Experimental Implementation and Validation of Consensus Algorithms on a Mobile Actuator and Sensor Network Platform

Wei Ren, Haiyang Chao, William Bourgeous, Nathan Sorensen, and YangQuan Chen

Abstract— **In this paper, we experimentally implement and validate distributed consensus algorithms on a mobile actuator and sensor network platform under directed, possibly switching interaction topologies to explore issues and challenges in distributed multi-vehicle cooperative control. Distributed consensus algorithms are applied to three target applications namely rendezvous, axial alignment, and formation maneuvering. In the rendezvous application, multiple mobile robots simultaneously arrive at a common *a priori* unknown target location determined through team negotiation. In the axial alignment application, multiple mobile robots collectively align their final positions along a line. In the formation maneuvering application, multiple mobile robots form a rigid geometric shape and maneuver as a group with a given group velocity. The experimental results show the effectiveness and robustness of the consensus algorithms even in the presence of platform physical limitations, packet loss, information delay, etc.**

I. INTRODUCTION

As an inherently distributed strategy that only requires local neighbor-to-neighbor information exchange among vehicles, information consensus has received significant attention in the cooperative control community recently [1]. The basic idea for information consensus is that each vehicle updates its information state based on the information states of its local, possibly time-varying neighbors in such a way that the final information state of each vehicle converges to a common value. This basic idea can be extended to a variety of different scenarios that incorporate group behaviors and dynamics.

Theoretical aspects of consensus algorithms have recently been studied extensively in the literature (see e.g., [2]–[4]). Consensus algorithms have applications in rendezvous, formation control, flocking, attitude alignment, and sensor networks. However, in the current literature, most of the research activities in information consensus have focused on theoretical aspects, and most of the applications are demonstrated by means of simulations. Recent efforts in experimental implementation of multi-robot flocking and cyclic pursuit are reported, respectively, in [5] and [6], where flocking assumes undirected information exchange and the interaction topology for cyclic pursuit forms a unidirectional ring. Experimental implementation and validation of general consensus algorithms under a general (possibly directed switching) interaction topologies play an important role for studying distributed cooperative control schemes.

The main purpose of the current paper is to experimentally implement and validate consensus algorithms under directed, possibly switching interaction topologies to explore issues and challenges in distributed multi-vehicle cooperative control. In particular, distributed consensus algorithms are applied to three target applications namely rendezvous, axial alignment, and formation maneuvering. In the rendezvous application, multiple robots are required to simultaneously arrive at a common *a priori* unknown target location determined through team negotiation. The rendezvous case is directly relevant to unmanned air vehicle (UAV) cooperative timing missions, where multiple UAVs are controlled to converge on the boundary of a radar detection area simultaneously to maximize the element of surprise. In the axial alignment application, multiple robots are required to be evenly distributed on a line with given separation distance through team negotiation. The axial alignment case is directly relevant to sensor deployment and satellite attitude alignment applications. In the formation maneuvering application, multiple robots are required to form a geometric shape and move as a group with a given group velocity. The formation maneuvering case is directly relevant to cooperative surveillance tasks involving a team of vehicles and spacecraft formation flying missions. The three target applications are validated on a low-cost mobile actuator and sensor network platform. The experimental results show the effectiveness and robustness of consensus algorithms even in the presence of platform physical limitations, packet loss, information delay, etc.

II. BACKGROUND AND PRELIMINARIES

It is natural to model the information exchange between vehicles by directed/undirected graphs. A directed graph consists of a pair $(\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is a finite nonempty set of nodes and $\mathcal{E} \in \mathcal{N} \times \mathcal{N}$ is a set of ordered pairs of nodes, called edges. An edge (i, j) in a directed graph denotes that vehicle j can obtain information from vehicle i, but not necessarily vice versa. In contrast, the pairs of nodes in an undirected graph are unordered, where an edge (i, j) denotes that vehicles i and j can obtain information from one another. Note that an undirected graph can be considered a special case of a directed graph, where an edge (i, j) in the undirected graph corresponds to edges (i, j) and (j, i) in the directed graph. If there is a directed edge from node i to node j, then i is defined as the parent node and j is defined as the child node.

A directed path is a sequence of ordered edges in a directed graph of the form $(i_1, i_2), (i_2, i_3), \ldots$, where $i_j \in \mathcal{N}$. A directed tree is a directed graph, where every node has exactly one parent except for one node, called the root, which has no parent, and the root has a directed path to every

W. Ren, H. Chao, W. Bourgeous, N. Sorensen, and Y. Chen are with the Department of Electrical and Computer Engineering, Utah State University, Logan, UT 84322, USA wren@engineering.usu.edu

1-4244-0991-8/07/$25.00/©2007 IEEE

other node. A directed spanning tree of a directed graph is a directed tree formed by graph edges that connect all of the nodes of the graph. A graph has or contains a directed spanning tree if there exists a directed spanning tree as a subset of the graph. The union of a group of graphs is a graph with nodes given by the union of the node sets and edges given by the union of the edge sets of the group of graphs.

Suppose that there are n vehicles in the team. The adjacency matrix $A = [a_{ij}] \in I\!R^{n\times n}$ of a weighted directed graph is defined as $a_{ii} = 0$ and $a_{ij} > 0$ if $(j,i) \in \mathcal{E}$, where $i \neq j$. The adjacency matrix of a weighted undirected graph is defined accordingly except that $a_{ij} = a_{ji}$, $\forall i \neq j$, since $(j,i) \in \mathcal{E}$ implies $(i,j) \in \mathcal{E}$.

Let matrix $L = [\ell_{ij}] \in I\!R^{n\times n}$ be defined as $\ell_{ii} = \sum_{j\neq i} a_{ij}$ and $\ell_{ij} = -a_{ij}$, where $i \neq j$. The matrix L satisfies the following conditions:

$$\ell_{ij} \leq 0,\ i \neq j, \quad \sum_{j=1}^{n} \ell_{ij} = 0,\ i = 1,\ldots,n. \tag{1}$$

For an undirected graph, L is the Laplacian matrix, which has the property that it is symmetric positive semi-definite. However, L for a directed graph does not have this property. In both the directed and the undirected cases, 0 is an eigenvalue of L with an associated eigenvector $\mathbf{1}$, where $\mathbf{1}$ is a column vector of all ones. In the case of undirected graphs, all of the nonzero eigenvalues of L are positive. In the case of directed graphs, all of the nonzero eigenvalues of L have positive real parts from Gershgorin disc theorem [7]. In the case of undirected graphs, 0 is a simple eigenvalue of L if and only if the undirected graph is connected [8]. In addition, the second smallest eigenvalue of L is known as the algebraic connectivity of the undirected graph. In the case of directed graphs, 0 is a simple eigenvalue of L if and only if the directed graph contains a directed spanning tree [9].

III. CONSENSUS ALGORITHMS

In this section we first review consensus algorithms for vehicles modeled by single-integrator dynamics, and then extend these algorithms to account for relative state separations and desired state tracking.

Consider vehicles with single-integrator dynamics given by

$$\dot{\xi}_i = u_i, \quad i = 1,\ldots,n, \tag{2}$$

where $\xi_i \in I\!R^m$ is the information state of the i^{th} vehicle, and $u_i \in I\!R^m$ is the control input. A consensus algorithm is proposed in [2]–[4] as

$$u_i = -\sum_{j\in\mathcal{J}_i(t)} k_{ij}(\xi_i - \xi_j), \quad i = 1,\ldots,n, \tag{3}$$

where $\mathcal{J}_i(t)$ represents the set of vehicles whose information is available to vehicle i at time t, and k_{ij} is a positive weighting factor. The objective of (3) is to drive the information state of each vehicle toward the state of its neighbor. In the following, we assume that $i \notin \mathcal{J}_i$.

For (3), consensus is said to be reached asymptotically among the n vehicles if $\xi_i(t) \to \xi_j(t)$, $\forall i \neq j$, as $t \to \infty$ for all $\xi_i(0)$.

Let $L = [\ell_{ij}] \in I\!R^{n\times n}$ be defined as

$$\ell_{ij} = -k_{ij},\ j \in \mathcal{J}_i,\ \ell_{ij} = 0,\ j \notin \mathcal{J}_i \bigcup \{i\},\ \ell_{ii} = \sum_{j\in\mathcal{J}_i} k_{ij}. \tag{4}$$

Note that L satisfies (1). Also note that with (3), (2) can be written in matrix form as $\dot{\xi} = -(L \otimes I_m)\xi$, where $\xi = [\xi_1^T,\ldots,\xi_n^T]^T$, $\otimes$ denotes the Kronecker product, and I_m is the $m \times m$ identity matrix.

We have the following two lemmas for (3) under time-invariant and switching interaction topologies respectively.

Lemma 3.1: [9] Under a time-invariant interaction topology, (3) achieves consensus exponentially if and only if the interaction topology contains a directed spanning tree. In the case that the interaction topology contains a directed spanning tree, the final consensus equilibrium is equal to the weighted average of the initial conditions of those vehicles that have a directed path to all of the other vehicles. That is, $\xi_i \to \sum_{j=1}^{n} \nu_j \xi_j(0)$, $\forall i$, where $\nu = [\nu_1,\ldots,\nu_n]^T$ is a nonnegative left eigenvector of L, given by (4), satisfying the condition that $\nu_i > 0$ if vehicle i has a directed path to every other vehicle and $\nu_i = 0$ otherwise, and $\sum_{j=1}^{n} \nu_j = 1$.

Lemma 3.2: [4] Under switching interaction topologies, (3) reaches consensus asymptotically if there exist infinitely many consecutive uniformly bounded time intervals such that the union of the interaction topologies across each time interval has a directed spanning tree.

Note that (3) represents the fundamental form of consensus algorithms. The algorithm can be extended to achieve different convergence results. For example, (3) can be extended to guarantee that the differences of the information states converge to desired values, i.e., $\xi_i - \xi_j \to \Delta_{ij}(t)$, where $\Delta_{ij}(t)$ denotes the desired (time-varying) separation between ξ_i and ξ_j. We apply the following algorithm for relative separations:

$$u_i = \dot{\delta}_i - \sum_{j\in\mathcal{J}_i(t)} k_{ij}[(\xi_i - \xi_j) - (\delta_i - \delta_j)], \quad i = 1,\ldots,n, \tag{5}$$

where $\delta_i - \delta_j$, $\forall i \neq j$, denotes the desired separation between the information states. Note that by appropriately choosing δ_ℓ, $\ell = 1,\ldots,n$, we can guarantee that the differences of the information states converge to desired values. The algorithm (5) has applications in formation control, where the team forms a certain formation shape by maintaining relative positions between vehicles. Also note that (3) corresponds to the case that $\Delta_{ij} = 0$, $\forall i \neq j$.

We have the following corollary for relative separations:

Corollary 3.1: Suppose that the interaction topology is time invariant. With (5), $\xi_i - \xi_j \to \delta_i - \delta_j$ exponentially if and only if the interaction topology has a directed spanning tree.

Proof: With (5), (2) can be written as

$$\dot{\hat{\xi}}_i = -\sum_{j\in\mathcal{J}_i} k_{ij}(\hat{\xi}_i - \hat{\xi}_j), \quad i = 1,\ldots,n,$$

where $\hat{\xi}_i = \xi_i - \delta_i$. From Lemma 3.1, we know that $\hat{\xi}_i \to \hat{\xi}_j$ exponentially if and only if the interaction topology has a directed spanning tree. The rest of the proof then follows the fact that $\hat{\xi}_i \to \hat{\xi}_j$ is equivalent to $\xi_i - \xi_j \to \delta_i - \delta_j$. ■

The consensus algorithm (3) can also be extended to guarantee that $\xi_i \to \xi^d$, $\forall i$. We apply the following algorithm:

$$u_i = \dot{\xi}^d - \alpha_i(\xi_i - \xi^d) - \sum_{j \in \mathcal{J}_i} k_{ij}(\xi_i - \xi_j), \quad i = 1, \ldots, n, \tag{6}$$

where $\alpha_i > 0$.

We have the following corollary for desired state tracking:

Corollary 3.2: Under an arbitrary time-invariant interaction topology, algorithm (6) guarantees that $\xi_i(t) \to \xi^d(t)$, $\forall i$, exponentially as $t \to \infty$.

Proof: With (6), (2) can be written in matrix form as $\dot{\tilde{\xi}} = -[(L+\Gamma)\otimes I_m]\tilde{\xi}$, where Γ is a diagonal matrix with α_i being the diagonal entries and $\tilde{\xi} = [\tilde{\xi}_1^T, \ldots, \tilde{\xi}_n^T]^T$ with $\tilde{\xi}_i = \xi_i - \xi^d$. From Gershgorin disc theorem [7], it is straightforward to see that all eigenvalues of $-(L + \Gamma)$ have negative real parts. Therefore, it follows that $\tilde{\xi} \to 0$ exponentially, that is, $\xi_i \to \xi^d$, $\forall i$. ■

In (6) the first two terms are used to guarantee that $\xi_i \to \xi^d$, $\forall i$. Note that the argument of Corollary 3.2 does not rely on the interaction topology between the vehicles. Even if there is no information exchange between the vehicles (i.e., $\mathcal{J}_i$ is empty, $\forall i$), the conclusion of the theorem is still valid. However, the last term in (6) with the interaction topology having a directed spanning tree is important to guarantee good transient performance (i.e., $\xi_i \to \xi_j$, $\forall i \neq j$, during the transition when $\xi_i \to \xi^d$).

Note that although we only deal with time-invariant interaction topologies above, similar results can be extended to the case of switching topologies.

IV. EXPERIMENTAL VALIDATION

In this section, we experimentally validate the consensus algorithms via three cooperative control applications namely rendezvous, axial alignment, and formation maneuvering. We will describe, in detail, the experimental platform, implementation of the three applications, and experimental results.

A. Experimental Platform

The Mobile Actuator and Sensor Network (MASnet) platform in the Center for Self-Organizing and Intelligent Systems (CSOIS) at Utah State University combines wireless sensor networks with mobility [10]. That is, a large number of robots can serve both as actuators and sensors. Although each robot has limited sensing, computation, and communication ability, they can coordinate with each other as a team to achieve challenging cooperative control tasks such as formation keeping and environment monitoring.

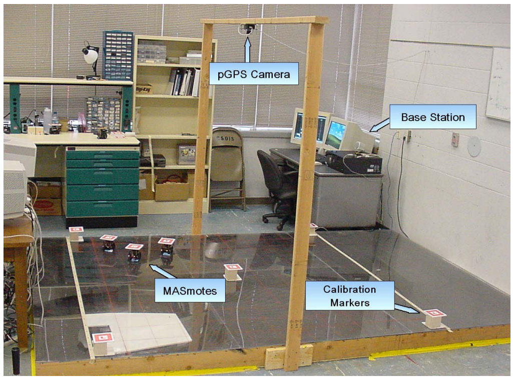


Fig. 1. MASnet experimental platform.

The MASnet platform is comprised of MASmotes, an overhead camera, and a base station PC as shown in Fig. 1. MASmotes are actually two-wheel differentially steered robots that can carry sensors and actuators wireless networked via Micaz from Crossbow. The functionality of MASmotes includes inter-mote and mote to base station communication, data collecting, PWM signal generation, and encoder counting. An overhead CCD camera is used to identify each robot and determine its position and orientation (i.e., pseudo-GPS information). Images from the camera are processed by the base station. The functionality of the base station includes image processing, serial to programming board communication, pseudo-GPS information broadcasting, and decision making. The base station communicates with a gateway mote mounted on a programming board through a serial port. The gateway mote then communicates with all of the MASmotes over a 2.4 GHz wireless mesh network. Note that the gateway mote serves as a gateway between wireless communication and serial port communication, and its only purpose is to forward all messages between the serial port and the RF port. Through communication the base station can send commands and pseudo-GPS information to each MASmote. All the MASmotes can also communicate with each other over the 2.4 GHz wireless mesh network.

B. Implementation of Three Target Applications on MASnet Platform

Because both inter-mote and mote to base station communication are available, the MASnet platform can be used to experimentally test both centralized and decentralized cooperative control schemes. For a centralized cooperative control scheme, each MASmote is only responsible for its low-level motor control while the base station, served as a centralized station, broadcasts pseudo-GPS information to each MASmote robot, implements cooperative control algorithms, and sends control commands based on the information gathered from the whole team. For a decentralized cooperative control scheme, each MASmote implements its own cooperative control algorithm based on the pseudo-GPS information provided by the base station.

In our experiments, all of the control algorithms are implemented on the MASmotes, and each MASmote only uses the pseudo-GPS information of its own and its local neighbors even if the pseudo-GPS information of all of the MASmotes is provided by the base station. By doing so, we can test distributed cooperative control algorithms involving only local neighbor-to-neighbor information exchange via communication or sensing for multi-vehicle systems. The feature of local information exchange is important in applications, where communication or sensing topologies are usually not fully connected, vehicles only have limited communication range and bandwidth, power consumption of the team may be constrained, and the stealth of the team may need to be increased.

We focus on three target applications in our experiments including rendezvous, axial alignment, and formation maneuvering. In all of these three applications, we only allow local neighbor-to-neighbor information exchange.

Let $r_i = [x_i, y_i]^T$ and $r_i^d = [x_i^d, y_i^d]^T$ denote the actual and desired position of MASmote robot i respectively. For rendezvous, the following consensus algorithm is applied

$$\dot{r}_i^d = -\sum_{j\in\mathcal{J}_i}(r_i - r_j).$$

For axial alignment, the following extended consensus algorithm is applied:

$$\dot{r}_i^d = -\sum_{j\in\mathcal{J}_i}[(r_i - r_j) - (\delta_i - \delta_j)],$$

where $\delta_i = [\delta_{ix}, \delta_{iy}]^T$ has been chosen to guarantee that the robots align on a horizontal line with a separation distance of 24 cm along the x axis between two adjacent neighboring robots. For formation maneuvering, the following extended consensus algorithm is applied

$$\dot{r}_i^d = \dot{r}_g^d - \alpha_i(r_i - r_g^d - \delta_i) - \sum_{j\in\mathcal{J}_i}[(r_i - r_j) - (\delta_i - \delta_j)],$$
$$i = \ell,$$
$$\dot{r}_i^d = \frac{1}{|\mathcal{J}_i|}\sum_{j\in\mathcal{J}_i}\{\dot{r}_j - [(r_i - r_j) - (\delta_i - \delta_j)]\}, \quad i \neq \ell,$$

where $\alpha_i > 0$, r_g^d denotes the desired trajectory of the formation, $|\mathcal{J}_i|$ denotes the cardinality of $\mathcal{J}_i$, ℓ denotes the index of the vehicle that has the knowledge of r_g^d (i.e, team leader), and $\delta_i = [\delta_{ix}, \delta_{iy}]^T$ has been chosen to ensure a separation distance of 24 cm between two adjacent neighboring robots along both x axis and y axis.

In our experiments, discrete-time versions of the above algorithms are used. The control input to each MASmote is the desired position r_i^d. Each MASmote robot updates its desired position at each time instant when the robot receives the position and orientation information of its own and its local neighbors. The update period depends on the pseudo-GPS information update period, which is between 0.1 and 0.2 seconds on average. Low-level PID control algorithms have been developed to achieve accurate position control so that r_i tracks r_i^d.

C. Experimental Results

In this subsection, we show experimental results for rendezvous, axial alignment, and formation maneuvering on our MASnet platform.

1) Rendezvous: For the rendezvous application, we study rendezvous of four MASmote robots under time-invariant and dynamic interaction topologies respectively. The motivation for studying dynamic interaction topologies comes from the following observations. In real-world applications, the interaction topology between vehicles will likely be dynamic. For instance, communication links between vehicles may be unreliable due to disturbances, or they may be subject to communication range limitations. Alternatively, if information is exchanged via direct sensing, the visible neighbors of a vehicle will likely change over time.

Fig. 2 shows six different time-invariant interaction topologies associated with Cases I-VI. In particular, Cases I corresponds to an undirected connected graph, Case II corresponds to a directed tree (i.e., leader-follower graph), Case III corresponds to an undirected graph having separated subgroups, Case IV corresponds to a directed graph having multiple leaders (i.e., vehicles 1 and 3), Case V corresponds to a cyclic pursuit graph, and Case VI corresponds to a general directed graph containing a directed spanning tree.

Fig. 3 shows the experimental results of the rendezvous captured by the overhead camera for Cases I-VI, where the circles denote the initial positions of the robots, and the dots denote the trajectories of the robots updated at each sample period of the vision system. The trajectories of the robots are captured between $t = 0$ and $t = t_f$ seconds.

We can see from Fig. 3 that the four robots rendezvous in all cases except Cases III and IV where only a subgroup rendezvous. This experimental result is consistent with the first argument of Lemma 3.1 since only the interaction graphs of Cases III and IV do not contain a directed spanning tree. In addition, we can see from Fig. 3 that the final rendezvous in Cases I and V are weighted averages of all four robots' initial positions. As a comparison, the final rendezvous in Cases II and VI are, respectively, robot 1's initial position and a weighted average of the initial positions of robots 1, 2, and 3. This experimental result is consistent with the second argument of Lemma 3.1 since in Cases I and V each robot has a directed path to every other robot, in Case II only robot 1 has a directed path to all of the other robots, and in Case VI every robot except robot 4 has a directed path to all the other robots. Furthermore, by comparing Cases I, II, V, and VI, we can see that in Case I the four robots rendezvous the fastest while in Case V the four robots rendezvous the slowest. This experimental result is also consistent with the theoretical result. The fact that consensus under a cyclic pursuit topology converges slower than under an undirected connected topology can be seen by comparing the eigenvalues of L given by (4) in two cases. Let λ_2 be the eigenvalue of L whose real part is the second smallest, which characterizes the convergence speed of the consensus algorithms in some sense. It is straightforward to show that $\lambda_2 = 2$ in Case I while $\lambda_2 = 1 + i$ in Case V assuming $k_{ij} = 1$ in (4).

To test rendezvous in the case of switching topologies, we assume that the interaction topologies for the four robots switch randomly from the set $\bar{\mathcal{G}}_s = \{\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3, \mathcal{G}_4, \mathcal{G}_5\}$ as

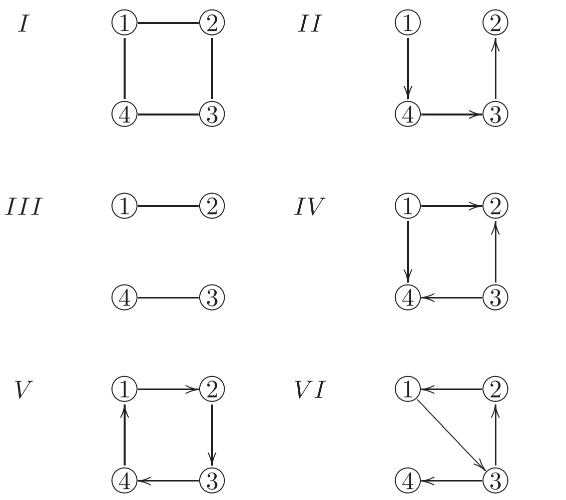


Fig. 2. Interaction topologies between four robots for the rendezvous application.

shown in Fig. 4. Note that each directed graph in $\mathcal{G}_s$ does not have a directed spanning tree but that the union of these graphs denoted by $\mathcal{G}_u$ does have a directed spanning tree as shown in Fig. 4. As the switching between graphs in $\bar{\mathcal{G}}_s$ is random, the condition for consensus in Lemma 3.2 is generically satisfied.

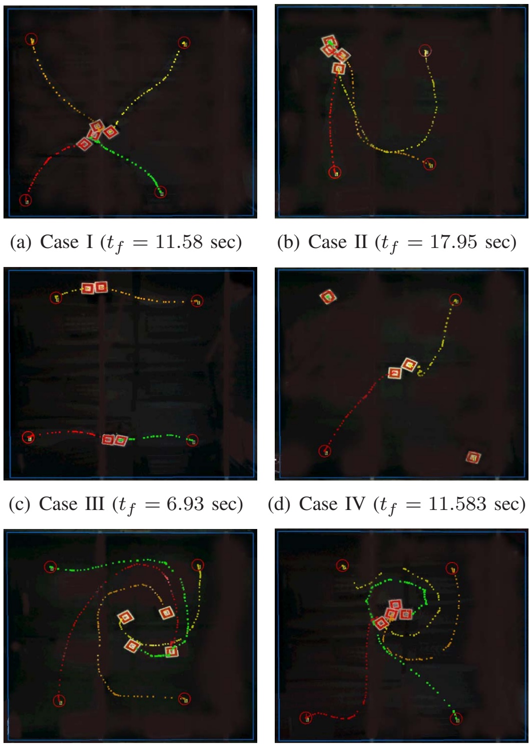
Fig. 5 (a) shows the experimental result of rendezvous when the interaction topologies switch randomly from set $\bar{\mathcal{G}}_s$ with switching periods randomly chosen between 1 and 3 seconds while Fig. 5 (b) shows the experimental result under a time-invariant interaction topology $\mathcal{G}_u$. Note that the four robots rendezvous even in the case that the topologies switch randomly with time, which is consistent with the argument of Lemma 3.2. By comparing Figs. 5 (a) and (b), we can see that convergence in the case of switching topologies is slower than in the time-invariant case. In addition, the switching topologies result in sudden drastic changes in robot directions as shown in Fig. 5 (a).

2) Axial Alignment: For the axial alignment application, we study the case that four robots are evenly distributed along a straight line under a time-invariant interaction topology.

Fig. 6 shows the undirected interaction topology between the four robots. Fig. 7 shows the experimental result of the axial alignment. As shown in Fig. 7, although each robot starts at arbitrary initial positions, their final positions are evenly distributed along a horizontal line with a separation distance of approximately 24 centimeters. The experimental result is consistent with the argument of Corollary 3.1.

3) Formation Maneuvering: For the formation maneuvering application, we study the case that the five robots maintain a desired V-shape formation geometry under a time-invariant information exchange topology.

Fig. 8 shows the interaction topology between five robots. Note that the interaction topology has a directed spanning tree in the sense that robot 1 has a directed path to all of the other robots. Compared to the traditional leader-follower approach, where information only flows from leaders to followers, information flows from followers to leaders as a form of feedback are naturally taken into account in the consensus algorithms. As shown in Fig. 8, information also flows from robot 5 to robot 3 and from robot 4 to robot 2. Fig. 9 shows the experimental result of formation maneuvering. As shown in Fig. 9, the five robots maintain the desired V-shape formation with desired x-axis and y-axis separation distances of 24 centimeters and move as a whole.



(a) Case I ($t_f = 11.58$ sec)

(b) Case II ($t_f = 17.95$ sec)

(c) Case III ($t_f = 6.93$ sec)

(d) Case IV ($t_f = 11.583$ sec)

(e) Case V ($t_f = 19.80$ sec)

(f) Case VI ($t_f = 18.89$ sec)

Fig. 3. Experimental results of rendezvous for Cases I-VI.

4) Lessons Learned: In our attempt to use the MASnet platform to validate cooperative control applications based on consensus algorithms, several practical challenges are encountered.

In terms of hardware, the dynamic characteristics of our MASmote robots impose performance limitation not accounted for in highly idealized simulations. One challenge comes from the low-level PID controllers for position control. Another challenge comes from the nonholonomic constraints of the robots, which cause the loss of orientation information for the robots with the use of low-level position controllers.

In terms of software, the computationally complex task of processing the image, finding MASmote markers, and extracting position and orientation information introduce a delay of 0.1 to 0.2 seconds between image capture and position and orientation information broadcast. When the MASmotes are moving slowly, the delay has little effect but at full speed the difference between the actual and the

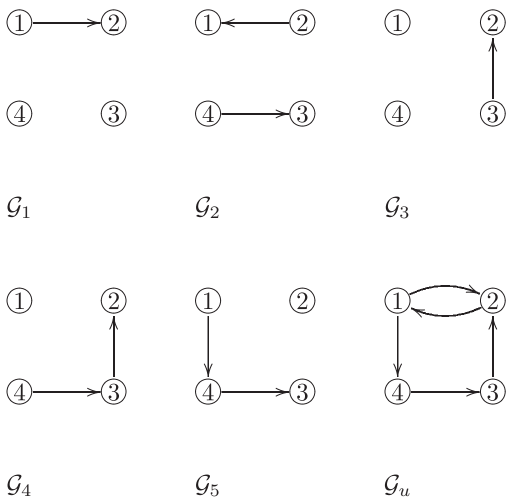
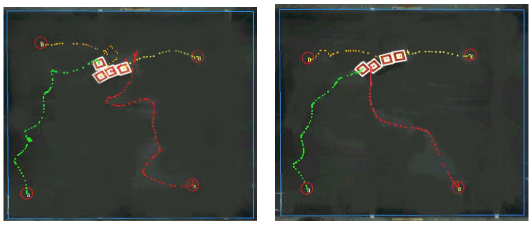


Fig. 4. Switching information topologies $\mathcal{G}_1$-$\mathcal{G}_5$ and their union $\mathcal{G}_u$ for rendezvous.



(a) Rendezvous with topologies randomly switching from $\bar{\mathcal{G}}_s = \{\mathcal{G}_1, \ldots, \mathcal{G}_5\}$ ($t_f = 45$ sec)

(b) Rendezvous with a time-invariant topology $\mathcal{G}_u$ ($t_f = 24$ sec)

Fig. 5. Experimental results of rendezvous with topologies randomly switching from $\bar{\mathcal{G}}_s$ vs a time-invariant topology $\mathcal{G}_u$.

broadcast position can be quite large. This is most noticeable when a MASmote is rotating.

Despite those practical limitations existing in the MASnet platform, the three cooperative control tasks based on consensus algorithms function well even under frequent pseudo-GPS packet loss (roughly $2\% - 5\%$) and average position measurement error of 13.2 mm. These results demonstrate the robustness of the consensus algorithms and their effectiveness in designing distributed cooperative control schemes. Consensus algorithms provide a promising method for distributed multi-vehicle cooperative control even in presence of robot physical limitations, packet loss, information delay, etc.

V. CONCLUSION AND FUTURE WORK

We have applied consensus algorithms to three cooperative control problems including rendezvous, axial alignment, and formation maneuvering. The experimental results of the three applications on our MASnet platform have demonstrated the effectiveness and robustness of the consensus algorithms to cooperative control. The lessons learned from implementing the consensus algorithms on the MASnet platform have

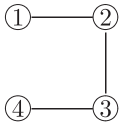


Fig. 6. Interaction topology for axial alignment.

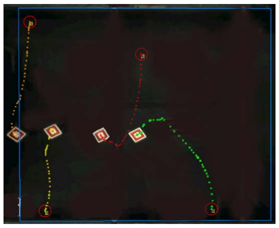


Fig. 7. Experimental result of axial alignment.

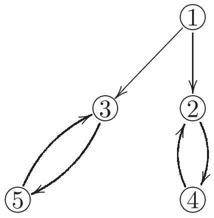


Fig. 8. Interaction topology for formation maneuvering.

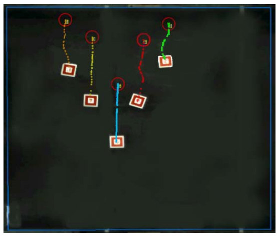


Fig. 9. Experimental result of formation maneuvering.

been summarized. Future work will introduce a collision avoidance mechanism for formation maneuvering of the robots in the experiment. The videos of the experiments can be found at `http://mechatronics.ece.usu.edu/mas-net/movies/0605/`.

REFERENCES

[1] W. Ren, R. W. Beard, and E. 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.

[2] 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.

[3] 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.

[4] 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.

[5] A. Regmi, R. Sandoval, R. Byrne, H. Tanner, and C. Abdallah, "Experimental implementation of flocking algorithms in wheeled mobile robots," in *Proceedings of the American Control Conference*, Portland, OR, 2005, pp. 4917–4922.

[6] J. A. Marshall, T. Fung, M. E. Broucke, G. M. T. D'Eleuterio, and B. A. Francis, "Experiments in multirobot coordination," *Robotics and Autonomous Systems*, vol. 54, no. 3, pp. 265–275, 2006.

[7] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 1985.

[8] F. R. K. Chung, *Spectral Graph Theory*. American Mathematical Society, 1997.

[9] 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. Springer, 2005, pp. 171–188.

[10] W. Bourgeous, L. Ma, P. Chen, Z. Song, and Y. Chen, "Simple and efficient extrinsic camera calibration based on a rational model," in *Proceedings of the IEEE Conference on Mechatronics and Automation*, Luoyang, China, June 2006.