desired attitude q^d and maintain the same attitudes during the transition. Fig. 9 shows the quaternion attitudes of spacecraft #1, #3, and #5, where $q_i^{(j)}$ denotes the j^{th} component of quaternion q_i.

V. Conclusion

We have proposed cooperative control design strategies based on the concept of information consensus. These strategies only require local information exchange between vehicles. We have also applied these strategies to several cooperative control problems including cooperative timing, formation maintenance, rendezvous, altitude alignment, and synchronized rotations.

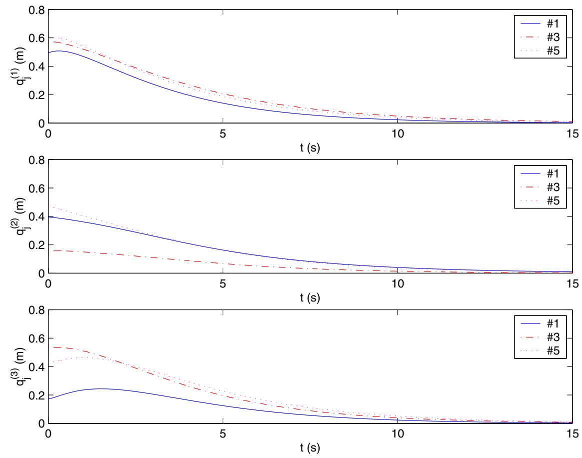

Fig. 9. Synchronized rotations among six spacecraft.

Acknowledgment

The author would like to gratefully acknowledge Profs. Randy Beard and Tim McLain for their technical guidance on the subject.

References

[1] T. W. McLain and R. W. Beard, "Coordination variables, coordination functions, and cooperative timing missions," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 28, no. 1, pp. 150–161, January 2005.

[2] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, June 2003.

[3] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, September 2004.

[4] Z. Lin, M. Broucke, and B. Francis, "Local control strategies for groups of mobile autonomous agents," *IEEE Transactions on Automatic Control*, vol. 49, no. 4, pp. 622–629, 2004.

[5] L. Moreau, "Stability of multi-agent systems with time-dependent communication links," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 169–182, February 2005.

[6] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 655–661, May 2005.

[7] W. Ren, R. W. Beard, and E. M. Atkins, "A survey of consensus problems in multi-agent coordination," in *Proceedings of the American Control Conference*, Portland, OR, June 2005, pp. 1859–1864.

[8] W. Ren, R. W. Beard, and T. W. McLain, "Coordination variables and consensus building in multiple vehicle systems," in *Cooperative Control: A Post-Workshop Volume 2003 Block Island Workshop on Cooperative Control*, V. Kumar, N. E. Leonard, and A. S. Morse, Eds., vol. 309. Springer-Verlag Series: Lecture Notes in Control and Information Sciences, 2004, pp. 171–188.

[9] W. Ren and E. M. Atkins, "Distributed multi-vehicle coordinated control via local information exchange," *International Journal of Robust and Nonlinear Control*, 2005, (in review).

[10] W. Ren and R. W. Beard, "Decentralized scheme for spacecraft formation flying via the virtual structure approach," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 27, no. 1, pp. 73–82, January-February 2004.