October 2003

Formation Control with Configuration Space Constraints

Guilherme A. S. Pereira
Universidade Federal de Minas Gerais

Aveek K. Das
University of Pennsylvania

R. Vijay Kumar
University of Pennsylvania, kumar@grasp.upenn.edu

Mario F. M. Campos
Universidade Federal de Minas Gerais

Formation Control with Configuration Space Constraints

Abstract
We address the problem of controlling a team of robots subject to constraints on relative positions. We adopt the general framework of leader-follower control in [1], [2] in which a network of controllers is used to control the position and orientation of the team and its shape. We propose two improvements to this scheme. First, we introduce cooperative leader-following where the motion of a robot is determined not only by its leader, but also by other robots including their followers. Second, we allow constraints that are induced by limitations on ranges of sensors and wireless network cards. Our approach is based on potential field controllers for each robot and the on-line modification of these controllers to accommodate motion constraints induced by other robots in the group. We present experimental results with a team of three car-like robots equipped with omnidirectional cameras and 802.11b network cards.

Proceedings of the 2003 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
Las Vegas, Nevada · October 2003

Formation Control with Configuration Space Constraints

Guilherme A. S. Pereira[§] Aveek K. Das[†] Vijay Kumar[†] Mario F. M. Campos[§]
[§]VERLab, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brazil.
[†]GRASP Lab., University of Pennsylvania, Philadelphia, PA, USA.
{gpereira, mario}@dcc.ufmg.br, {aveek, kumar}@grasp.cis.upenn.edu

***Abstract*— We address the problem of controlling a team of robots subject to constraints on relative positions. We adopt the general framework of leader-follower control in [1], [2] in which a network of controllers is used to control the position and orientation of the team and its shape. We propose two improvements to this scheme. First, we introduce cooperative leader-following where the motion of a robot is determined not only by its leader, but also by other robots including their followers. Second, we allow constraints that are induced by limitations on ranges of sensors and wireless network cards. Our approach is based on potential field controllers for each robot and the on-line modification of these controllers to accommodate motion constraints induced by other robots in the group. We present experimental results with a team of three car-like robots equipped with omnidirectional cameras and 802.11b network cards.**

I. Introduction

Many approaches for motion coordination of large scale multi-robot systems use the leader-following framework [1], [3], [4]. In this framework, each robot has at least one designated leader. Leaders can be other robots in the group or virtual robots that represent pre-computed trajectory supplied by a higher level planner. Thus, each robot is a follower that tries to maintain a specified relative configuration (a fixed separation and bearing for example) to its leader(s).

One disadvantage with this framework is that there is a explicit dependence of the motion of followers on their leaders, but the leaders' motion is independent of their followers. If, for example, a robot fails or slows down, its followers' motion will be directly affected by this behavior, while its leaders will continue their task without modifying their plans. In situations where it is important to maintain a sensing or communication network, a single failure could result in the failure of the task.

In this paper we modify the notion of leader-following and present a framework where robots change their motion plans in real time in order to satisfy constraints related to other robots. These constraints may have to do with a task of maintaining a pre-specified formation. Alternatively, robots may have constraints because of limited ranges or fields of view of sensors, or of communication radios and antenna. Thus we introduce *cooperative leader-following*, a modification of the standard leader-following approach, where the motion of the robots can be dependent not only on their leaders but also on other robots including their followers.

In our previous work [5], [6], we solved a similar problem assuming that all robots had prior knowledge of the motion plan for the group in the form of a navigation function. Further, all the robot velocities derived from the gradients of the navigation function were assumed to be equal or close to each other. Each robot could deviate from this motion plan in order to satisfy *formation constraints*, inequality constraints on individual robots induced by the other robots in the team. This approach was used for cooperative manipulation [5], and to maintain communication and sensing constraints [6]. In this paper we use a similar approach for formation control. We still use the same definition of formation constraint, but now we consider formation setpoints as in [1], [2], leading to equality constraints. Thus, our constraints are divided into two types: (i) equality constraints that specify the relative position between a robot and its leader, and (ii) inequalities constraints that characterize a configuration space between a robot and other robots in the group. The equality constraints (i) are standard constraints in formation control [1], [2], [3], [4]. Constraints of type (ii) are used to maintain communication and sensor constraints allowing the group to deviate from the prescribed formation in (i). This allows, for example, a leader to wait for a possibly slow follower, allowing the team to adapt to failures. This kind of behavior can be found in centralized formation control approaches such as the one presented in [7], but are not explored in any decentralized control policies [1], [3]. Our goal in this paper, is to address both types of specifications (i) and (ii) with decentralized control policies.

Our framework uses potential field controllers [8], [9], [10]. For a single robot navigating an obstacle field, a potential function with a single minimum in the goal position provide a Lyapunov function that guarantees the robot's convergence to the goal [9]. Potential functions can be locally modified to accommodate unmodeled obstacles or dynamic constraints [11]. When the domain is convex, these modifications lead to guarantees on global performance. In order to leverage these results, we design potential functions for a robot to enable it to follow a leader. Each potential function is a function of the leader's and follower's position. We change the potential functions in real time to accommodate dynamic constraints while providing results on convergence for a team of holonomic, fully-actuated robots. We also point to extensions to

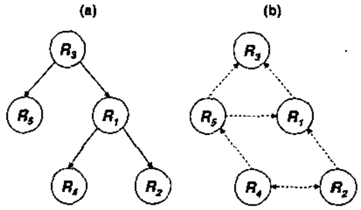

Fig. 1. Graph modeling for a group of 5 robots: (a) – formation control graph; (b) – constraint graph.

non-holonomic robots and present experiments with our team of car-like robots with omnidirectional cameras and wireless network cards.

II. Problem Definition

Consider a planar world, $\mathcal{W} = \mathbb{R}^2$, occupied by a group $\mathcal{R} = \{R_1, R_2, \dots R_n\}$ of n robots. The i^{th} robot R_i is represented by the configuration q_i in the configuration space $\mathcal{C}$. A formation of n robots is represented by a directed graph called *formation control graph*[1], $G_f = (\mathcal{R}, \mathcal{E}_f)$, and a second directed graph called *constraint graph*, $G_c = (\mathcal{R}, \mathcal{E}_c)$, where $\mathcal{R}$ is the set of nodes and $\mathcal{E}_f$ and $\mathcal{E}_c$ are edge sets.

For the formation control graph, G_f, each edge $e_{ij} = (R_i, R_j) \in \mathcal{E}_f$ is associated with a specification for R_j following R_i. For each edge, R_i is the *leader* and R_j is the *follower*. The robot that does not have any leaders and is responsible for guiding the others through the environment is called the *lead robot* [1]. Only one lead robot is allowed in our approach. Also, the robots that do not have any followers are called *terminal followers*. Figure 1(a) shows an example of a formation control graph where R_3 is the lead robot and R_2, R_4 and R_5 are terminal followers. Robot R_1 follows R_3 and is followed by R_2 and R_4.

The edges $e_{ij} = (R_i, R_j) \in \mathcal{E}_c$ of G_c are associated with constraints on relative position and orientation. While $\mathcal{E}_f$ describes leader-following relationships and set-points for the shape of the formation, $\mathcal{E}_c$ describes inequalities that reflect constraints such as communication and sensing constraints. Figure 1(b) shows an example of a constraint graph. In this figure R_3, for example, needs to maintain constraints with respect to R_1 and R_5. The bidirectional edge between R_2 and R_4 indicates that these robots need to maintain constraints with each other.

With the previous model, the control problem can be divided in two parts namely *graph assignment* and *controller design*. The first problem involves designing G_f and G_c and is not the main focus of this paper. Measures of performance that depend on G_f are discussed in [12] and heuristics for selecting edges are described in [13]. This paper is concerned with the problem of maintaining the formation described by G_f and the constraints described by G_c. We assume that graphs themselves are preassigned and focus our attention on controlling the robots to satisfy the edge specifications. For G_f, the specification for each edge is a configuration for robot R_j with respect to its leader R_i. On the other hand, the specification for each edge in G_c is a convex function $g(q_i, q_j)$ that represents the allowable configuration space for R_j parameterized by the configuration of R_i. While G_f specifies, for each robot (except the lead), a unique point in configuration space, G_c specifies the allowable subset of configuration space.

Although G_f and G_c are apparently independent, in order to allow robot R_i to reach its set-point $q_i^d(q_j)$ specified by G_f and still satisfy the constraints specified by G_c, we need to guarantee that, except for the lead robot, q_i^d is inside the allowable configuration space $\mathcal{C}_i^d(q_1, \dots, q_{i-1}, q_{i+1}, \dots, q_n)$ defined by all constraints in G_c. Thus, the edge definition for the two graphs must satisfy the following condition:

$$(q_1^d, q_2^d, \dots, q_n^d) \in \mathcal{C}_1^d \times \mathcal{C}_2^d \times \dots \times \mathcal{C}_n^d. \quad (1)$$

Moreover, since each $\mathcal{C}_i^d$ is an intersection of convex sets, the right hand side of (1) is also a convex set. Therefore, if the robots are initially inside this set, they can always reach their goal configurations without going out of the set.

Our goal in this paper is to design control laws that take in account the formation set-points and the allowable configuration spaces. Before continuing any further we will make three assumptions:

Assumption 1 All robots are identical in terms of geometry, and in terms of capabilities and constraints related to sensing, communication, control, and mobility.

Assumption 2 The robots are fully-actuated, holonomic, point robots. For the i^{th} robot, the dynamical model is then given by: $\dot{q}_i = u_i$, where $q_i = (x_i, y_i)$.

Assumption 3 G_f is acyclic and the in-degree at each node is 1. In other words, every follower has only one leader[2].

III. Potential Functions

In this paper, we use artificial potential fields to plan and control the robots' motion. Potential field methods yield closed loop controllers that allow convergence to the goal in the presence of actuator and sensor noise and other disturbances [8]. Thus, for a potential function, ϕ_i, robot R_i's input is given by $u_i = -k\nabla\phi_i(q_i)$ where $\nabla\phi_i(q_i)$ is the gradient of ϕ_i computed in the configuration q_i. The integral curves of the vector field formed by $-\nabla\phi_i(q_i)$, define implicity paths from every start configuration in $\mathcal{C}$ to the goal configuration q_i^d. As pointed out in [11], a

[1] The term *control graph* is used in [1], [2] to describe what we are calling a formation control graph.

[2] This is somewhat restrictive since the in-degree for systems with two inputs can be up to two [1], [2].

potential function with a single minimum in q_i^d can be thought of as a Lyapunov function for the system $\dot{q}_i = u_i(q_i)$, $u_i(q_i) = -\nabla\phi_i(q_i)$, because $\phi_i(q_i)$ is positive definite and its value is, by definition, always decreasing along system trajectories.

In the leader-following problem, we chose a navigation function [9] as a potential function for the lead robot. Navigation functions are constructed as the functions that solve the non-cooperative problem of steering a individual robot towards the goal while avoiding the static obstacles in the environment. The methodology presented in [9] can be directly applied for designing the function. While these functions can be very complicated, we limit the class of functions to quadratic functions in this paper.

For the robots that have at least one leader, the potential function is constructed as a function of the leaders' position. In the case that R_i follows R_j, we can describe the follower's relative configuration in local coordinates as $\bar{q} = (q_j - q_i)$. We consider a quadratic Lyapunov function candidate of the form:

$$\phi_i(\bar{q}) \doteq \frac{1}{2}\|\bar{q}^d - \bar{q}\|^2 .$$

If ϕ_i is a Lyapunov function we can use it as a leader-following potential function. The input for the follower robot R_i is then given by the negative of the gradient of $\phi_i(q_i, q_j)$ as:

$$u_i = -k\nabla\phi_i = -k(\bar{q}^d - \bar{q}) ,$$

where k is a positive constant and $\nabla\phi_i = \partial\phi_i/\partial q_i$. The derivative of the potential function for this input is given by:

$$\begin{aligned}\dot{\phi}_i &= -\nabla\phi_i \cdot \dot{\bar{q}} = -\nabla\phi_i \cdot (\dot{q}_j - \dot{q}_i) = -\nabla\phi_i \cdot (\dot{q}_j - u_i) \\ &= -\nabla\phi_i \cdot (\dot{q}_j + k\nabla\phi_i) = -k\|\nabla\phi_i\|^2 - \nabla\phi_i \cdot \dot{q}_j .\end{aligned}$$

Observe that $\dot{\phi}_i$ decreases along the system trajectory if:

$$k\|\nabla\phi_i\|^2 > -\nabla\phi_i \cdot \dot{q}_j .$$

In the worst case, $\dot{q}_j$ and $-\nabla\phi_i$ are parallel and the previous condition can be written as the following sufficient condition:

$$k\|\nabla\phi_i\| > \|\dot{q}_j\| . \tag{2}$$

Because a real robot is subject to dynamics, there is a practical limit on its velocities. We assume each robot (*i.e.*, all leaders) have a maximum velocity of $\dot{q}_{max}$. From Equation (2), it is clear that if we exclude the region given by the ball:

$$\|\bar{q}^d - \bar{q}\| < \gamma = \frac{\dot{q}_{max}}{k} ,$$

$\dot{\phi}_i$ decreases along the trajectories of the system. Thus we can show that trajectories that start outside the ball (*i.e.*, when $\|\bar{q}^d - \bar{q}\| > \gamma$), will converge to the ball. In other words, $\dot{\phi}_i < 0$ for $\|\bar{q}^d - \bar{q}\| \geq \gamma$. The constant γ is the maximum allowable steady state error in $\bar{q}$.

We note that it is possible to make γ arbitrarily small by allowing for feedforward control. If the follower input is given by:

$$u_i = -k\nabla\phi_i + \dot{q}_j = -k(\bar{q}^d - \bar{q}) + \dot{q}_j ,$$

where $\dot{q}_j$ is feedforward information, the controller exponentially converges to $\bar{q} = 0$. The feedforward velocity requires estimation of the leader's velocity by the follower robot and is discussed elsewhere [2].

IV. CONSTRAINTS

As mentioned before, with each edge $(R_i, R_j) \in \mathcal{E}_c$, we associate a configuration constraint for R_j induced by R_i as a inequality of the form $g(q_i, q_j) \leq 0$. For example, if R_j must keep R_i in sight using a omnidirectional camera, $g(q_i, q_j) = (x_i - x_j)^2 + (y_i - y_j)^2 - r^2$ [6].

In this paper each constraint $g(q_i, q_j)$ defines three regions in the configuration spaces of R_i and R_j (see Figure 2). In the *safe region*, $g(q_i, q_k) < \delta$, where the small negative number δ can be thought of as a threshold. The region defined by $\delta \leq g(q_i, q_k) < 0$ is the *critical region* for the robot. The constant δ is designed in order to guarantee that the constraint is still satisfied in the critical region and also to ensure that the robot does not leave this region. We say that a constraint is active when $g(q_i, q_k) \geq \delta$. If $g(q_i, q_k) \geq 0$ the robot is in the *unsafe region*. Depending on the nature of the constraints, the robots may not be able to return to the safe region of the configuration space. Our decentralized controllers are designed with the objective of keeping the robots in the safe configuration space.

V. CONTROLLERS

Our control system is decentralized and implemented using a set of three reactive controllers. Based on the two graphs, G_f and G_c, described in Section II we define a third graph that will govern the switching between the controllers. We call this time dependent graph that changes with the state of the robots, the *formation graph* $H = (\mathcal{R}, \mathcal{E}_h)$, where $\mathcal{E}_h$ is defined as the union of two subsets of $\mathcal{E}_f$ and $\mathcal{E}_c$:

$$\mathcal{E}_h = \bar{\mathcal{E}}_f \cup \bar{\mathcal{E}}_c ,$$
$$\bar{\mathcal{E}}_f = \{e_{ij} | [e_{ij} \in \mathcal{E}_f] \wedge [g(k, i) < 0 \forall e_{ki} \in \mathcal{E}_c, R_k \in \mathcal{R}]\} ,$$
$$\bar{\mathcal{E}}_c = \{e_{ij} | [e_{ij} \in \mathcal{E}_c] \wedge [g(i, j) \geq \delta]\} .$$

Thus based on H, each robot R_i has three basic behaviors or modes depending on the number and type of incoming edges at R_i. If there is only one incoming control edge ($e_{ji} \in \mathcal{E}_f$), the robot is in the SAFE mode, corresponding to the safe region in Section IV. The control law in this mode is given by:

$$u_i = -k\nabla\phi_i , \tag{3}$$

where $\nabla\phi_i$ is the gradient vector of the potential function ϕ_i, and k is a positive gain. For the lead robot it is

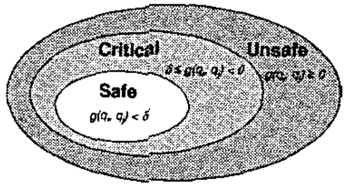

Fig. 2. The activation of the constraints define three regions in the robots' configuration space.

a deliberative controller with a pre-planned navigation function that guides the robot toward the goal. For the followers it is a reactive controller designed to maintain the edge of the graph as shown in Section III.

When all incoming edges are constraint edges ($e_{ji} \in \mathcal{E}_c$), R_i is in the UNSAFE mode. The robot tries to move in order to satisfy the constraints without using the potential function. In other words, the constraints themselves act as potential fields attracting the robots to each other and forcing them into a feasible configuration that satisfies all the constraints. The control input in this mode is:

$$u_i = -k\,\nabla g^j\,, \tag{4}$$

where ∇g^j is the gradient of the constraint $g^j(q_i, q_j)$ defined by $\partial g^j/\partial q_i$ and g^j is the constraint induced on R_i by R_j. If R_i has more than one active constraint to be satisfied, ∇g^j represent the sum of their gradients.

The third mode is a linear combination of the other two. A robot switches to this mode if it has one incoming control edge ($e_{ji} \in \mathcal{E}_f$) and at least one constraint edge ($e_{ji} \in \mathcal{E}_c$). The robot must navigate toward its goal while maintaining the constraint whose boundary it is closest to. This mode is called the CRITICAL mode. The input in the mode is:

$$u_i = -k(\alpha_i\,\nabla g^j + \nabla\phi_i)\,, \tag{5}$$

where $0 < \alpha_i \leq 1$. The constant α_i determines how much each robot will deviate to its main objective in order to preserve a constraint. It must be chosen so that:

$$\alpha_i < \frac{\|\nabla\phi_i\|^2}{|\nabla\phi_i \nabla g^j|}\,, \tag{6}$$

if $\nabla\phi_i\nabla g^j < 0$ and 1 otherwise. This condition guarantees that u_i has positive projection along $\nabla\phi_i$.

As an example of how the switching among the control modes is governed, consider a possible H generated by the combination of the two graphs of Figure 1. In Figure 3, the dotted arrows show active constraints, while solid arrows denote equality specifications. R_5 is in the UNSAFE mode, R_1 is in the SAFE mode, and R_2, R_3, and R_4 are in the CRITICAL mode of the controller.

It can be shown that the above switched control system solves the n problems of individually controlling the robots while guaranteeing the constraints are satisfied in the following sense:

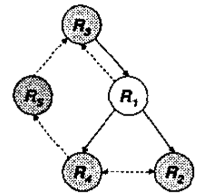

Fig. 3. A formation graph based on the combination of the graphs of Figure 1. Based on incoming edges, R_5 has configuration space constraints on its position relative to R_4, R_1 follows a potential function to acquire a position relative to R_3, while R_2, R_3, and R_4 must execute a combination of two reactive behaviors.

If the robots initially satisfy the constraints and never enter the unsafe region of the configuration space, the controllers given by Equations 3 and 5 guarantee that the lead robot goes to its destination and the team achieves the desired formation.

The proof for this is straight forward. Observe that ϕ_i, which is locally positive definite, is a common Lyapunov function for both behaviors:

$$\frac{d\phi_i(q_i)}{dt} = \dot{\phi}_i = \nabla\phi_i \cdot \dot{q}_i = \nabla\phi_i \cdot u_i$$

In the SAFE mode:

$$\dot{\phi}_i = \nabla\phi_i \cdot u_i = -k\nabla\phi_i \cdot \nabla\phi_i = -k\|\nabla\phi_i\| \leq 0$$

In the CRITICAL mode:

$$\begin{aligned}\dot{\phi}_i &= -k\nabla\phi_i \cdot (\alpha_i\,\nabla g^j + \nabla\phi_i)\\ &= -k(\alpha_i\,\nabla\phi_i \cdot \nabla g^j + \nabla\phi_i \cdot \nabla\phi_i) \leq 0\end{aligned}$$

since α_i is properly chosen. Therefore, in these two modes the control law is free of local minima since $\nabla\phi_i = 0$ if and only if $q_i = q_i^d$, by the definition of ϕ_i.

However, one shortcoming of the above analysis is that it fails to include an analysis of the stability of the system. While we have shown that under the stated assumptions the value of potential function for all the robots decreases as a function of time, we have not shown the system is Lagrange stable.

The above analysis lends itself to stronger results for specific graphs. Consider, for example, a group of robots in a linear formation where each robot has to satisfy a constraint with its immediate follower. The system would never enter in the unsafe mode if for a generic active constraint, $g(q_i, q_j)$, the control input ensure $\dot{g}(q_i, q_j) \leq 0$. The time derivative of $g(q_i, q_j)$ is given by:

$$\dot{g}(q_i, q_j) = \frac{\partial g}{\partial q_i}\dot{q}_i + \frac{\partial g}{\partial q_j}\dot{q}_j\,. \tag{7}$$

Denote $\partial g/\partial q_i$ by ∇g^i, and $\partial g/\partial q_j$ by ∇g^j. The key observation is $\partial g/\partial q_j = -\partial g/\partial q_i$, or $\nabla g^i = -\nabla g^j$.

In the worst case, both the leader R_i and the follower R_j are in a critical mode. Assume that there is a constraint $g(q_j, q_k)$ with gradient ∇g^k active for the follower R_k.

Substituting $\dot{q}_i$ and $\dot{q}_j$ in (7) by the control inputs in (5) we rewrite the time derivative of $g(q_i, q_j)$ as:

$$\dot{g}(q_i, q_j) = -k\nabla g^j(\alpha_i \nabla g^j + \nabla\phi_i) + \\ + k\nabla g^j(\alpha_j \nabla g^k + \nabla\phi_j).$$

For the specific case of circular constraints ($g(q_i, q_j) = (x_i - x_j)^2 + (y_i - y_j)^2 - r^2$), observe that ∇g^j is anti-parallel to $\nabla\phi_j$. Also, notice that, for a linear formation in steady-state, $\nabla\phi_i$ and ∇g^k are anti-parallel as well. Grouping those vectors together we rewrite the previous equation as:

$$\dot{g}(q_i, q_j) = -k\nabla g^j(\alpha_i \nabla g^j - \nabla\phi_j) + \\ + k\nabla g^j(\alpha_j \nabla g^k - \nabla\phi_i), \qquad (8)$$

where the first term is contributing to satisfy the constraints and the second has the opposite effect.

By (8), observe that if $\nabla g^k = 0$ there is always a value of α_i that guarantees that the constraint is satisfied. Thus, starting with the terminal follower, for which ∇g^k does not exist, and finishing in the lead robot, it is easy to see that all constraints can be satisfied. Again considering the chain of robots, notice that small velocities of the lead robot (small $\nabla\phi_i$) contribute to satisfy the constraints. Also, observe that if robot R_j fails, causing $\nabla\phi_j = \nabla g^k = 0$, the constraint will eventually be violated and R_i will enter its unsafe configuration space. In practice, this situation causes all interconnected robots to continuously switch between the UNSAFE and CRITICAL modes forcing the group to stop. It is in some way a desirable condition since the main idea of the methodology is forcing the robots to wait for their teammates. Other similar observations can be made depending on the values of the other terms in (8).

The controllers of this section have considered fully actuated robots. Since in practice most of the robots are underactuated, it is also natural to ask if the methodology can be extended to non-holonomic robots. For non-holonomic robots we can derive controllers that will allow a reference point (x, y) to follow a desired trajectory including those that are specified by potential field controllers. We rely on this idea for implementation of the methodology on our real world robot platforms. In the next section we present experimental results demonstrating the performance of our cooperative leader following approach on three of our car-like robot platforms.

VI. Experimental Results

Our car-like robots are equipped with omnidirectional cameras as their primary sensors (see Figure 4). The communication among the robots relies on IEEE 802.11b networking. A calibrated overhead camera is used to localize the robots in the environment. Because with this camera we do not estimate the robots' orientation, we use communication between the robots in order to construct a

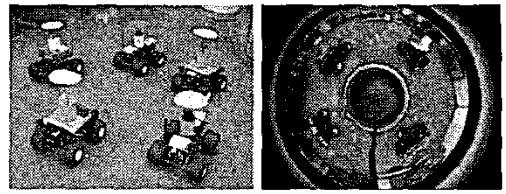

Fig. 4. The GRASP Lab. robots (left) and a sample image from an omni-directional camera (right).

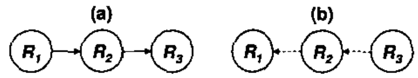

Fig. 5. (a) – The control graph and (b) – the constraint graph for the experiment.

complete knowledge of the robots configuration. The communication is essentially used for cooperative localization but is not used for control or decision making. See [2] for details.

A limitation of the omnidirectional cameras used by the robots is that their resolution decreases with the distance of the objects. At $2\,m$, for instance, the projection of an observed robot in the image plane is only one pixel in size. Since visibility of other robots is important for orientation estimation, the three robots must maintain sensing constraints with their neighbors. Thus, in the experiments, the three robots are commanded to maintain a line formation as shown by G_f and G_c in Figure 5. The function $g(q_i, q_j)$ was set as a circle of radius $1.6\,m$. Observe that this radius is much smaller than the distance where the robots are actually blind ($2\,m$) in order to guarantee that the task is completed even if the robots enter in their unsafe configuration spaces. The threshold δ for the critical region was chosen to make it a circle of $1.3\,m$ radius.

Figure 6 shows four snapshots of our experiment. In (c) the last robot (R_3) was manually stopped. In what follows all the robots switch to their UNSAFE modes. When R_3 starts moving again the robots switch back to their CRITICAL and then SAFE modes and complete their tasks. Figure 7 shows the y coordinates of the robots for the same experiment.

VII. Conclusions and Future Work

We have presented a methodology that allows robots to maintain constraints while pursuing a specified formation. The robots follow potential field controllers to achieve the desired formation, but modify their motion to accommodate constraints. When the specified formation satisfies the constraints, we can show that the robots eventually reach the desired formation. Because the robots are sensitive to constraints, they stop as a team to accommodate failed robots. Thus, they are in one sense, more robust to failures. Although, our approach considers fully actuated, point robots, we have presented encouraging experimental

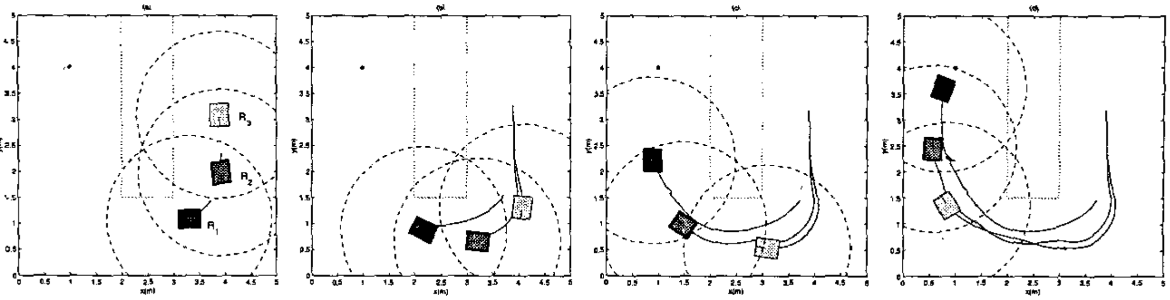

Fig. 6. Four snapshots of an experiment where three robots are in line formation and keeping visibility constraints with their followers. The goal configuration for the lead robot is marked with a (*). The dashed circumferences represent the sensors' field of view. In (c) robot R_3 was manually stopped for 7 seconds. The robots stop following their potential functions and wait for R_3 so that the constraints are preserved.

results with a group of non-holonomic robots arranged in a simple formation.

Future work includes methodologies and algorithms for the assignment of control and constraint graphs and the analysis of the system behavior for complex formations.

VIII. Acknowledgments

This work was supported in part by DARPA MARS grant NBCH1020012, AFOSR grant F49620-01-1-0382, NSF grant IIS02-22927, and CNPq-Brazil grants 200765/01-9 and 350750/1994-7.

IX. References

[1] J. P. Desai, J. Ostrowski, and V. Kumar, "Controlling formations of multiple mobile robots," in *Proc. IEEE Int'l Conf. on Robot. and Automat.*, pp. 2864–2869, 1998.

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

[3] T. Balch and R. C. Arkin, "Behavior-based formation control for multi-robot teams," *IEEE Trans. on Robot. and Automat.*, vol. 14, no. 6, pp. 1–15, 1998.

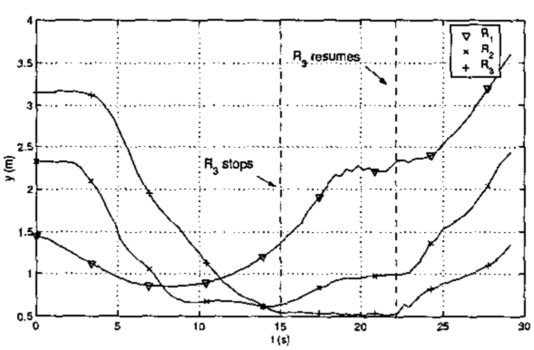

Fig. 7. The y coordinate for the experiment in Fig. 6. The terminal follower, R_3, was stopped for approximately $7\,s$ at the time $15\,s$. It causes the other robots to switch to their UNSAFE modes and stop, as was expected.

[4] J. Lawton, B. Young, and R. Beard, "A decentralized approach to elementary formation maneuvers," in *Proc. IEEE Int'l. Conf. Robot. Automat.*, pp. 2728–2733, 2000.

[5] G. A. S. Pereira, V. Kumar, and M. F. M. Campos, "Decentralized algorithms for multirobot manipulation via caging," in *Int'l Workshop on Algorithmic Foundations of Robotics*, 2002.

[6] G. A. S. Pereira, A. Das, V. Kumar, and M. F. M. Campos, "Decentralized motion planning for multiple robots subject to sensing and communication constraints," in *Multi-Robot Systems Volume II: From Swarms to Intelligent Automata* (A. Schultz, L. E. Parker, and F. Schneider, eds.), Kluwer Academic Press, 2003.

[7] M. Egerstedt and X. Hu, "Formation constrained multi-agent control," in *Proc. IEEE Int'l Conf. on Robot. and Automat.*, pp. 3961–3966, 2001.

[8] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int'l Journal of Robotics Reaserch*, vol. 5, no. 1, pp. 90–98, 1986.

[9] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potencial functions," *IEEE Trans. on Robot. and Automat.*, vol. 8, no. 5, pp. 501–517, 1992.

[10] N. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Proc. IEEE Conf. on Decision and Control*, 2001.

[11] J. M. Esposito and V. Kumar, "A method for modifying closed-loop motion plans to satisfy unpredictable dynamic constraints at runtime," in *IEEE Int'l. Conf. on Robotics and Automation*, pp. 1691–1696, 2002.

[12] H. G. Tanner, V. Kumar, and G. J. Pappas, "The effect of feedback and feedforward on formation ISS," in *Proc.IEEE Int'l Conf. on Robot. and Automat.*, pp. 3448–3453, 2002.

[13] A. K. Das, R. Fierro, and V. Kumar, "Control graphs for robot networks," in *Cooperative Control and Optimization* (R. Murphey and P. Pardalos, eds.), Applied Optimization, ch. 4, Kluwer Academic Press, 2002.