Controlling Shapes of Ensembles of Robots of Finite Size with Nonholonomic Constraints

Nathan Michael and Vijay Kumar
University of Pennsylvania
Philadelphia, Pennsylvania 19104-6228
Email: {nmichael, kumar}@grasp.upenn.edu

***Abstract*—In this paper we focus on the construction of distributed formation control laws that permit the control of individual mobile ground robots in a formation to a desired distribution with minimal knowledge of the global state. As in previous work, we consider an abstraction of the team that is derived from a shape descriptor of the ensemble and the position and orientation of the ensemble. We consider the control of the abstract state with decentralized control laws which are independent of the number of agents. However, we incorporate an important departure from previous work by explicitly modeling the shape of the robot, the geometric, non-interpenetration constraints and nonholonomic, kinematic constraints. Further, we propose a motion planning technique to plan motions for ensembles of robots and a technique for the splitting and merging of groups and subgroups. We demonstrate the effectiveness of the algorithms on a team of differential drive robots in simulation and on real hardware.**

I. Introduction

Effective strategies for controlling large teams of robots in complex environments are becoming increasingly relevant as the development of pervasive embedded computing, sensing, and wireless communication enables the application of multi-agent systems to challenging tasks such as environmental monitoring [1], surveillance and reconnaissance for security and defense [2], and support for first responders in a search and rescue operation [3]. In such scenarios, it is necessary to apply control strategies that allow robots to adapt to different environments and execute complex tasks, while avoiding collisions. Further, robot controllers must be robust to permit robot failures or changes in the team size.

Several methodologies exist to control large teams of robots. One way of reducing the complexity of the controller is to require the team to conform to a geometric rigid virtual structure [4]. Most of the recent works on stabilization and control of virtual structures model formations using *formation graphs* [5]. The controllers guaranteeing local asymptotic stability of a given rigid formation can be derived using standard techniques such as input-output linearization [6], input-to-state stability [7], Lyapunov energy-type functions [8], and biologically-inspired artificial potential functions [9]. Virtual structures unnecessarily constrain the problem, making this approach inappropriate for tasks in complex environments. Additionally, graph formulations and leader-follower architectures require identification and ordering of robots, which makes the overall architecture sensitive to failures.

The problem of controlling the trajectory of the group and shape of a large team of point robots was studied in [10], [11]. The authors defined an abstraction of the team that has a product structure of the Euclidean group and a shape space, and is independent of the number of robots. The group captures the pose of an ellipsoid spanning the team with semi-axes given by the shape variables. The overall abstract description is invariant to robot permutations. In addition, the model and the formulation is invariant to left actions of the group. This description allows one to define and control the behavior of the abstract state or the abstract description of the team at a high level, with automatic generation of individual robot control laws based only on the feedback of this abstract state. However, the control laws do not account for the physical constraints of the robots and ignore inter-agent interactions.

Coverage control schemes proposed by [12], [13] and their variants have a similar flavor. They enable large groups of robots to use local information to distribute themselves so that a suitable integral over this distribution is maximized. However, this formulation does not lend itself to the control of the position and orientation of the overall team.

In this paper we focus on a basic problem, the control of the position and orientation of a formation of mobile robots and the adaptation of the shape to the environment. Two related problems, the planning of the shape and trajectory of the ensemble and the development of effective coordination strategies to split the team into subgroups and to merge two subgroups, are also considered. We view these problems and their solutions to be building blocks that can enable a robot team to navigate an environment, adapting to the constraints imposed by obstacles in the environment. In contrast to most previous work we model the physical shape of the robot and consider controllers that are guaranteed to avoid collisions between the robots. Although the approach assumes global observation of the abstract state, it is possible to develop an estimator for individual robots to estimate this abstract state [14]. However, in the context of our work, we emphasize the control of the team of robots and decouple the challenges of considering the estimation problem. Further, we believe that some element of centralization is essential to command a large team of robots.

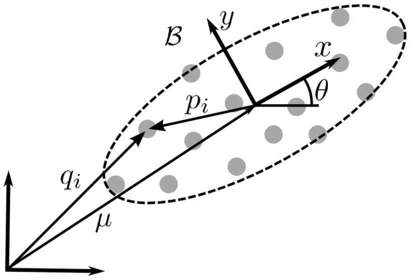


Fig. 1. The frame $\mathcal{B}$ fixed to the group of robots moves with respect to the inertial frame.

II. Background

This paper builds on the previous work in [10], [11], [15]. In these papers, an *abstraction* map is used to transform the high-dimensional state space into a smaller, tractable state space which captures only the position, orientation, and shape of the formation. The main advantages of this abstract representation are: (a) its dimension is independent of the number of robots in the team; and (b) it lends itself to planning in a lower dimensional space.

The state space of the N-robot system is constructed by creating N copies of Q_i, the state space of the i^{th} robot:

$$Q = Q_1 \times Q_2 \times \ldots \times Q_N.$$

The abstract space, M, whose dimension is smaller and independent of the dimension of Q, is defined by a smooth, differentiable map

$$\phi : Q \to M, \quad \phi(q) = \boldsymbol{x}, \tag{1}$$

where ϕ is a mapping of the higher-dimensional state $q \in Q$ to the lower-dimensional abstract state $\boldsymbol{x} \in M$. In this paper, we consider kinematic robots in the plane (see [11] for a treatment of the three-dimensional case). Thus, q reflects the collection of the positions of the robots, $q = [q_1, \ldots, q_i, \ldots, q_N]^{\mathrm{T}}$ where $q_i \in \mathbb{R}^2$.

As in previous work, the shape is modeled by characterizing the distribution of robots about the mean position. The centroid of the group is given by:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} q_i.$$

We can define a local frame, $\mathcal{B}$, whose origin is at the centroid, as shown in Fig. 1, by requiring the orientation to be such that the coordinates of the robots in this frame, $p_i = [x_i, y_i]$, satisfy

$$\sum_{i=1}^{N} x_i y_i = 0.$$

The distribution of robots in this local frame can be approximated by the inertia tensor (assuming uniform unit mass) or by a matrix of second moments:

$$\mathcal{I} = \sum_{i=1}^{N} p_i \, p_i^T = \left[\begin{array}{cc} \mathcal{I}_{11} & 0 \\ 0 & \mathcal{I}_{22} \end{array} \right].$$

We define two shape variables proportional to the diagonal elements:

$$s_1 = \kappa \mathcal{I}_{11}, \quad s_2 = \kappa \mathcal{I}_{22},$$

where $\kappa \neq 0$. Choosing $\kappa = \frac{1}{N-1}$ gives the shape variables a geometric interpretation. They become the semi-major and semi-minor axes for a concentration ellipse for a group of robots whose coordinates in the plane are chosen to satisfy a normal distribution. Alternatively, the shape variables may be defined such that abstract shape is described by a bounding rectangle as discussed in [10].

The abstract description of the team of robots, $\boldsymbol{x}$, is given by the position and orientation of the team, g, and the shape s. In this paper, we take g to be the position and orientation of $\mathcal{B}$:

$$g = \left[\begin{array}{ccc} \cos\theta & -\sin\theta & \mu_1 \\ \sin\theta & \cos\theta & \mu_2 \\ 0 & 0 & 1 \end{array} \right]$$

where $\mu = (\mu_1, \mu_2)$ are the components of the centroid in the inertial frame and the shape $s = (s_1, s_2)$. The map ϕ defined in this way can be easily shown to be a submersion [10].

The abstract space, M, is naturally decomposed into a shape space, S, and a Lie group, G, which in our case is $SE(2)$. Since ϕ is a submersion, it follows that there is a unique $\dot{x}$ for every $\dot{q}$ but not the other way around.

Using the natural kinetic energy metric on Q, it is possible to derive the optimal velocity (tangent vector) at any point $q \in Q$ for a desired $\dot{\boldsymbol{x}}$ at the corresponding point $\boldsymbol{x} = \phi(q) \in M$. It was shown in [10] that this input $u^\star$, for the system can be found by considering the time derivative of the transformation described by (1),

$$d\phi \dot{q} = \dot{\boldsymbol{x}}. \tag{2}$$

From (1), the definitions of (μ, θ, s_1, s_2), and algebraic simplification, the transformation $d\phi$ becomes

$$d\phi = \kappa \begin{bmatrix} \frac{1}{\kappa N} I_2 & \cdots & \frac{1}{\kappa N} I_2 \\ \frac{(q_1-\mu)^{\mathrm{T}}}{s_1 - s_2} H_3 & \cdots & \frac{(q_N-\mu)^{\mathrm{T}}}{s_1 - s_2} H_3 \\ (q_1 - \mu)^{\mathrm{T}} H_1 & \cdots & (q_N - \mu)^{\mathrm{T}} H_1 \\ (q_1 - \mu)^{\mathrm{T}} H_2 & \cdots & (q_N - \mu)^{\mathrm{T}} H_2 \end{bmatrix}. \tag{3}$$

Here H_1, H_2, and H_3 are defined by

$$H_1 = I_2 + R^2 E_2, \quad H_2 = I_2 - R^2 E_2, \quad H_3 = R^2 E_1,$$

where I_2 is the 2×2 identity matrix and

$$E_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad E_2 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

In [10] (and the extension to three-dimensions in [11]) it was shown that the minimum-energy solution satisfying (2) is obtained using the Moore-Penrose Inverse:

$$u^\star = d\phi^{\mathrm{T}} (d\phi d\phi^{\mathrm{T}})^{-1} \dot{\boldsymbol{x}}. \tag{4}$$

Further algebraic simplification of (4) using (3) results in the control law for each individual agent, $u_i = \dot{q}_i$,

$$\begin{aligned} u_i^\star = \dot{\mu} &+ \frac{s_1 - s_2}{s_1 + s_2} H_3 (q_i - \mu) \dot{\theta} \\ &+ \frac{1}{4 s_1} H_1 (q_i - \mu) \dot{s}_1 + \frac{1}{4 s_2} H_2 (q_i - \mu) \dot{s}_2. \end{aligned} \tag{5}$$

Although the description of shape variables is fairly simple, it is generalizable to include higher moments (beyond second order). However, the development of minimum-norm control inputs, such as (5), are harder with more general shapes.

In the next section, we will pursue a slightly different formulation by writing these equations in the moving frame $\mathcal{B}$.

III. Problem Formulation

A. Dynamics in the moving frame

At any point $\boldsymbol{x} = (g, s) \in M$ in the abstract space, the derivative can be written as:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{g} \\ \dot{s} \end{bmatrix} = \begin{bmatrix} g & 0 \\ 0 & I_2 \end{bmatrix} \begin{bmatrix} \xi \\ \sigma \end{bmatrix}. \tag{6}$$

$\dot{\boldsymbol{x}} = (\dot{g}, \dot{s})$ is the time derivative of the abstract space in the inertial frame while $\zeta = (\xi, \sigma)$ is the time derivative in the moving frame $\mathcal{B}$, and

$$\Gamma = \begin{bmatrix} g & 0 \\ 0 & I_2 \end{bmatrix}$$

is a non-singular 5×5 transformation matrix. If ν_i is the robot velocity in the frame $\mathcal{B}$ so that $u_i = R\nu_i$,

$$\kappa \begin{bmatrix} \frac{I_2}{\kappa N} & \cdots & \frac{I_2}{\kappa N} \\ \frac{1}{s_1 - s_2} {p_1}^{\mathrm{T}} E_1 & \cdots & \frac{1}{s_1 - s_2} {p_N}^{\mathrm{T}} E_1 \\ {p_1}^{\mathrm{T}}(I_2 + E_2) & \cdots & {p_N}^{\mathrm{T}}(I_2 + E_2) \\ {p_1}^{\mathrm{T}}(I_2 - E_2) & \cdots & {p_N}^{\mathrm{T}}(I_2 - E_2) \end{bmatrix} \begin{bmatrix} \nu_1 \\ \vdots \\ \nu_N \end{bmatrix} = \begin{bmatrix} \xi \\ \sigma \end{bmatrix}.$$

The minimum-energy solution (5) can be written as:

$$\nu^\star = d\phi^{\mathrm{T}} (d\phi d\phi^{\mathrm{T}})^{-1} \zeta, \tag{7}$$

with the simplification:

$$\begin{aligned} \nu_i^\star = & \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} + \frac{s_1 - s_2}{s_1 + s_2} E_1 p_i \xi_3 \\ & + \frac{1}{4s_1}(I_2 + E_2) p_i \sigma_1 + \frac{1}{4s_2}(I_2 - E_2) p_i \sigma_2. \end{aligned} \tag{8}$$

The control law defined by (8) does not consider inter-agent collisions or the spatial size of individual robots. Although [15] proposed an extension that resolved collisions, the strategy requires communication and negotiations during collisions, which adds unnecessary complexity. In the next subsection we address an approach in which collision avoidance is done without explicit arbitration.

B. Collision avoidance

The separation distance between the reference points on robots i and j is:

$$\delta_{ij} = \|p_i - p_j\|.$$

To avoid collisions between robots, we define a safe separation distance between two robots:

$$\epsilon = 2\rho + \epsilon_s, \tag{9}$$

where ρ is the radius of each robot and ϵ_s is a specified safety region.

We define the neighborhood $\mathcal{N}_i$ as the set of all robots sensed by or communicating with robot i such that i is able to gain knowledge of its neighbors' positions and velocities, $\{p_j, \nu_j\}$, $\forall j \in \mathcal{N}_i$. To ensure that the robots do not collide, we require that

$$(p_i - p_j) \cdot (\nu_i - \nu_j) \geq 0, \tag{10}$$

for all $j \in \mathcal{N}_i$ such that $\delta_{ij} \leq \epsilon$.

C. Asymptotic convergence to a desired abstract state

In the absence of collisions, the easiest way to guarantee convergence to a time-invariant abstract state $\boldsymbol{x}^{des}$ is to require the error $\tilde{\boldsymbol{x}} = (x^{des} - x)$ to converge exponentially to zero:

$$\dot{\boldsymbol{x}} = K\, \tilde{\boldsymbol{x}},$$

or equivalently,

$$\zeta = \Gamma^{-1}\, K\, \tilde{\boldsymbol{x}}, \tag{11}$$

where K is any positive-definite matrix, and use (7, 8) to obtain robot velocities that guarantee globally asymptotic convergence to any abstract state.

We next discuss the first contribution of this paper, where we propose a control law that guarantees convergence to an abstract state satisfying certain conditions, while guaranteeing safety (i.e., there are no inter-agent collisions).

IV. Control with collision avoidance

A. Monotonic convergence

We relax the requirement of exponential convergence to an abstract state and replace it with a slightly different notion of convergence in order to accommodate the safety constraints in (10). Specifically instead of insisting on the minimum-energy solution, (7), we find the solution closest to the minimum-energy solution satisfying the safety constraints.

First, we require that the error in the abstract state decrease *monotonically*:

$$\tilde{\boldsymbol{x}}^{\mathrm{T}} K \dot{\boldsymbol{x}} \geq 0. \tag{12}$$

Substituting (2) into (12), this inequality reduces to

$$\tilde{\boldsymbol{x}}^{\mathrm{T}}\, K\, \Gamma \begin{bmatrix} I_2 & \cdots & I_2 \\ \frac{1}{s_1 - s_2} {p_1}^{\mathrm{T}} E_1 & \cdots & \frac{1}{s_1 - s_2} {p_N}^{\mathrm{T}} E_1 \\ {p_1}^{\mathrm{T}}(I_2 + E_2) & \cdots & {p_N}^{\mathrm{T}}(I_2 + E_2) \\ {p_1}^{\mathrm{T}}(I_2 - E_2) & \cdots & {p_N}^{\mathrm{T}}(I_2 - E_2) \end{bmatrix} \begin{bmatrix} \nu_1 \\ \nu_2 \\ \cdots \\ \nu_N \end{bmatrix} \geq 0. \tag{13}$$

A sufficient condition to satisfy this *monotonic convergence condition* in (13) is that each robot select inputs that satisfy:

$$\tilde{\boldsymbol{x}}^{\mathrm{T}}\, K\, \Gamma \begin{bmatrix} I_2 \\ \frac{1}{s_1 - s_2} {p_i}^{\mathrm{T}} E_1 \\ {p_i}^{\mathrm{T}}(I_2 + E_2) \\ {p_i}^{\mathrm{T}}(I_2 - E_2) \end{bmatrix} \nu_i \geq 0. \tag{14}$$

If all robots choose controls satisfying (14), the error in the abstract state will monotonically decrease. It is useful to show that the minimum-energy control law (8) satisfies this inequality.

Proposition 1. *The minimum-energy control law (8) with ζ given by (11) satisfies the monotonic convergence condition (14).*

Proof: We define g_i and m_i such that

$$m_i = \left[I_2,\, \frac{s_1 - s_2}{s_1 + s_2} E_1 p_i,\, \frac{1}{4s_1}(I_2 + E_2)p_i,\, \frac{1}{4s_2}(I_2 - E_2)p_i \right]$$

$$g_i = \left[I_2,\, \frac{1}{s_1 - s_2} p_i^T E_1,\, {p_i}^{\mathrm{T}}(I_2 + E_2),\, {p_i}^{\mathrm{T}}(I_2 - E_2) \right]^{\mathrm{T}}.$$

Substituting (8) and (11) into the left hand side of (14) gives the quadratic form:

$$\tilde{\boldsymbol{x}}^{\mathrm{T}}\, K\, \Gamma\, [g_i\, m_i]\, \Gamma^{-1}\, K\, \tilde{\boldsymbol{x}}.$$

The 5×5 matrix $[g_i\, m_i]$, although asymmetric, can be shown to be positive semi-definite with the two non-zero eigenvalues to be given by:

$$\lambda_1 = 1 + \frac{\|p_i\|^2}{s_1 + s_2} \quad \text{and} \quad \lambda_2 = 1 + \frac{p_{i,x}^2}{s_1} + \frac{p_{i,y}^2}{s_2}.$$

Since K is chosen to be positive definite the inequality (14) is satisfied. ■

B. A safe minimum-energy control law

In this subsection we derive a decentralized control law that selects a control input as close as possible to the minimum-energy controls while satisfying the monotonic convergence inequality and the safety constraints.

Proposition 2. *Equation (15) is a decentralized control law that selects a unique control input that has the smallest energy instantaneously while satisfying the monotonic convergence inequality and the safety constraints.*

$$\nu_i = \underset{\hat{\nu}_i \in U}{\arg\min} \|\nu_i^\star - \hat{\nu}_i\|^2, \quad s.t. \quad (10, 14) \qquad (15)$$

Proof: The constraints in (10, 14) provide the safety guarantees and the monotonic convergence condition. The function being minimized is the discrepancy from the minimum-energy input. Since the inequality constraints are linear in ν_i and the function being minimized is a positive-definite, quadratic function of ν_i, (15) is a convex, quadratic program with a unique solution. Further, since each robot only relies on its own state and knowledge of the error in the abstract state, it is a decentralized control law. ■

Convergence properties of (15)

To investigate the global convergence properties, we introduce the Lyapunov function

$$V(q) = \frac{1}{2}\tilde{\boldsymbol{x}}^{\mathrm{T}}\tilde{\boldsymbol{x}}.$$

Since the solution of (15) must satisfy the inequality (14), we know that $\tilde{\boldsymbol{x}}^{\mathrm{T}} K \dot{\boldsymbol{x}} \geq 0$. If K is chosen to be diagonal with positive entries, this condition also implies $\tilde{\boldsymbol{x}}^{\mathrm{T}}\dot{\boldsymbol{x}} \geq 0$. In other words,

$$\dot{V}(q) = -\tilde{\boldsymbol{x}}^{\mathrm{T}}\dot{\boldsymbol{x}} \leq 0.$$

From [10], we know that q is bounded given that $\boldsymbol{x}$ is bounded and that $V(q) \to \infty$ as $\|q\| \to \infty$. Further, $V(q)$ is globally uniformly asymptotically stable. Therefore, from LaSalle's invariance principle, we know that the abstract state will converge to the largest invariant set given by $\tilde{\boldsymbol{x}}^{\mathrm{T}}\dot{\boldsymbol{x}} = 0$. From (2), we know that $\dot{\boldsymbol{x}} = 0$ only when $\nu = 0$. Thus the invariant set is characterized by the set of conditions that lead to the system of inequalities given by (10, 14) to have $\nu = 0$ as the only solution.

Proposition 3. *For any desired change in the abstract state $\tilde{\boldsymbol{x}}$, subject to the condition $\tilde{\boldsymbol{x}}_4 \geq 0$, $\tilde{\boldsymbol{x}}_5 \geq 0$, (i. e., a condition where the size of the shape of the formation is not decreasing), there is a non-zero solution to the inequalities (10, 14).*

Proof: Consider the solution given by the minimum-energy control law (8). In component form,

$$\nu_i^\star = \begin{bmatrix} \xi_1 + \frac{s_1 - s_2}{s_1 + s_2} y_i \xi_3 + \frac{x_i}{4s_1}\sigma_1 \\ \xi_2 + \frac{s_1 - s_2}{s_1 + s_2} x_i \xi_3 + \frac{y_i}{4s_2}\sigma_2 \end{bmatrix}.$$

It is easy to see that this satisfies the collision constraints (10) for every pair of robots $(i,\, j)$:

$$[(x_i - x_j)\ (y_i - y_j)] \begin{bmatrix} \nu_{i,x}^\star - \nu_{j,x}^\star \\ \nu_{i,y}^\star - \nu_{j,y}^\star \end{bmatrix} \geq 0.$$

As shown earlier in Proposition 1, (8) also satisfies the monotonic convergence inequality. ■

Remark 1. *It is clear from the above proof that there are no guarantees when the shape in the abstract state is shrinking in area. If $\tilde{\boldsymbol{x}}_4 < 0$ or $\tilde{\boldsymbol{x}}_5 < 0$, there may not be a non-zero velocity vector ν that satisfies the inequalities (10, 14). It is only in this condition that the system will reach an equilibrium away from the desired abstract state.*

We next discuss the second contribution of this paper, where we propose an energy metric for motion planning of a deformable ellipse. Such a metric permits the computation of optimal motion plans in complex environments.

V. Motion Planning in the Abstract Space

The abstract representation of the team of robots permits the planning of motions that only require consideration of an abstract state space of fixed dimension, rather than one that scales with the number of robots. In this section we consider the problem of generating reference trajectories in the abstract space.

We start by defining a metric on M. On $SE(2)$, we can define a Riemannian metric as a bi-linear form derived by an inner product on $se(2)$. Given two twists $\{\xi_1,\, \xi_2\} \in se(2)$, we can define [16]:

$$< \xi_1,\, \xi_2 >= \xi_1^T\, W\, \xi_2,$$

where W is a positive definite matrix. At an arbitrary element $g \in SE(2)$, the inner product between two velocities or tangent vectors $\dot{g}_1$, $\dot{g}_2$ is obtained by left translation:

$$< \dot{g}_1,\, \dot{g}_2 >_g = < g^{-1}\dot{g}_1,\, g^{-1}\dot{g}_2 >_e,$$

where $g^{-1}\dot{g}_i$ are tangent vectors at the identity element e (the 3×3 identity homogeneous transformation) and therefore lie in $se(2)$. A metric defined in this way is a left-invariant Riemannian metric. Following [17], we can use the inertia tensor of a rigid body and its kinetic energy to define W_g:

$$W_g = \begin{bmatrix} mI_2 & 0 \\ 0 & \mathcal{I}_{11} + \mathcal{I}_{22} \end{bmatrix},$$

in the body-fixed coordinate system $\mathcal{B}$.

The above treatment was for a rigid shape. However, since $M = G \times S$ is a product space, we treat the shape space independently. We assume a constant metric $W_s = \alpha I_2$ to model the "cost" in changing the shape. Thus the rate of change of the abstract shape in $\mathcal{B}$ given by ζ has the norm:

$$\|\zeta\| = \frac{1}{2}\zeta^T \begin{bmatrix} W_g & 0 \\ 0 & W_s \end{bmatrix} \zeta, \tag{16}$$

which is well-defined everywhere on M.

Realistically one must also model the potential energy associated with deforming the shape. The simplest approach to creating an abstract model for potential energy storage is to think of the expansion or contraction as a reversible, adiabatic process in which no energy is lost. Compression results in an increase of internal energy which can then be recovered during expansion. It is well known that in such processes the pressure p and the volume v are related by the ratio of specific heats γ by the equation:

$$pv^\gamma = constant$$

and the work done to effect a change in volume from v_1 to v_2, and therefore an increase in internal energy, is given by:

$$\Delta V = k\left(\frac{1}{v_2^{\gamma-1}} - \frac{1}{v_1^{\gamma-1}}\right),$$

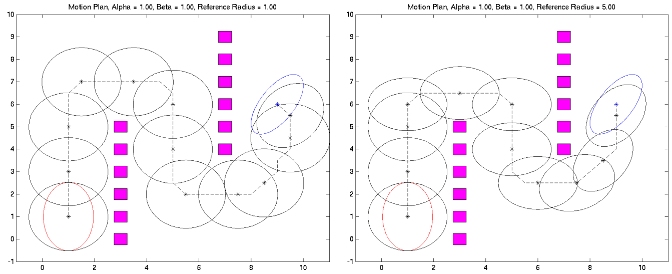
where k is a constant. In the plane, we can use the area of the ellipse (with a unit depth) instead of volume, which we know to be $\pi\sqrt{s_1 s_2}$. We define a reference shape $s^0(N)$ as a circular shape for N robots with zero potential energy. It is intuitively clear that the radius of a zero-energy circular shape, $r_0(N)$, must increase with N. In this paper, we take $r_0(N) = \frac{N\epsilon}{2}$. The potential energy associated with any shape $s \in S$ is given by:

$$V(s) = \beta\left(\frac{1}{(s_1 s_2)^{\frac{\gamma-1}{2}}} - \frac{1}{(r_0^2(N))^{\gamma-1}}\right), \tag{17}$$

where β is a constant. Thus, the total energy associated with any motion at any configuration is given by:

$$E(g, s, \zeta) = \frac{1}{2}\zeta^T \begin{bmatrix} W_g & 0 \\ 0 & W_s \end{bmatrix} \zeta + V(s). \tag{18}$$

Equation (18) gives us a principled approach to determine the cost of changes in configuration using a kinetic and potential energy with two constants α and β. It also allows us to formulate trajectory generation and motion planning problems as problems of finding geodesics.



(a)

(b)

Fig. 2. Two motion plans depict the effects of changing the radius of zero-energy, r_0, from $r_0 = 1$ (Fig. 2(a)) to $r_0 = 5$ (Fig. 2(b)). Although $\alpha = \beta = 1$ in both cases and the initial (red) and final (blue) abstract states are consistent, two different optimal motion plans are computed (using a Bellman-Ford search in a discretized abstract space). Obstacles are shown in magenta.

It is possible to split the space and derive the trajectories for $\mu \in \mathbb{R}^2$ and $(\theta, s_1, s_2) \in \mathbb{R}^2 \times SO(2)$ separately in open environments because of the product structure of the metric (18). Instead, we propose the design of motion plans using a discretization of the abstract space with constraints defined by obstacles and admissible abstract states. Many discrete optimal planning algorithms exist that permit the application of the energy metric defined by (18) to solve the for the minimum-energy path through open or cluttered environments [18].

VI. Splitting and Merging of Groups of Robots

When an ensemble of robots are forced to squeeze through constrained spaces, the only allowable shapes have small areas and correspond to large values of internal energy $V(s)$ as seen from (17). Therefore we define a threshold so that when $V(s) > V_{max}$ the group splits into two subgroups with shapes identical to the original shape but each with half the number of robots. This allows the group to reset its energy to a lower level by reducing the number of constraints, while remaining in the same configuration. Alternatively, a supervisory agent can decide when to divide the robots into subgroups. The formulation of Sect. IV can be applied to multiple subgroups with the new abstraction manifold $M = M_1 \times M_2$, where $M_i = G_i \times S_i$. The only additional mechanism that is required is a protocol for each robot to determine to which subgroup it belongs. Clearly the motion plan for each subgroup, $\boldsymbol{x}_i^{des}(t)$, can be generated and this desired trajectory can be broadcast to the group. With the feedback of the abstract state, $\boldsymbol{x}_i$, each robot can compute its own controls with the knowledge of the specific subgroup i to which it belongs. We propose two different event-triggered techniques for splitting and merging teams of robots in a decentralized and distributed manner.

A. Market-based auctioning method

Our first approach is based on the market-based auctioning method proposed in [19] which is guaranteed to converge in polynomial time but requires communication between the robots. It allows a team of robots to divide into subgroups by defining an auction determined by the desired abstract subgroup states $\{\boldsymbol{x}_1^{des}, \ldots, \boldsymbol{x}_k^{des}\}$, and the maximum number of agents allowed in each subgroup $\{n_1, \ldots, n_k\}$, where k

is the number of desired subgroups. The end result of the auction is that all agents are associated with a subgroup and distributed in agreement with the maximum number of agents in each group. To ensure correctness, the algorithm requires that $N = \sum_{i=1}^{k} n_i$.

B. Stochastic policy for splitting

As the number of robots grows, it is beneficial to use a mean-field model to model the distribution of robots and develop stochastic switching rules that guarantee the desired ensemble properties [20]. As $N \to \infty$, the ensemble properties of the group of robots using the stochastic switching rules converge to the desired properties. In other words, if each robot uses a probability distribution to select one subgroup versus the other, the ensemble properties of the group can be inferred from this probability distribution. From a practical standpoint, a split between k subgroups with $\{n_1, \ldots, n_k\}$ robots can be implemented approximately by each robot preferring group i with probability $p_i = \frac{n_i}{N}$.

C. Merging of groups

The merging of separate groups is trivial in the proposed framework using the controller (15). The redefinition of $\tilde{\boldsymbol{x}}$ to account for the desired merged abstract state results in a single group, while accounting for inter-agent collision avoidance.

VII. Simulation and Experimental Results

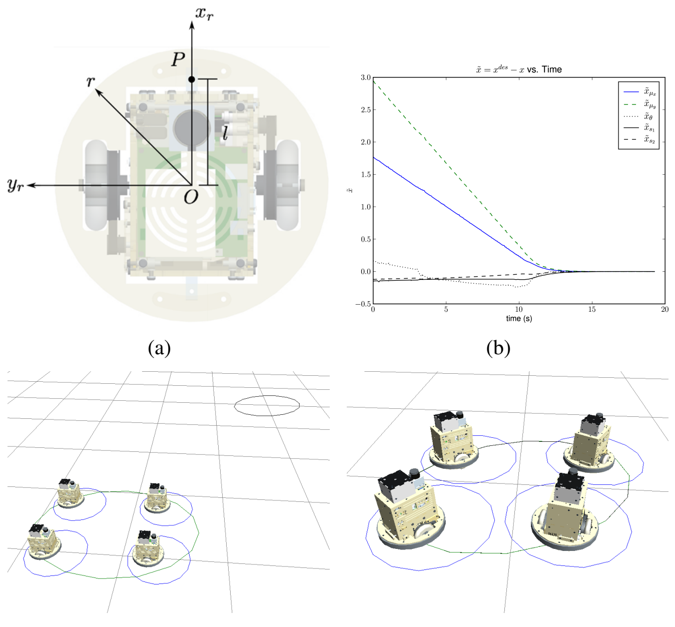
The remainder of the paper is dedicated to verifying the effectiveness of the control algorithm with collision avoidance presented in Sect. IV. We begin by discussing implementation details relevant to the analysis and experiment discussion that follows.

A. Implementation Details

The control algorithm was implemented in *C++* using the open-source robotics software *Player*, part of the *Player/Stage/Gazebo* project [21]. The *Player* server enables network communications between multiple robots. *Gazebo* is a three-dimensional simulation environment incorporating a dynamics engine and collision detection. *Player* also permits integration with *Gazebo*, allowing the same code base to be used in both simulation and experimentation on the real hardware.

The algorithm was tested in simulation via *Gazebo* and on an experimental infrastructure consisting of a team of small differential-drive robots, an indoor tracking system for ground-truth purposes, and a computer infrastructure to support wireless communication and data logging. Accurate models of the robots were created for use in simulation to emulate the real robots. Further, the asynchronous and distributed nature of the hardware was emulated by creating separate execution threads for each agent where all inter-agent communication was accomplished through the *Player* server.

Although robot dynamics play a significant role when considering inter-agent interactions, we are able to ignore these effects due to the fact that the robot platforms use stepper motors which permit "instantaneous" changes in velocity for sufficiently small magnitudes given the mass of the robot. We ensure during simulation and experimentation that the control velocities respect these thresholds.



(a) (b)

(c) (d)

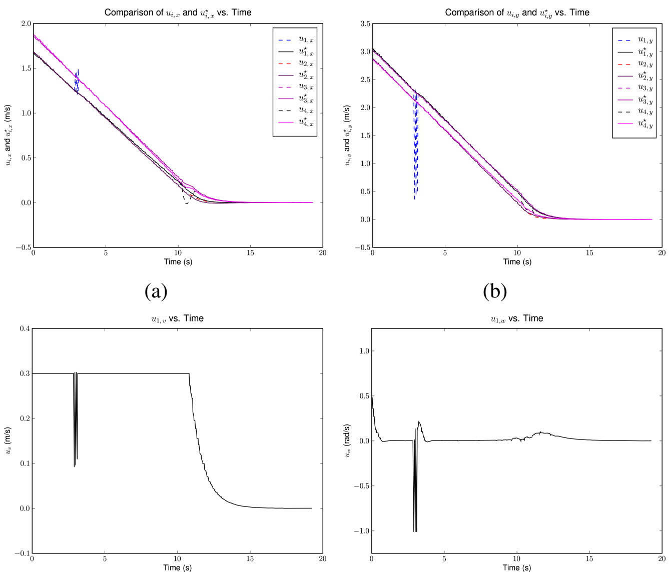
Fig. 3. A top view of the robot model used in both simulation and experimentation showing the body-fixed coordinate system. P is a reference point on the robot whose position is regulated by the vector fields (Fig 3(a)). The convergence of a team of four robots in simulation to $\boldsymbol{x}^{des} = x$ (Fig. 3(b)). The team of robots controlling to the desired abstract state $\boldsymbol{x}^{des} = \{2, 2, 0.5, 0.2, 0.1\}$ (Figs. 3(c)–3(d)). The abstract states $\boldsymbol{x}$ and $\boldsymbol{x}^{des}$ are shown in green and black, respectively. The ϵ radius safety region is shown in blue, which for the differential drive robots is defined with respect to the feedback linearization point.

In both simulation and experimentation, the abstract state $\boldsymbol{x}$ (assessed using simulation data or the tracking system) and the desired abstract state $\boldsymbol{x}^{des}$ were broadcast to the robots. As the control law updates, each robot broadcasts its current pose and velocity while listening for the pose and velocity of its neighbors. The optimization defined by (15) is solved using the quadratic program routines provided by the open-source *Computational Geometry Algorithms Library* [22].

We consider a simple model of a point robot with coordinates (x, y) in the world coordinate system. On the differential-drive robot in Fig. 3(a), these are the coordinates of a reference point P on the robot which is offset from the axle by a distance l. Its velocities in the inertial frame and moving frame $\mathcal{B}$ are u_i and ν_i, respectively. The velocity of the reference point P can be converted into linear and angular velocities for the robot through the equations below:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\frac{\sin\theta}{l} & \frac{\cos\theta}{l} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}.$$

It is well-known that if a robot's reference point is at the point P, and if r is the radius of a circle circumscribing the robot, all points on the robot lie within a circle of radius $r + l$ centered at P. In other words, if the reference point tracks a trajectory



(a) (b)

(c) (d)

Fig. 4. The optimal control, $u_i^\star$, and u_i from (15) defined in the robot's local frame (Figs. 4(a)–4(b)). In general, the optimal control law is the solution to (15). However, during inter-agent interactions the resulting control varies from the optimal solution. The linear and angular velocities resulting from feedback linearization (Figs. 4(c)–4(d)).

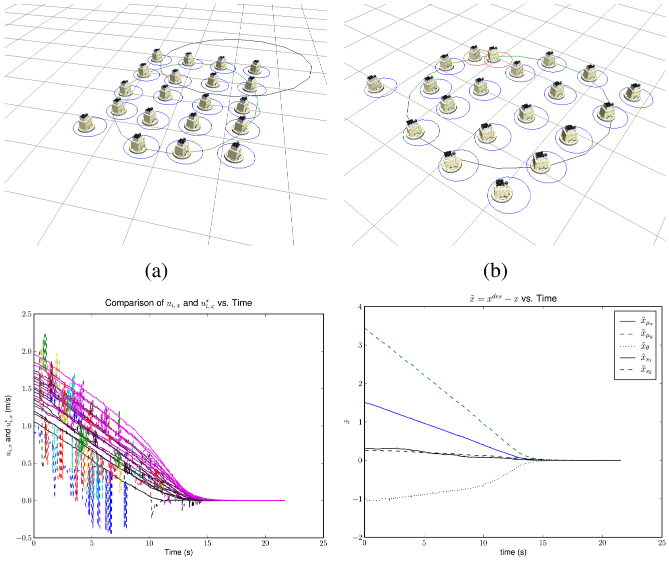
$(x_d(t), y_d(t))$, the physical confines of the robot are within a circle of radius $r + l$ of this trajectory. This allows us to use a geometric abstraction of a circular robot at P with a radius $\rho = r + l$. By ensuring that the reference points of adjacent robots are at least ϵ away as in (9), we guarantee that there are no collisions for the real robot.

B. Simulation

In this section we present two representative trials of the control law for collision avoidance: a basic system with limited interactions, and a large team of robots with numerous interactions. In both cases, we let $K = \text{diag}(1, 1, 0.8, 0.8, 0.8)$, $r = 0.15\,\text{m}$, $\epsilon_s = 0.1\,\text{m}$, $l = 0.1\,\text{m}$, and the maximum linear and angular velocities are $0.3\,\text{m/s}$ and $1\,\text{rad/s}$, respectively.

1) Controlling four robots with limited interactions: A small team of four stationary differential-drive ground robots were sent a desired abstract state (see Fig. 3). The convergence of the controller to the desired abstract state is shown in Fig. 3(b). A comparison of the resulting optimal control law (5) and the collision avoidance control law (15) in the separate kinematic controller dimensions is depicted in Figs. 4(a)–4(b). Indeed, as stated in Proposition 1, in general the optimal control law, $u_i^\star$, is applied. However, during collisions, the control varies from the optimal control to account for the inter-agent interactions. The linear and angular velocities of the first agent resulting from feedback linearization and velocity saturation are shown in Figs. 4(c)–4(d). These interactions are apparent after feedback linearization.

2) Controlling a team of twenty robots: The team of twenty robots depicted in Fig. 5 demonstrates the effectiveness of the controller in large groups to ensure collision avoidance while controlling to a desired shape. From Fig. 5(c) it is clear that numerous interactions occur between the agents and yet the system converges to the desired shape. In general, agents will control to a separation distance of ϵ since each agent is individually computing the collision constraints (10). However, it is possible that robots remain within the ϵ safety region of other robots at the time of convergence (see Fig. 5(b)) but do not collide since the solution to (15) when $\tilde{\boldsymbol{x}} = 0$ is $u_i = 0$.



(a) (b)

(c) (d)

Fig. 5. A team of twenty robots control to $\boldsymbol{x}^{des} = \{3, 3, 0.5, 1.5, 1\}$. A red ϵ safety region indicates a state where $\delta_{ij} < \epsilon$, but in this case $\dot{\boldsymbol{x}} = 0$ (Fig. 5(b)). A comparison of $u_{i,x}$ from (15) and the optimal control, $u^\star_{i,x}$ defined in the robot's local frame (Fig. 5(c)). A comparison of $u_{i,y}$ to $u^\star_{i,y}$ results in similar control variations. Although there are numerous interactions between the agents, as depicted by the many variations of $u_{i,x}$ from $u^\star_{i,x}$, the system converges to the desired shape (Fig. 5(d)).

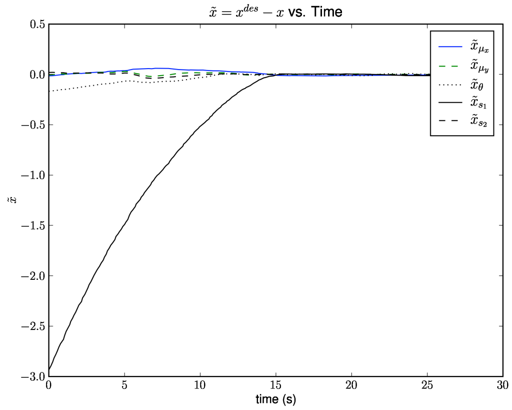


Fig. 6. The convergence of the team of seven robots in experimentation to $\boldsymbol{x}^{des}$. The trial represents a merging scenario where the robots were distributed in distinct groups separated by several meters.

C. Experiments

The control algorithm was verified on a team of seven nonholonomic robots. A trial run is depicted in Fig. 7. The algorithm was tested with several trials over a variety of desired states and scenarios with performance consistent with the convergence shown in Fig. 6. The noise associated with

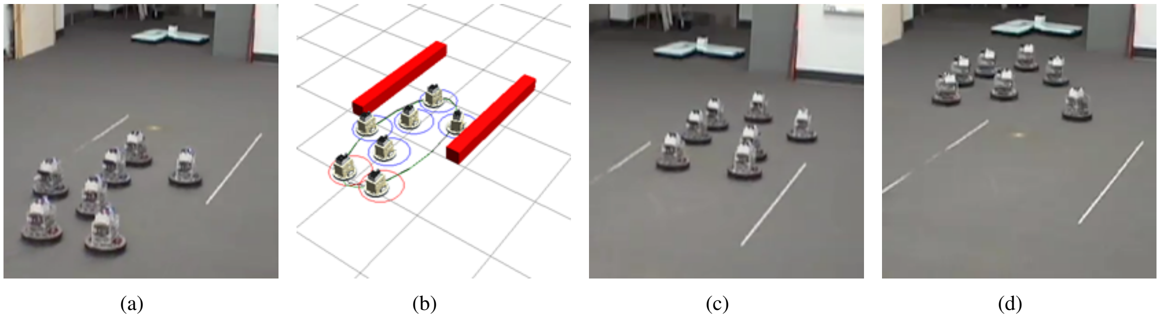


Fig. 7. A team of seven robots control through a series of abstract states in a corridor passing maneuver. A visualization of the experiment (following the representation defined in Fig. 3) corresponding to the system state in Fig. 7(a) depicts the current and desired abstract states, robot states, and corridor location (Fig. 7(b)).

the estimation of the abstract state via the localization system is observable in the smoothness of $\tilde{x}$ in Fig. 6.

VIII. Conclusion and Future Work

We presented solutions to the problem of planning and controlling the position, orientation, and shape of a formation of a team of robots. In contrast to most previous work, we model the physical shape of the robot and consider controllers that are guaranteed to avoid collisions between the robots. We also derive software abstractions that lend themselves to implementation on real platforms and to experimentation. In addition, we present a metric for the planning of deformable shapes and trajectories of the ensemble and the development of effective coordination strategies to split the team into subgroups and to merge subgroups. We view these problems and their solutions to be building blocks that can enable a robot team to navigate an environment adapting to the constraints imposed by obstacles in the environment. Simulation and experimental results demonstrate the effectiveness of the control algorithm when applied to nonholonomic robots.

As a direction of future research, we are interested in pursuing time-parameterized trajectories in the abstract space. We have also assumed the existence of a global observer in past work and are currently considering distributed estimation algorithms to relax this requirement.

Acknowledgment

We gratefully acknowledge partial support from NSF grants IIS-0427313, IIP-0742304, and IIS-0413138, ARO grant W911NF-05-1-0219, and ONR grant N00014-07-1-0829.

References

[1] G. S. Sukhatme, A. Dhariwal, B. Zhang, C. Oberg, B. Stauffer, and D. A. Caron, "Design and development of a wireless robotic networked aquatic microbial observing system," *Environmental Engineering Science*, vol. 24, no. 2, pp. 205–215, Mar. 2007.

[2] P. E. Rybski, S. A. Stoeter, M. D. Erickson, M. Gini, D. F. Hougen, and N. Papanikolopoulos, "A team of robotic agents for surveillance," in *Proc. of Int. Conf. on Autonomous Agents*, Barcelona, Spain, June 2000, pp. 9–16.

[3] V. Kumar, D. Rus, and S. Singh, "Robot and sensor networks for first responders," *IEEE Pervasive Computing*, vol. 3, no. 4, pp. 24–33, Oct. 2004.

[4] T. Eren, P. N. Belhumeur, and A. S. Morse, "Closing ranks in vehicle formations based on rigidity," in *Proc. of the IEEE Conf. on Decision and Control*, vol. 3, Las Vegas, NV, Dec. 2002, pp. 2959–2964.

[5] R. Olfati-Saber and R. M. Murray, "Distributed cooperative control of multiple vehicle formations using structural potential functions," in *Proc. of the IFAC World Congress*, Barcelona, Spain, July 2002.

[6] J. P. Desai, J. Ostrowski, and V. Kumar, "Controlling formations of multiple mobile robots," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, vol. 4, Leuven, Belgium, May 1998, pp. 2864–2869.

[7] H. Tanner, G. J. Pappas, and V. Kumar, "Input-to-state stability on formation graphs," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Las Vegas, NV, Dec. 2002, pp. 2439–2444.

[8] M. Egerstedt and X. Hu, "Formation constrained multi-agent control," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 947–951, Dec. 2001.

[9] P. Ogren, E. Fiorelli, and N. Leonard, "Formations with a mission: stable coordination of vehicle group maneuvers," in *Proc. of Int. Symposium on Mathematical Theory Networks and Systems*, Notre Dame, IN, Aug. 2002.

[10] C. Belta and V. Kumar, "Abstraction and control for groups of robots," *IEEE Transactions on Robotics*, vol. 20, no. 5, pp. 865–875, Oct. 2004.

[11] N. Michael, C. Belta, and V. Kumar, "Controlling three dimensional swarms of robots," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Orlando, FL, May 2006, pp. 964–969.

[12] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, Apr. 2004.

[13] M. Schwager, J. McLurkin, and D. Rus, "Distributed coverage control with sensory feedback for networked robots," in *Robotics: Science and Systems*, Philadelphia, PA, Aug. 2006.

[14] R. A. Freeman, P. Yang, and K. M. Lynch, "Distributed estimation and control of swarm formation statistics," in *Proc. of the American Control Conf.*, Minneapolis, MN, June 2006, pp. 749–755.

[15] N. Michael, J. Fink, and V. Kumar, "Controlling a team of ground robots via an aerial robot," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, San Diego, CA, Oct. 2007, pp. 965–970.

[16] R. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Florida: CRC Press, 1993.

[17] M. Zefran, V. Kumar, and C. B. Croke, "On the generation of smooth three-dimensional rigid body motions," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 576–589, Aug. 1998.

[18] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006.

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

[20] A. Martinoli, K. Easton, and W. Agassounon, "Modeling swarm robotic systems: A case study in collaborative distributed manipulation," *The Int. Journal of Robotics Research*, vol. 23, no. 4-5, pp. 415–436, Apr. 2004.

[21] B. P. Gerkey, R. T. Vaughan, and A. Howard, "The player/stage project: Tools for multi-robot and distributed sensor systems," in *Proc. of the Int. Conf. on Advanced Robotics*, Coimbra, Portugal, June 2003, pp. 317–323.

[22] "CGAL, Computational Geometry Algorithms Library," http://www.cgal.org.