A Framework for Scalable Cooperative Navigation of Autonomous Vehicles

Rafael Fierro, Peng Song, Aveek Das, and Vijay Kumar
GRASP Laboratory, University of Pennsylvania
3401 Walnut Street , Philadelphia, PA 19104
{rfierro, pengs, aveek, kumar}@grasp.cis.upenn.edu

Abstract

We describe a general framework for controlling and coordinating a group of nonholonomic mobile robots equipped with range sensors, with applications ranging from scouting and reconnaissance, to search and rescue and manipulation tasks. We first describe a set of control laws that allows each robot to control its position and orientation with respect to neighboring robots or obstacles in the environment. We then develop a coordination protocol that allows the robots to automatically switch between the control laws to follow a specified trajectory. Finally, we describe two simple trajectory generators that are derived from potential field theory. The first allows each robot to plan its reference trajectory based on the information available to it. The second scheme requires sharing of information and results in a trajectory for the designated leader. Numerical simulations illustrate the application of these ideas and demonstrate the scalability of the proposed framework for a large group of robots.

1 Introduction

It has long been recognized that there are several tasks that can be performed more efficiently and robustly using multiple robots. In fact, there is extensive literature on mobile robot control and the coordination of multiple robots, see for example [21]. Topics include cooperative manipulation, multi-robot navigation and planning, collaborative mapping and exploration, software architectures, and formation control. We are particularly interested in multi-robot navigation and planning, and formation control. The problem of multi-robot navigation is to generate collision-free paths for mobile robots to reach their desired destinations. Previous approaches in this area can be broadly divided into two classes including graph based planners [4] and potential field methods [16, 17]. Graph based planners generally require an expensive precomputation step to construct the connectivity graph –the set of the collision free configurations of the robot, before the search for a path can actually start. Global knowledge of the environment and other robots is assumed in order to build the connectivity graph. As an elegant alternative, the potential field method applies

repulsive potential functions around the obstacles while trying to place the goal location at the global minimum of the potential field. But the construction of a potential field with no other local minima than the goal configuration turns out to be difficult. Various techniques have been developed to overcome these difficulties [24, 5, 3]. But largely because of the computational limitations, most of the work to date in the field of mobile robot navigation has been conducted for small scale laboratory environments.

Formation control of multiple autonomous vehicles arises in many scenarios. For instance, military applications and intelligent vehicle highway systems (IVHS) require that vehicles maneuver while keeping a prescribed formation. In recent years, formation control approaches have also been applied to the coordination of spacecraft and aircraft [13]. Two main approaches have been developed: *leader-following* and *behavioral-based*. In the leader-following approach one robot acts as a leader and generates the reference trajectory for the team of robots. In the *behavioral-based* approach [2] a number of basic behaviors is prescribed, *e.g.*, obstacle avoidance, keep formation, goal seeking. The overall control action (*emergent behavior*) is a weighted average of the control actions for each basic behavior. In this case, deriving control strategies for competing behaviors and implementing them in a decentralized fashion can be straightforward. However, formal analysis of the emergent team behavior is difficult and, in general, stability and performance cannot be guaranteed.

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. In contrast, we also know that it is relatively easy to design *reactive* controllers or behaviors that react to simple stimuli from the environment. This is the basis for the *subsumption* architecture [6] and the paradigm for behavior-based robotics. While control and estimation theory allows us to model each behavior as a dynamical system, it does not give us the tools to model switches in behavior or the hierarchy that might be inherent in the switching behavior.

The lack of a formal analysis of switching-based cooperative control has motivated this paper. Here we describe a framework for decentralized cooperative control of multi-robotic systems that emphasizes simplicity in planning, coordination, and control. The framework incorporates a two-level control hierarchy for each robot consisting of a trajectory generation level and a coordination level as illustrated in Figure 1. The trajectory generator derives the reference trajectory for the robot while the coordination level selects the appropriate controller (behavior) for the robot.

The availability and sharing of information between the robots greatly influences the design of each level. This is particularly true at the trajectory generation level. The trajectory generator can be completely decentralized so that each robot generates its own reference trajectory based on the information available to it, through its sensors and through the communication network. Alternatively, a designated leader plans its trajectory and the other group members are able to organize themselves to following the leader. The trajectory generators are derived from potential field theory. Unlike [22], they are simple goal-directed fields that are not specifically designed to avoid obstacles or neighboring robots. Instead, when a robot is close to an obstacle, it adopts a behavior that simulates the dynamics of a visco-elastic collision with the obstacle guaranteeing that the actual collision never happens.

At the coordination level we assume range sensors that allow the estimation of position of neighboring robots and obstacles. This model is motivated by our experimental platform consisting of mobile robots equipped with omnidirectional cameras described in [8, 1]. Each robot chooses from a finite set of *modes* or control laws that describe its interactions with respect its neighbors (robots and obstacles) and allow it to go to a desired goal position. Thus the overall goal of this level is to prescribe the rules of mode switching and thus the dynamics of the switched system [19].

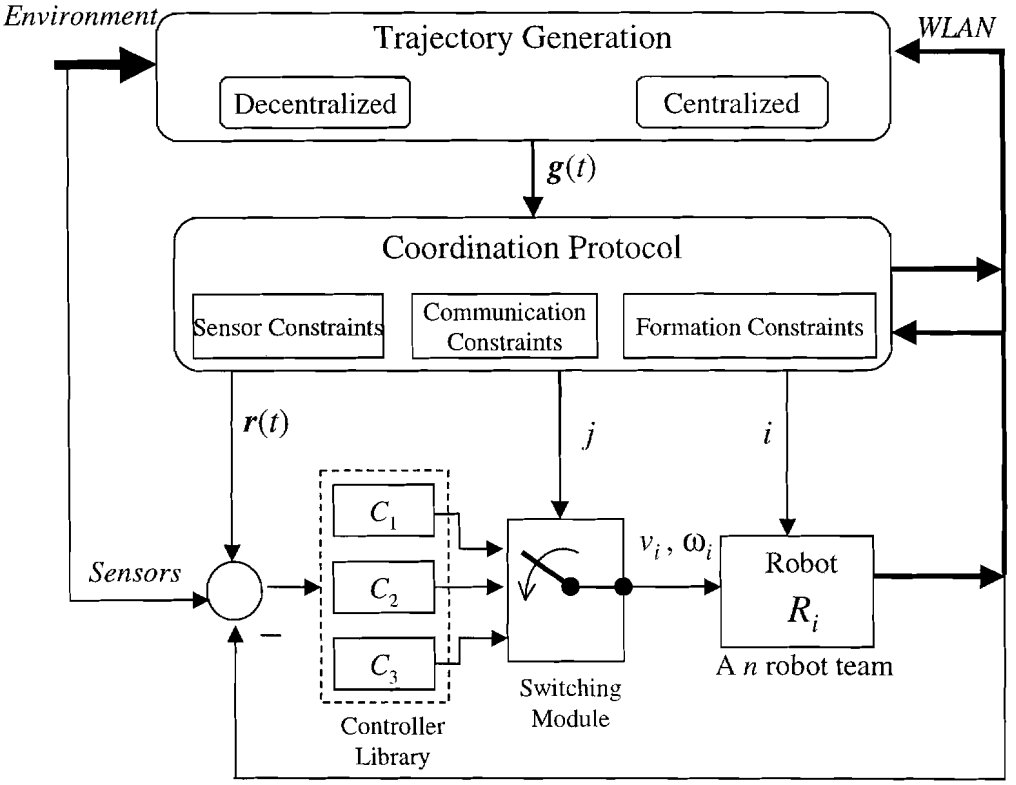

Figure 1: A formation control framework.

In the paper, we first described the three main components shown in Figure 1. First, in section 2 we present the suite of control modes and the strategy for switching between these modes in a stable manner. Second, in section 3, we describe an algorithm for selecting control modes based on the available information and the geometric, kinematic, and dynamic constraints of the robot system. The third component is the *trajectory generator*. In section 4, we present a novel approach that combine potential functions and the dynamics of visco-elastic contact to generate the trajectory either for a designated leader or for each robot. In this way we are able to hierarchically compose *planning* and *control* in a distributed fashion. Finally, simulation results illustrate the benefits and the limitations of this methodology underlying the implementation of cooperative control of robot formations in section 5.

2 Formation Control

In this section, we consider a group of n nonholonomic mobile robots and describe the controllers that specify the interactions between each robot and its neighbor. We will make two assumptions. First, we will assume that the robots are planar and have two independent inputs. This means we have to restrict the robot control laws to those that regulate two outputs. Second, we assume that the robots are assigned integer valued labels from 1 through n which restrict the choice of control laws. Robot 1 is the leader of the group. A robot with a label i, ignores the movements of robots with labels that have values higher than i. Thus, it can control its position and orientation with robots whose labels are lower than i. The assignment and dynamic re-assignment of these labels are discussed later.

We adopt a simple kinematic model for the nonholonomic robots. The kinematics of the *ith* robot are given by

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

where $x_i \equiv (x_i, y_i, \theta_i) \in SE(2)$. Most commercially available robots do not allow the direct control of forces or torques. Instead they incorporate motor controllers that allow the specification of v_i and ω_i. Thus we will treat these as our inputs. Again we point the reader to our previous work [11] to illustrate the advantages and limitations of this simple model.

In Figure 2, we show subgroups of two and three robots. Robot j can be designated as a follower of Robot i if $i < j$. Let $i < j < k$. We first describe two controllers that allow robot j to follow i (Figure 2 (left)), and robot k to follow robots i and j (Figure 2 (right)). We then describe a third controller that describes possible interactions with an obstacle. (Figure 3).

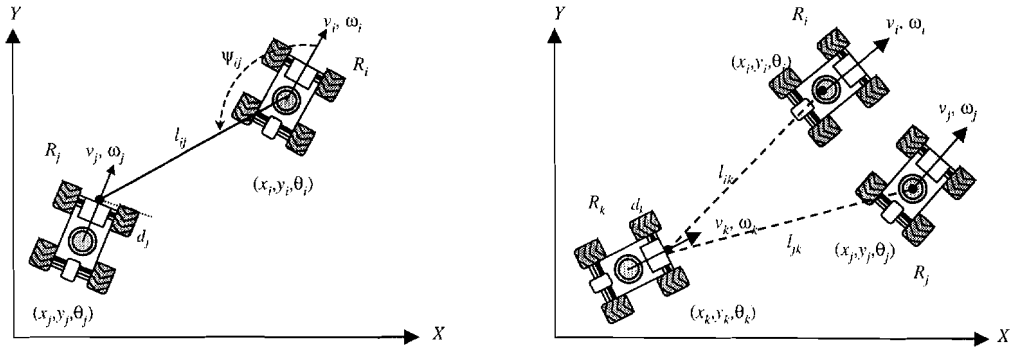

Figure 2: The *S*eparation *B*earing and *S*eparation *S*eparation *C*ontrollers.

Separation Bearing Control By using this controller (denoted $SB_{ij}C$ here), robot R_j follows R_i with a desired separation l_{ij}^d and desired relative bearing ψ_{ij}^d, see Figure 2(left). The control velocities for the *follower* are given by [10]

$$v_j = s_{ij}\cos\gamma_{ij} - l_{ij}\sin\gamma_{ij}(b_{ij} + \omega_i) + v_i\cos(\theta_i - \theta_j) \tag{2}$$

$$\omega_j = \frac{1}{d}[s_{ij}\sin\gamma_{ij} + l_{ij}\cos\gamma_{ij}(b_{ij} + \omega_i) + v_i\sin(\theta_i - \theta_j)] \tag{3}$$

where d is a distance from the wheel axis to a reference point on the robot, and

$$\gamma_{ij} = \theta_i + \psi_{ij} - \theta j, \tag{4}$$
$$s_{ij} = k_1(l_{ij}^d - l_{ij}), \tag{5}$$
$$b_{ij} = k_2(\psi_{ij}^d - \psi_{ij}), \quad k_1, k_2 > 0 \tag{6}$$

The closed-loop linearized system is

$$\dot{l}_{ij} = k_1(l_{ij}^d - l_{ij}), \quad \dot{\psi}_{ij} = k_2(\psi_{ij}^d - \psi_{ij}), \quad \dot{\theta}_j = \omega_j \tag{7}$$

Separation Separation Control By using this controller (denoted $S_{ik}S_{jk}C$), robot R_k follows R_i and R_j with a desired separations l_{ik}^d and l_{jk}^d, respectively, see Figure 2(right). In this case the control velocities for the follower robot become

$$v_k = \frac{s_{ik}\sin\gamma_{jk} - s_{jk}\sin\gamma_{ik} + v_i\cos\psi_{ik}\sin\gamma_{jk} - v_j\cos\psi_{jk}\sin\gamma_{ik}}{\sin(\gamma_{jk} - \gamma_{ik})} \tag{8}$$

$$\omega_k = \frac{-s_{ik}\cos\gamma_{jk} + s_{jk}\cos\gamma_{ik} - v_i\cos\psi_{ik}\cos\gamma_{jk} + v_j\cos\psi_{jk}\cos\gamma_{ik}}{d\sin(\gamma_{jk} - \gamma_{ik})} \tag{9}$$

The closed-loop linearized system is

$$\dot{l}_{ik} = k_1(l_{ik}^d - l_{ik}), \quad \dot{l}_{jk} = k_1(l_{jk}^d - l_{jk}), \quad \dot{\theta}_k = \omega_k. \tag{10}$$

Separation Distance–To–Obstacle Control In this case (denoted SD_oC), the outputs of interest are the separation l_{ij} between the follower and the leader, and the distance δ from an obstacle to the follower. We define a *virtual* robot R_o as shown in Figure 3, which moves on the obstacle's boundary with linear velocity v_o and orientation θ_o. For this case the velocity inputs for the follower robot R_j are given by [11]

$$v_j = \frac{s_{ij}\cos\gamma_{oj} + s_{oj}\sin\gamma_{ij} + v_i\cos\psi_{ij}\cos\gamma_{oj}}{\cos(\gamma_{oj} - \gamma_{ij})} \tag{11}$$

$$\omega_j = \frac{s_{ij}\sin\gamma_{oj} - s_{oj}\cos\gamma_{ij} + v_i\cos\psi_{ij}\sin\gamma_{oj}}{d\cos(\gamma_{oj} - \gamma_{ij})} \tag{12}$$

Thus, the linearized kinematics become

$$\dot{l}_{ij} = k_1(l_{ij}^d - l_{ij}), \quad \dot{\delta} = k_o(\delta_o - \delta), \quad \dot{\theta}_j = \omega_j. \tag{13}$$

where $s_{oj} \equiv k_o(\delta_o - \delta)$, δ_o is the desired distance from the robot R_j to an obstacle, and k_i's are positive controller gains.

It is worth noting that feedback I/O 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 $\vec{\delta}$ and $\vec{l}_{ij}$ are collinear. Moreover, 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. Complex environments (*e.g.*, star-like obstacles) that require a different strategy where a leader tracking may not be guaranteed are beyond the scope of this paper.

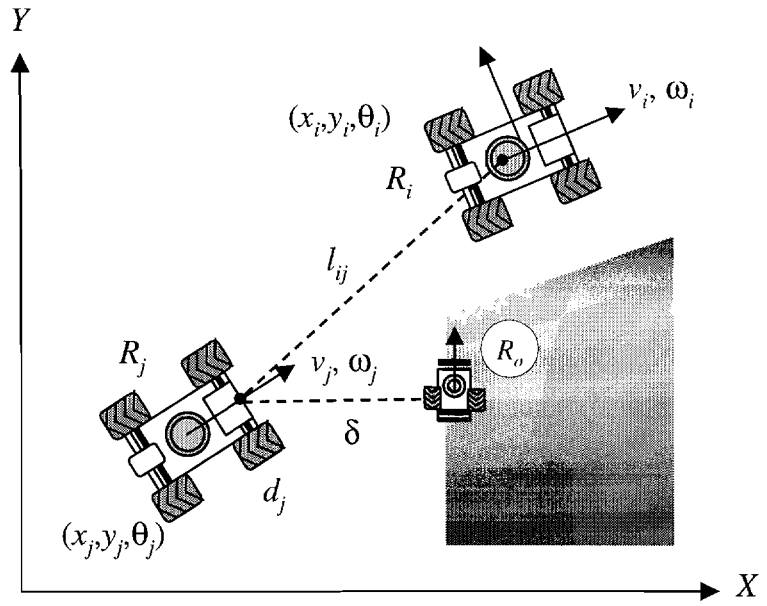

Figure 3: The Separation Distance to Obstacle Control *SDoC*.

2.1 Stability Analysis

In this section we develop a general approach to build formations in a modular fashion. The *low-level* control is coordinated by the protocol presented in next section. To be more specific, since each robot in the team is nonholonomic, it is able to control up two output variables [9], *i.e.*, a robot can follow another robot maintaining a desired separation and bearing, or follow two robots maintaining desired separations. Thus, a basic formation building block consists of a *lead* robot R_i, a *first follower* robot R_j, and a *follower* robot R_k. Figure 4 illustrates the basic formation and the actual robots we use in our experimental testbed. The basic idea is that R_i follows a given trajectory $g(t) \in SE(2)$, R_j and R_k use *SBC* and *SSC*, respectively. In the following, we prove that the *basic formation* is stable, that is, distances and bearings

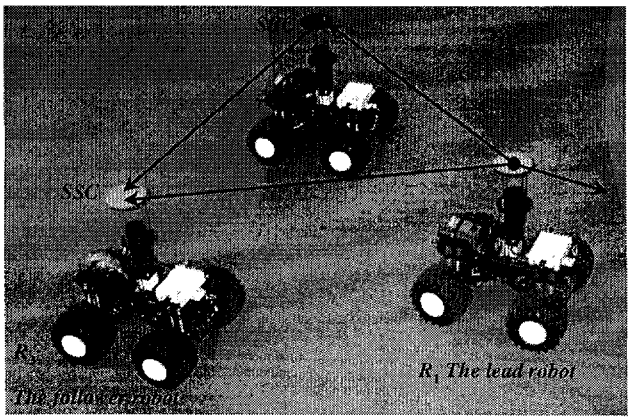

Figure 4: The basic formation configuration.

reach their desired values asymptotically. Notice we are not showing that the whole group of robots will reach the goal position, instead the group navigates in formation going wherever the lead robot is going. Since we are using I/O feedback linearization [14], the linearized systems are given by (7) and (10) with outputs

$$\boldsymbol{z}_1 = [l_{ij} \quad \psi_{ij}]^T, \qquad \boldsymbol{z}_2 = [l_{ik} \quad l_{jk}]^T$$

It is straightforward to show that the output vectors $\boldsymbol{z}_{1,2}$ will converge to the desired values arbitrarily fast. However, a complete stability analysis requires the study of the internal dynamics of the robots *i.e.*, the heading angles θ_j and θ_k which depend on the controlled angular velocities ω_j and ω_k.

Theorem 2.1 *Assume that the lead vehicle's linear velocity along the path is lower bounded i.e., $v_i \geq V_{\min} > 0$, its angular velocity is also bounded i.e., $\|\omega_i\| < W_{\max}$, the relative velocity $\delta_v \equiv v_i - v_j$ and relative orientation $\delta_\theta \equiv \theta_i - \theta_j$ are bounded by small positive numbers ε_1, ε_2. If the control velocities (2)- (3) are applied to R_j, and the control velocities (8)- (9) are applied to R_k then the formation is stable, and the system outputs l_{ij}, ψ_{ij}, l_{ik}, and l_{jk} converge exponentially to the desired values.*

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

$$\begin{aligned} e_1 &= l_{ij}^d - l_{ij}, \quad e_2 = \psi_{ij}^d - \psi_{ij}, \quad e_3 = \theta_i - \theta_j \\ e_4 &= l_{ik}^d - l_{ik}, \quad e_5 = l_{jk}^d - l_{jk}, \quad e_6 = \theta_i - \theta_k \end{aligned} \tag{14}$$

We need to show that the internal dynamics of R_j and R_k are stable which in formation control, is equivalent to show that the orientation errors e_3, e_6 are bounded. For the first follower R_j, we have

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

after some algebraic simplification, we obtain

$$\dot{e_3} = -\frac{v_i}{d} \sin e_3 + g_1(e_3, \omega_i, e_1, e_2) \tag{15}$$

where

$$g_1(t, e_3) = (1 - \frac{l_{ij}}{d} \cos \gamma_{ij})\omega_i - \frac{1}{d}(k_1 e_1 \sin \gamma_{ij} + k_2 e_2 l_{ij} \cos \gamma_{ij})$$

The nominal system *i.e.*, $g_1(t, e_3) = 0$ is given by

$$\dot{e_3} = -\frac{v_i}{d} \sin e_3 \tag{16}$$

which is (locally) exponentially stable provided that the velocity of the lead robot $v_i > 0$. Since ω_i is bounded, it can be shown that $\|g_1(t, e_3)\| \leq \delta_1$. By using stability theory of perturbed systems [15], and assuming that $\|e_3(t_0)\| < c_1 \pi$ for some positive constant $c_1 < 1$, then

$$\|e_3(t)\| \leq b_1, \qquad \forall\, t \geq t_1$$

for some finite time t_1. Now for the follower R_k, the error system becomes

$$\dot{e}_6 = \omega_i - \omega_k$$

as before and after some work, we obtain

$$\dot{e_6} = -\frac{v_i}{d}\sin e_6 + g_2(e_6, \omega_i, e_4, e_5, \delta_v, \delta_\theta) \tag{17}$$

where

$$\begin{aligned} g_2(t, e_6) &= \omega_i - \frac{v_i\delta_\theta \sin\psi_{jk}\cos(e_6 - \psi_{jk}) + \delta_v \cos(e_6 + \psi_{ik})\cos\psi_{jk}}{d[\delta_\theta\cos(\psi_{jk} - \psi_{ij}) + \sin(\psi_{jk} - \psi_{ij})]} - \\ &\quad - \frac{k_5 e_5 \cos\gamma_{ij} + k_4 e_4 \cos\gamma_{jk}}{d[\delta_\theta\cos(\psi_{jk} - \psi_{ij}) + \sin(\psi_{jk} - \psi_{ij})]} \end{aligned}$$

Again, the nominal system is given by (16) *i.e.*, $g_2(t, e_6) = 0$, and it is (locally) exponentially stable provided that the velocity of the lead robot $v_i > 0$. Since $\|\omega_i\| < W_{\max}$, $\|\delta_v\| < \varepsilon_1$, and $\|\delta_\theta\| < \varepsilon_2$, it can be shown that $\|g_2(t, e_6)\| \leq \delta_2$. Assuming that $\|e_6(t_0)\| < c_2\pi$ for some positive constant $c_2 < 1$, then

$$\|e_6(t)\| \leq b_2, \qquad \forall\, t \geq t_2$$

for some finite time t_2.

□

The above theorem shows that, under some reasonable assumptions, the three-robot formation system is stable *i.e.*, there exists a Lyapunov function $V(t, \boldsymbol{e})$ in $[0, \infty) \times D$, where $D = \{\boldsymbol{e} \in \Re^6 | \, \|\boldsymbol{e}\| < c\}$, such that $\dot{V}(t, \boldsymbol{e}) \leq 0$. Let

$$V = \boldsymbol{e}_{12}^T \boldsymbol{P}_{12}\boldsymbol{e}_{12} + \frac{1}{2}e_3^2 + \boldsymbol{e}_{45}^T \boldsymbol{P}_{45}\boldsymbol{e}_{45} + \frac{1}{2}e_6^2 \tag{18}$$

be a Lyapunov function for the system error (14) then

$$\begin{aligned} \dot{V} &= -\boldsymbol{e}_{12}^T \boldsymbol{Q}_{12}\boldsymbol{e}_{12} - \boldsymbol{e}_{45}^T \boldsymbol{Q}_{45}\boldsymbol{e}_{45} - \frac{v_i}{d}e_3 \sin e_3 \\ &\quad - \frac{v_i}{d}e_6 \sin e_6 + g_1(t, e_3)e_3 + g_2(t, e_6)e_6 \end{aligned} \tag{19}$$

where $\boldsymbol{e}_{12}^T \equiv [e_1 \quad e_2]$, $\boldsymbol{e}_{45}^T \equiv [e_4 \quad e_5]$, and $\boldsymbol{P}_{12}$, $\boldsymbol{P}_{45}$, $\boldsymbol{Q}_{12}$, and $\boldsymbol{Q}_{45}$ are 2×2 positive definite matrices. By looking at (18)-(19), we can study some particular formations of practical interest.

- Let us assume two robots in a *linear motion* leader-following formation *i.e.*, v_i is constant, and $\omega_i = 0$. Thus the Lyapunov function and its derivative become

$$V_2 = \boldsymbol{e}_{12}^T \boldsymbol{P}_{12}\boldsymbol{e}_{12} + \frac{1}{2}e_3^2 \tag{20}$$

$$\dot{V}_2 = -\boldsymbol{e}_{12}^T \boldsymbol{Q}_{12} \boldsymbol{e}_{12} - \frac{v_i}{d} e_3 \sin e_3 \tag{21}$$

then the two-robot 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. This result can be extended to n robots in a *convoy-like* formation (*c.f.*, [7]. Let us consider a team of n robots where R_i follows R_{i-1} under *SBC*. A Lyapunov function and its derivative can be given by

$$V_{1\ldots n} = \sum_{i=2}^{n} \boldsymbol{e}_{i-1,i}^T \boldsymbol{P}_{i-1,i} \boldsymbol{e}_{i-1,i} + \frac{1}{2} e_{\theta i}^2 \tag{22}$$

$$\dot{V}_{1\ldots n} = -\sum_{i=2}^{n} (\boldsymbol{e}_{i-1,i}^T \boldsymbol{Q}_{i-1,i} \boldsymbol{e}_{i-1,i} + \frac{v_i}{d} e_{\theta i} \sin e_{\theta i} - g_i(t, e_{\theta i})) \tag{23}$$

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.

- A similar analysis can be carried out for the case of three robots in a *parallel* linear motion where $v_i = v_j = constant$, $\omega_i = \omega_j = 0$, and $\theta_i(t_0) = \theta_j(t_0)$. The Lyapunov function and its derivative are given by

$$V_3 = \boldsymbol{e}_{45}^T \boldsymbol{P}_{45} \boldsymbol{e}_{45} + \frac{1}{2} e_6^2 \tag{24}$$

$$\dot{V}_3 = -\boldsymbol{e}_{45}^T \boldsymbol{Q}_{45} \boldsymbol{e}_{45} - \frac{v_i}{d} e_6 \sin e_6 \tag{25}$$

then the three-robot system is (locally) asymptotically stable *i.e.*, $e_6 \to 0$ as $t \to \infty$ provided that $v_i > 0$, $\|e_6\| < \pi$ and $l_{ij} < l_{ik} + l_{jk}$. Again, this result can be extended to n robots in parallel linear formation.

So far, 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. These problems have motivated the development of a *switching parading* that allows robots change the shape of the formation *on-the-fly*. The basic idea is as follows. Suppose a two-robot (R_1, R_2) formation is following a predefined trajectory using *SBC*. If there is an obstacle in the field-of-view of the follower, it switches to *SDoC*. When the obstacle has been successfully negotiated, R_2 switches back to *SBC*. Assume now a third robot R_3 joins the formation. Since R_3 has some sensor constraints, it may *see* or *follow* R_1, R_2 or both. For avoiding inter-robot collisions, the preferred configuration is that R_3 follows R_1 and R_2 using *SSC*. Thus, if R_3 sees only R_2, it will follow R_2 with desired values (*i.e.* l_{23}^d, ψ_{23}^d) selected in a way that R_3 is driven to the domain of controller *SSC*. Similarly, if R_3 sees only R_1, the desired output values (l_{13}^d, ψ_{13}^d) are chosen such that R_3 is driven to the domain of controller *SSC*. The interested reader is referred to [11] for details.

3 Coordination Protocol

At the *coordination* level, for an n robot formation to maintain a desired shape we need to model the choice of controllers between the individual robots as they move in a given environment. We use directed graphs as our tool for accomplishing this.

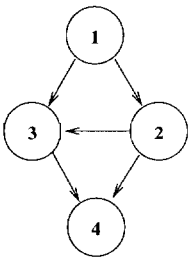

Figure 5: Formation graph for 4 robots

Formation control graphs When $n > 3$, we can construct more complex formations by using the three controllers discussed in Section 2. In Figure 5 for example, the formation of a group of four robots involves one separation-bearing control (R_2 following R_1) and two separation-separation controllers (R_3 following R_1 and R_2, and R_4 following R_2 and R_3). We call such a directed graph $\mathcal{H}$, with n nodes representing n robots and edges describing the control policy between the connected robots, a *control graph*. Any control graph can be represented by its *adjacency matrix* (see [20] for definition). For the example in Figure 5, this adjacency matrix is given by:

$$H = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{26}$$

Note that this is a directed graph with the control flow from leader i to follower j. If a column k has a non zero entry in row i, then robot k is following i. A robot can have up to 2 leaders. The column with all zeros corresponds to the lead robot. A row with all zeros corresponds to a terminal follower.

It is clear that the number of possible control graphs increases dramatically with the number of robots. For labeled robots with the constraint of leaders having lower labels than followers, $n = 3$ allows 3 control graphs, $n = 4$ results in 18 graphs, and $n = 5$ results in 180 graphs. The number of possible control graphs for 10 robots exceeds a billion. However, if we consider identical (unlabelled robots), the control graphs can be classified into a smaller number equivalence classes [10]. Note that H need not be upper triangular in such a case. In either case, labelled or unlabelled, identifying the appropriate control graph for a given situation is an important and challenging problem.

We now focus on the following problem. Given a distribution of n robots with known configurations with one lead robot and m obstacles, find an optimal formation control graph

$\mathcal{H}$ assigning a controller and leader(s) for each robot. The choice of $\mathcal{H}$ depends on constraints on the robot. We will consider the following two constraints.

- Sensor constraints: the range and field of view of the robot's sensor;
- Kinematic constraints: the relative position and orientation between neighboring robots and the rates of change of these quantities.

Maintaining Formation Constraints We will assume that each robot has perfect information about its own state. We will also make the very conservative assumption that the robots cannot communicate. Thus, the only channel of communication is indirect, via sensory observations. The sensor constraints indicate the observations that are possible. Control graphs that are compatible with the sensor constraints have to be identified. (For example, a robot cannot follow a robot that it cannot see). The primary consideration with kinematic constraints is the possibility of collision. We want to ensure that the separation c_{ij} between robots i and j is above a threshold. In addition, we will consider the rate of change of this separation and ensure that relative motion between the robots do not cause this separation to decrease below the threshold rapidly.

To formalize these ideas, consider first the dynamics of the group, where the formation configuration is given by

$$\boldsymbol{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_n]^T \tag{27}$$

Then we can write

$$\dot{\boldsymbol{x}} = [\dot{\boldsymbol{x}}_1, \dot{\boldsymbol{x}}_2, \ldots \dot{\boldsymbol{x}}_n]^T \tag{28}$$

where robot j with control inputs $\boldsymbol{u}_j$ has dynamics $\dot{\boldsymbol{x}}_j = \boldsymbol{f}_j(\boldsymbol{x}, \boldsymbol{u}_j)$. Suppose R_j has to maintain the formation separation constraint $c(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq 0$ with a neighboring robot R_i. In order for R_j to consider choosing R_i as a leader (assuming it is observable by R_j) we need to define a measure for the rate of change of separation of R_j relative to R_i. Let us replace $c(\boldsymbol{x}_i, \boldsymbol{x}_j)$ with c_{ij} for simplicity. We know:

$$\dot{c}_{ij} = \frac{\partial c_{ij}}{\partial \boldsymbol{x}_i}\dot{\boldsymbol{x}}_i + \frac{\partial c_{ij}}{\partial \boldsymbol{x}_j}\dot{\boldsymbol{x}}_j \tag{29}$$

which can be written as:

$$\dot{c}_{ij} = \mathcal{L}_{f_i} c_{ij} + \mathcal{L}_{f_j} c_{ij} \tag{30}$$

where, $\mathcal{L}_{f_i} c_{ij}$ denotes the Lie derivative of c_{ij} along $\boldsymbol{f}_i$. Now R_j can calculate *instantaneous time to violation* with R_i as:

$$\delta t_{ij} = \frac{c_{ij}}{\dot{c}_{ij}} \tag{31}$$

In order to calculate δt_{ij} explicitly we can either instantaneously consider R_i to be static ($\dot{\boldsymbol{x}}_i = 0$ in (29)) or assume R_j estimates $\dot{\boldsymbol{x}}_i$. We assume R_j knows $\dot{\boldsymbol{x}}_j$ accurately. The sign of δt_{ij} tells us if R_j is headed towards or away from R_i. $\delta t_{ij} = 0$ means violation has occurred. A smaller magnitude of δt_{ij} means violation is about to happen (negative sign) or has just happened (positive sign). This captures the fact that robots which are close but are facing

away from each other are less important candidates than ones which are farther apart but are headed towards constraint violation.

We are now ready to present our algorithm for assigning a feasible formation control graph H given n robot configurations with one lead robot and arbitrary labels. In addition the control graph is optimal with respect to the choice of the maximum possible number of leaders with stable controllers (Section 2) for every robot. First, we build three $n \times n$ matrices $C = [c_{ij}]$, $T = [\delta t_{ij}]$ and $\Phi = [\phi_{ij}]$. The first two are described above and ϕ_{ij} is the angular position of R_i relative to R_j (in its current configuration).

control_graph_assignment algorithm {
  for each robot $k \in \{1, 2, \ldots, n\}$ {
    Step 1
    find set P of robots from row k of C & Φ s.t.
      $c_{ik} < sensor\_range$ & $\phi_{ik} \in sensor\_field\_of\_view$ $\forall i \in P$;
    find set Q of robots from column k of C & Φ s.t.
      $c_{ik} < sensor\_range$ & $\phi_{ik} \in sensor\_field\_of\_view$ $\forall i \in Q$;
    if $(P \cup Q) = \emptyset$ next k, goto **Step 1**;
    sort $P \cup Q$ in ascending order of δt_{ik} $(i \in (P \cup Q))$;
    let $S = \{j : \delta t_{jk} \leq 0, j \in (P \cup Q)\}$;
    if $S = \emptyset$ {
      for $i \in (P \cup Q)$ s.t. δt_{ik} least positive, assign $H(i,k) = 1$;
      next k, goto **Step 1**; }
    if $numOfElements(S) = 1$ {
      assign $H(i,k) = 1$ for $S = \{i\}$; next k, goto **Step 1**; }
    if $numOfElements(S) \geq 2$ {
      pick last two $\{i,j\} \in S$ (with $i \leq j$);
      if $\{i,j,k\}$ satisfy triangle inequality {
        $H(i,k) = 1$ & $H(j,k) = 1$; next k, goto **Step 1**; }
      $H(j,k) = 1$; next k, goto **Step 1**; }
  }
}

There can be situations where a pair of robots (R_i, R_j), that are close to each other, are facing each other and have symmetric configurations. This might cause a "tie" where R_j chooses R_i as a reference and R_i chooses R_j as a reference. If either of them is the lead robot or has another robot as a reference, the tie can be resolved. If not, this is when communication between neighboring robots becomes necessary. It will be clear in the examples below that such ties can be easily detected in the control graph.

Examples We now look at two example situations with four robots. In one of them we will see the need for communication to break a tie as two robots try to follow each other. In both each individual robot assumes it has unit linear velocity at the given instant.

Example 1: The robot configurations (x, y in m., θ in rad.) in Figure 6 are $(0, 3, 0)$, $(2, 3, 0)$, $(0, 4, \pi)$, and $(0, 1, \frac{\pi}{4})$. The sensors look ahead of the robot for a range of 3 m. and

a field of view of π rad (facing forward). The robots themselves are assumed to occupy a circle of radius 0.5 m. The adjacency matrix of the output graph is given by

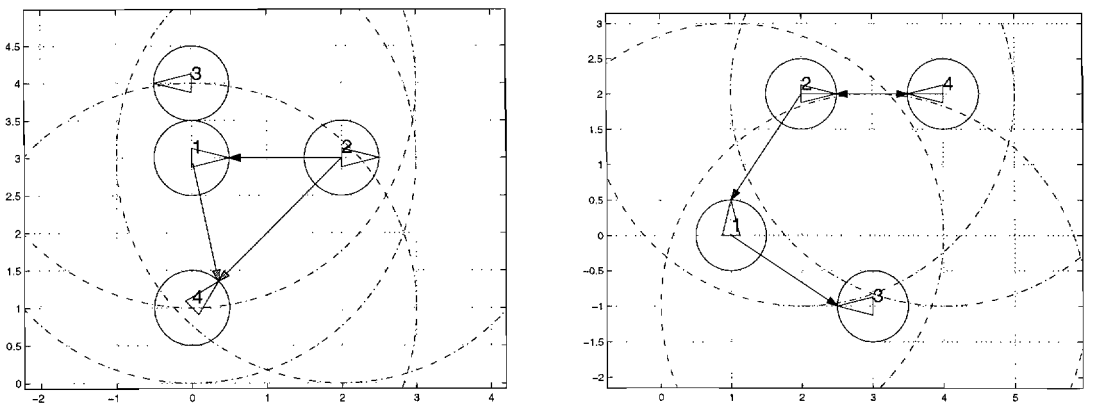

Figure 6: Optimal control graph for Example 1 (left) and Example 2 (right)

$$H_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{32}$$

The interesting fact is that R_3 is not a part of the formation as it cannot see any of the other robots and is outside the angular field of view of R_1 and R_2. Robot R_2 is the lead robot, R_1 follows R_2 and R_4 follows both R_1 and R_2.

Example 2: The robot configurations (same units as above) in Figure 6 are $(1, 0, \frac{\pi}{2})$, $(2, 2, 0)$, $(3, -1, \pi)$, and $(4, 2, \pi)$. The sensor constraints are same as in Example 1. The adjacency matrix of the output graph is given by

$$H_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{33}$$

The interesting thing to note in this example is that there is a tie regarding choosing a leader between R_2 and robot R_4 due to their symmetric configurations. However this can be easily detected in (33) as $H_2(2,4) = H_2(4,2) = 1$. Thus given a graph $\mathcal{H}$, communication is necessary between R_i and R_j whenever there is a tie *i.e.*, $H(i,j) = H(j,i) = 1$.

The above approach for choosing a formation control graph also works in the presence of obstacles which can be treated as virtual leaders (see section 2) if they satisfy the sensor constraints. We add an extra row before the present first row of the adjacency matrix for each obstacle and run the algorithm.

The control graph assignment algorithm models the decentralized decision making for each individual robot. It runs at the supervisory level and needs the current configurations

of all the robots. This is especially useful in simulations of large formations in complex scenarios (Section 4) to keep track of individual choice of controllers and switching between them.

4 Decentralized Navigation Using Contact Dynamics Models

In this section, we propose a decentralized scheme for sensor-based navigation systems. This approach demonstrates strong scalability and is flexible in terms of designing controllers for different navigation tasks. The key idea that distinguishes our approach from previous work is the use of rigid body contact dynamics models to embrace collisions between the robot and its surroundings instead of avoiding them.

4.1 Modeling collisions

Consider a group of mobile robots moving in an environment with the presence of obstacles, we first characterize the surrounding spatial division of each mobile robot with three zones as depicted in Figure 7. Use robot R_1 as an example, the sensing zone denotes the region within which a robot can detect obstacles and other robots. The contact zone is a collision warning zone. The robot starts estimating the relative positions and velocities of any objects that may appear inside its contact zone. The innermost circle is the protected zone which is modeled as a rigid core during a possible contact to provide a collision free environment for the actual robot. The ellipse within the protected zone represents the *reachable* region of the robot for a given time buffer. During the planning process, we will use the protected zone as an abstraction of the agent itself.

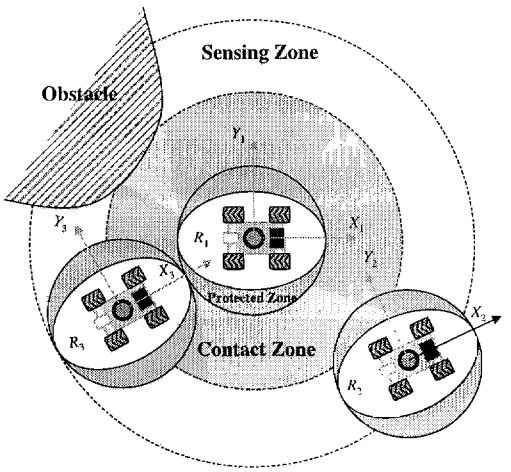

Figure 7: Zones for the computation of contact response.

For the planar case, the dynamics equations of motion for the ith agent in a n-robot

group are given by

$$M_i(q_i)\ddot{q}_i + h_i(q_i, \dot{q}_i) = u_i + \sum_{j=1}^{k} W_{ij} F_{ij}, \qquad i = 1 \ldots n, \tag{34}$$

where $q_i \in \Re^3$ ($\Re^6$ for the spatial case) is the vector of generalized coordinates for the ith agent, M_i is an 3×3 positive-definite symmetric inertia matrix, $h_i(q_i, \dot{q}_i)$ is a 3×1 vector of nonlinear inertial forces, and u_i is the 3×1 vector of applied (external) forces and torques which can be provided through the local controller. k is the number of the contacts between the ith agent and all other objects which could be either obstacles or other robots. $F_{ij} = (F_{N,ij}\ F_{T,ij})^T$ is a 2×1 vector of contact forces corresponding to the jth contact, and $W_{ij} \in \Re^{3\times 2}$ is the Jacobian matrix that relates the velocity at the jth contact point to the time derivatives of the generalized coordinates of the agent. For the time being, we will assume that nonholonomic constraints are not present.

We adopt a state-variable based compliant contact model described in [23] to compute the contact forces. At the jth contact of the agent i, the normal and tangential contact forces ($F_{N,ij}$ and $F_{T,ij}$ are given by

$$F_{N,ij} = f_N(\delta_{N,ij}) + g_N(\delta_{N,ij}, \dot{\delta}_{N,ij}),\ j = 1, \ldots, k, \tag{35}$$

$$F_{T,ij} = f_T(\delta_{T,ij}) + g_T(\delta_{T,ij}, \dot{\delta}_{T,ij}), \quad j = 1, \ldots, k, \tag{36}$$

where the functions f_N and f_T are the elastic stiffness terms and g_N and g_T are the damping terms in the normal and tangential directions respectively. Similar to handling rigid body contact, these functions can be designed to adjust the response of the robot when contact happens. $\delta_{N,ij}(q)$ and $\delta_{T,ij}(q)$ are the local normal and tangential deformations which can be uniquely determined by the generalized coordinates of the system. It is also possible to add on the frictional effects if desired. The details and variations on the compliant contact model are discussed in [18, 23]. A key feature of this model is that it helps us resolve the ambiguous situations when there are more than three objects came into contact with one agent.

Figure 8 shows an example of the army-ant scenario (marshaling) in which 25 robots trying to team around a goal. The grouping is done dynamically using decentralized decision making. While the team is initialized with two groups, the groups merge around the goal location. A quadratic well type of potential function [16, 17] is constructed to drive the robots toward the goal. The expression of the potential function is given by

$$\phi(q) = \frac{k_p}{2}(q - q_g)^T(q - q_g) \tag{37}$$

where q_g is the coordinates of the goal. The input u_i for the ith agent can be obtained by the gradient of the potential function

$$u_i = -\nabla\phi(q_i) = -k_p(q - q_g) \tag{38}$$

which is a proportional control law. Asymptotic stabilization can be achieved by adding dissipative forces to u [16]. Other types of potential fields, such as the conical well function [24], can be used to cover a larger environment.

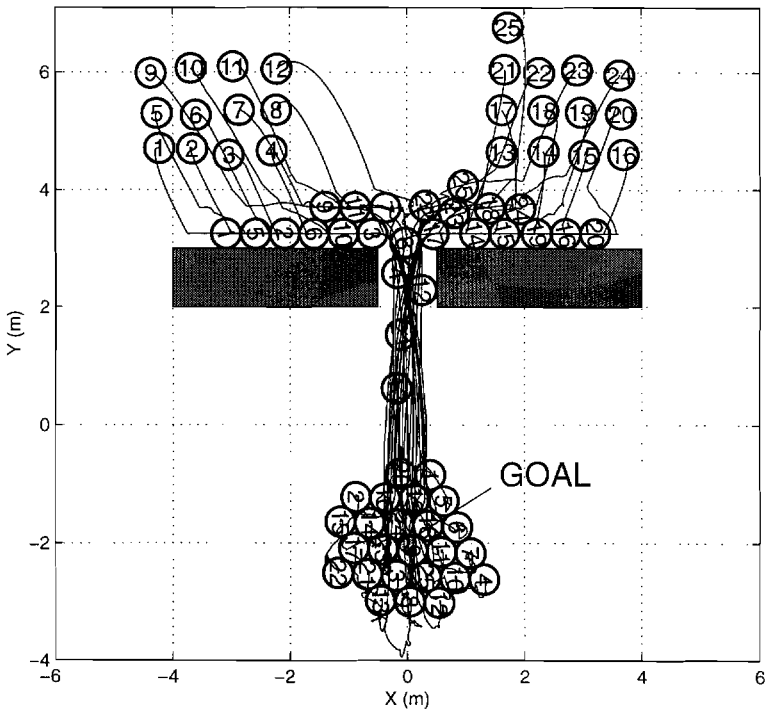

Figure 8: Example of large-scale exploration: the army-ant scenario with 25 holonomic agents

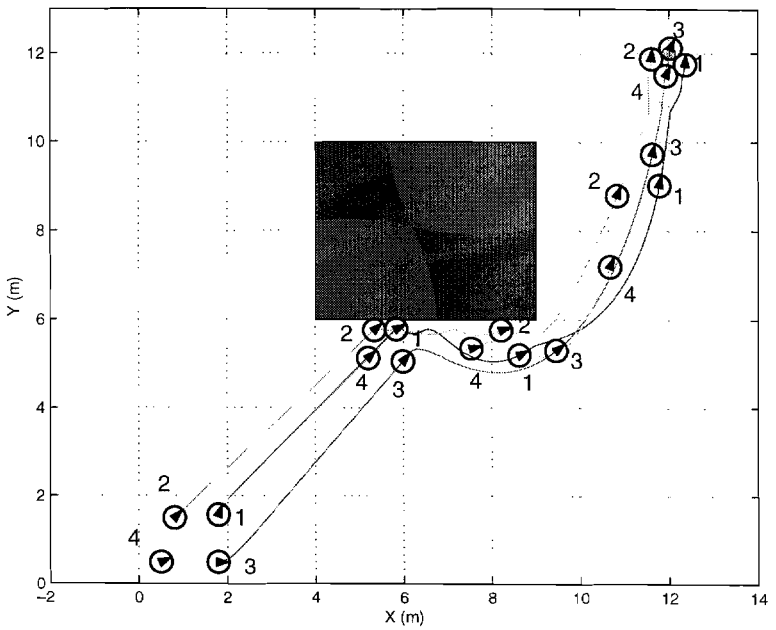

Figure 9: Grouping of nonholonomic robots.

This approach can be easily scaled up for even larger numbers of robots. A decentralized control structure appears naturally based on this approach. Each agent is guided by a local (decentralized) controller that uses the information obtained within its sensing zone. Explicit communications between robots are avoided.

4.2 Decentralized control of nonholonomic agents

The nonholonomic nature of most autonomous robots requires substantial care when developing the local level controllers [4, 5]. The dynamic model for a car-like nonholonomic agent can be expressed as

$$M_i(q_i)\ddot{q}_i + h_i(q_i, \dot{q}_i) = B(q_i)u_i - A_i(q_i)^T\lambda_i + \sum_{j=1}^{k} W_{ij}F_{ij}, \quad i = 1 \ldots n, \tag{39}$$

where B is an input transformation matrix. λ is the constraint forces due to the following nonholonomic constraints

$$A_i(q_i)\dot{q}_i = 0, \text{ or } \dot{q}_i = S_i(q_i)v_i. \tag{40}$$

We can project the contact forces $\sum_{j=1}^{k} W_{ij}F_{ij}$ onto the reduced space while eliminating the constraint forces $A_i(q_i)^T\lambda_i$ in (39). The complete dynamics of the reduced system is given by

$$\begin{cases} \dot{q} &= Sv \\ \dot{v} &= \bar{M}^{-1}(MS^TBu + S^T\sum_{j=1}^{k} W_jF_j - SM\dot{S}v - S^Th), \end{cases} \tag{41}$$

where $\bar{M} = S^TMS \in \Re^{2\times 2}$ is a symmetric, positive definite matrix. Note that the index i in the above equations is omitted for the sake of simplicity. We use I/O linearization techniques to generate a control law u that gives exponentially convergent solutions for the state variables (q, v) [12]. The projected contact forces are treated as external disturbances during this process. Figure 9 shows the simulation results of an example for the grouping behavior of 4 nonholonomic agents.

The contact mechanics based framework proposed here is highly flexible. Other navigation behaviors such as the formation behaviors can be fully integrated with this framework. We will demonstrate this in the next section.

5 Simulation Results

The results obtained in previous sections are applied to a simulation example which includes four robots and an obstacle. First, the initial control graph is given by the algorithm presented in section 3. Second, the trajectory for the lead robot, R_1, is generated by the technique discussed in section 4. Finally, the basic controllers and the switching strategy outlined in section 2 are implemented on $R_{2,3,4}$. The desired shape of the formation is a *diamond* with inter-robot separation of 1.2 m. As it is shown in Figure 10, the robots are able to negotiate the obstacle, avoid collisions and keep formation.

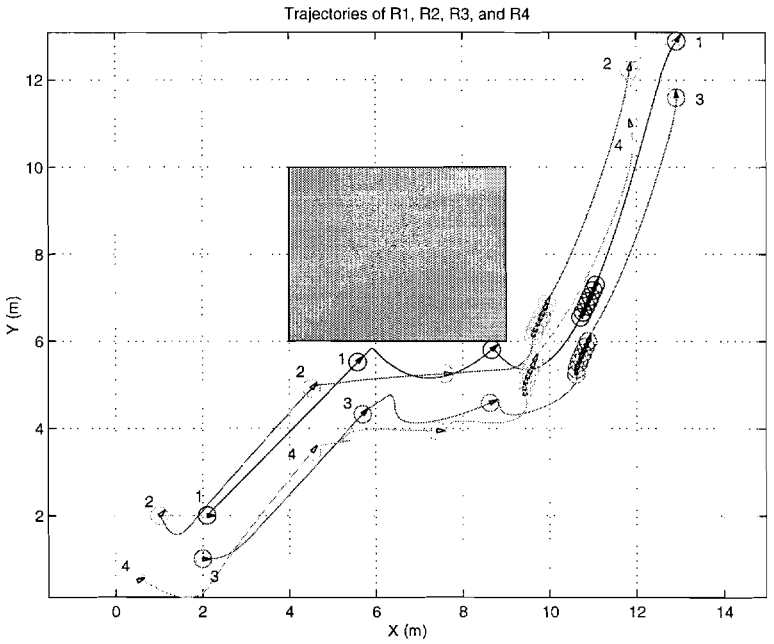

Figure 10: A 4-Robot example–Decentralized planner.

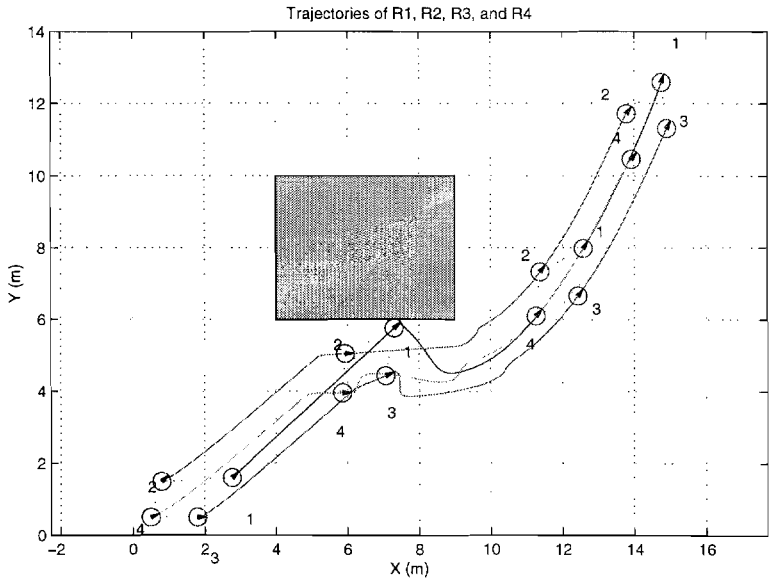

Figure 11: A 4-Robot example–Centralized planner.

We repeat the simulation experiment, but this time the lead robot's trajectory is generated taking into account masses and inertias of the whole group. Thus, as it can be seen in Figure 11 the behavior of the group after a collision is different from the previous case.

6 Conclusions

In this paper, we have presented a framework for controlling and coordinating a team of nonholonomic mobile robots. We have identified and integrated three fundamental components in formation control: reference trajectory generation, a coordination protocol that allows the robots to switch between control policies, and a suite of controllers that under reasonable assumptions guarantees stable formations. Our approach can easily scale to any number of vehicles and is flexible enough to support many formation shapes. The framework described here can also be applied to other types of unmanned vehicles (*e.g.*, aircrafts, spacecrafts, and underwater vehicles). Currently, we are conducting experiments on a team of nonholonomic mobile robots, and applying similar ideas to formation flight on $SE(3)$.

Acknowledgments

This research has been supported in part by the DARPA ITO MARS Program Grant No. 130-1303-4-534328-xxxx-2000-0000.

References

[1] 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.

[2] T. Balch and R. Arkin. Behavior-based formation control for multi-robotic teams. *IEEE Transactions on Robotics and Automation*, 14(6):926–934, 1998.

[3] J. Barraquand, B. Langlois, and J. Latombe. Numerical potential field techniques for robot path planning. *IEEE Trans. Syst., Man, and Cybern.*, 22(2):224–241, 1992.

[4] J. Barraquand and J. Latombe. Non-holonomic multibody mobile robots: controlability and motion planning in the presence of obstacles. *Algorithmica*, 10:121–155, 1993.

[5] A. Bemporad, A. De Luca, and G. Oriolo. Local incremental planning for a car-like robot navigating among obstacles. In *Proc. of the 1996 IEEE Int'l Conf. on Robotics and Automation*, pages 1205–1211, 1996.

[6] R. Brooks. A robust layered control system for a mobile robot. *IEEE J. Robotics and Automation*, 2(1):14–23, 1986.

[7] C. Canudas-de-Wit and A. D. NDoudi-Likoho. Nonlinear control for a convoy-like vehicle. *Automatica*, 36:457–462, 2000.

[8] 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., ICRA01, May 2001.

[9] A. De Luca, G. Oriolo, and C. Samson. Feedback control of a nonholonomic car-like robot. In J.-P. Laumond, editor, *Robot Motion Planning and Control*, pages 171–253. Springer-Verlag, London, 1998.

[10] J. Desai, J. P. Ostrowski, and V. Kumar. Controlling formations of multiple mobile robots. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 2864–2869, Leuven, Belgium, May 1998.

[11] R. Fierro, A. Das, V. Kumar, and J. P. Ostrowski. Hybrid control of formation of robots. To appear in IEEE Int. Conf. Robot. Automat., ICRA01, May 2001.

[12] R. Fierro and F. L. Lewis. Robot kinematics. In J. Webster, editor, *Wiley Encyclopedia of Electrical and Electronics Engineering.* John Wiley and Sons, Inc, 1999.

[13] F. Giulietti, L. Pollini, and M. Innocenti. Autonomous formation flight. *IEEE Control Systems*, 20(6):34–44, 2000.

[14] A. Isidori. *Nonlinear Control Systems.* Springer-Verlag, London, 3rd edition, 1995.

[15] H. Khalil. *Nonlinear Systems.* Prentice Hall, Upper Sadle River, NJ, 2nd edition, 1996.

[16] O. Khatib. Real-time obstacle avoidance for manipulatorsand mobile robots. *International Journal of Robotics Research*, 5:90–98, 1986.

[17] D. Koditschek. Exact robot navigation by means of potential functions: Some topological considerations. In *Proc. of the 1987 IEEE Int'l Conf. on Robotics and Automation*, pages 1–6, 1987.

[18] P. R. Kraus and V. Kumar. Compliant contact models for rigid body collisions. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1382–1387, Sept. 1997.

[19] D. Liberzon and A. S. Morse. Basic problems in stability and design of switched systems. *IEEE Control Systems*, 19(5):59–70, Oct. 1999.

[20] G. L. Nemhauser and L. A. Wolsely. *Integer and Combinatorial Optimization*, chapter I.3. Wiley, 1988.

[21] L. E. Parker. Current state of the art in distributed autonomous mobile robotics. In L. E. Parker, G. Bekey, and J. Barhen, editors, *Distributed Autonomous Robotic Systems*, volume 4, pages 3–12. Springer, Tokio, 2000.

[22] E. Rimon and D. Koditschek. Exact robot navigation using articial potential fields. *IEEE Trans. Robot. & Autom.*, 8(5):501–518, Oct. 1992.

[23] P. Song, P. Kraus, V. Kumar, and P. Dupont. Analysis of rigid body dynamic models for simulation of systems with frictional contacts. Technical Report MS-CIS-00-08, University of Pennsylvania - Department of Computer and Information Science, 2000.

[24] R. Volpe and P. Khosla. Manipulator control with superquadric artificial potential functions: Theory and experiments. *IEEE Trans. on Syst., Man, and Cybern.*, 20(6):1423–1436, 1990.