A Framework for Vision Based Formation Control

Aveek Das, Rafael Fierro, Vijay Kumar, Jim Ostrowski, John Spletzer, and Camillo Taylor

Abstract—**We describe a framework for controlling and coordinating a group of nonholonomic mobile robots for cooperative tasks ranging from scouting and reconnaissance to distributed manipulation. The framework allows us to build complex systems from simple controllers and estimators. This modular approach is attractive because of the potential for reusability. In addition, we show that our approach to composition also guarantees stability and convergence in a wide range of tasks. There are two key features in our approach. The first is a paradigm for switching between simple decentralized controllers thus allowing changes in the formation. Second, all the controllers use information from a single sensor — an omnidirectional camera. We describe estimators that abstract the sensory information at different levels enabling decentralized as well as cooperative control. Our results consist of numerical simulations as well as experiments on a platform of three nonholonomic robots.**

Keywords—**Hybrid control, formation control, cooperative localization, distributed manipulation, nonholonomic mobile robots.**

I. Introduction

The last few years have seen active research in the field of control and coordination for multiple mobile robots, and application to tasks such as exploration [1], surveillance [2], search and rescue [3], mapping of unknown or partially known environments [4], [5], distributed manipulation [6], [7] and transportation of large objects [8], [9]. An excellent review of contemporary work in this area is presented in [10].

While robot control is considered to be a well understood problem area [11], most of the current success stories in multi-robot coordination do not rely on or build on the results available in the control theory and dynamical systems literature. The reason for this is fairly clear. Traditional control theory mostly enables the design of controllers in a single mode of operation, in which the task and the model of the system are fixed [12], [13], [14]. When operating in unstructured or dynamic environments with many different sources of uncertainty, it is very difficult if not impossible to design controllers that will guarantee performance even in a local sense. A similar problem exists in developing estimators in the context of sensing, and mapping and motion planning.

In contrast, we know that it is relatively easy to design reactive controllers or behaviors that react to simple stimuli or commands from the environment. We can see successful applications of this idea in the subsumption architecture [15], in the paradigm for behavior-based robotics [16], [17], [18], and in other work [10].

In this paper we address the development of intelligent robot systems by composing simple building blocks in a bottom-up approach. The building blocks consist of controllers and estimators, and the framework for composition allows for tightly-coupled perception-action loops. While this philosophy is similar in spirit to the behavior based control paradigm [15], we differ in the control-theoretic approach to the development of the basic components, and our formal approach to their composition.

GRASP Labratory, University of Pennsylvania, Philadelphia PA, {aveek,rfi erro,kumar,jpo,spletzer,cjtaylor}@grasp.cis.upenn.edu

The goal of this paper is to develop a framework for composition of simple controllers and estimators to control the formation of a group of robots. By formation control, we simply mean the problem of controlling the relative positions and orientations of robots in a group, while allowing the group to move as a whole. We are particularly interested in problems of cooperative manipulation, where a "rigid" formation may be necessary to transport a grasped object to a prescribed location, and cooperative mapping, where the formation may be defined by a minimal (in comparison) set of constraints.

Problems in formation control that have been investigated include assignment of feasible formations [19], [20], [21], getting into formation [22], [23], [24], maintenance of formation shape [25], [26], [27] and switching between formations [28], [29]. Approaches to modeling and solving these problems have been diverse, ranging from paradigms based on combining reactive behaviors [17], [30] to those based on leader-follower graphs [25] and virtual structures [31], [32].

In this paper we describe a suite of controllers and estimators and a methodology for their composition that permits control of formations for a group of robots in all the above and other applications. This suite consists of centralized as well as decentralized algorithms — either can be used depending on the nature of the communication link. We consider situations in which there may be no access to any global positioning system and the main sensing modality is vision. All our estimators are derived from vision. Our platform of interest is a nonholonomic car-like robot with a single physical sensor - an omnidirectional camera. The wider field of view offered by this camera allows us to extract more information from the video signal than conventional cameras and makes it possible to implement a wider range of motion strategies on the same platform.

Our contributions in this paper are two-fold. First we develop a bottom-up approach to building controllers and estimators, and describe, in some detail, the components used for multi-robot coordination. These include simple decentralized, reactive controllers for obstacle avoidance, collision recovery and pursuing targets, and more complex controllers for maintaining formation, controllers that can be either centralized or decentralized and are derived from input-output linearization. Our second contribution is the framework for multi-robot coordination that allows robots to maintain or change formation while following a specified trajectory, and to perform cooperative manipulation tasks. We require a robot to be designated as the *reference* robot or leader. All other robots choose an appropriate controller depending on the their relative position and converge on the desired formation. Our framework involves a sequential composi-

tion of controllers, or modes, and we show that the dynamics of the resulting switched system are stable.

The rest of this paper is organized as follows. First we give a broad overview of our framework in Section II, and illustrate some of the salient features using our nonholonomic mobile robot experimental testbed. In Section III we present a set of controllers that serve as the building blocks for formation control. We discuss the assignment of formations, changes in formations, and stable switching strategies in Section IV using a group of three robots as an example. Section V addresses the centralized and decentralized schemes for sensing and estimation for implementation of formation control. Hardware details and experimental results illustrating the application of this methodology to cooperative tasks are in Section VI. Finally, some concluding remarks and directions for future work are provided in Section VII.

II. Framework

In this section, we describe a framework for cooperative control of multiple mobile robots. We are motivated by theoretical ideas and modern tools in software engineering, and the emerging theory of hybrid systems [1].

The software for each robot consists of components called *agents*. Agents operate concurrently and lend themselves to *parallel composition*. Each agent consists of a main *mode*, that may in turn have *submodes*. A mode is a discrete state characterized by a set of differential equations and constraints. The concept of modes allows us to formally define the notion of sequential composition. Definition of submodes implicitly define the notion of *hierarchical composition*.

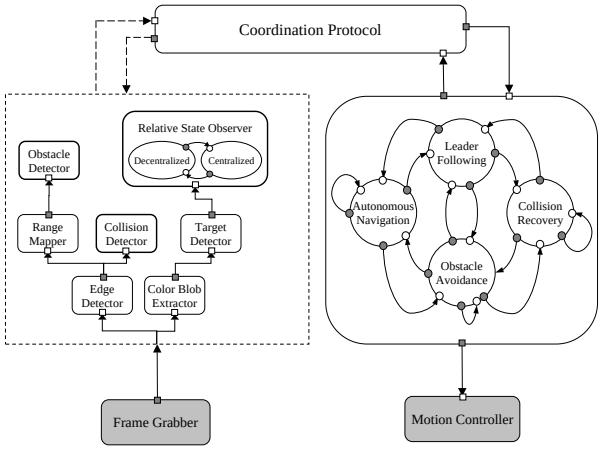

Fig. 1. Architecture diagram for formation control. Agents are indicated by rounded rectangles, and shading indicates off-the-shelf components. The estimator agents are grouped together in the dotted rectangle. To avoid clutter, the input/output channels are not shown explicitly. The controller agent shows the four main submodes (circles) with arrows denoting possible transitions

The basis for the software framework and formal definitions of composition are provided in [34]. Here, we will use the architecture diagram in Figure 1 to explain the definitions. All robots run the same software. In the figure, there are many estimator agents (shown enclosed within the dotted rectangle on the left) operating concurrently. Well-defined input and output ports describe the exchange of information between the agents. On the right, is the behavioral diagram for the controller agent. There are four main submodes (circles) within the agent, with the arrows indicating transitions between the submodes. The conditions (invariants and guards) for transitions, and the entry/exit points in each mode are not shown in the diagram, but are explained in following sections. Each submode can contain a hierarchy of submodes. In this framework, the coupling of one or more sensor agents with a controller agent defines the dynamic system, and the specification of the submodes within each agent defines a behavior in the sense of Arkin and others [16]. The coordination protocol instantiates each agent (all estimators and the controller) with parameters and provides a reference trajectory that are used in different modes.

Our low-level implementation in C++ uses Live Objects [35]. Live Objects have been developed as part of the software architecture for implementation on the hardware platforms. A live object encapsulates algorithms and data in the usual object-oriented manner together with control of a thread within which the algorithms will execute, and a number of events that allow communication with other live objects.

Thus far, we have restricted our definition to components, *i.e.*, agents and modes, for a single robot agent. Since our robots can communicate through a wireless ethernet, we can form two-robot or n-robot agents by parallel composition of robot agents. However, our scope in this paper is much less ambitious. The availability and sharing of information between the robots allows us to – (a) design modes within estimator agents that can exploit sensory information obtained from other robots; and (b) design the coordination protocol to initiate or trigger mode-switching within the controller agent.

Before we proceed with the description of the individual components, we list several important assumptions concerning the group of robots and the formation. We assume, as we do in [25], the robots are labeled and one of the robots, designated as R_1 (or simply 1), is the lead robot. The lead robot's motion defines the motion of the group. The motion of the group members within the formation can be characterized in terms of the motion of individual robots with respect to the lead robot, even though the robots may not follow the lead robot in a leader-follower configuration. As in [28], the relationship between a robot and its neighboring robots is described by a *control graph*. The control graph is a acyclic, directed graph with robots as nodes, R_1 as the parent node, and edges directed from nodes with smaller integer values to those with higher integer values, and no more than two incoming edges incoming at each node [2]. As explained later, each edge denotes a constraint between the robots connected by the edge, and therefore a controller that tries to maintain the constraint. We will consider two types of scenarios. In the first, the control graph is fixed and is not changed through the task. In the second approach, the control graph is adapted to changes in the environment and the relative robot positions. We illustrate both these scenarios through simulations and experiments in the following sections.

[1] A hybrid system here refers to a collection of digital programs that interact with each other in a physical world that is analog in nature [33].

[2] The numbering constraints on the lead robot and other robots is relaxed in other work [29], but we impose this constraint here to limit the scope of the paper.

III. Control Algorithms

A. Modeling

In this section, we describe control algorithms that specify the interactions between each robot and its neighbor(s) or the environment. The robots are velocity controlled nonholonomic car-like platforms and have two independent inputs. The control laws are based on input-output feedback linearization. This means we are able to regulate two outputs. The kinematics of the i^{th} robot are given by

$$\dot{x}_i = v_i \cos\theta_i, \quad \dot{y}_i = v_i \sin\theta_i, \quad \dot{\theta}_i = \omega_i, \tag{1}$$

where $\boldsymbol{x}_i \equiv (x_i,\ y_i,\ \theta_i) \in SE(2)$, and v_i and ω_i are the linear and angular velocities, respectively.

We consider a subgroup of two robots shown in Figure 2. First, we describe a controller for leader–following, adopted from [25], and derive a second controller that takes into account possible interactions with an obstacle.

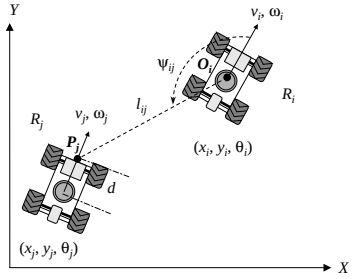

Fig. 2. Two robots in a leader–following configuration.

B. Basic Leader-Following Control

By using this controller (denoted $SB_{ij}C$ here), robot R_j follows R_i with a desired *S*eparation l_{ij}^d and desired relative *B*earing ψ_{ij}^d. Note that this relative bearing describes the heading direction of the follower with respect to the leader. The two-robot system is transformed into a new set of coordinates where the state of the leader is treated as an exogenous input. Thus the kinematic equations are given by

$$\dot{\boldsymbol{z}}_j = \boldsymbol{G}_1(\boldsymbol{z}_j)\boldsymbol{u}_j + \boldsymbol{F}_1(\boldsymbol{z}_j, \boldsymbol{u}_i), \quad \theta_j = \omega_j, \tag{2}$$

where $\boldsymbol{z}_j = [l_{ij} \quad \psi_{ij}]^T$ is the system output, $\boldsymbol{u}_j = [v_j \quad \omega_j]^T$ is the input for R_j, $\boldsymbol{u}_i = [v_i \quad \omega_i]^T$ is R_i's input, and

$$\begin{aligned} \boldsymbol{G}_1 &= \begin{pmatrix} \cos\gamma_{ij} & d\sin\gamma_{ij} \\ \frac{-\sin\gamma_{ij}}{l_{ij}} & \frac{d\cos\gamma_{ij}}{l_{ij}} \end{pmatrix} \\ \boldsymbol{F}_1 &= \begin{pmatrix} -v_i\cos\psi_{ij} \\ \frac{v_i\sin\psi_{ij}}{l_{ij}} - \omega_i \end{pmatrix} \\ \gamma_{ij} &= \theta_i + \psi_{ij} - \theta_j. \end{aligned}$$

By applying input-output feedback linearization, the control velocities for the *follower* are given by

$$\boldsymbol{u}_j = \boldsymbol{G}_1^{-1}(\boldsymbol{p}_1 - \boldsymbol{F}_1), \tag{3}$$

where d is the offset to an off-axis *reference* point P_j on the robot, and $\boldsymbol{p}_1$ is an auxiliary control input given by

$$\boldsymbol{p}_1 = \begin{pmatrix} k_1(l_{ij}^d - l_{ij}) \\ k_2(\psi_{ij}^d - \psi_{ij}) \end{pmatrix} \quad = \boldsymbol{k}(\boldsymbol{z}_j^d - \boldsymbol{z}_j),$$

$k_1,\ k_2 > 0$ are the design controller gains. The closed-loop linearized system is simply given by

$$\dot{\boldsymbol{z}}_j = \boldsymbol{p}_1, \quad \dot{\theta}_j = \omega_j. \tag{4}$$

In the following, we prove that under suitable assumptions on the motion of the lead robot, the closed-loop system is stable. Since we are using input-output feedback linearization [13], the output vector $\boldsymbol{z}_j$ will converge to the desired value $\boldsymbol{z}_j^d$ arbitrarily fast. However, a complete stability analysis requires the study of the internal dynamics of the robot, *i.e.*, the heading angle θ_j which depends on the controlled angular velocity ω_j.

Theorem 1: Assume that the lead vehicle's linear velocity along the path $g(t) \in SE(2)$ is lower bounded *i.e.*, $v_i \geq V_{\min} > 0$, its angular velocity is bounded *i.e.*, $\|\omega_i\| < W_{\max}$, and the initial relative orientation $\|\theta_i - \theta_j\| < c_1\pi$ with $c_1 < 1$. If the control input Eq. 3 is applied to R_j, then the system output $\boldsymbol{z}_j$ in Eq. 4 converges exponentially to the desired value $\boldsymbol{z}_j^d$.

Proof: Let the system error $\boldsymbol{e} = [e_1\ e_2\ e_3]^T$ be defined as

$$e_1 = l_{ij}^d - l_{ij}, \quad e_2 = \psi_{ij}^d - \psi_{ij}, \quad e_3 = \theta_i - \theta_j. \tag{5}$$

We need to show that the internal dynamics of R_j are stable which is equivalent to showing that the orientation error e_3 is bounded. Thus, we have

$$\dot{e}_3 = \omega_i - \omega_j,$$

and, after some algebraic simplification, we obtain

$$\dot{e_3} = -\frac{v_i}{d}\sin e_3 + \eta_1(\omega_i, e_1, e_2, e_3), \tag{6}$$

where

$$\begin{aligned} \eta_1(t, e_3) &= (1 - \frac{l_{ij}}{d}\cos(e_3 + \psi_{ij}))\omega_1 \\ &\quad -\frac{1}{d}(k_1 e_1 \sin(e_3 + \psi_{ij}) + k_2 e_2 l_{ij}\cos(e_3 + \psi_{ij})). \end{aligned}$$

The nominal system, *i.e.*, $\eta_1(t, e_3) = 0$ is given by

$$\dot{e_3} = -\frac{v_i}{d}\sin e_3, \tag{7}$$

which is (locally) exponentially stable provided that the velocity of the lead robot $v_i > 0$ and $\|e_3\| < \pi$. Since ω_i is bounded, it can be shown that $\|\eta_1(t, e_3)\| \leq \delta_1$. By using stability theory of perturbed systems [12] and the condition $\|e_3(t_0)\| < c_1\pi$, we can show that

$$\|e_3(t)\| \leq \sigma_1, \qquad \forall\, t \geq t_1$$

for some finite time t_1 and positive number σ_1. □

Remark 1: The above theorem shows that, under some reasonable assumptions, the formation system is stable, *i.e.*, there

exists a Lyapunov function $V(t,\boldsymbol{e}) \in [0,\infty) \times D$, where $D = \{\boldsymbol{e} \in \mathbb{R}^3 | \, \|\boldsymbol{e}\| < c\}$ and a positive number c, such that $\dot{V}(t,\boldsymbol{e}) \leq 0$. Let $\boldsymbol{e}_{12} = [e_1 \; e_2]^T$ and

$$V = \boldsymbol{e}_{12}^T \boldsymbol{P}_{12} \boldsymbol{e}_{12} + \frac{1}{2} e_3^2 \tag{8}$$

be a Lyapunov function for the system error Eq. 5. Then

$$\dot{V} \;\; = -\boldsymbol{e}_{12}^T \boldsymbol{Q}_{12} \boldsymbol{e}_{12} - \tfrac{v_i}{d} e_3 \sin e_3 + \eta_1(t,e_3) e_3 \tag{9}$$

where $\boldsymbol{P}_{12}$ and $\boldsymbol{Q}_{12}$ are 2×2 positive definite matrices.

By looking at Eq. 8-9, we can study some particular formations of practical interest. For example, if the leader travels in a straight line, *i.e.*, $\omega_i = 0$. It can be shown that the system is (locally) asymptotically stable *i.e.*, $e_3 \to 0$ as $t \to \infty$ provided that $v_i > 0$ and $\|e_3\| < \pi$. If ω_i is constant (*circular motion*), then e_3 is bounded. It is well-known that an optimal nonholonomic path can be planned by joining linear and circular trajectory segments. Hence any trajectory generated by such a planner for the leader will ensure stable leader-follower dynamics using the above controller.

This result can be extended to n robots in a *convoy-like* formation (*c.f.*, [36]). Let us consider a team of n robots where R_i follows R_{i-1} under $SB_{i-1,i}C$. Consider the Lyapunov function

$$V_{1\ldots n} = \sum_{i=2}^{n} \boldsymbol{e}_{i-1,i}^T \boldsymbol{P}_{i-1,i} \boldsymbol{e}_{i-1,i} + \frac{1}{2} e_{\theta i}^2 \tag{10}$$

Its derivative is

$$\dot{V}_{1\ldots n} = -\sum_{i=2}^{n} (\boldsymbol{e}_{i-1,i}^T \boldsymbol{Q}_{i-1,i} \boldsymbol{e}_{i-1,i} + \frac{v_i}{d} e_{\theta i} \sin e_{\theta i} - \eta_i(t, e_{\theta i})) \tag{11}$$

where $\boldsymbol{e}_{i-1,i} = [l_{i-1,i}^d - l_{i-1,i} \quad \pi - \psi_{i-1,i}]^T$ is the output error, and $e_{\theta i} = \theta_{i-1} - \theta_i$ is the orientation error between R_{i-1} and R_i. If the leader's trajectory is well-behaved (same assumptions as Theorem 1), then the convoy-like system can be shown to be stable.

C. Leader–Obstacle Control

This controller (denoted SD_oC) allows the follower to avoid obstacles while following a leader with a desired separation. Thus, the outputs of interest are the separation l_{ij}, and the distance δ between the reference point P_j on the follower and the closest point O on the object. We define a *virtual* robot R_o as shown in Figure 3, which moves on the obstacle's boundary. For this case the kinematic equations are given by

$$\dot{\boldsymbol{z}}_j = \boldsymbol{G}_2(\boldsymbol{z}_j)\boldsymbol{u}_j + \boldsymbol{F}_2(\boldsymbol{z}_j, \boldsymbol{u}_i), \quad \theta_j = \omega_j, \tag{12}$$

where $\boldsymbol{z}_j = [l_{ij} \quad \delta]^T$ is the system output, $\boldsymbol{u}_j = [v_j \quad \omega_j]^T$ is the input for R_j, and

$$\begin{aligned} \boldsymbol{G}_2 &= \begin{pmatrix} \cos\gamma_{ij} & d\sin\gamma_{ij} \\ \sin\gamma_{oj} & d\cos\gamma_{oj} \end{pmatrix} \\ \boldsymbol{F}_2 &= \begin{pmatrix} -v_i \cos\psi_{ij} \\ 0 \end{pmatrix} \\ \gamma_{oj} &= \theta_o - \theta_j. \end{aligned}$$

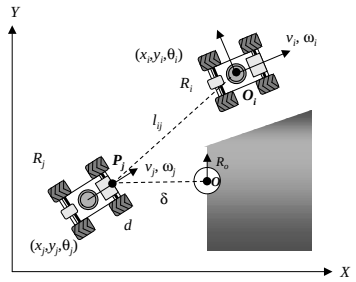

Fig. 3. The Leader–Obstacle Controller.

By applying input-output feedback linearization, the control velocities for the *follower* are given by

$$\boldsymbol{u}_j = \boldsymbol{G}_2^{-1}(\boldsymbol{p}_2 - \boldsymbol{F}_2), \tag{13}$$

where $\boldsymbol{p}_2$ is an auxiliary control input given by

$$\boldsymbol{p}_2 = \begin{pmatrix} k_1(l_{ij}^d - l_{ij}) \\ k_o(\delta_o - \delta) \end{pmatrix} \quad = \boldsymbol{k}(\boldsymbol{z}_j^d - \boldsymbol{z}_j),$$

$k_1, \; k_o > 0$ are the design controller gains, and δ_o is the desired distance from R_j to an obstacle. The closed-loop linearized system is simply given by

$$\dot{\boldsymbol{z}}_j = \boldsymbol{p}_2, \quad \dot{\theta}_j = \omega_j. \tag{14}$$

Remark 2: It is worth noting that feedback input-output linearization is possible as long as $d\cos(\gamma_{oj} - \gamma_{ij}) \neq 0$, *i.e.*, the controller is not defined if $\gamma_{oj} - \gamma_{ij} = \pm k\frac{\pi}{2}$. This occurs when vectors $O\vec{P}_j$ and $O_i\vec{P}_j$ are collinear, which should never happen in practice.

Remark 3: By using this controller a follower robot will avoid the nearest obstacle within its *field-of-view* while keeping a desired distance from the leader. This is a reasonable assumption for many outdoor environments of practical interest. While there are obvious limitations to this scheme in maze-like environments, it is not difficult to characterize the set of obstacles and leader trajectories for which this scheme will work.

We now consider a formation of three nonholonomic robots. There are three possible approaches to controlling the formation. The obvious approach is to use two basic lead-follower controllers: either $SB_{12}C$ and $SB_{13}C$, or $SB_{12}C$ and $SB_{23}C$ (assuming the leader has the label 1) [28]. The two other approaches are more attractive in terms of robustness to noise and are discussed next.

D. Dilation Control

This controller allows robot R_2 to maintain a desired separation ρ^d and desired bearing ψ_{12}^d with respect to R_1, and allows R_3 to follow R_1 and R_2 with desired relative bearings ψ_{13}^d and ψ_{23}^d, respectively (see Figure 4). By changing the *dilation* factor, ρ, the formation can be contracted or expanded in size while preserving the shape. The kinematic equations for this $SB_{12}B_{123}$ controller become

$$\dot{\boldsymbol{z}}_3 = \boldsymbol{G}_3(\boldsymbol{z}_3)\bar{\boldsymbol{u}}_3 + \boldsymbol{F}_3(\boldsymbol{z}_3, \boldsymbol{u}_1), \quad \dot{\theta}_2 = \omega_2, \quad \dot{\theta}_3 = \omega_3, \tag{15}$$

where $\boldsymbol{z}_3 = [\rho \quad \psi_{12} \quad \psi_{13} \quad \psi_{23}]^T$ is the system output, $\bar{\boldsymbol{u}}_3 = [v_2 \quad \omega_2 \quad v_3 \quad \omega_3]^T$ is the input vector, and

$$\boldsymbol{G}_3 = \begin{pmatrix} \cos\gamma_{12} & d\sin\gamma_{12} & 0 & 0 \\ \frac{-\sin\gamma_{12}}{l_{12}} & \frac{d\cos\gamma_{12}}{l_{12}} & 0 & 0 \\ 0 & 0 & \frac{-\sin\gamma_{13}}{l_{13}} & \frac{d\cos\gamma_{13}}{l_{13}} \\ \frac{\sin\psi_{23}}{l_{23}} & -1 & \frac{-\sin\gamma_{23}}{l_{23}} & \frac{d\cos\gamma_{23}}{l_{23}} \end{pmatrix}$$

$$\boldsymbol{F}_3 = \begin{pmatrix} -v_1\cos\psi_{12} \\ \frac{v_1\sin\psi_{12}}{l_{12}} - \omega_1 \\ \frac{v_1\sin\psi_{13}}{l_{13}} - \omega_1 \\ 0 \end{pmatrix}$$

$$\gamma_{ij} = \theta_i + \psi_{ij} - \theta_j.$$

Applying the same approach to control using input-output feedback linearization can as in the previous subsection, we have

$$\bar{\boldsymbol{u}}_3 = \boldsymbol{G}_3^{-1}(\boldsymbol{p}_3 - \boldsymbol{F}_3), \tag{16}$$

where $\boldsymbol{p}_3$ is an auxiliary control input given by

$$\boldsymbol{p}_3 = \begin{pmatrix} k_1(l_{12}^d - l_{12}) \\ k_2(\psi_{12}^d - \psi_{12}) \\ k_2(\psi_{13}^d - \psi_{13}) \\ k_2(\psi_{23}^d - \psi_{23}) \end{pmatrix} = \boldsymbol{k}(\boldsymbol{z}_3^d - \boldsymbol{z}_3),$$

$k_1,\ k_2 > 0$ are the design controller gains. The linearized closed-loop system becomes

$$\dot{\boldsymbol{z}}_3 = \boldsymbol{p}_3, \quad \dot{\theta}_2 = \omega_2, \quad \dot{\theta}_3 = \omega_3. \tag{17}$$

As it can be seen R_3 has two leaders to follow. Therefore, the leaders' trajectories must satisfy certain requirements as given in the next result.

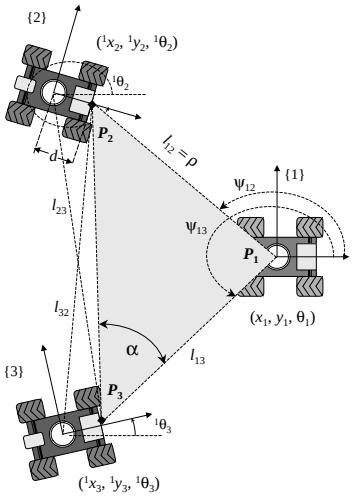

Fig. 4. 3-Robot Formation Control Geometry.

Theorem 2: Assume that the lead vehicle's linear velocity along the path $g(t) \in SE(2)$ is lower bounded, *i.e.*, $v_1 \geq V_{\min} > 0$, its angular velocity is also bounded, *i.e.*, $\|\omega_1\| < W_{\max}$, the relative velocity $\delta_v \equiv v_1 - v_2$, relative angular velocity $\delta_\omega \equiv \omega_1 - \omega_2$, and relative orientation $\delta_\theta \equiv \theta_1 - \theta_2$ are bounded by small positive numbers $\varepsilon_1,\ \varepsilon_2,\ \varepsilon_3$, and the initial relative orientation $\|\theta_1(t_0) - \theta_j(t_0)\| < c_j\pi$ with $c_j < 1$ and $j = 2,\ 3$. If the control input Eq. 16 is applied to $R_{2,3}$, then the formation is stable and the system output $\boldsymbol{z}_3$ in Eq. 17 converges exponentially to the desired value $\boldsymbol{z}_3^d$.

Proof: Let the system error $\boldsymbol{e} = [e_1 \cdots e_6]^T$ be defined as

$$e_1 = \rho^d - \rho, \quad e_2 = \psi_{12}^d - \psi_{12}, \quad e_3 = \theta_1 - \theta_2,$$
$$e_4 = \psi_{13}^d - \psi_{13}, \quad e_5 = \psi_{23}^d - \psi_{23}, \quad e_6 = \theta_1 - \theta_3. \tag{18}$$

By theorem 1, the internal dynamics of R_2 are stable *i.e.*, the orientation error e_3 is bounded. Now for R_3, we require to consider the conditions on relative velocities and orientations of R_1 and R_2.

$$\dot{e}_6 = \omega_1 - \omega_3$$

after some work, we have

$$\dot{e_6} = -\frac{v_1}{d}\sin e_6 + \eta_2(e_6, \omega_1, e_4, e_5, \delta_v, \delta_\theta, \delta_\omega) \tag{19}$$

where the nominal system *i.e.*, $\eta_2(t, e_6) = 0$ is (locally) exponentially stable provided that the velocity of the lead robot $v_1 > 0$ and $\|e_6\| < \pi$. Since $\|\omega_1\| < W_{\max}$, $\|\delta_v\| < \varepsilon_1$, $\|\delta_\omega\| < \varepsilon_2$ and $\|\delta_\theta\| < \varepsilon_3$, it can be shown that $\|\eta_2(t, e_6)\| \leq \delta_2$. Knowing that $\|e_6(t_0)\| < c_3\pi$ for some positive constant $c_3 < 1$, then

$$\|e_6(t)\| \leq \sigma_2, \qquad \forall\, t \geq t_2$$

for some finite time t_2 and positive number σ_2. □

Remark 4: The dilation control strategy is useful because it explicitly decouples the scale ρ from the shape (internal angles of the triangle). In applications like cooperative mapping [35], it is beneficial to preserve the scale. In cooperative manipulation, it is useful to keep the shape, which determines the grasp on the enclosed object, constant and vary the scale, which determines the maximum clearance between the robots and the enclosed object. In contrast to using two separation-bearing controllers, this formation uses only one estimate of distance and two estimates of angle. Vision sensors and in particular the geometry of our omnidirectional camera allows very accurate estimates of angle, while distance measurements are relatively poor.

E. Formation Shape Control

The formation shape controller (denoted $S_{13}S_{23}C$), allows robot R_3 to follow R_1 and R_2 with desired separations l_{13}^d and l_{23}^d, respectively, while R_2 follows R_1 with $SB_{12}C$. Again, the kinematic equations are given by

$$\dot{\boldsymbol{z}}_4 = \boldsymbol{G}_4(\boldsymbol{z}_4)\boldsymbol{u}_4 + \boldsymbol{F}_4(\boldsymbol{z}_4, \boldsymbol{u}_1), \quad \dot{\theta}_2 = \omega_2, \quad \dot{\theta}_3 = \omega_3, \tag{20}$$

where $\boldsymbol{z}_4 = [l_{12} \quad \psi_{12} \quad l_{13} \quad l_{23}]^T$ is the system output, $\boldsymbol{u}_4 = [v_2 \quad \omega_2 \quad v_3 \quad \omega_3]^T$ is the input vector, and

$$\boldsymbol{G}_4 = \begin{pmatrix} \cos\gamma_{12} & d\sin\gamma_{12} & 0 & 0 \\ \frac{-\sin\gamma_{12}}{l_{12}} & \frac{d\cos\gamma_{12}}{l_{12}} & 0 & 0 \\ 0 & 0 & \cos\gamma_{13} & d\sin\gamma_{13} \\ -\cos\psi_{23} & 0 & \cos\gamma_{23} & d\sin\gamma_{23} \end{pmatrix}$$

$$\boldsymbol{F}_4 = \begin{pmatrix} -v_1\cos\psi_{12} \\ \frac{v_1\sin\psi_{12}}{l_{12}} - \omega_1 \\ -v_1\cos\psi_{13} \\ 0 \end{pmatrix}$$

By applying input-output feedback linearization, the control velocities for the *follower* robots are given by

$$\boldsymbol{u}_4 = \boldsymbol{G}_4^{-1}(\boldsymbol{p}_4 - \boldsymbol{F}_4) \tag{21}$$

where $\boldsymbol{p}_4$ is an auxiliary control input given by

$$\boldsymbol{p}_4 = \begin{pmatrix} k_1(l_{12}^d - l_{12}) \\ k_2(\psi_{12}^d - \psi_{12}) \\ k_1(l_{13}^d - l_{13}) \\ k_1(l_{23}^d - l_{23}) \end{pmatrix} = \boldsymbol{k}(\boldsymbol{z}_4^d - \boldsymbol{z}_4).$$

The closed-loop linearized system is given by

$$\dot{\boldsymbol{z}}_4 = \boldsymbol{p}_4, \quad \dot{\theta}_2 = \omega_2, \quad \dot{\theta}_3 = \omega_3. \tag{22}$$

As before, we will show that the closed-loop system is stable, and the robots navigate keeping formation.

Theorem 3: Assume that the lead vehicle's linear velocity along the path $g(t) \in SE(2)$ is lower bounded *i.e.*, $v_1 \geq V_{\min} > 0$, its angular velocity is also bounded *i.e.*, $\|\omega_1\| < W_{\max}$, the relative velocity $\delta_v \equiv v_1 - v_2$ and relative orientation $\delta_\theta \equiv \theta_1 - \theta_2$ are bounded by small positive numbers ε_1, ε_2, and the initial relative orientation $\|\theta_1 - \theta_j\| < c_j\pi$ with $c_j < 1$ and $j = 2,\ 3$. If the control input Eq. 21 is applied to $R_{2,3}$, then the formation is stable and the system output $\boldsymbol{z}_4$ in Eq. 22 converges exponentially to the desired value $\boldsymbol{z}_4^d$.

The proof is similar to Theorem 2.

Remark 5: In contrast to the previous two three-robot formation controllers, this controller allows explicit control of all separations and minimizes the risk for collisions. This controller is preferred when the separations between robots are small, and when, coincidentally, the estimates of distance through vision are better.

IV. Coordination Protocol

A. Choice of Formations

In Section III, we have shown that under certain assumptions a group of robots can navigate maintaining a stable formation. However, in real situations mobile robotic systems are subject to sensor, actuator and communication constraints, and have to operate within unstructured environments. We describe a *switching paradigm* that allows robots to select the most appropriate controller (formation) depending on the environment.

In this work, we model the group of n autonomous mobile robots as a tuple $\mathcal{F} = (g, \boldsymbol{r}, \mathcal{H})$ where $g(t) \in SE(2)$ is the reference trajectory of the lead robot, $\boldsymbol{r}$ is a set of *shape* vectors describing the relative positions of each vehicle with respect to the reference formation frame $\{M\}$, and $\mathcal{H}$ is a *control graph* where nodes represent robots and edges represent relations between nodes (see details in following subsection) [25]. Thus, $\mathcal{F}$ describes a dynamical system evolving in continuous-time on the interval $T = [t_0, t_N] \subset \mathbb{R}^+$ in the configuration space $\mathcal{C} = SE(2)^n$. Without loss of generality, the formation reference frame $\{M\}$ is fixed to the lead robot; however, it is not a requirement in our method. Sometimes it is necessary to add *virtual* robots to the group to represent either moving targets, or trajectories that are along such features as walls, lanes, or obstacles. While this results in a change in $\mathcal{F}$, it does not change the configuration space of the system.

The control graphs describing the formation are designed from the basic controllers described in the previous section. The enumeration of control graphs for n robots is discussed in [28]. Let $U_j = \{\xi_{1,j}, \ldots, \xi_{p,j}\}$ be the set of available controllers for robot R_j. We consider the problem of selecting the controller, $\xi_{k,j} \in U_j$ for robot R_j, assuming that the controllers for robots $R_2, R_3, \ldots, R_{j-1}$ have been specified.

We illustrate this approach using three nonholonomic mobile robots $R_{1,2,3}$ equipped with range sensors. First, R_1, the *reference* robot, follows a given trajectory $g(t) \in SE(2)$. Since R_2 can only follow R_1 (because of the numbering constraint), $U_2 = \{SB_{12}C\}$. Thus R_2 follows R_1 with $SB_{12}C$. The set of for R_3 now has three controllers: $U_3 = \{SB_{13}C, SB_{23}C, S_{13}S_{23}C\}$. Thus, as shown in Figure 5, R_3 may follow R_1 or R_2 with $SB_{13}C$ or $SB_{23}C$, or follow both R_1 *and* R_2 with $S_{13}S_{23}C$. The palette of controllers for the three-robot group becomes $U = \{U_2 \times U_3\}$. Each member of this palette corresponds to a different control graph and a different mode.

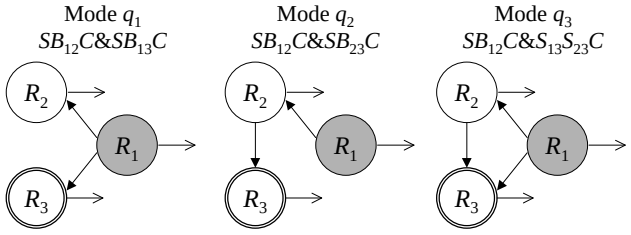

Fig. 5. The three control graphs for the 3-robot case. Each graph denotes a different mode (q_1, q_2, and q_3) for R_3.

If the assumptions in Theorems 1 and 3 hold, then each mode q_i with $i = 1,\ 2,\ 3$ is stable. We need to show that for a given switching strategy S_w, the hybrid system is stable, *i.e.*, given any initial mode q_i^0, a desired mode q_i^d is achieved in finite time.

Our switching strategy is guided primarily by our sensor (omnidirectional camera) constraints and the presence of obstacles. Figure 6 depicts the switching boundaries in Cartesian space where r_2 denotes the *maximum range* within which a neighbor robot can be detected. $r_1 < r_2$ is a predefined range where a robot may detect two possible leaders. To be more specific, R_3 may detect R_1, R_2 or both. In some cases, neither R_1 nor R_2 are within the field–of–view of R_3. Notice the triangle inequality $l_{ik} + l_{jk} > l_{ij}$ should be satisfied. If R_i with $i = 1, 2, 3$ were collinear, *SSC* would not be defined, then a *SBC* should be utilized.

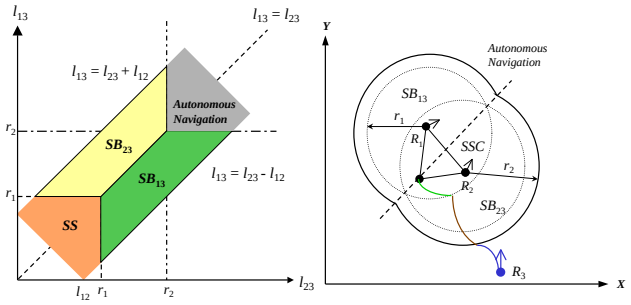

Fig. 6. Choice of controllers for R_3. The plot on the right shows the constraints and equilibrium point in cartesian $x - y$ coordinates.

The formation control objective is to drive R_3 to a region where it can detect both R_1 and R_2. Thus, the switching control strategy for R_3 can be summarized as follows

$$\begin{aligned}
\text{If } (l_{13} < l_{23})\&(l_{23} > r_1)\&(l_{13} < r_2) \text{ Then } SB_{13}C \\
\text{If } (l_{13} > l_{23})\&(l_{13} > r_1)\&(l_{23} < r_2) \text{ Then } SB_{23}C \\
\text{If } (l_{13} < r_1)\&(l_{23} < r_1) \text{ Then } S_{13}S_{23}C \\
\text{If } (l_{13} > r_2)\&(l_{23} > r_2) \text{ Then } AutonNavig
\end{aligned}$$

The set of control behaviors that a robot may exhibit when there is no leader within its field-of–view is called *Autonomous Navigation* here.

Figure 7 depicts the switching boundaries in the presence of obstacles. Here δ_{safe} denotes a *safety* region within which an obstacle can be detected, δ_o is the desired distance from the robot to the obstacle, and β_{1o} is the angle between $\vec{\delta}$ and $\vec{l_{12}}$.

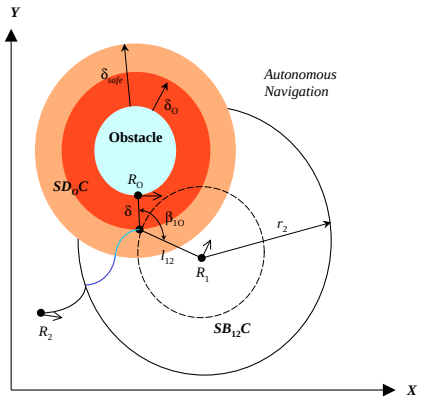

Fig. 7. Choice of controllers for R_2 in presence of obstacles.

Let us assume R_2 follows R_1 with $SB_{12}C$, if an obstacle is detected, then R_2 switches to SD_OC. Once the obstacle has been successfully negotiated, R_2 switches back to $SB_{12}C$ according to the following switching rules.

$$\begin{aligned}
\text{If } (\delta < \delta_{safe})\&(\beta_{1o} < \pi)\&(l_{12} < r_2) \text{ Then } SD_OC \\
\text{If } (l_{12} < r_2)\&(\delta > \delta_{safe}) \text{ Then } SB_{12}C \\
\text{If } (l_{12} > r_2)\&(\delta > \delta_{safe}) \text{ Then } AutonNavig
\end{aligned}$$

B. Stability Analysis

Since a palette of controllers and a switching strategy are given, we need to *verify* that the hybrid system is stable provided that each mode shares a common equilibrium point $\boldsymbol{x}_0$. One way to solve this verification problem is to find a common Lyapunov function, thus the switched system is stable for any arbitrary fast switching sequence. This is in general a difficult task. A number of approaches have been proposed in the literature to confront this problem (see [37] and the references therein). In our 3-robot formation example, it turns out that under some reasonable assumptions, there may exist a common Lyapunov function. Therefore, the equilibrium point is stable, and the system error of the desired formation mode converges to zero. However, the property of exponential convergence is lost in the switching process.

Let the system error be defined as

$$\begin{aligned}
e_1 &= l_{13}^d - l_{13} \quad e_2 = \psi_{13}^d - \psi_{13}, \quad e_3 = \theta_1 - \theta_3 \\
e_4 &= l_{23}^d - l_{23}, \quad e_5 = \psi_{23}^d - \psi_{23}, \quad e_6 = \theta_2 - \theta_3 \\
e_7 &= l_{12}^d - l_{12}, \quad e_8 = \psi_{12}^d - \psi_{12}, \quad e_9 = \theta_1 - \theta_2,
\end{aligned}$$

and a Lyapunov function candidate for the desired formation $\mathcal{F}_3$ be given by

$$V(\boldsymbol{e}) = V_3 + V_{12}$$

where

$$V_3 = \frac{1}{2}\left[e_1^2 + e_4^2 + e_3^2\right], \quad V_{12} = \frac{1}{2}\left[e_7^2 + e_8^2 + e_9^2\right] \tag{23}$$

V_{12} is a Lyapunov function candidate for subsystem $SB_{12}C$ *i.e.*, R_2 follows R_1 using a basic leader-following controller. If the assumptions in Theorem 1 are satisfied, then $\dot{V}_{12} \leq 0$. Moreover, if the assumptions in Theorem 3 are satisfied for subsystem $S_{13}S_{23}C$, then $\dot{V}_3 \leq 0$. Since $SB_{12}C$ is common for all modes, we only need to consider V_3 in Eq. 23 for studying the stability of the switched system.

By definition V_3 is a Lyapunov function for mode q_3. We would like to show that, V_3 is also a Lyapunov function for q_1 and q_2. Let us consider formation mode q_1. $SB_{13}C$ makes $e_1 \to 0$ and $e_2 \to 0$ exponentially as $t \to \infty$. But we need to show that $e_4 \to 0$. To accomplish this, let us define $V_4 = 1/2(e_4^2)$, then show that $\dot{V}_4 = e_4\dot{e}_4 \leq 0$ or $(l_{23}^d - l_{23})\dot{l}_{23} \geq 0$. The main idea here is to pick ψ_{13}^d such that $l_{23} \to l_{23}^d$ as $e_2 \to 0$. Thus, we have

$$\psi_{13}^d = \cos^{-1}\left(\frac{{l_{12}^d}^2 + {l_{13}^d}^2 - {l_{23}^d}^2}{2l_{12}^d l_{13}^d}\right) + \psi_{12}^d \tag{24}$$

Using the inequality constraint imposed by the geometry of the problem, *i.e.*, $l_{23}^d < l_{12}^d + l_{13}^d$, it is easy to show that $\dot{V}_4 = e_4\dot{e}_4 \leq 0$. Then V_3 is a Lyapunov function for q_1 (similarly for q_2).

Remark 6: It is well known that Lyapunov methods provide conservative stability regions, since we always consider the worst case. Simulation results reveal that the desired formation is achieved even when some of the assumptions discussed here are not satisfied, *e.g.*, position and orientation of $R_{2,3}$ are randomly initialized.

V. Sensing and Estimation

The sole physical sensor used by the robots in our experiments is the on-board catadioptric camera system. From the omnidirectional imagery acquired by these cameras, we have developed several *logical sensors* – an *obstacle detector*, a *collision detector*, a *decentralized state observer*, and a *centralized state observer*. All of these logical sensors trace their origins to one of two sets of features in the omnidirectional image – extracted edges and segmented colors (see Figure 1). The obstacle and collision detectors rely on edge images as input. Implementation details for these can be found in our previous work [35].

In contrast, the remaining sensors rely on color segmentation to identify targets in the image. To facilitate this, each robot is equipped with a colored cylindrical collar. This yields a 360°

symmetrical target about each robot's optical axis. We then segment the image for these colors by using an extractor operating in YUV colorspace. Our implementation takes advantage of YU and YV look-up tables to significantly reduce processing time, and segments up to 8 colors simultaneously. By applying a blob extractor to the color segmented image, each robot is able to isolate teammates within its own image.

Next, we exploit the characteristics of the omnidirectional cameras. One of their primary advantages in this application is that catadioptric camera systems afford a single effective point of projection. This means that, after an appropriate calibration, every point in the omnidirectional image can be associated with a unique ray through the focal point of the camera. As a result, by taking the center of gravity (CG) of the extracted collars in the color segmented image, each robot can compute reliable estimates of the direction vectors to its teammates. These directions provide the basis for both centralized and decentralized state observation.

A. Decentralized State Observation

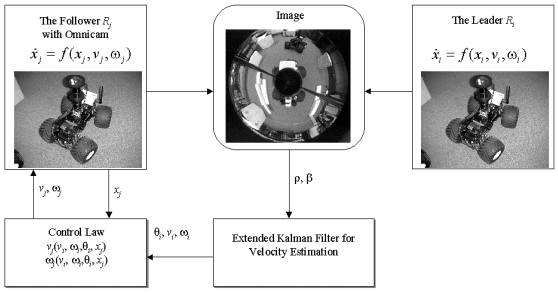

Fig. 8. Leader-follower estimation framework

The controllers described in Section III, require reliable estimation of the linear velocity $v_i(t)$ and angular velocity $\omega_i(t)$ of the leader mobile robot R_i by follower robot R_j, and relative orientation $(\theta_i - \theta_j)$. Our algorithm estimates a leader's velocity and the relative position and orientation using an extended Kalman filter [38]. It uses the omni-directional vision system to determine the range ρ_{ij} and the bearing β_{ij} of the observed leader R_i as needed by follower R_j for estimation of v_i and w_i. The linear angular velocities of the observed vehicle are treated as part of the state. In addition, the filter requires a sensor model, and the relative kinematics (see Eq. 1) of the leader R_i and follower R_j.

The image processing algorithms provide the following observations

$$\begin{aligned} \rho_{ij}^2 &= (x_i - x_j)^2 + (y_i - y_j)^2 \\ \beta_{ij} &= \frac{\pi}{2} + \mathrm{a\,tan\,2}(y_i - y_j, x_i - x_j) - \theta_j \end{aligned} \tag{25}$$

Next we differentiate (25) to obtain $\dot{\rho}_{ij}$ and $\dot{\beta}_{ij}$. Using the kinematic equations (1), our extended state vector then becomes

$$\dot{\bar{\boldsymbol{x}}} = f(\bar{\boldsymbol{x}}, \boldsymbol{u}, \boldsymbol{w}) \tag{26}$$

$$\begin{pmatrix} \dot{\theta}_i \\ \dot{v}_i \\ \dot{\omega}_i \\ \dot{\rho}_{ij} \\ \dot{\beta}_{ij} \\ \dot{\theta}_j \end{pmatrix} = \begin{pmatrix} \omega_i \\ 0 \\ 0 \\ v_i \sin\alpha_{ij} - v_j \sin\beta_{ij} \\ \frac{v_i \cos\alpha_{ij} - v_j \cos\beta_{ij}}{\rho_{ij}} - \omega_j \\ \omega_j \end{pmatrix} + \boldsymbol{w}(t) \tag{27}$$

where $\alpha_{ij} = \beta_{ij} + \theta_j - \theta_i$, $\boldsymbol{w}(t)$ is the process noise, and we assume $\dot{v}_i \approx 0, \dot{w}_i \approx 0$. The system output with sensor noise is given by

$$\boldsymbol{z}(t) = \boldsymbol{h}(\bar{\boldsymbol{x}}) + \boldsymbol{\eta}(t) = [\rho_{ij} \quad \beta_{ij}]^T \tag{28}$$

The discrete system becomes

$$\bar{\boldsymbol{x}}(k+1) = \boldsymbol{F}(\bar{\boldsymbol{x}}(k), \boldsymbol{u}(k)) + \boldsymbol{w}(k), \quad \boldsymbol{w}(k) \sim N(0, \boldsymbol{Q}(k)) \tag{29}$$

where $\boldsymbol{F}(\bar{\boldsymbol{x}}(k),\boldsymbol{u}(k))$ is the nonlinear state transition function. The input vector is given by $\boldsymbol{u} = [v_j \quad \omega_j]^T$. $\boldsymbol{w}(k)$ is a noise source assumed to be zero-mean Gaussian with covariance $\boldsymbol{Q}(k)$. We use a sampling interval $\Delta T \sim 50$ ms. The discrete (observation) output is given by

$$\boldsymbol{Z}(k) = \boldsymbol{h}(\bar{\boldsymbol{x}}(k)) + \boldsymbol{\eta}(k), \quad \boldsymbol{\eta}(k) \sim N(0, \boldsymbol{R}(k)) \tag{30}$$

The covariance $\boldsymbol{R}(k)$ is experimentally determined. The goal of the EKF algorithm is to estimate $\hat{\bar{\boldsymbol{x}}}(k+1|k+1)$ and its covariance $\boldsymbol{P}(k+1|k+1)$ given $\hat{\bar{\boldsymbol{x}}}(k|k)$ and $\boldsymbol{P}(k|k)$ at time k, and the current observation $\boldsymbol{Z}(k+1)$. We use a standard estimation algorithm, see for instance [39], where the observation vector and measurement prediction are given by

$$\boldsymbol{Z}(k+1) = [\rho(k+1) \quad \beta(k+1)]^T \tag{31}$$

$$\hat{\boldsymbol{Z}}(k+1) = \boldsymbol{H}\hat{\bar{\boldsymbol{x}}}(k+1|k) \tag{32}$$

with

$$\boldsymbol{H} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{33}$$

The decentralized state observer provides the follower with necessary information about the velocity of the leader for feedforward control, in addition to the relative state (position and orientation). This eliminates the need for explicit communication. The basic structure for this algorithm is shown in Figure 8.

B. Centralized State Observation

The centralized observer relies upon information sharing between robots to solve for the team pose (position and orientation) in closed form. The resulting estimate is more robust than that obtained in the decentralized case since the state is fully observable with each observation; the need to estimate the velocity for state prediction is eliminated. However, this comes at the cost of communication. In our implementation, the centralized observer uses two methods for estimating the team pose: *triangulation-based* and *pair-wise* localization.

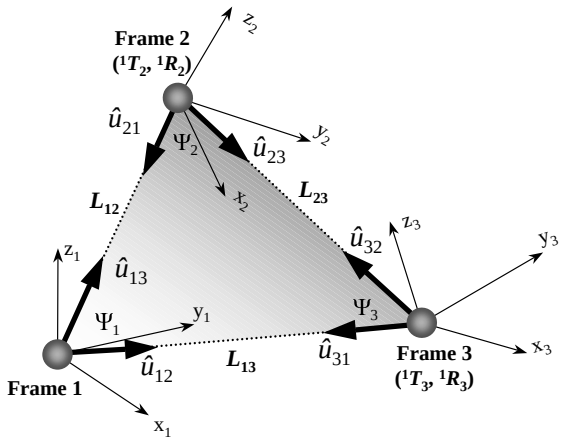

Fig. 9. Three-dimensional geometry for agent localization.

Using the triangulation-based method, a team of three (or more) robots is capable of localizing in $3D$-space when each can measure the direction vectors to the other team members. In Figure 9 the unit vectors $\hat{u}_{ij} \in \mathbb{R}^3$ denote the direction between robot i and robot j expressed in the coordinate frame of robot i. Let ${}^iT_j \in \mathbb{R}^3$ and ${}^iR_j \in SO(3)$ represent respectively the translation and rotation of robot j with respect to the frame of reference of robot i. These direction vectors are derived from the images using the procedure described in the previous paragraphs. Without loss of generality we can choose the reference frame of robot 1 as our base frame of reference and recover the configuration of the robot team by recovering the positions and orientations of the other robots with respect to this frame.

In each frame, the internal angle between the direction vectors to the other two robots (ψ_i) can be determined from their scalar product; for instance $\psi_2 = \cos^{-1}(\hat{u}_{21} \cdot \hat{u}_{23})$. With this angle information, the translation between the frames can readily be determined to a scale factor by applying the sine rule to the shaded triangle in Figure 9. Position vectors relative to other frames can also be obtained by using the corresponding unit vectors.

With the position of agents known, we only require the relative orientations of the frames to complete the localization procedure. To accomplish this, we note that the vectors ${}^jT_i$ and ${}^iT_j$ should have equal magnitude, but opposite direction when transformed to the same frame. We note a similar relationship between the vectors $({}^jTi - {}^jTk)$ and ${}^iT_k$. From these, we obtain the following pairs of equations.

$$\begin{aligned} -{}^1T_2 &= {}^1\boldsymbol{R}_2{}^2T_1, & {}^1T_3 - {}^1T_2 &= {}^1\boldsymbol{R}_2{}^2T_3 \\ -{}^1T_3 &= {}^1\boldsymbol{R}_3{}^3T_1, & {}^1T_2 - {}^1T_3 &= {}^1\boldsymbol{R}_3{}^3T_2 \end{aligned} \tag{34}$$

With all translation vectors known to a scale factor, the problem of solving for each rotation matrix reduces to the form:

$$\boldsymbol{R}a_i = b_i \qquad i \in [1,2] \tag{35}$$

This can be rephrased as the following optimization problem:

$$\min_{R \in SO(3)} \sum_i \left\| \boldsymbol{R}a_i - b_i \right\|^2 \tag{36}$$

The rotation matrix which minimizes this expression can be computed in closed form as follows:

$$\boldsymbol{R} = (M^T M)^{-1/2} M^T \tag{37}$$

where $M = \sum_i a_i b_i^T$ [40].

Again recall that this solution yields the pose of the team *to a scale factor*. In order to obtain metric results, a means to recover the scale is necessary. This can be accomplished if the length of any one of the translation vectors between frames can be determined. In our experiments the robots were constrained to move on a flat surface. Since the geometry of each robot was known, any robot could gauge the distance to its teammates based on the radial distance to the extracted blobs in the image. The smaller the range, the closer the blob will appear to the image center. As a result, we have a means by which each robot can provide two estimates of the scale (one for each of its visible partners). We use the redundant estimates from all three to obtain the overall scale factor and the relative pose of the team.

This solution offers an improvement over methods presented previously, in that we obtain the relative position and orientation of the robot team solely from angular measurements without requiring that the angular estimates be referenced to a common axis like the gravity vector. This eliminates the need for the additional sensors that were required to measure agent orientation in previous implementations [41]. However, it does not eliminate the singularity associated with linear formations. Additionally, it requires that all three robots maintain line-of-sight with each other. This is a stringent requirement that does not hold in an obstacle-cluttered environment. We note though that when the pose problem is reduced to $2D$-space, relative localization can be accomplished by a pair of robots. Using this fact, our implementation dynamically switches between triangulation-based and pair-wise localization estimation based on team geometry and the external environment.

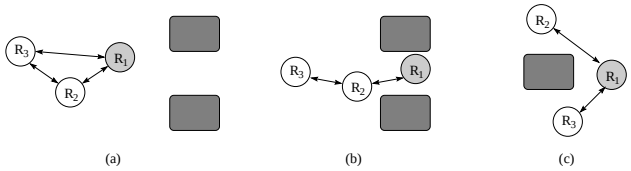

Fig. 10. Triangular to pair-wise localization switch resulting from team geometry (a-b) or occlusions in the environment (c).

Consider the case of a triangular formation approaching a narrow passage through obstacles shown in Figure 10. A formation switch is ordered to allow the team to proceed through the passage (Figure 10a). As the robots approach a linear formation, there comes a point where the improved accuracy afforded by the closed form solution of the triangulation-based localizer is compromised by operating in proximity to its singularity. This point is a function of the sensors used, but occurs when the error in estimating the interior angles becomes a significant percentage of the angle size itself (in our implementation, this occurred when interior angles approached 10 degrees). At this point, the centralized observer automatically switches to pair-wise localization mode (Figure 10b). Robot R_2 exchanges information

with the team leader (R_1) to localize relative to the leader's frame. R_3 performs a similar exchange with R_2, obtains a localization estimate relative to R_2, and as a result determines its pose relative to R_1.

While this mode switch resulted from the formation geometry, it can also be directly triggered by the environment. This is shown in Figure 10c, where the line-of-sight between two robots is occluded by an obstacle. This occlusion can be detected from a global visibility matrix, resulting in a pair-wise localization switch.

The pair-wise method serves as the secondary localization mode for the centralized observer. In most formation geometries, the constraint obtained by determining the relative formation scale – along with the redundant range measurements for estimating the absolute scale – result in improved performance in the triangulation-based mode. Mean range errors were typically 3-5%, compared to 10% for the pair-wise case.

The advantages resulting from this internal switching are twofold. It allows the centralized observer to robustly estimate the team state regardless of formation geometry. Additionally, it allows the team to react to an obstacle-cluttered environment with only a slight degradation in accuracy. Since the observer provides only state estimates for use by the controller modes, the switching is transparent to all users.

VI. Results

A. Simulation of Switching Strategy

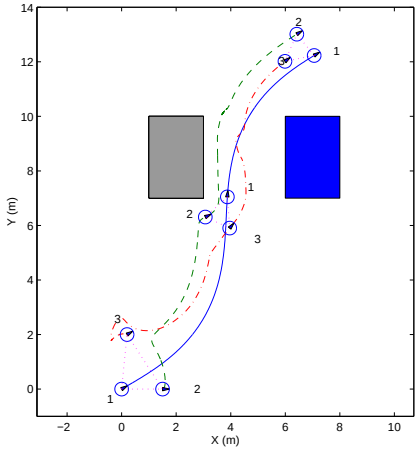

Fig. 11. The leader follows a sinusoidal trajectory while followers switch to avoid obstacle while maintaining desired triangle formation.

In Section IV, we discussed choosing formations and switching strategies for maintaining formation shape whilst ensuring a stable switched system. Here, we illustrate the application of these concepts to a simulation of 3 nonholonomic robots with one obstacle (Figure 11). Robot R_1 is the lead robot, and the desired shape is a triangle. The control mode switching in R_2 and R_3 are shown in Figure 12. The formation shape is achieved and the robots successfully negotiate the obstacle.

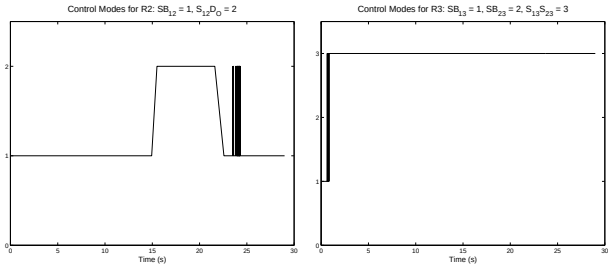

Fig. 12. Mode switching for robot R_2 (left plot) and R_3 (right plot) for trajectories in Figure 11. The numbering of the modes are shown on the top of each plot.

B. Experiments

B.1 Hardware Platform

The cooperative control framework was implemented on the GRASP Lab's Clodbuster^{TM}(CB) robots. The CB platform is based on the Tamiya Clodbuster^{TM} radio controlled $1/10$ scale model truck. Each CB is equipped with an omnidirectional camera (described in Section V) as its sole sensor [42]. The platform lacks on-board processing. As a result, video signals from the camera on-board are sent to a remote computer for processing via a wireless 2.4 GHz video transmitter. Velocity and heading control signals are sent from the host computer to the vehicles as necessary. This reduces the cost and size of the platform, and makes it simple to coordinate the data processing and control operations. Note that each robot can be independently controlled using different host computers. The CB team used for our multi-robot coordination experiments can be seen in Figure 13.

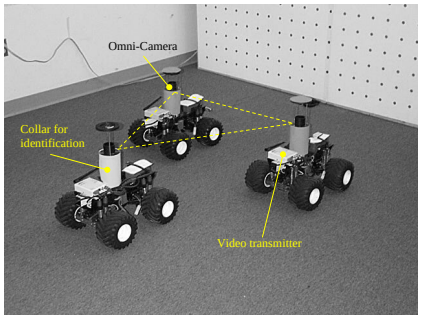

Fig. 13. The Clodbuster^{TM} team used for experiments.

B.2 Formation Control

Initial experiments in formation control were intended to validate the dynamic state estimation implementation and corresponding control approach. As a result, experiments first examined stable formations following trajectories of straight lines, gradual arcs and circles. Video data from these trials were recorded using a calibrated overhead camera. This allowed "ground-truth" position data of the formation to be recorded and analyzed off-line together with the state observer position estimates. Ground plane trajectories from a pair of representative trials can be found in Figure 14.

We next compared the state observer estimates with the ground-truth position data. As an example, in the trial on the left

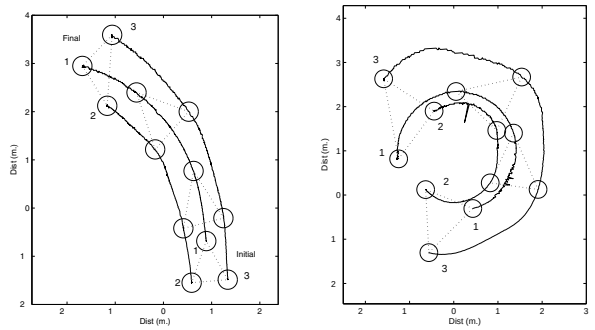

Fig. 14. Sample ground-truth data for trajectories for a triangular formation

side of Figure 14, the desired formation was an isosceles triangle where both followers maintained a distance of 1.0 m from the leader. Figure 15 contrasts the measured leader-follower separation distances with those calculated by the centralized state observer. Results are for the most part satisfactory, with mean separation errors of 3.2% and 5.5% for the two followers. Discontinuities in state observer estimates are due to corrupted image data resulting from the remote video transmission. Typical image corruption rates were 15-20% for each robot, leaving periods of time where no localization was possible.

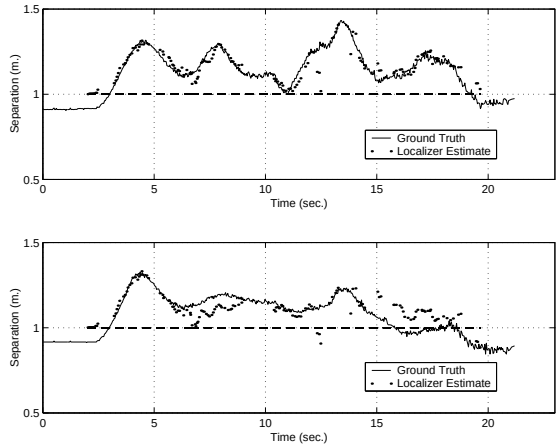

Fig. 15. Follower separation distances – ground-truth vs. centralized observer estimates for followers R_2 (top) and R_3.

A point of interest in these plots is that the actual separation distance is always greater than that desired during motion. This is due to the pure feedback controller used with the centralized observer. Additional experiments with the decentralized observer using an Extended Kalman Filter for velocity estimation were also conducted with improvements in tracking performance due to availability of feed-forward terms. This is shown in Figures 16-17, where the lead robot executed a circle and the follower attempted to maintain 0.6 m separation and a relative bearing (as defined in Section III) of 180 degrees. The controller response is significantly improved as a result. We also examined the robustness of the estimator by manually restraining the follower at $t \sim 65$ sec. As can be seen from the plots, the system recovered quickly.

From these results we conclude that both observers provide

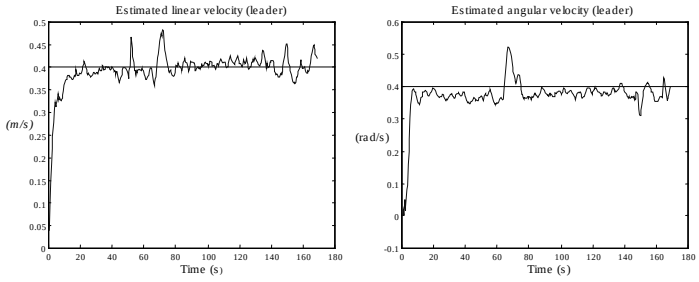

Fig. 16. Leader velocity estimation by the follower. Results are consistent with the actual linear and angular velocities for the leader doing a constant circle (0.4 m/s and circle radius 1.05 m).

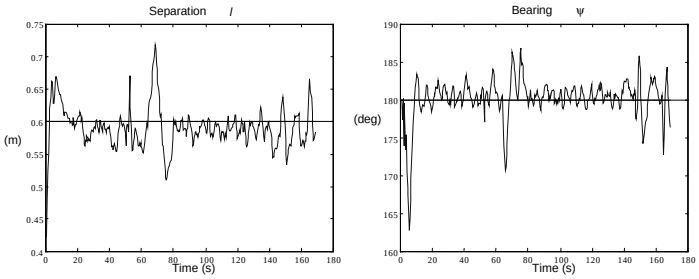

Fig. 17. Follower separation and relative bearing history for a feed-forward controller. Notice the sharp jump at $t \sim 65$ sec as we manually restrained the follower for 5 sec. The controller recovers within a few seconds of removing the restraint.

sufficiently good state estimates. However, despite the superior estimator performance, the control response for the centralized case is compromised by the lack of a feed-forward component. We are currently integrating a centralized velocity estimator to address this.

B.3 Switching Formations

Fig. 18. Triangular to in-line formation switch to avoid obstacles.

In these experiments, the goal was to allow the lead robot in a three robot formation to perform an exploratory mission while the formation shape changes in a decentralized fashion as required by the environment. We implemented this by running a simple reactive *obstacle avoider* [43] on the leader and allowing the team to choose between two formation shapes - an isosceles triangle and an in-line convoy. The role of the followers was to follow the leader while maintaining a triangle if

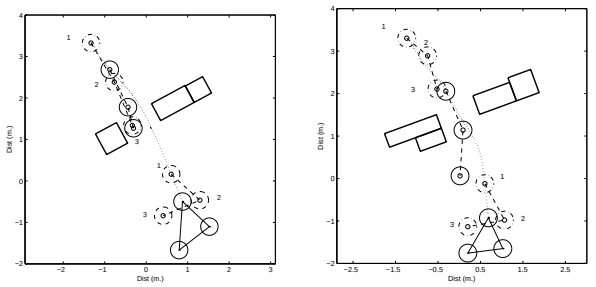

Fig. 19. Ground plane data for formation switching - two runs. The line change from solid to dotted corresponds to the initiation of the switch.

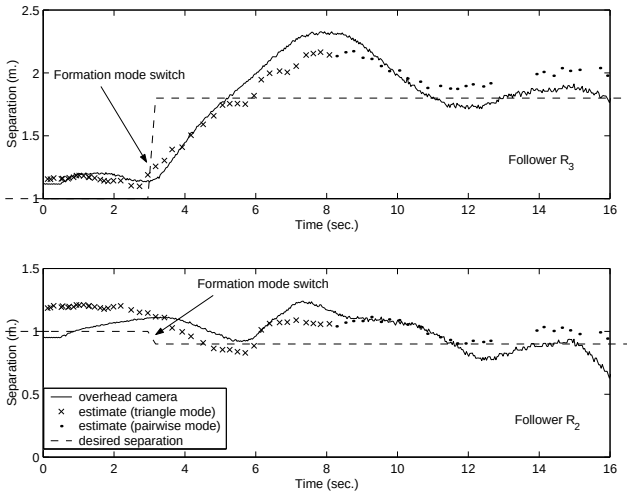

Fig. 20. Ground-truth vs. centralized observer estimates corresponding to the experiment in Figure 19 (right). After approximately 3 seconds, the leader detects the obstacles and triggers a formation switch (triangle to in-line). Note the observer mode switches internally from triangular to pair-wise independent of the formation switch, but dependent on the formation geometry.

there were no obstacles ahead. In the presence of obstacles, the followers switch to an in-line position behind the leader and hence negotiate the obstacles while following the leader. The results are summarized in Figures 18-19. The obstacles used were cardboard boxes and the arrangement shown is to mimic a narrow passageway causing a change in formation shape. We also demonstrate the internal mode switching in our centralized state observer in Figure 20.

B.4 Distributed Manipulation

The ability to maintain a prescribed formation allows the robots to "trap" objects in their midst and to flow the formation – guaranteeing that the object is transported to the desired position. With this in mind, we proceeded to apply this technique to a manipulation application. Experiments were conducted using a box as the object to be manipulated. In Figure 21, the initial team configuration is centered around the box, with the goal to flow the now encumbered formation along a trajectory generated by the leader. By choosing a constraining formation geometry, the box is kept in contact with all three robots during the formation flow. Several snapshots from a sample run are shown in Figure 21.

Despite the control strategy not accounting for changes in the object pose, the formation was typically successful in its manipulation task over the tested trajectories. These experiments, while not an exhaustive investigation of distributed manipulation, demonstrate the potential for a vision based formation control application.

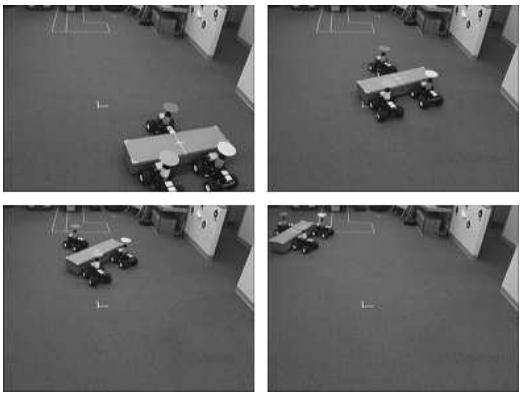

Fig. 21. Distributed manipulation demonstration

VII. Conclusions

In this work we propose a framework for the development of intelligent multi-robot systems by composing simple sensing, estimation, control and coordination blocks in a bottom-up approach. The main contributions are a suite of control and estimation algorithms, and a paradigm for switching that allows a group of robots to maintain a prescribed formation (shape and size) while following a planned trajectory. The switching paradigm also allows the robots to change formation in the presence of obstacles. A distinguishing feature of our work is the fact that each robot relies only on a single omnidirectional camera for sensory information. We demonstrate our framework through experiments with different multi-robot cooperative tasks like exploration and distributed manipulation. Because our controllers and estimators can be decentralized, and the framework allows the selection of the best controller and estimator in a given situation, our framework can potentially scale to groups of tens and hundreds of robots. Analyzing the effect of communication constraints, deciding the optimality of formation choices for a given environment, sensor planning for cooperative active vision and implementing multi-robot coordination tasks with a larger number of robots are important directions for our future work.

Acknowledgments

This work was supported by the DARPA ITO MARS Program, grant no. 130-1303-4-534328-xxxx-2000-0000 and NSF grant no. CDS-97-03220. We thank Ben Southall for his contributions to the development of the software framework and architecture, and Joel Esposito, Greg Grudic, Ken McIsaac, Peng Song, and Zhidong Wang for discussions on multi-robot cooperation and control.

References

[1] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, "Collaborative multi-robot exploration," in *Proc. IEEE Int. Conf. Robot. Automat.*, San

Francisco, CA, April 2000, pp. 476–481.
[2] J. Feddema and D. Schoenwald, "Decentralized control of cooperative robotic vehicles," in *Proc. SPIE Vol. 4364, Aerosense*, Orlando, Florida, April 2001.
[3] J. S. Jennings, G. Whelan, and W. F. Evans, "Cooperative search and rescue with a team of mobile robots," *Proc. IEEE Int. Conf. on Advanced Robotics*, 1997.
[4] C. J. Taylor, "Videoplus: A method for capturing the structure and appearance of immersive environment," *Second Workshop on 3D Structure from Multiple Images of Large-scale Environments*, 2000.
[5] L. Iochhi, K. Konolige, and M. Bayracharya, "A framework and architecture for multi-robot coordination," in *Proc. Seventh Int. Symposium on Experimental Robotics (ISER)*, Honolulu, Hawaii, Dec. 2000.
[6] D. Rus, B. Donald, and J. Jennings, "Moving furniture with teams of autonomous robots," in *IEEE/RSJ International Conf. on Intelligent Robots and Systems*, Pittsburgh, PA, Aug 1995, pp. 235–242.
[7] M. Mataric, M. Nilsson, and K. Simsarian, "Cooperative multi-robot box pushing," in *IEEE/RSJ International Conf. on Intelligent Robots and Systems*, Pittsburgh, PA, Aug 1995, pp. 556–561.
[8] D. Stilwell and J. Bay, "Toward the development of a material transport system using swarms of ant-like robots," in *IEEE International Conf. on Robotics and Automation*, Atlanta, GA, May 1993, pp. 766–771.
[9] T. Sugar and V. Kumar, "Control and coordination of multiple mobile robots in manipulation and material handling tasks," in *Experimental Robotics VI: Lecture Notes in Control and Information Sciences*, P. Corke and J. Trevelyan, Eds., vol. 250, pp. 15–24. Springer-Verlag, 2000.
[10] L. E. Parker, "Current state of the art in distributed autonomous mobile robotics," in *Distributed Autonomous Robotic Systems*, L. E. Parker, G. Bekey, and J. Barhen, Eds., vol. 4, pp. 3–12. Springer, Tokyo, 2000.
[11] A. De Luca, G. Oriolo, and C. Samson, "Feedback control of a nonholonomic car-like robot," in *Robot Motion Planning and Control*, J.-P. Laumond, Ed., pp. 171–253. Springer-Verlag, London, 1998.
[12] H. Khalil, *Nonlinear Systems*, Prentice Hall, Upper Sadle River, NJ, 2nd edition, 1996.
[13] A. Isidori, *Nonlinear Control Systems*, Springer-Verlag, London, 3rd edition, 1995.
[14] S. Shastry, *Nonlinear Systems; Stability, Analysis and Control*, Springer-Verlag, 1st edition, 1999.
[15] R. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robotics and Automation*, vol. 2, no. 1, pp. 14–23, 1986.
[16] R. Arkin and T. Balch, *Artificial Intelligence and Mobile Robots*, chapter Cooperative Multiagent Robot Systems, MIT Press, 1998.
[17] T. Balch and R. Arkin, "Behavior-based formation control for multirobotic teams," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926–934, 1998.
[18] M. Mataric, "Issues and approaches in the design of collective autonomous agents," *Robotics and Autonomous Systems*, vol. 16, no. 2-4, pp. 321–331, Dec 1995.
[19] P. Tabuada, G. Pappas, and P. Lima, "Feasible formations of multi-agent systems," American Control Conference, 2001.
[20] T. Balch, "Social potentials for scalable multi-robot formations," *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 73–80, April 2000.
[21] R. Fierro, P. Song, A. K. Das, and V. Kumar, "A framework for scalable cooperative navigation for autonomous vehicles," Tech. Rep. MS-CIS-01-09, GRASP Laboratory, University of Pennsylvania, PA, 2001, Available at http://www.cis.upenn.edu/techreports.html.
[22] H. Yamaguchi and T. Arai, "Distributed and autonomous control method for generating shape of multiple mobile robot group," in *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, 1994, vol. 2, pp. 800–807.
[23] Q. Chen and J. Y. S. Luh, "Coordination and control of a group of small mobile robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1994, vol. 3, pp. 2315–2320.
[24] G. Beni and P. Liang, "Pattern reconfiguration in swarms - convergence of a distributed asynchronous and bounded iterative algorithm," *IEEE Trans. Robot. Automat.*, vol. 12, no. 3, pp. 485–490, 1996.
[25] J. Desai, J. P. Ostrowski, and V. Kumar, "Controlling formations of multiple mobile robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, Leuven, Belgium, May 1998, pp. 2864–2869.
[26] H. Yamaguchi and J. W. Burdick, "Asymptotic stabilization of multiple nonholonomic mobile robots forming groups formations," in *Proc. IEEE Int. Conf. Robot. Automat.*, Leuven, Belgium, May 1998, pp. 3573–3580.
[27] F. E. Schneider, D. Wildermuth, and H.-L. Wolf, "Motion coordination in formations of multiple mobile robots using a potential field approach," in *Distributed Autonomous Robotic Systems*, L. E. Parker, G. Bekey, and J. Barhen, Eds., vol. 4, pp. 305–314. Springer, Tokyo, 2000.
[28] J. Desai, V. Kumar, and J. P. Ostrowski, "Control of changes in formation for a team of mobile robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, Detroit, Michigan, May 1999, pp. 1556–1561.
[29] R. Fierro, P. Song, A. K. Das, and V. Kumar, "Cooperative control of robot formations," Submitted to Cooperative Control and Optimization Series, Kluwer, Feb. 2001.
[30] R. Burridge, A. Rizzi, and D. Koditschek, "Sequential composition of dynamically dexterous robot behaviors," *Int. J. Robot. Research*, vol. 18, no. 6, pp. 534–555, June 1999.
[31] K. H. Tan and M. A. Lewis, "Virtual structures for high precision cooperative mobile robot control," *Autonomous Robots*, vol. 4, pp. 387–403, October 1997.
[32] J. Lawton, B. Young, and R. Beard, "A decentralized approach to elementary formation manuevers," in *Proc. IEEE Int. Conf. Robot. Automat.*, San Francisco, CA, April 2000.
[33] N. H. McClamroch and I. Kolmanovsky, "Performance benefits of hybrid control design for linear and nonlinear systems," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 1083–1096, July 2000, Invited paper.
[34] R. Alur, R. Grosu, Y. Hur, V. Kumar, and I. Lee, "Modular specifications of hybrid systems in CHARON," *Hybrid Systems: Computation and Control, LNCS 1790*, pp. 6–19, 2000.
[35] R. Alur, A. Das, J. Esposito, R. Fierro, Y. Hur, G. Grudic, V. Kumar, I. Lee, J. P. Ostrowski, G. Pappas, J. Southall, J. Spletzer, and C. J. Taylor, "A framework and architecture for multirobot coordination," in *Proc. ISER00, Seventh International Symposium on Experimental Robotics*, Honolulu, Hawaii, Dec. 2000.
[36] C. Canudas-de-Wit and A. D. NDoudi-Likoho, "Nonlinear control for a convoy-like vehicle," *Automatica*, vol. 36, pp. 457–462, 2000.
[37] D. Liberzon and A. S. Morse, "Basic problems in stability and design of switched systems," *IEEE Control Systems*, vol. 19, no. 5, pp. 59–70, Oct. 1999.
[38] G. M. Souris, *An engineering approach to optimal control and estimation theory*, John Wiley & Sons Inc., New York, 1996.
[39] J. J. Leonard and H. F. Durrant-Whyte, *Directed sonar sensing for mobile robot navigation*, Kluwer Academic Publishers, Boston, 1992.
[40] A. Nadas, "Least squares and maximum likelihood estimation of rigid motion," Tech. Rep., IBM, 1978.
[41] R. Kurazume and S. Hirose, "Study on cooperative positioning system - optimum moving strategies for CPS III," in *Proc. IEEE Int. Conf. Robot. Automat.*, Leuven, Belgium, May 1998, pp. 2896–2903.
[42] S. Baker and S. Nayar, "A theory of catadoptric image formation," in *International COnference on Computer Vision*, Bombay, India, Jan 1998, pp. 35–42.
[43] A. Das, R. Fierro, V. Kumar, J. Southall, J. Spletzer, and C. J. Taylor, "Real-time vision based control of a nonholonomic mobile robot," To appear in IEEE Int. Conf. Robot. Automat., May 2001.