Experiments on Formation Switching for Mobile Robots

Kiattisin Kanjanawanishkul, Marius Hofmeister, and Andreas Zell

Abstract— **In this paper, we address the problem of distributed role assignment for multiple mobile robots. This problem arises when a mobile robot in the team must decide what role to take on in a desired formation configuration. In some applications, in which the center and the orientation of a desired formation are not predetermined, the rotation and translation of the formation can be computed by using average consensus protocols. However, the conflict arises when the same role is assigned to more than one robot. This problem is resolved by using a negotiation strategy, while each assigned robot is traveling to the target position. We evaluate our proposed framework through two experiments on a team of physical nonholonomic mobile robots, i.e., (i) robots reconfigure themselves from one formation to another, and (ii) formation switching happens, while each robot is following a reference path.**

I. Introduction

Recent advances in computation and communication have led to distributed control of multi-robot systems (MRS) which, compared to centralized control, offers many advantages, such as increased flexibility and efficiency of operating a group of robots and failure tolerance due to redundancy [1]. Formation control is one of the active research topics in MRS and it has received significant attention during the last two decades (e.g., [2], [3], [4]). The goal of formation control is to maintain the position of a team of robots relative to each other or relative to a reference.

In this work, we address the problem of distributed role assignment. One of the key problems is what role to take on in a given formation when the individual robots do not have complete information about other robots in the team [5], [6], [7]. We propose a novel distributed and online solution to this problem by using consensus protocols and negotiation algorithms. We evaluate our proposed framework through two scenarios, i.e., (i) to drive each robot to the target position transformed through the common translation and rotation of the target formation, and (ii) to maintain a desired formation and to change one formation to another while the robots are following a reference path. It is worth noting that selecting a particular formation shape, in general, relies on a dynamical context, e.g., environments and tasks. There is still no underlying theory that handles this problem. However, in this paper, we assume to have such a mechanism that can choose a suitable formation (e.g., [8]).

The presentation of the paper is organized as follows. Section II presents the problem definition, while our distributed role assignment algorithms are proposed in Section III. The experiments on formation switching and on coordinated path following are given in Section IV and Section V, respectively. The last section contains a conclusion and a discussion of further work.

II. Problem Definition

Given a team of N mobile robots, whose positions are $\mathbf{x}_i \in \mathbb{R}^2,\ i = 1, ..., N$, and the target formation represented by the target positions $\mathbf{x}^d_k \in \mathbb{R}^2,\ k = 1, ..., M$, assuming that all robots have knowledge of the target formation, we need to find an appropriate mapping $p : 1, ..., N \rightarrow 1, ..., M$ that assigns robot i, located at $\mathbf{x}_i$ to target position $\mathbf{x}^d_k$. Furthermore, we are also interested in how to determine the rotation angle θ_f and translation $v_f \in \mathbb{R}^2$ of the target formation when the center and the orientation of the target formation are not predetermined.

A. Assignment Problems

The assignment problem is one of the most famous problems in combinatorial optimization. It consists of finding a minimum weight matching in a weighted bipartite graph. Depending on the form of the cost function, assignment problems can be classified as linear or quadratic. The Hungarian algorithm [9] is one of many algorithms that can be used to solve the linear assignment problem in polynomial time. However, the quadratic assignment problem is NP-hard and suboptimal solutions can be obtained by means of various relaxations.

Let $p \in P_N$, where P_N is the set of all possible permutations over N elements, be the assignment of the robots to the targets. As defined in [5], the centralized role assignment is given as follows:

$$\textstyle\sum_c(\mathbf{x}, \mathbf{x}^d) : \min_{(v_f,\theta_f,p)\in\mathbb{R}^2\times[0,2\pi)\times P_N} J_c(\mathbf{x}, \mathbf{x}^d, v_f, \theta_f, p), \tag{1}$$

where $J_c(\mathbf{x}, \mathbf{x}^d, v_f, \theta_f, p)$ is the cost

$$J_c(\mathbf{x}, \mathbf{x}^d, v_f, \theta_f, p) = \textstyle\sum_{i=1}^N c(\mathbf{x}_i, R(\theta_f)(\mathbf{x}^d_{p(i)} + v_f)), \tag{2}$$

and $R(\theta_f)$ in (2) is the rotation matrix, i.e.,

$$R(\theta_f) = \begin{bmatrix} \cos\theta_f & \sin\theta_f \\ -\sin\theta_f & \cos\theta_f \end{bmatrix}, \tag{3}$$

and c is a performance measure. The interpretation is that $c : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$ gives the cost of assigning the robot i at $\mathbf{x}_i$ to the target located at $R(\theta_f)(\mathbf{x}^d_{p(i)} + v_f)$.

When θ_f and v_f are provided, the optimal assignment satisfies

$$p^* = \underset{p\in P_N}{\operatorname{argmin}}\ J_c(\mathbf{x}, \mathbf{x}^d, v_f, \theta_f, p). \tag{4}$$

K. Kanjanawanishkul, M. Hofmeister, A. Zell are with the Department of Computer Architecture, University of Tübingen, Sand 1, 72076 Tübingen, Germany {kiattisin.kanjanawanishkul,marius.hofmeister, andreas.zell}@uni-tuebingen.de

Moreover, the problem

$$\min_{p\in P_N} J_c(\mathbf{x}, \mathbf{x}^d, v_f, \theta_f, p) \tag{5}$$

corresponds to the well-known linear assignment problem. p^* is easily computed by using the Hungarian method [9]. In [5], [7], the authors are interested in determining the rotation and translation of the target formation and finding an appropriate permutation that assigns target positions to robots. The solution results in a centralized off-line algorithm in the sense that the computation of the solutions will require complete information about all robots in the team. This algorithm has to be done before the team is deployed. However, when computation and information are distributed among multiple robots and the number of robots in the team is unknown, the need for a fully distributed control framework to solve the role assignment problem becomes vital. We propose a distributed solution to this problem using consensus protocols and negotiation algorithms. It has to be noted that although no global cost is optimized in this work, robots can always achieve the role assignment task, as shown in our experimental results.

B. Basic Graph Theory

The solution to the problem mentioned above greatly depends on notions from graph theory. Given the system of robots, we can define a dynamic graph $\mathcal{G}(t)$ as follows [6]:

Definition 1 (Dynamic Graph): We call $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ a dynamic graph consisting of a set of vertices $\mathcal{V} = \{1, ..., N\}$, indexed by the set of robots and a time varying set of links $\mathcal{E}(t) = \{(i, j) \in \mathcal{V} \times \mathcal{V} | \ \|\mathbf{x}_i - \mathbf{x}_j\|_2 < r\}$, meaning that edges are established between robots i and j, if and only if the robots are within distance r of each other.

Dynamic graphs $\mathcal{G}(t)$ such that $(i, j) \in \mathcal{E}(t)$ if and only if $(j, i) \in \mathcal{E}(t)$ are called undirected. Moreover, for any pair of vertices i and j such that $(i, j) \in \mathcal{E}(t)$ we say that vertices i and j are adjacent, or neighbors, at time t and the set of neighbors of robot i is denoted by $N_i \in \mathcal{V}$. A path between vertices (i, j) is a sequence of distinct vertices such that consecutive vertices are adjacent. A topological invariant of graphs that is of particular interest for the purposes of this work is graph connectivity.

Definition 2 (Graph Connectivity): We say that a dynamic graph $\mathcal{G}(t)$ is connected at time t if there exists a path between any two vertices in $\mathcal{G}(t)$.

Given any collection of m distinct instances of $\mathcal{G}(t)$, i.e., $\mathcal{G}(t_1), ..., \mathcal{G}(t_m)$, we say that the collection $\mathcal{G}(t_1), ..., \mathcal{G}(t_m)$ is *jointly connected* if the union of its members is a connected graph [10]. All graphs considered in this paper are undirected and a connectivity on the underlying communication network is assumed. In this case, the *Laplacian* matrix L, constructed from $L = D - A$, where $A = (a_{ij})$ is the adjacency matrix and D is the diagonal degree matrix, is symmetric and positive semi-definite. It has a simple zero eigenvalue and all the other eigenvalues are positive if and only if the graph is connected [11]. This matrix forms the basis for distributed consensus dynamics and captures many properties of the graph.

C. Mobile Robots

The mobile robots shown in Fig. 1 are used in real-world experiments in this paper. The robot controller is an ATMEGA644 microprocessor with 64 KB flash program memory, 16 MHz clock frequency and 4 KB SRAM. The robot orientation is measured by a Devantech CMPS03 compass and it is identified by a colored circle placed on the top of its platform. Using images from a camera looking down upon the robot's workplace, the robot position can be estimated through color segmentation and a Kalman filter. A desktop PC is used to compute the control inputs. The inputs are then sent to the robot via WLAN. Each mobile robot has the following kinematic equations

$$\dot{x} = v\cos\theta, \quad \dot{y} = v\sin\theta, \quad \dot{\theta} = \omega, \tag{6}$$

where $[x, y, \theta]^T$ denotes the state vector in the world frame of the robot. v and ω are the linear and angular velocities, respectively.

III. Distributed Role Assignment

We borrow the notations from [6] as follows. Let $\mathcal{I}_0 = 1, ..., M$ denote the index set of all available positions in a formation. We say that a position $k \in \mathcal{I}_0$ is being "taken" if there is at least one robot i such that $p(i) = k$. Let $\mathcal{I}^a(t)$ denote the index set of all positions that are "available", i.e., not "taken" at time t and $\mathcal{I}^t(t) = \mathcal{I}_0 \setminus \mathcal{I}^a(t)$ denote the index set of all "taken" positions. Similarly, let $\mathcal{I}_i^a$ and $\mathcal{I}_i^t$ denote the index sets of available and taken positions from the perspective of robot i, respectively.

At first, robot i exchanges the position information with its neighbor j, where $(i, j) \in \mathcal{E}(t)$ and then each robot starts locally the Hungarian algorithm to obtain its favorite position. However, in case that $N > M$, the robot may not get any favorite position from the Hungarian algorithm. In this situation, we assign the nearest position to this robot in order to ensure that every robot gets one assigned position. At each time step, every robot exchanges the information with its neighbors in order to update $\mathcal{I}_i^a$ and $\mathcal{I}_i^t$. If two robots realize that they are assigned to the same position, each robot adds the index of the other into its queue. Then, the robot with the higher index starts the pairwise negotiation strategy, given in Algorithm 1, to resolve the conflict. Note that d_i of robot i is the Euclidean norm of the difference between $\mathbf{x}_i$ and its target position.

Correctness of the proposed algorithm depends on the assumption that every robot requesting to be assigned to a

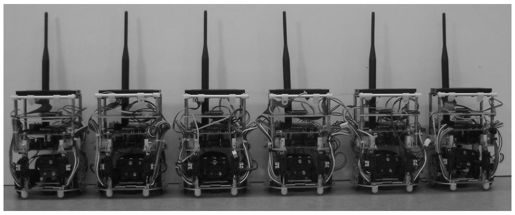


Fig. 1. The mobile robots (12 cm diameter) used in experiments.

Algorithm 1 Negotiation strategies for robot i

```
Input: $\mathcal{I}_i^a$, $\mathcal{I}_i^t$, $p(i)$, the descending index set of robots assigned
        to the same position, i.e., $\mathcal{A}_i = j \in N_i \mid p(j) = p(i)$
Exchange lists with neighbors, i.e., $\mathcal{I}_i^a = \mathcal{I}_i^a \setminus (\cup_{j \in N_i} \mathcal{I}_j^t)$ and
    $\mathcal{I}_i^t = \mathcal{I}_i^t \cup (\cup_{j \in N_i} \mathcal{I}_j^t)$
while $\mathcal{A}_i \neq$ NULL
    k = $\mathcal{A}_i.dequeue$
    if $d_i <= d_k$ then
        Robot $i$ wins and robot $k$ chooses a position from $\mathcal{I}_k^a$
        Update $\mathcal{I}_i^a$, $\mathcal{I}_i^t$, $\mathcal{I}_k^a$, $\mathcal{I}_k^t$
        Dequeue robot $k$ from $\mathcal{A}_i$ and empty $\mathcal{A}_k$
        Robot $k$ starts moving to the new assigned position
    else if $d_i > d_k$ then
        Robot $k$ wins and robot $i$ chooses a position from $\mathcal{I}_i^a$
        Update $\mathcal{I}_i^a$, $\mathcal{I}_i^t$, $\mathcal{I}_k^a$, $\mathcal{I}_k^t$
        Dequeue robot $i$ from $\mathcal{A}_k$ and empty $\mathcal{A}_i$
        Robot $i$ starts moving to the new assigned position
    end if
end while
```

formation role will eventually be able to communicate and negotiate with all other robots requesting to be assigned to the same role. Only one robot will take the role when that conflict is resolved. The robots that do not get that position will choose the new position from $\mathcal{I}^a(t)$ in such a way that it is the nearest position and connectivity is still maintained. In case that the number of robots is more than the number of roles, i.e., $N > M$, the unassigned robots have to perform the task of maintaining connectivity, e.g., [12] in order to keep the connectivity assumption satisfied, and also they have to move away from the "taken" positions. In this work, we simply apply artificial attractive and repulsive forces to meet these requirements. However, connectivity may be lost in case that $N < M$. This issue, solved by simultaneously maintaining connectivity and performing role assignment, is left to the future.

IV. Formation Switching

The goal of this section is to develop a distributed control framework able to drive the robots to the target formation transformed through the common translation vector v_f and rotation matrix $R(\theta_f)$, while the permutation P is determined dynamically by means of distributed negotiation strategies, using only local information. In some applications, in which the center and the orientation of the target formation are not given, we employ consensus protocols to obtain v_f and θ_f that are agreed by all robots in the team, as shown in Section IV-A. To drive the robots to the target positions safely, we integrate the navigating control law with the artificial potential field, as described in Section IV-B.

A. Consensus on the Rotation and Translation

In the consensus problem, a variety of algorithms have been proposed such that a group of robots can agree upon certain quantities of interest, such as direction, position, etc. with only local information (see [11], [13] and see [14] for implementing consensus protocols over an asynchronous communication network).

To minimize the displacement of the whole group from the initial positions to the target positions, we extract the first PCA (principal components analysis) axis from the robot locations and use it as a rotation variable. This line goes through the centroid and also minimizes the square of the distance of each point to that line. We estimate the largest eigenvector of the covariance matrix of distributed positions in a decentralized fashion, assuming that the positions are measured by the localization module and the communication links are locally connected. The idea proposed in [15] is based on the decomposition of the so-called power method into a decentralized iterative protocol. The eigenvectors of the covariance matrix $\hat{C}$ can be derived by using a power method as follows

$$\mathbf{q}(n+1) = \frac{\hat{C}\mathbf{q}(n)}{\|\hat{C}\mathbf{q}(n)\|} \quad , \tag{7}$$

with $\hat{C} = (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T, i = 1, ..., N$, where $\bar{\mathbf{x}}$ is the centroid. This method converges to the maximum eigenvector of $\hat{C}$ as long as the maximum eigenvalue of $\hat{C}$ is strictly greater than the other eigenvalues and the vector $\mathbf{q}(0)$, an initial random vector, has a non-zero component in the direction of the eigenvector associated to the largest eigenvalue [15]. For simplicity, let us assume that $\bar{\mathbf{x}} = \mathbf{0}$. The recursive equation (7) can be expanded as follows:

$$\begin{aligned} \mathbf{q}_1(n+1) &= \frac{\hat{C}\mathbf{q}_1(n)}{\|\hat{C}\mathbf{q}_1(n)\|} \\ &= \frac{\sum_{i=1}^{N} \mathbf{x}_i(\mathbf{x}_i \cdot \mathbf{q}_1(n))}{\|\sum_{i=1}^{N} \mathbf{x}_i(\mathbf{x}_i \cdot \mathbf{q}_1(n))\|} \\ &= \frac{\overline{\mathbf{x}(\mathbf{x} \cdot \mathbf{q}_1(n))}}{\|\overline{\mathbf{x}(\mathbf{x} \cdot \mathbf{q}_1(n))}\|} \end{aligned} \quad . \tag{8}$$

Subscript 1 of $\mathbf{q}_1(n)$ denotes the first eigenvector. Unlike [15], we perform the average consensus protocols for $\mathbf{x}_i(\mathbf{x}_i \cdot \mathbf{q}_1(n)),\ i = 1, ..., N$, instead of only the inner products and, in our algorithm, each robot does not need to know the number of robots in the team. Finally, we can obtain the estimate of $\mathbf{q}_1$.

B. Navigating Control with Collision Avoidance

In fact, any controller with the capability to avoid inter-robot collisions and to navigate through waypoints can be used. In this work, we employ the controller from [16]. The control architecture combines two feedback loops: a motion control loop and a new-target control loop. The latter loop provides a modification of the target position when an obstacle appears on the path of the mobile robot. By representing the robot position in polar coordinates, and considering the error vector e, as well as by letting $\alpha = \phi - \theta$ be the angle measured between the distance vector e and the main robot axis, the above kinematic equations (6) can be rewritten

$$\dot{e} = -v\cos\alpha, \qquad \dot{\alpha} = -\omega + v\frac{\sin\alpha}{e} \quad . \tag{9}$$

Then, the control laws for v and ω are given by

$$\begin{aligned} v &= \gamma \tanh e \cos\alpha \\ \omega &= k_c\alpha + \gamma\frac{\tanh e}{e}\sin\alpha\ \cos\alpha \quad \text{with } k_c > 0 \end{aligned} \quad , \tag{10}$$

where $\gamma = |v_{\max}|$ and $|\omega_{\max}| = k_c\pi + \gamma/2$. Since we do not have the prescribing orientation of the target position, these control laws can drive the distance vector e and α to zero asymptotically and θ becomes constant when $t \to \infty$. To avoid obstacles, the concept of attractive and repulsive forces to avoid obstacles, as proposed in [17], is applied to modify the target position when an obstacle suddenly appears on the path during navigating towards a target position.

C. Task Sequencing

Since the mission of this section is that a team of robots can reconfigure themselves from one formation to another, the whole formation-switching task can be controlled by passing from one to the next as each formation task is completed. When global communication is not available, the robots must measure the degree of consensus in their team. Once the consensus reaches a preset threshold, meaning that some proportion of the team members agree that the current task is completed, they should induce their system to abandon the current task and start the next one. This threshold highly depends on task requirements. If it is too high, the probability of a robot making an error and prematurely activating the switch to the next task may increase. To attain this goal, we introduce a new state, $z_i(t)$ for robot i, to measure the degree of task completion of the current task. It becomes zero when all robots agree that the current task is completed. The following proportional-integral consensus estimator, based on [18], is used in this work:

$$\begin{aligned} \dot{z} &= -Lz + k_z(w - z) + L\eta \\ \dot{\eta} &= -Lz \end{aligned} \quad , \tag{11}$$

where L is the Laplacian matrix and η is an integrator variable. $w_i(t)$ is a dynamic input. It becomes 0, if robot i detects the completion of its current task, and 1, otherwise. k_z is the forgetting factor. Large values of k_z mean that we get rid of old information quickly. We initialize $z_i(0) = w_i(0)$ for each robot. Unlike static consensus, in which all robots must converge to the average of their initial states, i.e, $\frac{1}{N}\sum_i^N z_i$, $w_i(t)$ can be seen as a dynamic input in dynamic consensus. All robots must track the time-varying average of the w_i terms, i.e., they have to reach $\bar{w} = \frac{1}{N}\sum_i^N w_i$. Therefore, z_i can be considered a time-varying estimate of the instantaneous average value $\bar{w}$. We use z_i to measure the degree of task completion. When z_i reaches a preset threshold, this implies that robot i should abandon the current task and start the next one.

D. Experimental Results

Real-world experiments of the distributed role assignment as established in this section are carried out to evaluate the performance of our framework. We consider a navigation task in $\mathbb{R}^2$, where $N = 6$ robots shown in Fig. 1, starting from randomly chosen initial configurations but satisfying the connectivity assumption, are required to reach the desired formation configurations. We evaluate our algorithms with three different formations as shown in Fig. 2. The robots have to switch from one formation to another in sequence. The communication range r is 1.2 m. In consensus-based task sequencing, the threshold of the state z is 0.1 and the forgetting factor is set to 0.5.

As shown in Fig. 3, all robots can successfully complete the formation switching task, in which the center and the orientation of the target formation are not given. They use the average consensus protocol to obtain an agreement on the translation as well as the rotation, and then they start negotiation to get the role, if a conflict arises. Each robot can move to the target position without any collision by using the controller described in Section IV-B. The translation and rotation may be changed from one formation to another if there are some unassigned robots.

V. Coordinated Path Following

A control strategy for coordinated path following of multiple mobile robots is presented in this section. A virtual vehicle concept (see [19]) is combined with a path following approach to achieve formation tasks (the pioneering work in path following control can be found in [20]). Our formation controller is designed in such a way that the path derivative is employed as an additional control input to synchronize the robot's motion with neighboring robots. A second-order consensus algorithm with a reference velocity under undirected information exchange is introduced to derive the control law for synchronization.

A. Controller Design

The error dynamics of robot i with respect to the robot frame are

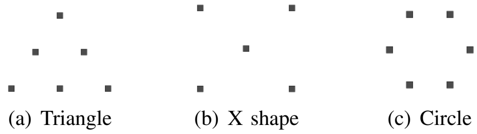
$$\begin{aligned} \dot{x}_{ei} &= y_{ei}\omega_i - v_i + \dot{s}_i\cos\theta_{ei} \\ \dot{y}_{ei} &= -x_{ei}\omega_i + \dot{s}_i\sin\theta_{ei} \\ \dot{\theta}_{ei} &= \kappa_i\dot{s}_i - \omega_i \end{aligned} \tag{12}$$

where $[x_{ei},\ y_{ei},\ \theta_{ei}]^T$ denotes the error state vector, κ_i is the path curvature, and $\dot{s}_i$ is the velocity of a virtual vehicle.

Let us define $\dot{\tilde{s}}_i = \dot{s}_i - v_{di}$, where $\dot{\tilde{s}}_i$ represents the formation speed tracking error of robot i and v_{di} is the desired forward velocity, and then we choose

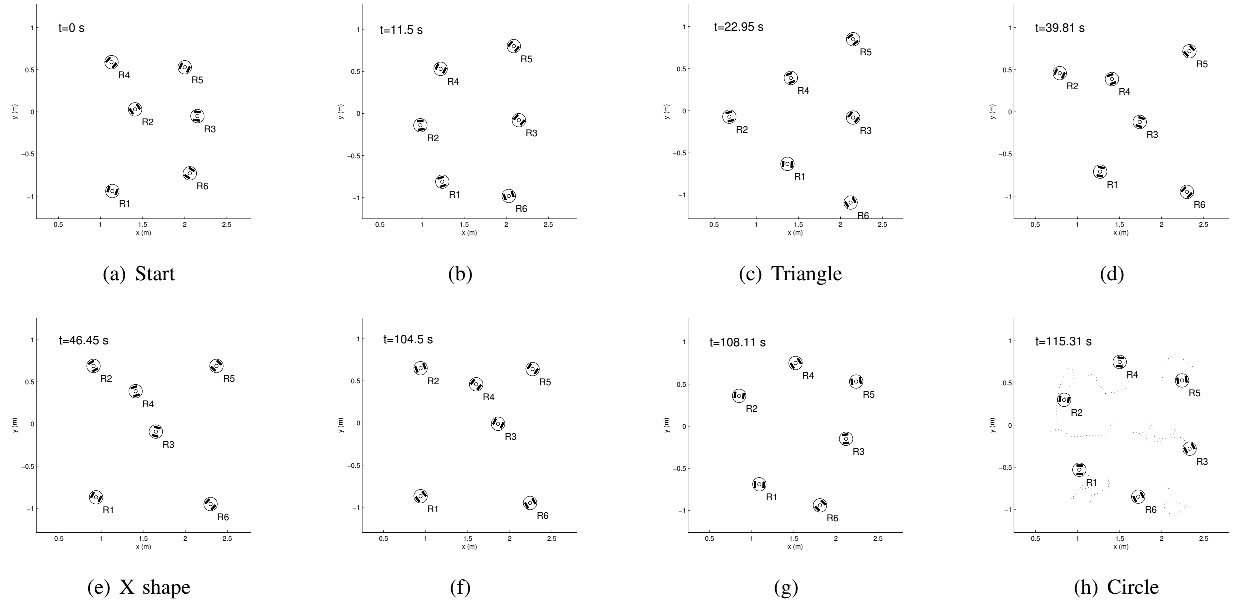
$$V = \frac{1}{2}\sum_{i=1}^{N}\left(x_{ei}^2 + y_{ei}^2 + \frac{1}{k_1}\left(\theta_{ei} - \delta_i(y_{ei}, v)\right)^2 + \dot{\tilde{s}}_i^2 + k_2\bar{s}_i^2\right) \tag{13}$$

as a candidate Lyapunov function, where k_1 and k_2 are positive gains. $\bar{s}_i = \sum_{j\in N_i}(s_i - s_j - s_{dij})$ is the coordination error of robot i and s_{dij} is the desired distance between two neighbors i and j. The function δ_i can be interpreted



(a) Triangle (b) X shape (c) Circle

Fig. 2. Illustration of three different formation configurations. Squares represent the target positions in the formation. These formations must satisfy the connectivity assumption.



(a) Start (b) (c) Triangle (d)

(e) X shape (f) (g) (h) Circle

Fig. 3. The experimental results on formation switching (a) each robot starts at a different random position, (c),(e) and (h) are the snapshots when the formation task is completed, corresponding to Fig. 2(a), Fig. 2(b), and Fig. 2(c), respectively. Note that robot $R4$ in (e) is an unassigned robot.

as the desired value for the orientation θ_{ei} during transients [20]. It is assumed that $\lim_{t\to\infty} v(t) \neq 0$, $\delta_i(0, v) = 0$, and $y_{ei} v \sin(\delta_i) \leq 0, \forall y_{ei} \forall v$. The function $\delta_i(y_{ei}, v)$ taken from [19] is $\delta_i(y_{ei}, v) = -\text{sign}(v_{di})\theta_a \tanh y_{ei}$ with $\theta_a = \frac{\pi}{4}$.

The derivative of V can be computed to give

$$\dot{V} = \sum_{i=1}^{N} \Bigg[x_{ei}\dot{x}_{ei} + y_{ei}\dot{y}_{ei} + \frac{1}{k_1}(\theta_{ei} - \delta_i)\left(\dot{\theta}_{ei} - \dot{\delta}_i\right) + \dot{\bar{s}}_i \ddot{\bar{s}}_i + k_2 \bar{s}_i \dot{\bar{s}}_i \Bigg]. \tag{14}$$

Let the control laws for v_i, ω_i, and $\ddot{s}_i$ be defined as

$$v_i = k_4 x_{ei} + v_{di}\cos\theta_{ei} \tag{15}$$

$$\omega_i = k_5(\theta_{ei} - \delta_i) + \omega_{di} - \dot{\delta}_i + k_1 y_{ei} v_{di}\left[\frac{\sin\theta_{ei} - \sin\delta_i}{\theta_{ei} - \delta_i}\right] \tag{16}$$

$$\ddot{s}_i = \dot{v}_{di} - k_6\dot{\bar{s}}_i - x_{ei}\cos\theta_{ei} - y_{ei}\sin\theta_{ei} - \frac{\kappa_i}{k_1}(\theta_{ei} - \delta_i) - 2k_2\sum_{j\in N_i}(s_i - s_j - s_{dij}) - k_3\sum_{j\in N_i}(\dot{s}_i - \dot{s}_j) \tag{17}$$

where $k_3, k_4, k_5, k_6 > 0$ and $\omega_{di} = \kappa_i v_{di}$. Then we can achieve

$$\dot{V} = \sum_{i=1}^{N}\left[-k_4 x_{ei}^2 - \frac{k_5}{k_1}(\theta_{ei} - \delta_i)^2 - k_6\dot{\bar{s}}_i^2 + y_{ei}v_{di}\sin\delta_i\right] - k_3\dot{s}^T L\dot{s} \leq 0 \tag{18}$$

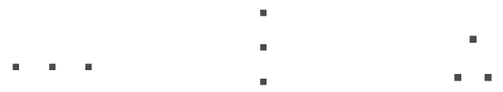
where $\dot{s} \in \mathbb{R}^N$ is the stack vector of the robots' path derivative. We omit the proof due to limited space[1].

B. Experimental Results

The C^2 reference path and the initial positions of three robots are plotted in Fig. 5(a). The robots have to switch one formation to another, see Fig. 4. We assume that they have a mechanism that can make a decision on which formation they want to select according to the environment. To avoid collisions, the concept of attractive and repulsive forces is applied to modify the reference position. As seen in Fig. 5(b) and Fig. 5(c), the robots maintain a line formation, illustrated in Fig. 4(a), while following a reference path. Fig.5(d) and Fig. 5(e) show the snapshots where they switch to a column formation, shown in Fig. 4(b), and the robots then switch to a triangular formation, as depicted in Fig.5(i) and Fig. 5(j). At the steady state, the distance errors and the coordination errors are less than 10 cm, while the orientation errors are less than 20 degrees. The main sources of disturbances include sensor distortion, vision-system delays, and communication delays.

VI. Conclusions and Future Work

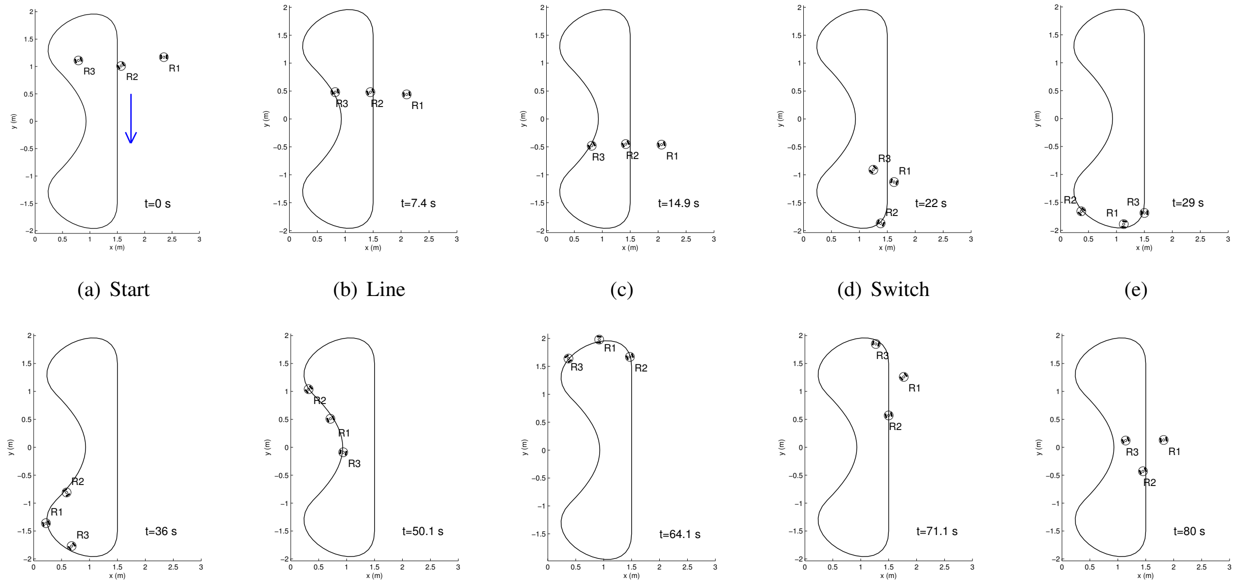
In this paper, we presented a novel solution for the distributed role assignment in formation switching tasks. Our framework is mainly based on the average consensus protocols and the negotiation strategies. From the experimental results of the first scenario, we can conclude that each



(a) Line (b) Column (c) Triangle

Fig. 4. Illustration of three different formation configurations. Squares represent the target positions in the formation. These formations must satisfy the connectivity assumption.

[1] The proof can be found at: http://www.ra.cs.uni-tuebingen.de/mitarb/kanjana/aim2010ext.pdf.



(a) Start (b) Line (c) (d) Switch (e)

(f) Column (g) (h) (i) Switch (j) Triangle

Fig. 5. The experimental results on coordinated path following and formation switching.

robot can reconfigure itself from one formation to another successfully. It can also detect the completion of each task and automatically move onto the next task in its queue. In the second scenario, we can achieve coordinated path following, i.e., each robot can be steered along a set of given spatial paths, while it keeps a desired inter-vehicle coordination pattern and switches formations.

Currently, we are investigating the computational complexity issue and theoretical proof. Future research also includes the extension of our results to robots in more complex environments. For example, environmental obstacles may appear on the robot's path.

References

[1] R. M. Murray, "Recent research in cooperative-control of multivehicle systems," *Journal of Dynamics, Systems, Measurement and Control*, vol. 129, no. 5, pp. 571–583, 2007.

[2] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor, "A vision-based formation control framework," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 5, pp. 813–825, Oct. 2002.

[3] J. R. Lawton, R. W. Beard, and B. Young, "A decentralized approach to formation maneuvers," *IEEE Trans. on Robotics and Automation*, vol. 19, no. 6, pp. 933–941, Dec. 2003.

[4] R. Ghabcheloo, A. Aguiar, A. Pascoal, C. Silvestre, I. Kaminer, and J. Hespanha, "Coordinated path-following in the presence of communication losses and time delays," *SIAM - Journal on Control and Optimization*, vol. 48, no. 1, pp. 234–265, 2009.

[5] M. Ji, S. Azuma, and M. Egerstedt, "Role-assignment in multi-agent coordination," *Int. Journal of Assistive Robotics and Mechatronics*, vol. 7, no. 1, pp. 32–40, Mar. 2006.

[6] N. Michael, M. Zavlanos, V. Kumar, and G. Pappas, "Distributed multi-robot task assignment and formation control," in *Proc. of Int. Conf. on Robotics and Automation*, Pasadena, CA, May 2008, pp. 128–133.

[7] J. C. Derenick and J. R. Spletzer, "Convex optimization strategies for coordinating large-scale robot formations," *IEEE Trans. on Robotics*, vol. 23, no. 6, pp. 1252–1259, Dec. 2007.

[8] M. D. Rocco, S. Panzieri, and A. Priolo, "Formation control through environment pattern recognition for a multi-robot architecture," in *Proc. of European Conf. on Mobile Robots*, Mlini/Dubrovnik, Croatia, Sept. 2009, pp. 241–246.

[9] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics*, vol. 2, no. 1, pp. 83–97, 1955.

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

[11] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Trans. on Automatic Control*, vol. 49, no. 9, pp. 1465–1476, Sept. 2004.

[12] M. Zavlanos and G. Pappas, "Potential fields for maintaining connectivity of mobile networks," *IEEE Trans. on Robotics*, vol. 23, no. 4, pp. 812–816, Aug. 2007.

[13] W. Ren, R. Beard, and E. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control System Magazine*, vol. 27, no. 2, pp. 71–82, Apr. 2007.

[14] M. Mehyar, D. Spanos, J. Pongsajapan, S. H. Low, and R. M. Murray, "Asynchronous distributed averaging on communication networks," *IEEE/ACM Trans. on Networking*, vol. 15, no. 3, pp. 512–520, June 2007.

[15] A. Scaglione, R. Pagliari, and H. Krim, "The decentralized estimation of the sample covariance," in *Proc. of IEEE Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, Oct. 2008, pp. 1722–1726.

[16] R. Carelli, H. Secchi, and V. Mut, "Algorithms for stable control of mobile robots with obstacle avoidance," *Latin American Applied Research*, vol. 29, no. 3/4, pp. 191–196, 1999.

[17] J. Velagic, B. Lacevica, and B. Perunicica, "A 3-level autonomous mobile robot navigation system designed by using reasoning/search approaches," *Robotics and Autonomous Systems*, vol. 54, no. 12, pp. 989–1004, Dec. 2006.

[18] R. A. Freeman, P. Yang, and K. M. Lynch, "Distributed estimation and control of swarm formation statistics," in *Proc. of American Control Conference*, Minneapolis, MN, June 2006.

[19] D. Soeanto, L. Lapierre, and A. Pascoal, "Adaptive non-singular path-following, control of dynamic wheeled robots," in *Proc. of Int. Conf. on Advanced Robotics*, Coimbra, Portugal, June 30 - July 3 2003, pp. 1387–1392.

[20] A. Micaelli and C. Samson, "Trajectory-tracking for unicycle-type and two-steering-wheels mobile robots," INRIA Sophia-Antipolis, Tech. Rep. 2097, Nov. 1993.