Available online at www.sciencedirect.com
ScienceDirect
Robotics and Autonomous Systems 56 (2008) 324–333
Robotics and Autonomous Systems
www.elsevier.com/locate/robot

Distributed coordination architecture for multi-robot formation control

Wei Ren*, Nathan Sorensen

Department of Electrical and Computer Engineering, Utah State University, Logan, UT 84322-4120, USA

Received 23 October 2006; received in revised form 17 August 2007; accepted 17 August 2007
Available online 31 August 2007

Abstract

In the exploration and implementation of formation control strategies, communication range and bandwidth limitations form a barrier to large scale formation control applications. The limitations of current formation control strategies involving a leader–follower approach and a consensus-based approach with fully available group trajectory information are explored. A unified, distributed formation control architecture that accommodates an arbitrary number of group leaders and arbitrary information flow among vehicles is proposed. The architecture requires only local neighbor-to-neighbor information exchange. In particular, an extended consensus algorithm is applied on the group level to estimate the time-varying group trajectory information in a distributed manner. Based on the estimated group trajectory information, a consensus-based distributed formation control strategy is then applied for vehicle level control. The proposed architecture is experimentally implemented and validated on a multi-robot platform under local neighbor-to-neighbor information exchange with a single or multiple leaders involved.

Keywords: Formation control; Consensus algorithm; Coordination; Multi-robot systems

1. Introduction

In the field of cooperative control, approaches for achieving of formation maintenance among multiple vehicles have received significant attention. Given the limitations of communication bandwidth and communication range in many applications, the need for distributed algorithms that require only local neighbor-to-neighbor information exchange is apparent.

A typical leader–follower formation control approach (e.g., [1]) assumes only one group leader within the team. In this case, only the group leader has the knowledge of group trajectory information, which is either preprogrammed in the group leader or provided to the group leader by an external source. The formation is then built on the reaction of the other group members to the motion of the group leader. The fact that only a single group leader is involved in the team implies that the leader–follower approach is simple to implement and understand, and the requirement on communication bandwidth is reduced. This is, however, a single point of massive failure type system because the loss of the group leader causes the entire group to fail. Another issue with the typical leader–follower approach is the lack of inter-vehicle information feedback throughout the group. For example, feedback from the followers is not used by the leader so the formation can become disjoint and followers can be left behind if they are not able to track the motion of the leader accurately.

In order to overcome this type of single point of failure tendency, much research has been focusing on decentralized or distributed cooperative control strategies where vehicle control laws are coupled and each vehicle makes its own decision according to the states of its neighbors (e.g., [2–17]). This allows the group to continue on to achieve an objective even in the presence of failure of any group member.

Among the decentralized or distributed cooperative control strategies, consensus algorithms (e.g., [6–12,15]) focus on driving the information states of all vehicles to a common value. For formation stabilization with a static formation centroid, if each vehicle in a group can reach consensus on the center point of the desired formation and specify a corresponding desired deviation from the center point, then vehicle formations can be achieved. To apply consensus algorithms to achieve formation maneuvering with a time-varying formation centroid trajectory, either the common formation velocity for the group or the desired group trajectory is assumed to be known by

* Corresponding author. Tel.: +1 435 797 2831; fax: +1 435 797 3054.
E-mail address: wren@engineering.usu.edu (W. Ren).

0921-8890/$ - see front matter
doi:10.1016/j.robot.2007.08.005

each vehicle in the group as in [3,5,17,18]. In particular, [5] assumes that a sequence of constant, desired formation centroid states are preprogrammed on each vehicle. However, this approach cannot account for dynamically changing formation centroid states in response to dynamically changing situational awareness. While a flocking behavior is achieved in [4,19] when no vehicle has the knowledge of group formation velocity, an accurate formation geometry is not specified. In this paper, we focus on applications that require accurate formation geometry maintenance with desired group trajectory information involved.

The requirement that each vehicle have the knowledge of the desired group trajectory may not be realistic for many applications. For example, communication bandwidth and range limitations may prevent each vehicle in the group having access to the group trajectory information. Also, to increase stealth and flexibility, only a portion of the vehicles in the team may be provided with the desired group trajectory information. In addition, it is also possible that only a portion of the vehicles are able to detect a target or dangerous source at a certain time instant, and those vehicles in turn serve as the group leaders to guide the behaviors of the other group members.

Given the strength of the consensus algorithms for formation control with coupling involved between neighboring vehicles and the effectiveness of a traditional leader–follower approach when group trajectory information is limited in the formation, integrating the two approaches yields the strength of both strategies. Bandwidth limitations for the group can be handled in limiting the amount of group trajectory information availability within the group, while robustness is achieved with distributed nature of the consensus algorithms.

The main contributions of the current paper are twofold. First, we propose a unified, distributed formation control architecture that accommodates an arbitrary number of group leaders and allows for arbitrary information flow among vehicles without adding complexity to the control law design and analysis. In particular, an extended consensus algorithm is applied on the group level to estimate the time-varying group trajectory information in a distributed manner. Based on the estimated group trajectory information, a consensus-based distributed formation control strategy is then applied for vehicle level control. Second, the proposed formation control architecture is experimentally implemented and validated on a multi-robot platform and the results are discussed. It is worthwhile to mention that although various strategies for decentralized or distributed formation control have been studied in the literature, few have been systematically verified on experimental platforms. A preliminary version of the work has been presented at the 2007 American Control Conference [20].

2. Background and preliminaries

2.1. Graph theory notations

It is natural to model information exchange among vehicles by directed or 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 $\mathcal{E} \in \mathcal{N} \times \mathcal{N}$ is a set of ordered pairs of nodes, called edges. An *edge* (i, j) in a digraph 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 digraph, where an edge (i, j) in the undirected graph corresponds to edges (i, j) and (j, i) in the digraph. If there is an edge from node i to node j in a digraph, then i is the *parent node*, and j is the *child node*. A *directed path* is a sequence of edges of the form $(v_{i_1}, v_{i_2}), (v_{i_2}, v_{i_3}), \ldots$, where $v_{i_j} \in \mathcal{N}$, in a digraph. An *undirected path* in an undirected graph is defined analogously. In a digraph, a *cycle* is a directed path that starts and ends at the same node. A digraph is *strongly connected* if there is a directed path from every node to every other node. An undirected graph is *connected* if there is a path between any distinct pair of nodes. A *directed tree* is a digraph, 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 other node. Note that in a directed tree, each edge has a natural orientation away from the root, and no cycle exists. In the case of undirected graphs, a *tree* is a graph in which every pair of nodes is connected by exactly one path. A *directed spanning tree* of a digraph 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 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 at least one node having a directed path to all of 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 tree is a weaker condition than being strongly connected.

The *adjacency matrix* $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ of a digraph is defined as $a_{ii} = 0$ and $a_{ij} > 0$ if $(j, i) \in \mathcal{E}$ where $i \neq j$. The adjacency matrix of an undirected graph is defined analogously except that $a_{ij} = a_{ji}, \forall i \neq j$, since $(j, i) \in \mathcal{E}$ implies $(i, j) \in \mathcal{E}$. Let matrix $L = [\ell_{ij}] \in \mathbb{R}^{n\times n}$ 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:

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

For an undirected graph, L is called the *Laplacian matrix* [21], which is symmetric positive semi-definite. However, L for a digraph does not have this property.

Let $\mathbf{1}$ and $\mathbf{0}$ denote the $n \times 1$ column vector of all ones and all zeros respectively. Let I_n denote the $n \times n$ identity matrix. Let $M_n(\mathbb{R})$ represent the set of all $n \times n$ real matrices. Given a matrix $S = [s_{ij}] \in M_n(\mathbb{R})$, the digraph of S, denoted by $\Gamma(S)$, is the digraph on n nodes v_i, $i \in \{1, 2, \cdots, n\}$, such that there is an edge in $\Gamma(S)$ from v_j to v_i if and only if $s_{ij} \neq 0$ (cf. [22]).

2.2. Consensus algorithms

Consider vehicles with single-integrator dynamics given by

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

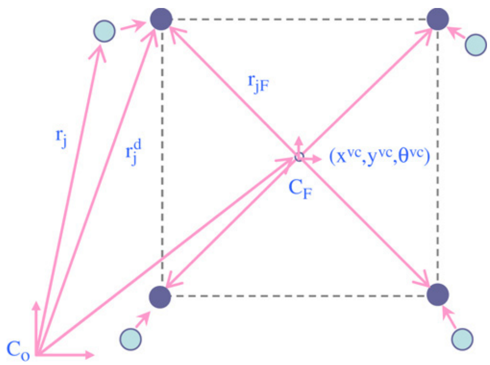


Fig. 1. A formation composed of four vehicles with a known virtual center.

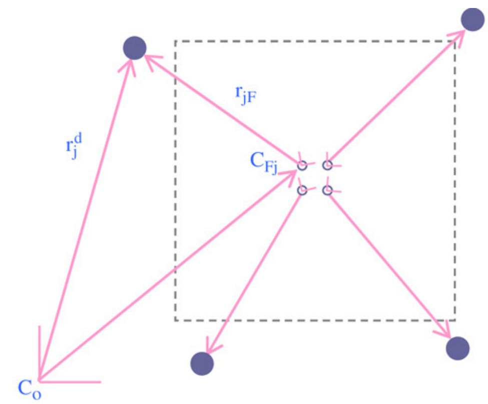


Fig. 2. A formation composed of four vehicles with inconsistent understanding of the virtual coordinate frame.

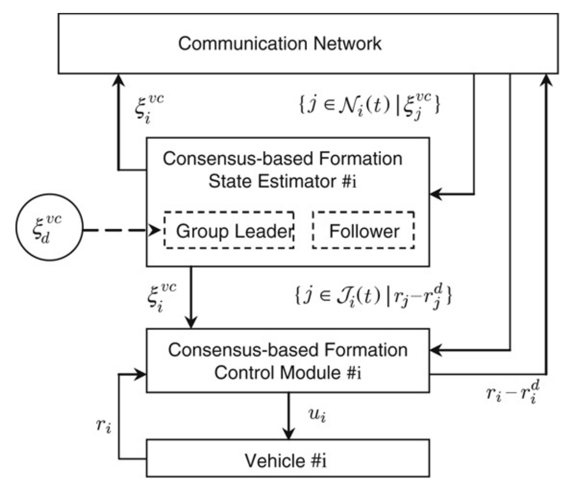


Fig. 3. A unified, distributed architecture for formation control.

where $r_i \in \mathbb{R}^m$ is the information state of the ith vehicle and $u_i \in \mathbb{R}^m$ is the control input to the ith vehicle. A consensus algorithm is proposed in [2,6–8,10] as

$$u_i = -\sum_{j=1}^{n} g_{ij} k_{ij} (r_i - r_j), \quad i = 1, \dots, n, \tag{3}$$

where k_{ij} is a positive weight, and $g_{ij} = 1$ if information flows from vehicle j to vehicle i and 0 otherwise. The objective of (3) is to drive the information state of each vehicle toward the information states of its local neighbors. For (3), *consensus* is reached asymptotically among the n vehicles if $r_i(t) \to r_j(t)$, $\forall i \neq j$, as $t \to \infty$ for all $r_i(0)$.

3. Distributed formation control architecture

In this section, we propose a distributed formation control architecture that accommodates an arbitrary number of group leaders and ensures accurate formation maintenance through information coupling between local neighbors.

One solution to formation control is the virtual structure approach [23,24]. Similar approaches include the action reference scheme [25] and the virtual leader approach [26,27]. The basic idea is to specify a virtual leader or a virtual coordinate frame located at the virtual center of the formation as a reference for the whole group such that each vehicle's desired states can be defined relative to the virtual leader or the virtual coordinate frame. As a result, single vehicle path planning and trajectory generation techniques can be employed for the virtual leader or the virtual coordinate frame while trajectory tracking strategies can be employed for each vehicle.

Fig. 1 shows an illustrative example of the virtual structure approach with a formation composed of four vehicles with planar motions, where C_o represents the inertial frame and C_F represents a virtual coordinate frame located at a virtual center (x^{vc}, y^{vc}) with an orientation θ^{vc} relative to C_o. In Fig. 1, $r_j = [x_j, y_j]^{\mathrm{T}}$ and $r_j^d = [x_j^d, y_j^d]^{\mathrm{T}}$ represent, respectively, the jth vehicle's actual and desired position, and $r_{jF} = [x_{jF}, y_{jF}]^{\mathrm{T}}$ represent the desired deviation of the jth vehicle relative to C_F, where

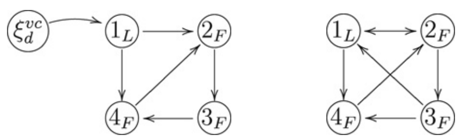
$$\begin{bmatrix} x_j^d(t) \\ y_j^d(t) \end{bmatrix} = \begin{bmatrix} x^{vc}(t) \\ y^{vc}(t) \end{bmatrix} + \begin{bmatrix} \cos(\theta^{vc}(t)) & -\sin(\theta^{vc}(t)) \\ \sin(\theta^{vc}(t)) & \cos(\theta^{vc}(t)) \end{bmatrix} \times \begin{bmatrix} x_{jF}(t) \\ y_{jF}(t) \end{bmatrix}.$$

If each vehicle can track its desired position accurately, then the desired formation shape can be preserved accurately. Note that Fig. 1 relies on the assumption that each vehicle knows the state of the virtual coordinate frame (i.e., virtual center position and orientation), denoted as $\xi^{vc} = [x^{vc}, y^{vc}, \theta^{vc}]^{\mathrm{T}}$, called *formation state* hereafter. However, this assumption is rather restrictive as described in Section 1.

When each vehicle has inconsistent understanding or knowledge of ξ^{vc} due to dynamically changing situational awareness or unreliable/limited information exchange, the desired formation geometry cannot be maintained as shown in Fig. 2, where C_{Fj} represents the jth vehicle's understanding of the virtual coordinate frame with state denoted as $\xi_j^{vc} = [x_j^{vc}, y_j^{vc}, \theta_j^{vc}]^{\mathrm{T}}$. While it may be intuitive to apply the consensus algorithm Eq. (3) to guarantee that $\xi_i^{vc} \to \xi_j^{vc}$, this approach is only applicable for problems where the virtual center and orientation are constant (e.g., formation stabilization).

Next we propose a unified, distributed architecture for formation control as shown in Fig. 3. The hierarchical architecture consists of three layers: consensus-based formation

Fig. 4. Multi-robot experimental platform at USU.



(a) Estimation level topology.

(b) Vehicle control level topology.

Fig. 5. Information-exchange topologies with a single group leader and three followers.

state estimator, consensus-based formation control module, and physical vehicle. In Fig. 3, $\mathcal{N}_i(t)$ and $\mathcal{J}_i(t)$ denote, respectively, the set of vehicles whose formation state estimates and position tracking errors are available to vehicle i at time t. The objective of the formation state estimator is to drive ξ_i^{vc} to $\xi_d^{vc} = [x_d^{vc}, y_d^{vc}, \theta_d^{vc}]^{\mathrm{T}}$, which represents the desired state of the virtual coordinate frame available only to the group leaders. The local control law u_i for each vehicle is based on its formation state estimate and the position tracking errors of its local neighbors.

On the formation state estimation level, each vehicle estimates the state of the virtual coordinate frame via an extended consensus algorithm as

$$\dot{\xi}_i^{vc} = \frac{\dot{\xi}_d^{vc} - \gamma(\xi_i^{vc} - \xi_d^{vc}) + \sum_{j=1}^{n} g_{ij}^{vc}[\dot{\xi}_j^{vc} - \gamma(\xi_i^{vc} - \xi_j^{vc})]}{1 + \sum_{j=1}^{n} g_{ij}^{vc}}, i \in \mathcal{L}$$

$$\dot{\xi}_i^{vc} = \frac{\sum_{j=1}^{n} g_{ij}^{vc}[\dot{\xi}_j^{vc} - \gamma(\xi_i^{vc} - \xi_j^{vc})]}{\sum_{j=1}^{n} g_{ij}^{vc}}, \quad i \notin \mathcal{L}, \tag{4}$$

where $\mathcal{L}$ denotes the set of group leaders that have knowledge of ξ_d^{vc}, $g_{ij}^{vc} = 1$ if vehicle j's formation state estimate is available to vehicle i and 0 otherwise, and $\gamma > 0$. Note that only the group leaders have direct access to ξ_d^{vc}, which may be time varying, and the number of the group leaders can be any number from 1 to n. We have the following theorem for convergence analysis of (4).

Theorem 3.1. *Let $G^{vc} = [g_{ij}^{vc}] \in \mathbb{R}^{(n+1)\times(n+1)}$ be the adjacency matrix, where g_{ij}^{vc}, $\forall i, j \in \{1, \ldots, n\}$, are defined after (4), $g_{i(n+1)}^{vc} = 1$ if $i \in \mathcal{L}$ and 0 otherwise, and $g_{(n+1)k}^{vc} = 0$, $\forall k \in \{1, \ldots, n+1\}$. Then the estimation algorithm (4) guarantees that $\xi_i^{vc} \to \xi_d^{vc}$, $\forall i$, asymptotically if and only if the graph of G^{vc} has a directed spanning tree.*

Proof. Let $\xi_{n+1}^{vc} \equiv \xi_d^{vc}$, then the two equations in (4) can be rewritten as

$$\dot{\xi}_i^{vc} = \frac{1}{\sum_{j=1}^{n+1} g_{ij}^{vc}} \sum_{j=1}^{n+1} g_{ij}^{vc}[\dot{\xi}_j^{vc} - \gamma(\xi_i^{vc} - \xi_j^{vc})], \quad i = 1, \ldots, n.$$

After some manipulation, we get that

$$\sum_{j=1}^{n+1} g_{ij}^{vc}(\dot{\xi}_i^{vc} - \dot{\xi}_j^{vc}) = -\gamma \sum_{j=1}^{n+1} g_{ij}^{vc}(\xi_i^{vc} - \xi_j^{vc}), \quad i = 1, \ldots, n,$$

which implies that

$$\sum_{j=1}^{n+1} g_{ij}^{vc}(\xi_i^{vc} - \xi_j^{vc}) \to 0, \quad i = 1, \ldots, n. \tag{5}$$

By adding a dummy equation $0 = 0$, $i = n+1$, to (5), we can rewrite (5) in matrix form as $(L_{n+1} \otimes I_3)\xi^{vc} \to 0$, where $\xi^{vc} = [\xi_1^{vc\mathrm{T}}, \ldots, \xi_{n+1}^{vc\mathrm{T}}]^{\mathrm{T}}$, $L_{n+1} = [\ell_{ij}] \in \mathbb{R}^{(n+1)\times(n+1)}$ is defined as $\ell_{ii} = \sum_{j\neq i} g_{ij}$, $\ell_{ij} = -g_{ij}$, $\forall i \in \{1, \ldots, n\}$, $\forall j \in \{1, \ldots, n+1\}$, and $\ell_{(n+1)i} = 0$, $\forall i$. Note that all of the entries of the $(n+1)$th row of L_{n+1} are zero. Also note that L_{n+1} satisfies the property (1) and the graph of L is equivalent to that of G^{vc}, which has a directed spanning tree. Therefore, from [28], we know that $\xi_i^{vc} \to \xi_j^{vc}$, $\forall i, j \in \{1, \ldots, n+1\}$, if and only if the graph of G^{vc} has a directed spanning tree. Equivalently, it follows that $\xi_i^{vc} \to \xi_d^{vc}$, $\forall i$, since $\xi_{n+1}^{vc} \equiv \xi_d^{vc}$. ■

The proof to Theorem 3.1 is from [29]. We have included the proof for completeness. Compared to (3), where consensus is reached on a constant value equal to the weighted average of the initial information states of all vehicles, the estimation algorithm (4) reaches consensus on any time-varying, desired formation state.

On the vehicle control level, we apply the extended consensus algorithm as

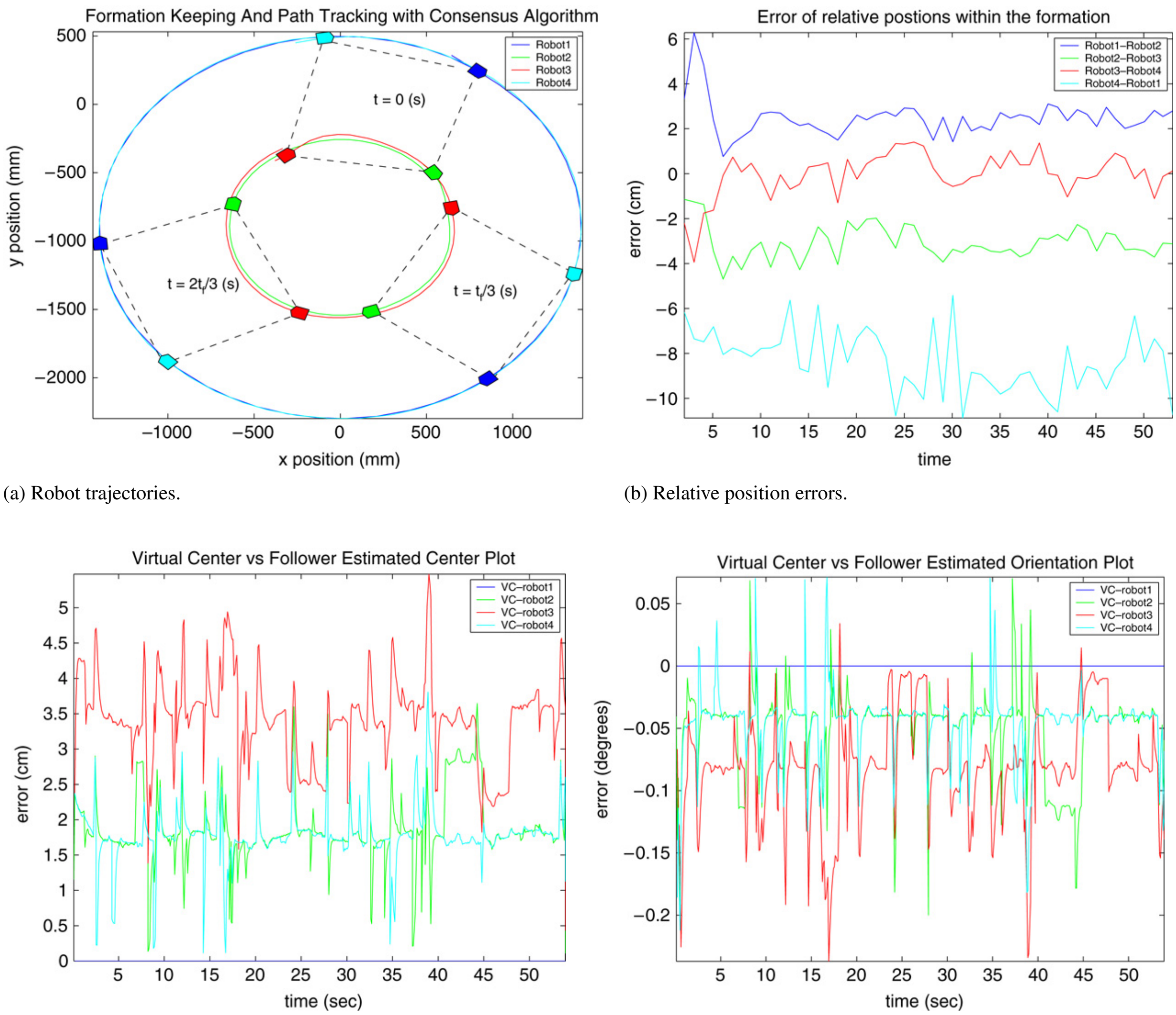
$$u_i = \dot{r}_i^d - \alpha_i(r_i - r_i^d) - \sum_{j=1}^{n} g_{ij}k_{ij}[(r_i - r_i^d) - (r_j - r_j^d)], \tag{6}$$

where $\alpha_i > 0$, $k_{ij} > 0$, $g_{ij} = 1$ if information flows from vehicle j to vehicle i and 0 otherwise,[1] and $r_i^d = [x_i^d, y_i^d]^{\mathrm{T}}$ with

$$\begin{bmatrix} x_i^d \\ y_i^d \end{bmatrix} = \begin{bmatrix} x_i^{vc} \\ y_i^{vc} \end{bmatrix} + \begin{bmatrix} \cos(\theta_i^{vc}) & -\sin(\theta_i^{vc}) \\ \sin(\theta_i^{vc}) & \cos(\theta_i^{vc}) \end{bmatrix} \begin{bmatrix} x_{iF} \\ y_{iF} \end{bmatrix}.$$

With (6) and (2) can be written in matrix form as $\dot{\tilde{r}} = -[(L + \Gamma) \otimes I_2]\tilde{r}$, where L is given as $\ell_{ii} = \sum_{j\neq i} g_{ij}k_{ij}$ and $\ell_{ij} =$

[1] Note that the information-exchange topology on the formation state estimation level defined by g_{ij}^{vc} may be different from the information-exchange topology on the vehicle control level defined by g_{ij}.



(a) Robot trajectories.

(b) Relative position errors.

(c) Virtual center position estimation errors.

(d) Virtual center orientation estimation errors.

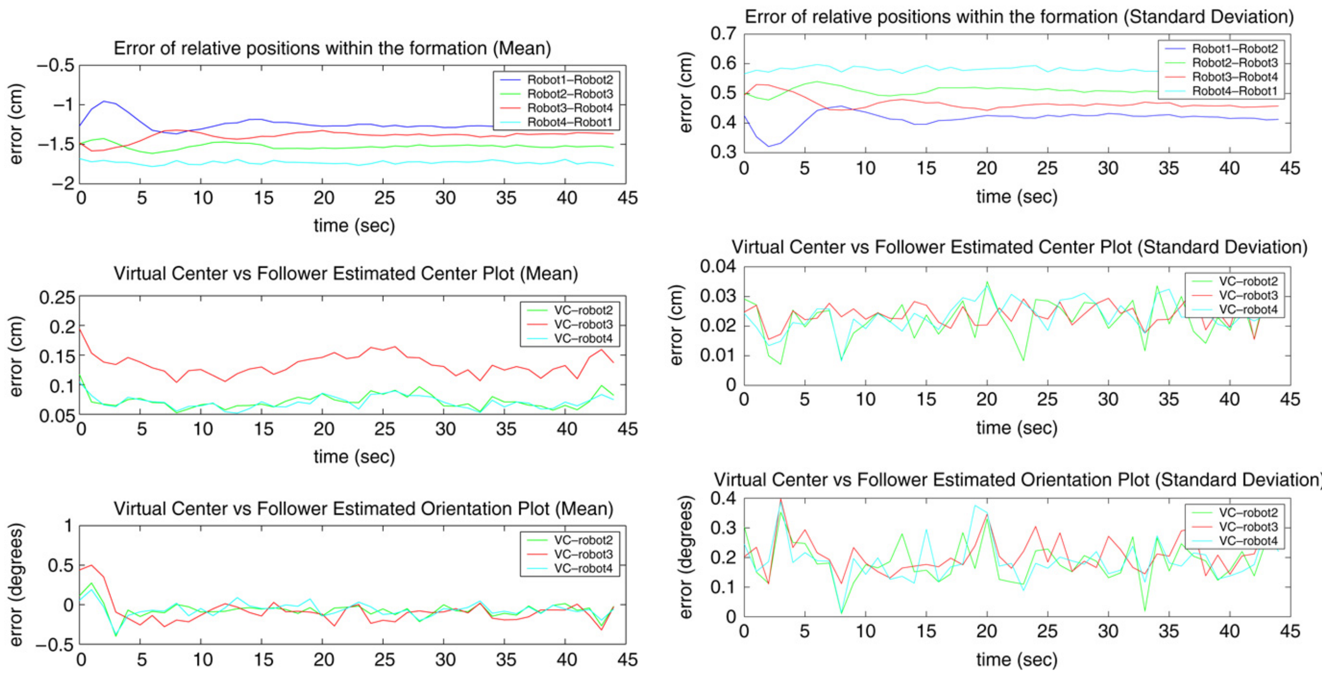
Fig. 6. Experimental result of a square formation with a single group leader and three followers where the virtual coordinate frame follows the circle trajectory.

$-g_{ij}k_{ij}$, $\forall i \neq j$, Γ is a diagonal matrix with α_i being the diagonal entries, and $\tilde{r} = [\tilde{r}_1^{\mathrm{T}}, \ldots, \tilde{r}_n^{\mathrm{T}}]^{\mathrm{T}}$ with $\tilde{r}_i = r_i - r_i^d$. Note that L satisfies the property (1). From Gershgorin disc theorem [22], it is straightforward to see that all eigenvalues of $-(L + \Gamma)$ have negative real parts. Therefore, under an arbitrary time-invariant information-exchange topology, it follows that $\tilde{r}(t) \to 0$ exponentially, that is, $r_i(t) \to r_i^d(t)$, $\forall i$, exponentially as $t \to \infty$. In other words, even a control law $u_i = \dot{r}_i^d - \alpha_i(r_i - r_i^d)$ is sufficient to guarantee that $r_i(t) \to r_i^d(t)$, $\forall i$, as $t \to \infty$. However, the coupling between neighboring vehicles denoted by the third term in (6) improves group robustness and reduces formation maintenance error.

Note that both (4) and (6) are distributed in the sense that only information exchange with local neighbors is required. The architecture denoted by Fig. 3 accommodates an arbitrary number of group leaders and arbitrary coupling among vehicles on both the formation state estimation level and the vehicle control level. The distributed nature of (4) and (6) ensures robustness of the group to failure of follower vehicles. The introduction of multiple group leaders complexify neither the control algorithms (4) and (6) nor their convergence analysis. Also the introduction of multiple group leaders reduces a single point failure existing in a team with a single group leader. In addition, with (4) and (6) each vehicle simply exchanges information with its local neighbors without the need to identify the group leaders. Approaches in [1,5,18] can be considered as special cases of the architecture denoted by Fig. 3. In particular, the approach in [1] corresponds to the case where only one group leader exists and each follower uses only the information from its unique parent node on the vehicle control level. The approach in [5] corresponds to the case where each vehicle behaves as a group leader and coupling on the vehicle control level occurs between one vehicle and its two adjacent neighbors. The approach in [18] corresponds to the case that each vehicle behaves as a group leader and coupling on the vehicle control level can occur between any local neighbors.

4. Experimental results on a multi-robot platform

In this section, we experimentally implement and validate the proposed distributed formation control architecture on a



(a) Mean. (b) Standard deviation.

Fig. 7. Means and standard deviations of ten experimental runs with a single group leader and three followers where the virtual coordinate frame follows the circle trajectory.

multi-robot platform. We conduct experiments with a single group leader and multiple group leaders, respectively.

4.1. Experimental platform and implementation

An AmigoBot and Pioneer 3-DX based multi-robot platform as shown in Fig. 4 has been developed at Utah State University (USU) for exploration of cooperative control strategies. The robots can communicate with each other through ethernet with TCP/IP protocols. The robots rely on encoder data for their position and orientation information.

In our experiments, we emulate limited inter-robot information exchange by simply disallowing the use of information obtained from certain members of the group although every robot can share information with every other robot. By doing so, we can test distributed cooperative control algorithms that involve only local neighbor-to-neighbor information exchange due to limited communication or sensing.

Let (r_{xi}, r_{yi}), θ_i, and (v_i, ω_i) denote the Cartesian position, orientation, and linear and angular velocity of the ith robot, respectively. The kinematic equations for the ith robot are

$$\dot{r}_{xi} = v_i \cos(\theta_i), \qquad \dot{r}_{yi} = v_i \sin(\theta_i), \qquad \dot{\theta}_i = \omega_i. \tag{7}$$

One challenge to implementing the consensus algorithm (6) on our platform is that (6) requires single-integrator dynamics. To focus on the main issue, we feedback linearize (7) for a fixed point off the center of the wheel axis denoted as (x_i, y_i), where $x_i = r_{xi} + d_i \cos(\theta_i)$ and $y_i = r_{yi} + d_i \sin(\theta_i)$ with $d_i = 0.15$ m. Letting

$$\begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) \\ -\dfrac{1}{d_i}\sin(\theta_i) & \dfrac{1}{d_i}\cos(\theta_i) \end{bmatrix} \begin{bmatrix} u_{xi} \\ u_{yi} \end{bmatrix},$$

gives

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} = \begin{bmatrix} u_{xi} \\ u_{yi} \end{bmatrix},$$

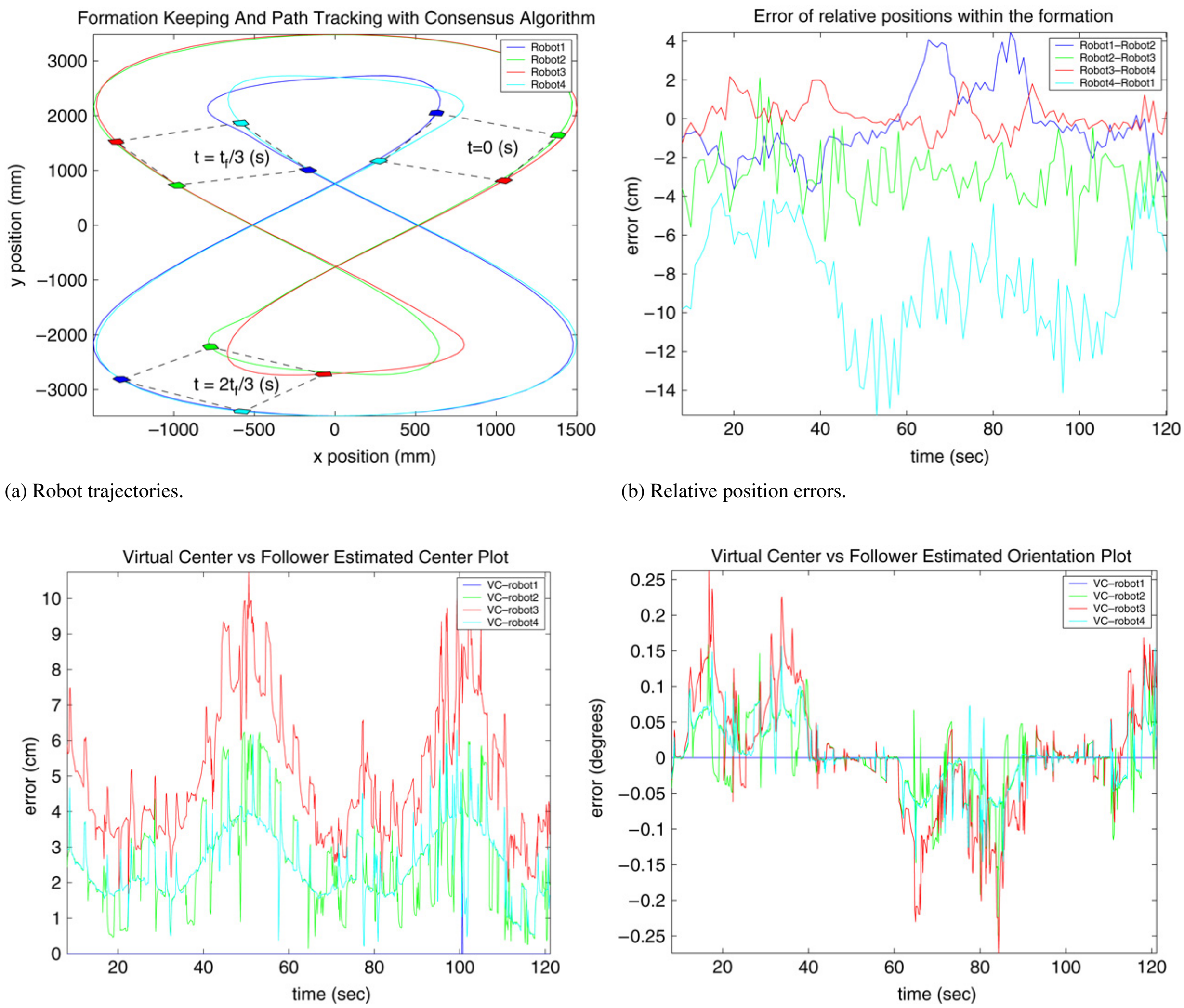
which is a simplified kinematic equation but is sufficient for the purpose of this paper.

In our experiments, a team of four AmigoBots are required to maintain a square formation with lateral length of 0.85 m. We let $x_{jF} = \ell_j \cos(\phi_j)$ and $y_{jF} = \ell_j \sin(\phi_j)$, where $\ell_j = 0.6$ m and $\phi_j = \pi - \frac{\pi}{4} j$ rad, $j = 1, \ldots, 4$. Two different trajectories will be tested. In the first case, the virtual coordinate frame located at the center of the square follows a circle of 0.9 m radius. In the second case, the virtual coordinate frame located at the center of the square follows a figure-8 trajectory. The virtual coordinate frame is initially located at $(x_d^{vc}(0), y_d^{vc}(0)) = (0, 0)$ m with an orientation $\theta_d^{vc}(0) = 0$ rad. Each robot applies (4) to estimate the formation state and (6) to derive u_{xi} and u_{yi}.

4.2. Formation control with a single group leader

In this subsection, we consider a single group leader with three followers. Fig. 5(a) shows the information-exchange topology on the formation state estimation level, where a subscript L denotes a group leader, a subscript F denotes a follower, and a link from node j to node i denotes that $g_{ij}^{vc} = 1$ in (4). Fig. 5(b) shows the information-exchange topology on the vehicle control level, where a link from node j to node i denotes that $g_{ij} = 1$ in (6).

Fig. 6 shows the experimental result with a single group leader and three followers where the virtual coordinate frame follows the circle trajectory. In particular, Fig. 6(a) shows the trajectories of the four robots at $t \in [0, t_f]$ sec and snapshots at



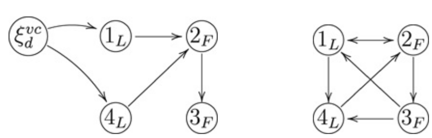
(a) Robot trajectories.

(b) Relative position errors.

(c) Virtual center position estimation errors.

(d) Virtual center orientation estimation errors.

Fig. 8. Experimental result of a square formation with a single group leader and three followers where the virtual coordinate frame follows the figure-8 trajectory.



(a) Estimation level topology.

(b) Vehicle control level topology.

Fig. 9. Information-exchange topologies with two group leaders and two followers.

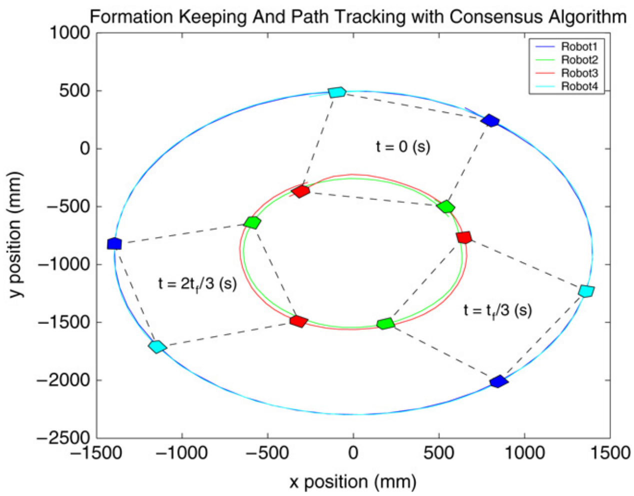
$t = 0, \frac{t_f}{3}, \frac{2t_f}{3}$ s, where t_f is the ending time of the experiment. Fig. 6(b) shows the relative position errors, defined as the difference between the desired and actual separation distance between the robots. Fig. 6(c) shows the virtual center position estimation errors, defined as $\sqrt{(x_d^{vc} - x_i^{vc})^2 + (y_d^{vc} - y_i^{vc})^2}$, where (x_d^{vc}, y_d^{vc}) is the desired virtual center position known by the group leader. Fig. 6(d) shows the virtual center orientation estimation errors, defined as $\theta_d^{vc} - \theta_i^{vc}$, where θ_d^{vc} is the desired virtual center orientation known by the group leader. Note that the group is able to travel in tight formation around the circle as shown in Fig. 6(a) with relative position errors between −12 and 7 cm as shown in Fig. 6(b). Also note that the virtual center position estimation errors are below 6 cm as shown in Fig. 6(c), and the virtual center orientation estimation errors are between −0.25 and 0.1 cm as shown in Fig. 6(d).

Fig. 7 shows the means and standard deviations of ten experimental runs with a single group leader and three followers where the virtual coordinate frame follows the circle trajectory. We can see that the experiments are consistent with each other.

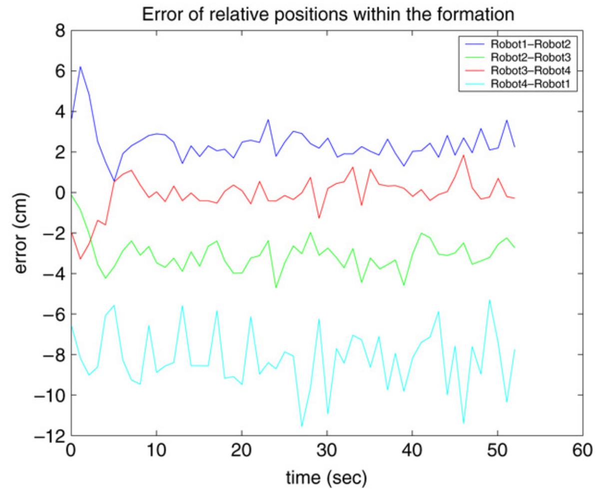
Fig. 8 shows the experimental result with a single group leader and three followers where the virtual coordinate frame follows the figure-8 trajectory. We can see that good performance is still achieved in this case.

4.3. Formation control with multiple group leaders

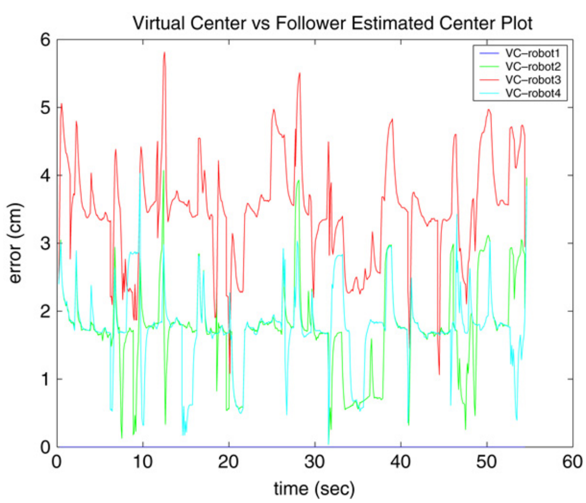
In this subsection, we consider two group leaders with two followers. Fig. 9(a) and (b) show the information-exchange topologies on the formation state estimation level and the vehicle control level, respectively.



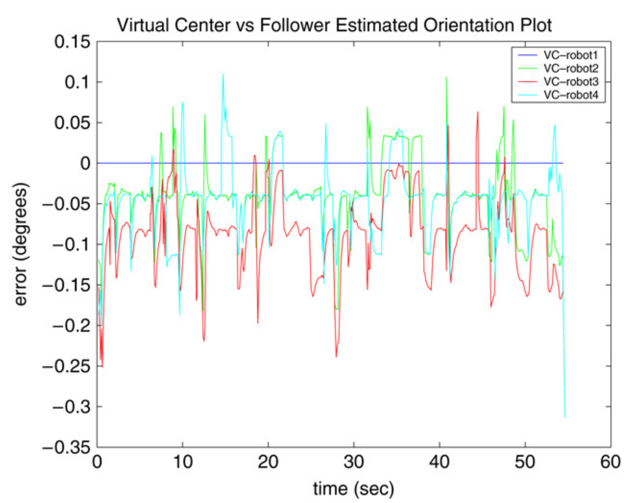
(a) Robot trajectories.



(b) Relative position errors.



(c) Virtual center position estimation errors.



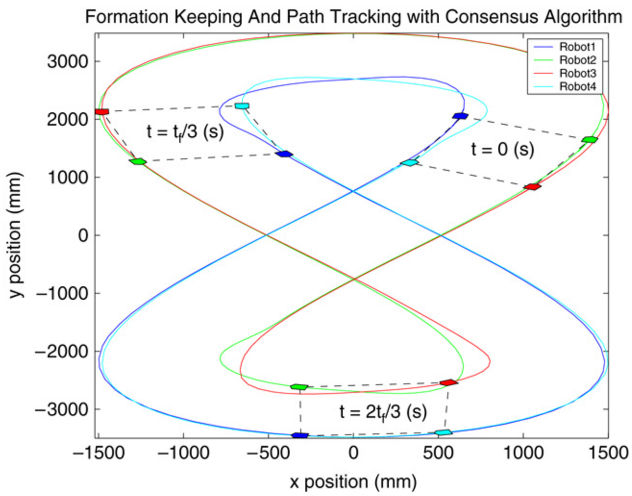
(d) Virtual center orientation estimation errors.

Fig. 10. Experimental result of a square formation with two group leaders and two followers where the virtual coordinate frame follows the circle trajectory.

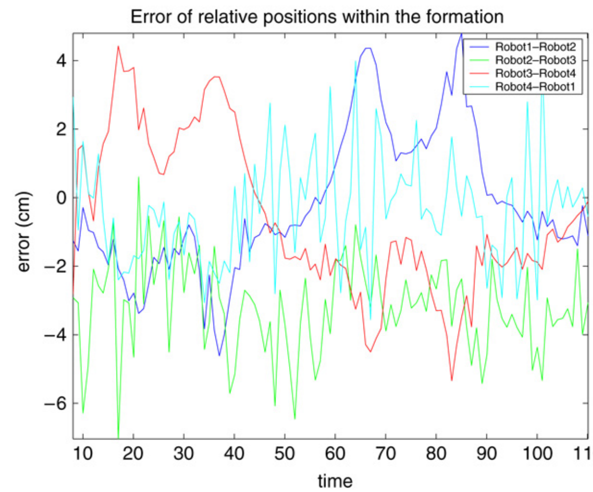
Figs. 10 and 11 show, respectively, the experimental results where the virtual coordinate frame follows, respectively, the circle trajectory and the figure-8 trajectory. Note that good formation maintenance performance is also achieved with multiple group leaders involved. The introduction of multiple group leaders increases the robustness of the whole group in the case of failure of a certain group leader and increases the formation maintenance performance.

5. Conclusion and future research

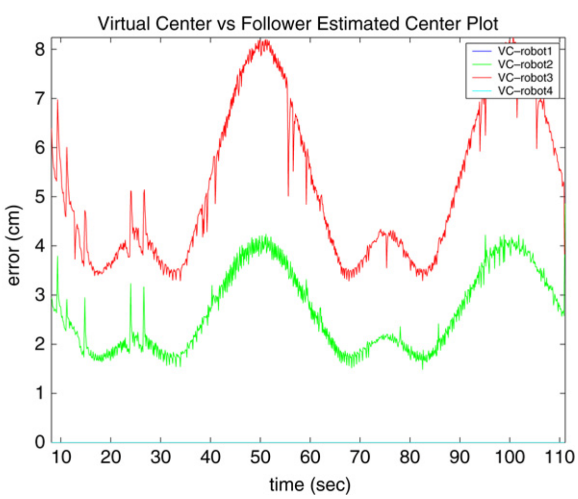
We have proposed a unified, distributed architecture to formation control that accommodates an arbitrary number of group leaders and allows for arbitrary inter-robot coupling on both the formation state estimation level and the vehicle control level. By using an extended consensus-based estimation algorithm, the vehicles come into agreement on the time-varying position and orientation of the virtual center. The vehicles then apply a consensus-based formation control algorithm to track their desired positions and preserve the formation geometry with their neighbors. The introduction of multiple group leaders and inter-robot coupling allows for individual group leader and follower failure in the presence of limited information exchange within the formation. By increasing the number of group leaders within the formation, robustness against a single point of failure is improved. Experimental results on a multi-robot platform have shown the effectiveness of the architecture. An experimental demonstration of the proposed algorithms on a team of four AmigoBots can be found at http://www.engineering.usu.edu/ece/faculty/wren/AmigoBots.htm. Future research will experimentally test the robustness of the architecture to group leader switching or failure and inter-robot switching information-exchange topologies. Research extending the experiments to a UAV platform will also be conducted. In addition, the issues of time delay, effects of robot dynamics, and data loss caused by the TCP/IP protocols also need to be addressed.



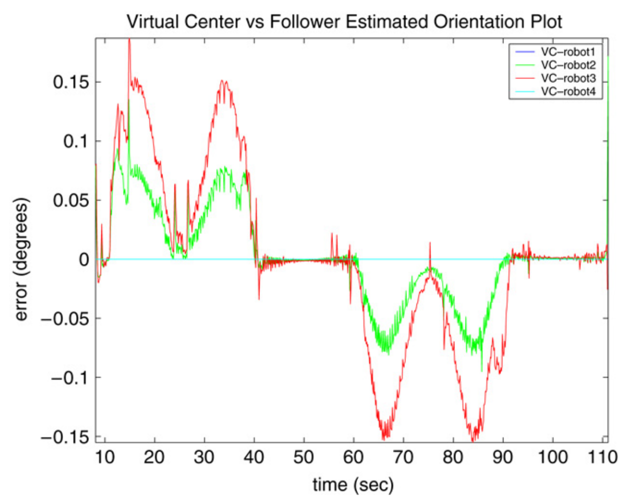
(a) Robot trajectories.



(b) Relative position errors.



(c) Virtual center position estimation errors.



(d) Virtual center orientation estimation errors.

Fig. 11. Experimental result of a square formation with two group leaders and two followers where the virtual coordinate frame follows the figure-8 trajectory.

Acknowledgements

This work was supported in part by the Utah Water Research Laboratory and the CURI Grant (2007–2008). The authors would like to gratefully acknowledge Larry Ballard and Yongcan Cao for their assistance in obtaining the experimental results.

References

[1] P.K.C. Wang, F.Y. Hadaegh, K. Lau, Synchronized formation rotation and attitude control of multiple free-flying spacecraft, Journal of Guidance, Control, and Dynamics 22 (1) (1999) 28–35.

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

[3] J.R. Lawton, R.W. Beard, B. Young, A decentralized approach to formation maneuvers, IEEE Transactions on Robotics and Automation 19 (6) (2003) 933–941.

[4] H.G. Tanner, A. Jadbabaie, G.J. Pappas, Flocking in fixed and switching networks, IEEE Transactions on Automatic Control 52 (5) (2007) 863–868.

[5] W. Ren, R.W. Beard, Decentralized scheme for spacecraft formation flying via the virtual structure approach, Journal of Guidance, Control, and Dynamics 27 (1) (2004) 73–82.

[6] J.A. Fax, R.M. Murray, Information flow and cooperative control of vehicle formations, IEEE Transactions on Automatic Control 49 (9) (2004) 1465–1476.

[7] R. Olfati-Saber, R.M. Murray, Consensus problems in networks of agents with switching topology and time-delays, IEEE Transactions on Automatic Control 49 (9) (2004) 1520–1533.

[8] Z. Lin, M. Broucke, B. Francis, Local control strategies for groups of mobile autonomous agents, IEEE Transactions on Automatic Control 49 (4) (2004) 622–629.

[9] L. Moreau, Stability of multi-agent systems with time-dependent communication links, IEEE Transactions on Automatic Control 50 (2) (2005) 169–182.

[10] W. Ren, R.W. Beard, Consensus seeking in multiagent systems under dynamically changing interaction topologies, IEEE Transactions on Automatic Control 50 (5) (2005) 655–661.

[11] L. Fang, P.J. Antsaklis, Information consensus of asynchronous discrete-time multi-agent systems, in: Proceedings of the American Control Conference, Portland, OR, 2005, pp. 1883–1888.

[12] D. Bauso, L. Giarre, R. Pesenti, Distributed consensus in networks of dynamic agents, in: Proceedings of the IEEE Conference on Decision and Control, Seville, Spain, 2005, pp. 7054–7059.

[13] D.P. Spanos, R. Olfati-Saber, R.M. Murray, Dynamic consensus on mobile networks, in: IFAC World Congress, Prague, Czech Republic, 2005. Paper code We-A18-TO/1.
[14] D. Lee, M.W. Spong, Stable flocking of multiple inertial agents on balanced graphs, in: Proceedings of the American Control Conference, Minneapolis, MN, 2006, pp. 2136–2141.
[15] D. Lee, M.W. Spong, Agreement with non-uniform information delays, in: Proceedings of the American Control Conference, Minneapolis, MN, 2006, pp. 756–761.
[16] J.A. Marshall, T. Fung, M.E. Broucke, G.M.T. D'Eleuterio, B.A. Francis, Experiments in multirobot coordination, Robotics and Autonomous Systems 54 (3) (2006) 265–275.
[17] R. Olfati-Saber, Flocking for multi-agent dynamic systems: Algorithms and theory, IEEE Transactions on Automatic Control 51 (3) (2006) 401–420.
[18] W. Ren, Consensus strategies for cooperative control of vehicle formations, IET Control Theory and Applications 1 (2) (2007) 505–512.
[19] C.W. Reynolds, Flocks, herds, and schools: A distributed behavioral model, Computer Graphics 21 (1987) 25–34.
[20] N. Sorensen, W. Ren, A unified formation control scheme with a single or multiple leaders, in: Proceedings of the American Control Conference, New York, 2007, pp. 5412–5418.
[21] G. Royle, C. Godsil, Algebraic graph theory, in: Springer Graduate Texts in Mathematics #207, New York, 2001.
[22] R.A. Horn, C.R. Johnson, Matrix Analysis, Cambridge University Press, 1985.
[23] M.A. Lewis, K.-H. Tan, High precision formation control of mobile robots using virtual structures, Autonomous Robots 4 (1997) 387–403.
[24] R.W. Beard, J.R. Lawton, F.Y. Hadaegh, A coordination architecture for spacecraft formation control, IEEE Transactions on Control Systems Technology 9 (6) (2001) 777–790.
[25] W. Kang, N. Xi, A. Sparks, Formation control of autonomous agents in 3D workspace, in: Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, CA, 2000, pp. 1755–1760.
[26] N.E. Leonard, E. Fiorelli, Virtual leaders, artificial potentials and coordinated control of groups, in: Proceedings of the IEEE Conference on Decision and Control, Orlando, FL, 2001, pp. 2968–2973.
[27] M. Egerstedt, X. Hu, A. Stotsky, Control of mobile platforms using a virtual vehicle approach, IEEE Transactions on Automatic Control 46 (11) (2001) 1777–1782.
[28] W. Ren, R.W. Beard, T.W. McLain, Coordination variables and consensus building in multiple vehicle systems, in: V. Kumar, N.E. Leonard, A.S. Morse (Eds.), Cooperative Control, in: Lecture Notes in Control and Information Sciences, vol. 309, Springer-Verlag, Berlin, 2005, pp. 171–188.
[29] W. Ren, Multi-vehicle consensus with a time-varying reference state, Systems and Control Letters 56 (7–8) (2007) 474–483.

Wei Ren received the B.S. degree in electrical engineering from Hohai University, China, in 1997, the M.S. degree in mechatronics from Tongji University, China, in 2000, and the Ph.D. degree in electrical engineering from Brigham Young University, Provo, UT, in 2004. From October 2004 to July 2005, he was a research associate in the Department of Aerospace Engineering at the University of Maryland, College Park, MD. Since August 2005, he has been an assistant professor in the Electrical and Computer Engineering Department at Utah State University, Logan, UT. His research focuses on cooperative control of multi-vehicle systems and autonomous control of robotic vehicles. Dr. Ren is currently an Associate Editor for the IEEE Control Systems Society Conference Editorial Board.

Nathan Sorensen received the B.S. degree in electrical engineering in 2005 and the M.S. degree in electrical engineering in 2006, both from Utah State University. Since 2007, he has been with the United Launch Alliance doing design and analysis in avionics for delta series rockets. His research interests include coordinated control of multiple vehicles, nonlinear control systems, and space systems.