Control of Many Agents Using Few Instructions

Timothy Bretl
Department of Aerospace Engineering
University of Illinois at Urbana-Champaign
tbretl@uiuc.edu

Abstract— **This paper considers the problem of controlling a group of agents under the constraint that every agent must be given the same control input. This problem is relevant for the control of mobile micro-robots that all receive the same power and control signals through an underlying substrate. Despite this restriction, several examples in simulation demonstrate that it is possible to get a group of micro-robots to perform useful tasks. All of these tasks are derived by thinking about the relationships between robots, rather than about their individual states.**

I. INTRODUCTION

A growing number of applications require coordinated control of multiple agents. For example, applications that have received considerable interest from the robotics community include remote monitoring with mobile sensor networks, collision and congestion avoidance with automated air traffic control systems, and cooperative search and rescue with fleets of unmanned vehicles. Control architectures used in these applications vary widely—they might be deliberative or reactive, they might be computed online or offline, they might use global or local information, and they might be implemented using centralized or distributed processing. But regardless of the control architecture, we typically assume that each agent is capable either of acting independently or of following a distinct control input.

In this paper, we are interested in multi-agent systems in which agents are not capable of acting independently. In particular, we are interested in *multi-agent systems in which every agent must be given the same control input.*

We are motivated by recent work in the development of untethered, steerable micro-robots [8]. These robots consist of an untethered scratch drive actuator (used for forward locomotion) and a cantilevered steering arm (used to turn, through frictional contact with the substrate). They are globally controllable, their motion resembling that of unicycles and other car-like vehicles. However, these robots do not have onboard sensing or processing, do not receive commands on distinct communication channels, and do not act independently. Instead, every robot receives the same power and control signal through an underlying electrical grid.

Some work by [8] has focused on removing this restriction. Electromechanical hysteresis is already used to independently send "forward" and "turn" commands by snapping the cantilever up or down. By adding more cantilever arms and making the hysteresis more sophisticated, it is possible to create distinct robots that each respond to some sequences of commands but not to others. The limits of this technique,

particularly in terms of how many robots can be controlled simultaneously, have yet to be defined.

In this paper we take a different approach, showing that it is possible to get a group of micro-robots to perform useful tasks even if every robot receives the same control signal.

We begin by considering point robots with simple kinematics (and without nonholonomic constraints). In Section II we show that it is impossible to move two point robots to arbitrary locations if the same control input is sent to both. In Section III we show that a small change in how the control input is interpreted (with respect to local coordinate frames rather than a global coordinate frame) makes it possible to change the relative location between two point robots, even if we still cannot move them to arbitrary locations. In particular, we show that the way in which we can change the relationship between the two robots has distinct structure—for example, we can direct them to meet at the same location, but this location is both unique and fixed. In Section IV, we apply these ideas to enable not just two, but a much larger number of point robots to perform useful tasks. In Section V, we return to the micro-robot example and consider a group of unicycles that must roll without slipping. Finally, in Section VI, we discuss related work and in particular point out several other applications of future interest (micro-robots are only one example).

It is important to understand that we are not advocating any particular control architecture or computational approach in this paper. Indeed, not only do we focus entirely on open-loop control (the steering or motion-planning problem), but we also neglect many phenomena that might influence control design, such as process noise, sensor uncertainty, and the presence of obstacles in the environment. These phenomena may even present alternative ways of differentiating state in response to a common input. However, it is difficult to determine how to take advantage of these phenomena in general—instead, we choose to focus on phenomena that exhibit distinct structure.

In particular, the key message here is that by thinking about *relationships* between robots (rather than about their individual states), we are led to consider tasks that are not only possible, but that admit algorithmic solutions with provable levels of performance. We frame several interesting motion planning problems as a result—for some, we present a method of solution; for others, we leave open questions for future work.

II. UNIFORM CONTROL INPUT, UNIFORM RESPONSE

Consider two agents moving in a plane with no obstacles. Imagine that we can direct their motion by telling them to
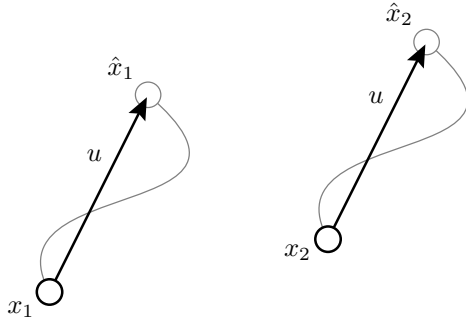
Fig. 1. Two agents are instructed to follow the same path, specified with respect to a global coordinate frame. The total displacement is represented by the vector u. The relative location between agents remains fixed.

move a certain distance north, south, east, or west (or in some arbitrary direction relative to these coordinates). We will assume that they follow our instructions exactly, without error. However, we will also assume that the same set of instructions is given to both agents—we cannot tell one to move east and the other west, for example. Under this restriction, is it possible to simultaneously direct both agents to arbitrary locations on the field? Alternatively, is it possible to direct both agents to meet at the same location?

We can easily show that the answer to both questions is *no*. In particular, let the position of each agent be $x_1, x_2 \in \mathbb{R}^2$. Let the vector $u \in \mathbb{R}^2$ represent our set of instructions, telling both agents to move a distance $\|u\|$ in a direction $u/\|u\|$ that is relative to a fixed coordinate frame. So the position of agent i after following our instructions is $x_i + u$. First, given arbitrary goal locations $\hat{x}_1$ and $\hat{x}_2$ in $\mathbb{R}^2$, we want to know if some u exists that satisfies

$$\hat{x}_1 = x_1 + u$$
$$\hat{x}_2 = x_2 + u.$$

Such a u exists only if $\hat{x}_1 - x_1 = \hat{x}_2 - x_2$. So the goal locations may not be chosen arbitrarily—they must be in the same place relative to both agents. Second, we want to know if some u exists that satisfies

$$x_1 + u = x_2 + u.$$

Such a u exists only if $x_1 = x_2$. So the two agents can be directed to meet at the same location only if they begin at the same location.

In short, it seems we are considering an example that is not interesting. The two agents are moving in formation—the relative location between them is fixed (see Fig. 1).

III. Uniform control input, selective response

Consider the same two agents moving in a plane with no obstacles. Again assume that we can direct their motion by telling them to move a certain distance in a certain direction, that they follow our instructions exactly, and that the same set of instructions is given to both agents. But now, rather than specify the direction of motion with respect to a global coordinate frame (north, south, east, west), imagine that we
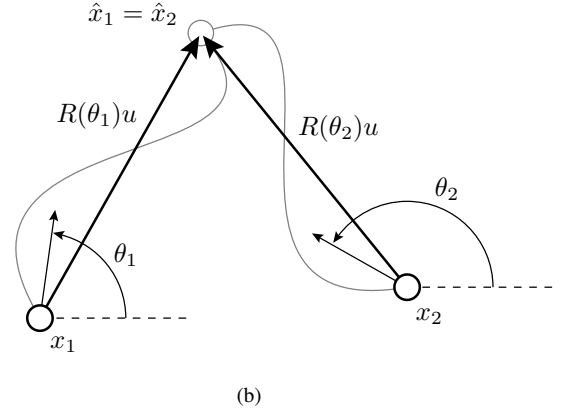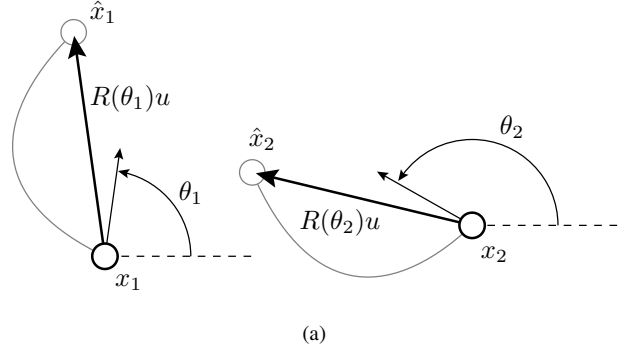

(a)

(b)

Fig. 2. Two agents are instructed to follow the same path, specified with respect to local coordinate frames. The total displacement is u in local coordinates, which corresponds to $R(\theta_1)u$ and $R(\theta_2)u$ in global coordinates. (a) The relative location between agents does not remain fixed. (b) In fact, the agents can be directed to meet at the same location.

can specify it with respect to a local coordinate frame (forward, back, right, left). So if we tell both agents to move forward, they may move in different directions, depending on which way they are initially facing. We assume that neither agent changes their orientation as they move, so that their local coordinate frames are fixed.

It is still impossible to simultaneously direct both agents to arbitrary locations. Let the position of each agent be $x_1, x_2 \in \mathbb{R}^2$. Let the initial (and fixed) orientation of each agent be $\theta_1, \theta_2 \in (-\pi, \pi]$. Let the vector $u \in \mathbb{R}^2$ represent our set of instructions, telling both agents to move a distance $\|u\|$ in a direction $u/\|u\|$ that is relative to their local coordinate frame. Define the rotation matrix

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

So the position of agent i after following our instructions is $x_i + R(\theta_i)u$. Given arbitrary goal locations $\hat{x}_1$ and $\hat{x}_2$ in $\mathbb{R}^2$, we want to know if some u exists that satisfies

$$\hat{x}_1 = x_1 + R(\theta_1)u$$
$$\hat{x}_2 = x_2 + R(\theta_2)u.$$

Such a u exists only if $R(\theta_1)^T(\hat{x}_1 - x_1) = R(\theta_2)^T(\hat{x}_2 - x_2)$. So again, the goal locations may not be chosen arbitrarily—they must be in the same place relative to both agents with respect to their local coordinate systems.

However, it is now possible to direct both agents to meet at the same location (as long as we do not care where that location is). We want to know if some u exists that satisfies

$$x_1 + R(\theta_1)u = x_2 + R(\theta_2)u.$$

Such a u is given by

$$u = -\left(R(\theta_2) - R(\theta_1)\right)^{-1}\left(x_2 - x_1\right).$$

The inverse $\left(R(\theta_2) - R(\theta_1)\right)^{-1}$ exists if and only if $\theta_1 \neq \theta_2$. So if each agent has a different initial orientation (or if both agents begin at the same location, the trivial case), then both agents can be directed to meet at the same location. Of course, this location may not be chosen arbitrarily. Instead, it is unique and is fixed by x_1, x_2, θ_1, and θ_2. We denote it by

$$u_{12} = -\left(R(\theta_2) - R(\theta_1)\right)^{-1}\left(x_2 - x_1\right)$$

in local coordinates and by

$$\hat{x}_{12} = x_1 + R(\theta_1)u_{12}$$
$$= x_2 + R(\theta_2)u_{12}$$

in global coordinates. But although u_{12} is fixed, the path taken to reach u_{12} is not. For example, consider a set of instructions specified by an arbitrary continuous curve $u \colon [0,1] \mapsto \mathbb{R}^2$ such that $u(0) = 0$. Then for all $t \in [0,1]$ such that $u(t) = u_{12}$, the two agents are at the same location, meeting at $\hat{x}_{12}$. Conversely, for all $t \in [0,1]$ such that $u(t) \neq u_{12}$, the two agents are at different locations.

We have made only a small change to our example from Section II: interpreting instructions in local coordinate frames rather than in a global coordinate frame. But now, even though we still give the same set of instructions to two agents, we can get something interesting to happen—we can direct them to meet at the same location (see Fig. 2). This result may seem surprising, even though it was easy to prove.

IV. CONTROL OF MANY AGENTS

In Sections II-III, we showed that it is possible to change the relative location between two agents (modeled as points in the plane) even if the same control input is sent to both. In particular, we showed that the way in which we can change the relationship between two agents has distinct structure—for example, we can direct them to meet at the same location, but this location is both unique and fixed. In this section we apply these ideas to enable a group of agents to perform useful tasks.

A. Sequential information passing with no propagation

Consider n agents moving in a plane with no obstacles. Again assume that we can direct their motion by telling them to move a certain distance in a certain direction, that they all follow our instructions exactly, and that the same set of instructions is given to everyone. Also, assume as in Section III

that we specify the direction of motion with respect to a local coordinate frame (forward, back, right, left).

Imagine that one agent has an important piece of information. We would like this agent to pass the information on to everyone else. We assume that the only way this agent can pass information to another agent is by meeting them at the same location. So we would like to compute a set of instructions that directs the agent with information to meet everyone else while traveling the minimum distance possible. We call this task *sequential information passing with no propagation*.

Note that we are using the term "information" as an abstraction for something physical that is passed around. For example, one might imagine micro-robots propagating a marker (such as a colored dye or a chemical substance) or exchanging micro-scale construction materials. It is not assumed that the robots can communicate, or in fact do anything on their own other than follow a common control input.

Let the position and initial orientation of each agent be $x_i \in \mathbb{R}^2$ and $\theta_i \in (-\pi, \pi]$, respectively, for all $i \in \{1, \ldots, n\}$. We assume that $\theta_i \neq \theta_j$ for all $i, j \in \{1, \ldots, n\}$ such that $i \neq j$. Let the continuous curve $u \colon [0,1] \mapsto \mathbb{R}^2$ represent our set of instructions, followed with respect to each agent's local coordinate frame. Define the rotation matrix

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

So the position of agent i after following our instructions for time $t \in [0,1]$ is $x_i + R(\theta_i)u(t)$. Assume without loss of generality that agent 1 has the "important information" and must meet everyone else. Recall from Section III that agent 1 can only meet agent $i \neq 1$ at the fixed location

$$\hat{x}_i = x_1 + R(\theta_1)u_i$$
$$= x_i + R(\theta_i)u_i$$

where

$$u_i = -\left(R(\theta_i) - R(\theta_1)\right)^{-1}\left(x_i - x_1\right).$$

We also define the initial location by $\hat{x}_1 = x_1$ and $u_1 = 0$. So agent 1 must visit $\hat{x}_i$ for all $i \in \{1, \ldots, n\}$. Since the total distance traveled is equal to the length of the curve u, our task is to find u of minimum length that passes through u_i for all $i \in \{1, \ldots, n\}$. The optimal u is clearly a sequence of straight lines between meeting points. It can be specified by an ordering $i(1), \ldots, i(n)$ of the set $\{1, \ldots, n\}$ and found by solving the following problem:

$$\text{minimize} \quad \sum_{k=1}^{n-1} \|u_{i(k+1)} - u_{i(k)}\|$$

$$\text{subject to} \quad k \in \{i(1), \ldots, i(n)\} \text{ for all } k \in \{1, \ldots, n\}$$
$$i(k) \in \{1, \ldots, n\} \text{ for all } k \in \{1, \ldots, n\}$$
$$i(1) = 1$$

In fact, this problem can be recast as a traveling salesman problem. Define a graph by a vertex set $V = \{1, \ldots, n\}$ and an
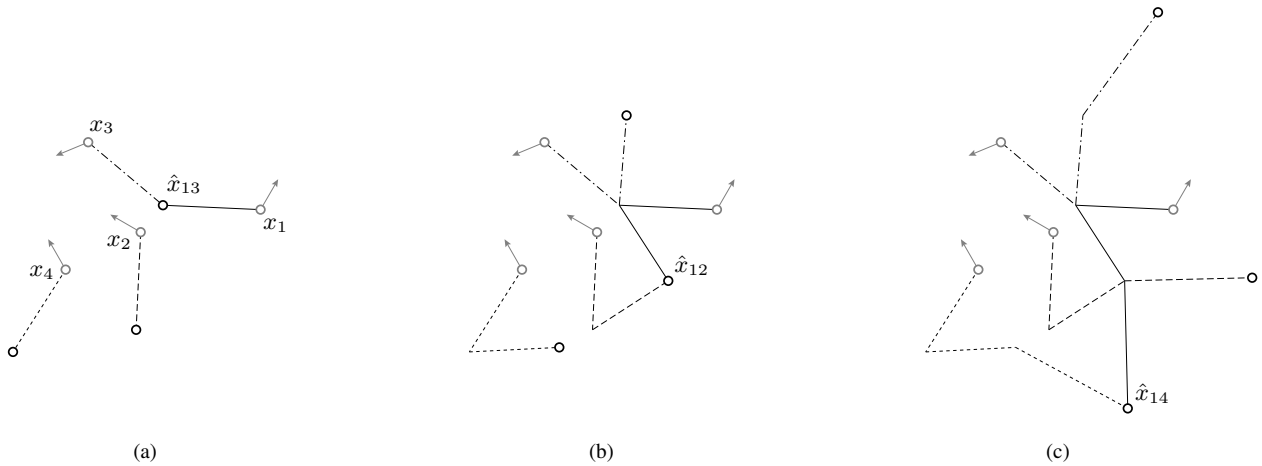
Fig. 3. *Information passing without propagation.* An optimal tour for one agent (beginning at x_1) to pass information to three other agents by meeting them each in turn, under the restriction that all agents must be instructed to follow the same path, specified with respect to local coordinate frames. In this tour, the first agent is directed sequentially to the meeting points $\hat{x}_{13}$, $\hat{x}_{12}$, and $\hat{x}_{14}$.
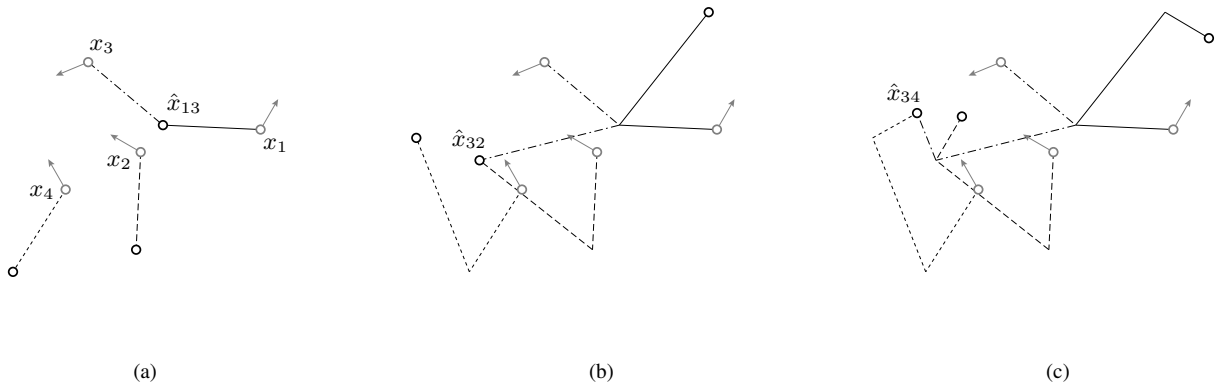

Fig. 4. *Information passing with propagation.* An optimal tour for the same example shown in Fig. 3, again under the restriction that all agents must be instructed to follow the same path, specified with respect to local coordinate frames. In this case, however, agents that have been given information are allowed to pass it along to other agents themselves. As a result, the agents are directed sequentially to different meeting points: $\hat{x}_{13}$, $\hat{x}_{32}$, and $\hat{x}_{34}$.

arc set $A = \{(i, j)$ for all $i, j \in V$ such that $i \neq j\}$. Define a cost matrix with elements

$$c_{ij} = \begin{cases} 0 & \text{if } j = 1 \\ \|u_i - u_j\| & \text{otherwise} \end{cases}$$

for all $i, j \in V$ such that $i \neq j$. Then the least-cost Hamiltonian circuit through the graph defined by V and A defines an optimal ordering $i(1), \ldots, i(n)$ of the set $\{1, \ldots, n\}$.

Figure 3 shows a simple example for $n = 4$ agents. Since each agent follows the same path (interpreted with respect to a local coordinate frame), it is impossible for them all to meet at the same location. So, the first agent meets each of the other three sequentially. The curve u consists of three straight lines: from $u_1 = 0$ to u_3, from u_3 to u_2, and from u_2 to u_4.

B. Sequential information passing with propagation

In the previous section we considered an information-passing task for a group of n agents, that required one agent

to meet each of the others in turn. But what if that first agent is not the only one that can pass information to others? Indeed, it may be natural to assume that once a second agent is given information, they too are able to pass it along. So in this section, we would like to compute a set of instructions that directs each agent to be met by someone who has already received the information, while traveling the minimum distance possible (where as usual we assume that the same set of instructions is given to everyone and interpreted with respect to local coordinate frames). We call this task *sequential information passing with propagation*.

Again we denote each agent's position by x_i and initial orientation by θ_i, and assume that $\theta_i \neq \theta_j$ for all $i \neq j$. The continuous curve $u \colon [0, 1] \mapsto \mathbb{R}^2$ represents our set of instructions, followed with respect to each agent's local coordinate frame as represented by a rotation matrix $R(\theta_i)$. So as before, the position of agent i after following our instructions for time $t \in [0, 1]$ is $x_i + R(\theta_i)u(t)$. Assume that

agent 1 is initially given the information to be passed on. Two agents i and $j \neq i$ can only meet at the fixed location

$$\hat{x}_{ij} = x_i + R(\theta_i)u_{ij}$$
$$= x_j + R(\theta_j)u_{ij}$$

where

$$u_{ij} = -\left(R(\theta_j) - R(\theta_i)\right)^{-1}\left(x_j - x_i\right).$$

We also define initial locations by $\hat{x}_{ii} = x_{ii}$ and $u_{ii} = 0$ for all $i \in \{1, \ldots, n\}$. Although $\hat{x}_{ij} = \hat{x}_{ji}$ and $u_{ij} = u_{ji}$, by convention we will write $\hat{x}_{ij}$ and u_{ij} if information is being passed from agent i to agent j, and $\hat{x}_{ji}$ and u_{ji} if information is being passed from agent j to agent i. So for all $j \in \{1, \ldots, n\}$, each agent j must visit and receive information at $\hat{x}_{ij}$ (equivalently, u must pass through u_{ij}) from some agent $i \in \{1, \ldots, n\}$ that has already received it.

Since the total distance traveled is equal to the length of u, the optimal path is a sequence of straight lines between meeting points. It can be specified by orderings $i(1), \ldots, i(n)$ and $j(1), \ldots, j(n)$ of the set $\{1, \ldots, n\}$ and found by solving the following problem:

minimize $\displaystyle\sum_{k=1}^{n-1}\|u_{i(k+1)j(k+1)} - u_{i(k)j(k)}\|$

subject to $\quad k \in \{j(1), \ldots, j(n)\}$ for all $k \in \{1, \ldots, n\}$
$\qquad\qquad i(k) \in \{j(1), \ldots, j(k-1)\}$ for all $k \in \{2, \ldots, n\}$
$\qquad\qquad i(k) \in \{1, \ldots, n\}$ for all $k \in \{1, \ldots, n\}$
$\qquad\qquad j(k) \in \{1, \ldots, n\}$ for all $k \in \{1, \ldots, n\}$
$\qquad\qquad i(1) = 1$
$\qquad\qquad j(1) = 1$

The first constraint ensures that each agent is met; the second constraint ensures that they are met by someone who already has information. The other constraints set the variable domains and the initial conditions.

It seems that, as in Section IV-A, this problem might be recast as a traveling salesman problem. It resembles the "generalized" traveling salesman problem in which cities are divided into groups and the salesman must visit at least one city in each group. However, the ordering constraints in our information passing problem make this formulation difficult. In any case, this problem is harder to solve than the one in Section IV-A, having $((n-1)!)^2$ possible orderings rather than $(n-1)!$. On the other hand, this problem has distinct structure that may be useful: for example, denoting $R_{ij} = R(\theta_i) - R(\theta_j)$, we see that

$$R_{12}u_{12} + R_{23}u_{23} + R_{34}u_{34} + R_{41}u_{41} = 0.$$

Figure 4 shows a simple example for $n = 4$ agents. In fact, this is the same example as in Fig. 3. However, here we take advantage of the ability to propagate information (rather than requiring one agent to meet all the others) to make the total distance traveled smaller. After receiving the information, the third agent (not the first) passes it along to the second and fourth agents. The curve u consists of three straight lines:
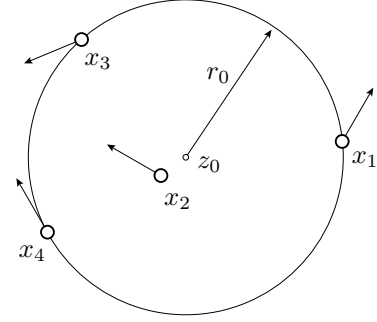
(a)
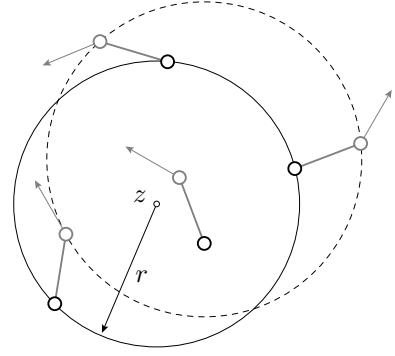
(b)

Fig. 5. Moving four agents inside a disc of minimum radius, for the same example shown in Figs. 3 and 4. As before, all agents follow the same path, interpreted with respect to local coordinate frames. In this case $r/r_0 \approx 0.9$.

from $u_{11} = 0$ to u_{13}, from u_{13} to u_{32}, and from u_{32} to u_{34}. This path is about 10% shorter than the one in Fig. 3.

C. Moving everyone close together

Again imagine that we are directing the motion of n agents in a plane. In previous sections, we have emphasized the fact that if everyone follows the same path with respect to their local coordinate frame, it is impossible for them all to meet at the same location at the same time (they can only meet pairwise). But what if we are interested in having everyone move close together rather than having everyone actually meet? In this section, we show that an optimal policy to accomplish this task can readily be computed.

As usual we denote each agent's position by x_i and initial orientation by θ_i. Let the vector $u \in \mathbb{R}^2$ represent our instructions, telling all agents to move a distance $\|u\|$ in a direction $u/\|u\|$. These instructions are followed with respect to each agent's local coordinate frame, as represented by a rotation matrix $R(\theta_i)$. So, the position of agent i after
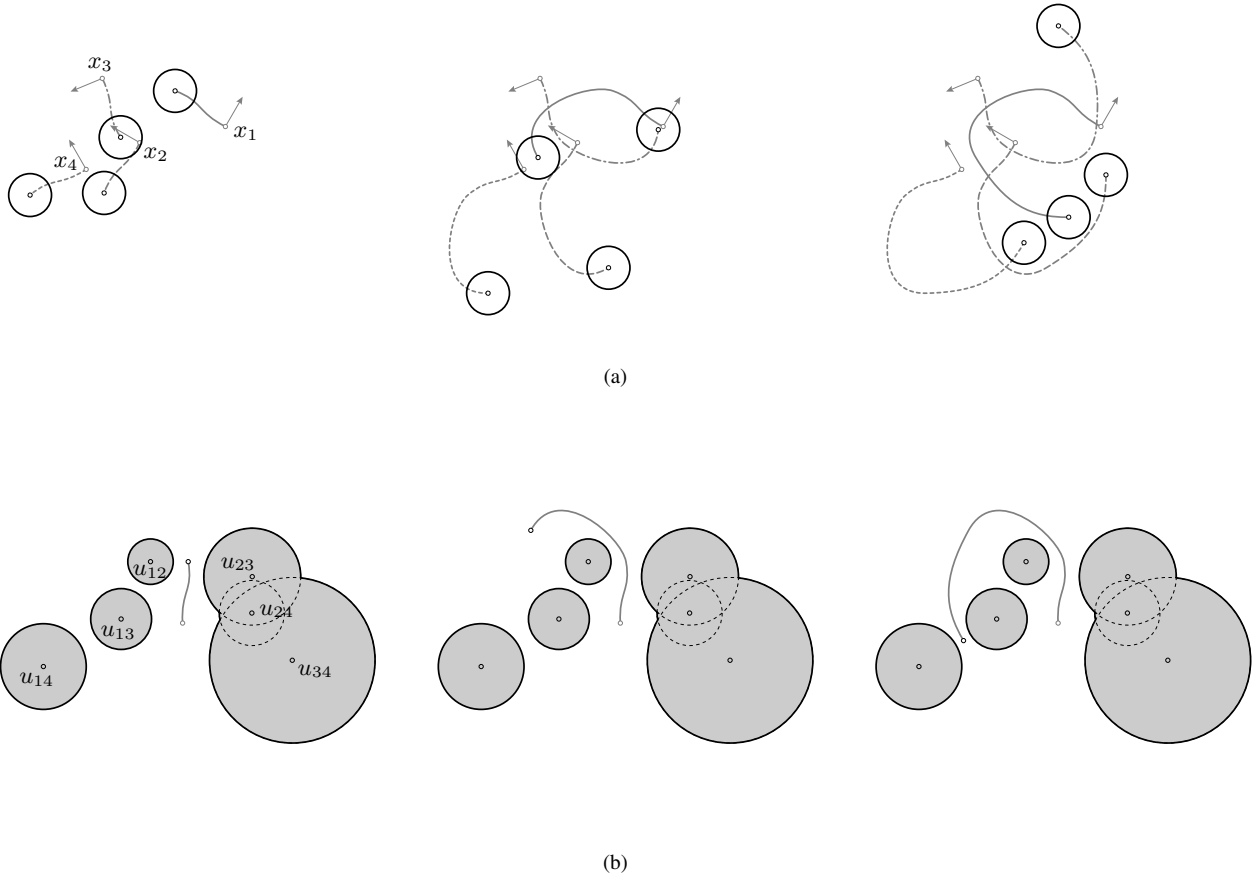
Fig. 6. Motion that avoids collision between four agents, for the same example shown in Figs. 3-5. Each agent has the same radius, and follows the same path (interpreted with respect to its local coordinate frame). (a) The paths $x_i + R(\theta_i)u(t)$ followed by each agent i in the workspace. (b) The curve $u(t)$ given as input. The shaded region is the set of all $u \in \mathbb{R}^2$ that would place two robots in collision.

following our instructions is $x_i + R(\theta_i)u$.

Subsequently, a disc with center $z \in \mathbb{R}^2$ and radius $r > 0$ contains the entire group of agents if $\|R(\theta_i)u + x_i - z\| \le r$ for all $i = 1, \ldots, n$. We measure the *distribution* of the group by the radius of the smallest such disc (for any z). We would like to find an input u that minimizes this distribution. We can do this by solving the following second-order cone program (SOCP):

minimize $\quad r$

subject to $\quad \|R(\theta_i)u + x_i - z\| \le r \quad$ for all $i = 1, \ldots, n$.

This SOCP is convex, and can be solved efficiently (by using an interior-point or primal-dual algorithm, for example [3]).

Figure 5 shows a simple example for $n = 4$ agents (the same agents as in Figs. 3 and 4). In this case, the optimal input u reduces the radius of the smallest disc containing all four agents by approximately 10%.

D. Avoiding collisions

Up to now, we have considered tasks that require agents to meet at the same location (in other words, to collide) or at least to move closer together. In this section, we consider the problem of *avoiding* collision. As before, assume that we are directing the motion of n agents in the plane by specifying a single path that each agent follows with respect to their local coordinate frame. But now, also assume that each agent has some physical shape—for example, we might model agents as discs of fixed radius. We would like to plan a collision-free path between any reachable start and goal configurations.

Denote each agent's position by x_i and initial orientation by θ_i. Assume that $\theta_i \ne \theta_j$ for all $i \ne j$. The continuous curve $u \colon [0, 1] \mapsto \mathbb{R}^2$ represents our set of instructions, followed with respect to each agent's local coordinate frame as given by a rotation matrix $R(\theta_i)$. So, the position of agent i after following our instructions for time t is $x_i + R(\theta_i)u(t)$. Likewise, the distance between any two agents i and j in the workspace is

$$d_{ij}(u) = \|(R(\theta_i) - R(\theta_j))u + (x_i - x_j)\|$$

Level sets of d_{ij} as a function of $u \in \mathbb{R}^2$ are circles centered at

$$u_{ij} = -(R(\theta_j) - R(\theta_i))^{-1}(x_j - x_i).$$

In particular, if we model each agent by a disc of radius r in the workspace, one can show that the set of all $u \in \mathbb{R}^2$ placing agents i and j in collision is a disc centered at u_{ij} with radius

$$\hat{r}_{ij} = \frac{2r}{\sqrt{2 - 2\cos(\theta_1 - \theta_2)}}.$$

Assume we are given $u(0)$ and $u(1)$, corresponding to reachable start and goal configurations $x_i + R(\theta_i)u(0)$ and $x_i + R(\theta_i)u(1)$, respectively, for each agent i. Then the problem of planning a collision-free path for all agents between start and goal is a *path planning* problem in a two-dimensional configuration space $\mathbb{R}^2$ with $n(n-1)/2$ circular obstacles, each a disc of radius $\hat{r}_{ij}$ centered at u_{ij} for some i and j. Many different algorithms are available to solve such problems [12], [6], [13].

Figure 6 shows a simple example for $n = 4$ agents (the same as in Figs. 3-5). The path of each agent in the workspace is shown in Fig. 6(a); the curve u (our set of instructions) is shown in Fig. 6(b). Since there are four agents, there are six circular obstacles—shown as shaded regions—that must be avoided by u. Notice that although the radius of each agent is the same, the radius of each obstacle can be arbitrarily large, depending on the difference $\theta_i - \theta_j$ between each pair of initial orientations.

V. THE MICRO-ROBOT EXAMPLE

Thus far we have assumed a simple kinematic model for each agent. However, one of the motivational applications with which we began this paper involved car-like micro-robots subject to a nonholonomic constraint. If we again assume that two agents are directed to follow the same path (interpreted with respect to local coordinate frames), but now assume that they roll without slipping, is it still possible to direct them both to meet at the same location? Is that location still a single fixed point? Is it still impossible to direct both agents to arbitrary locations?

In fact, if we model each agent as a unicycle (alternatively, a kinematic car or a differential drive vehicle), then the answer to each question is *yes*. This result follows simply from the invariance of unicycle trajectories under rotation and translation. In particular, consider a unicycle that moves forward at constant (unit) velocity, so we can direct its motion only by changing the turning rate $v(t) \in \mathbb{R}$ as a function of time. Denote the position and heading of the unicycle by $u \in \mathbb{R}^2$ and $\theta \in S^1$, respectively. Then we can write the equations of motion as

$$\dot{u} = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$$
$$\dot{\theta} = v$$

For a given control input $v(t)$, these equations can be integrated to find $u(t)$ and $\theta(t)$. Assume that both $u(0) = 0$ and $\theta(0) = 0$. If we apply the same control input $v(t)$ to a second unicycle with states u' and θ' and with different initial
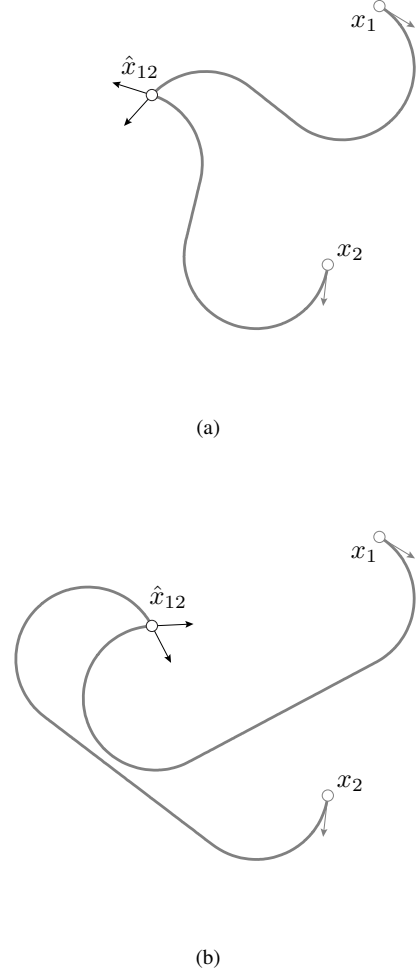

(a)

(b)

Fig. 7. Examples of different ways for two agents to meet at the same location, when their motion is subject to a nonholonomic constraint. Both agents are modeled as unicycles that can only drive forward and that have a minimum turning radius. Their final angle depends on the path taken; their meeting point does not.

conditions $u'(0) = u_0'$ and $\theta'(0) = \theta_0'$, then we can show that

$$u'(t) = u_0' + R(\theta_0')u(t)$$
$$\theta'(t) = \theta_0' + \theta(t),$$

where $R(\theta)$ is defined as the rotation matrix

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

Consequently, if we model each agent i as a unicycle with initial position $x_i \in \mathbb{R}^2$ and orientation $\theta_i \in (-\pi, \pi]$, then its position after time t is $x_i + R(\theta_i)u(t)$, exactly as before. The only difference now is that the input $u(t)$, a curve in $\mathbb{R}^2$, must itself satisfy the equations of motion of a unicycle.

Figure 7 shows an example of directing two agents to meet at the same location (exactly as described in Section III), assuming that they are both unicycles with a minimum turning radius. Notice that the final heading of both agents depends

on the path taken to reach the meeting point. So, despite the fact that our set of instructions is now only one-dimensional (turning rate as a function of time), we can still get a group of agents to perform useful tasks.

VI. Related work

A growing number of applications require the coordinated control of multiple agents. For many of these applications, it is possible to engineer the behavior of, and interactions between, individual agents. For example, one strategy for controlling large groups of mobile robots is to take advantage of flocking or swarming behavior. By following simple control rules based on local sensor measurements (such as regulating distance from nearest neighbors [20]), it is possible for a group of identical robots to accomplish complex global tasks such as movement synchronization [11], rendezvous [7], platooning [15], [10], and distributed manipulation [16]. Moreover, few external inputs are needed to influence this group behavior. For example, a formation can be controlled by commanding the location of its centroid or the variance of its distribution [1] or by directing the movement of a "virtual leader" [14].

In this paper, we are interested in applications for which it is impossible to engineer the behavior of individual agents. In particular, we have focused on one multi-agent system [8] in which every agent receives the same control input. A variety of other multi-agent systems are subject to similar constraints. For example, recent work has considered problems such as herding cattle with a robotic sheepdog [22] or with "virtual fences" [4], interacting with cockroaches using a mobile robot [5], guiding crowds during emergency egress [19], and even using groups of micro-organisms to manipulate small objects [18], [21]. These applications involve considerable uncertainty, so existing solution approaches seek to control the distribution of a group, modeled either by moments [9] or by a probability density function [17] using the formalism of stochastic optimal control. We have presented a different approach that is computationally more tractable but that does not yet address uncertainty. Our approach was based on thinking about *relationships* between agents, and deriving a set of useful tasks in that context.

VII. Conclusion

In this paper we considered the problem of controlling an entire team of agents using only a single control input. Initially we were motivated to pursue this problem by the development of untethered, steerable micro-robots [8], all of which receive the same power and control signals through an underlying electrical grid. We showed through several examples that, despite this restriction, it is possible to get the micro-robots to perform useful tasks. These tasks include meeting at the same location, sequentially passing "information," gathering close together, and avoiding collision.

There are several opportunities for future work. For example, in this paper we focused entirely on open-loop control. These results could be extended to closed-loop control, including a consideration of process noise, sensor uncertainty, and the presence of obstacles in the environment. These results could also be applied to other untethered micro-robots, such as [2], or to the other applications mentioned in Section VI. Finally, several of the motion planning problems raised in Section IV remain to be solved.

References

[1] G. Antonelli and S. Chiaverini. Kinematic control of platoons of autonomous vehicles. *IEEE Trans. Robot.*, 22(6):1285–1292, Dec. 2006.

[2] D. Bell, S. Leutenegger, K. Hammar, L. Dong, and B. Nelson. Flagella-like propulsion for microrobots using a nanocoil and a rotating electro-magnetic field. In *Int. Conf. Rob. Aut.*, pages 1128–1133, Roma, Italy, Apr. 2007.

[3] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[4] Z. Butler, P. Corke, R. Peterson, and D. Rus. From robots to animals: virtual fences for controlling cattle. *Int. J. Rob. Res.*, 25(5-6):485–508, 2006.

[5] G. Caprari, A. Colot, R. Siegwart, J. Halloy, and J.-L. Deneubourg. Insbot: Design of an autonomous mini mobile robot able to interact with cockroaches. In *Int. Conf. Rob. Aut.*, 2004.

[6] H. Choset, K. Lynch, S. Hutchinson, G. Kanto, W. Burgard, L. Kavraki, and S. Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, 2005.

[7] J. Cortés, S. Martínez, and F. Bullo. Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. *IEEE Trans. Automat. Contr.*, 51(8):1289–1296, Aug. 2006.

[8] B. R. Donald, C. G. Levey, C. D. McGray, I. Paprotny, and D. Rus. An untethered, electrostatic, globally controllable MEMS micro-robot. *J. Microelectromech. Syst.*, 15(1):1–15, Feb. 2006.

[9] R. A. Freeman, P. Yang, and K. M. Lynch. Distributed estimation and control of swarm formation statistics. In *American Control Conference*, pages 749–755, Minneapolis, MN, June 2006.

[10] V. Gazi. Swarm aggregations using artificial potentials and sliding-mode control. *IEEE Trans. Robot.*, 21(6):1208–1214, Dec. 2005.

[11] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. Automat. Contr.*, 48(6):988–1001, June 2003.

[12] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, MA, 1991.

[13] S. M. LaValle. *Planning algorithms*. Cambridge University Press, New York, NY, 2006.

[14] N. E. Leonard and E. Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. In *IEEE Conf. Dec. Cont.*, pages 2968–2973, Orlando, FL, Dec. 2001.

[15] Y. Liu, K. M. Passino, and M. M. Polycarpou. Stability analysis of m-dimensional asynchronous swarms with a fixed communication topology. *IEEE Trans. Automat. Contr.*, 48(1):76–95, Jan. 2003.

[16] A. Martinoli, K. Easton, and W. Agassounon. Modeling swarm robotic systems: a case study in collaborative distributed manipulation. *Int. J. Rob. Res.*, 23(4-5):415–436, 2004.

[17] D. Milutinović and P. Lima. Modeling and optimal centralized control of a large-size robotic population. *IEEE Trans. Robot.*, 22(6):1280–1285, Dec. 2006.

[18] N. Ogawa, H. Oku, K. Hashimoto, and M. Ishikawa. Microrobotic visual control of motile cells using high-speed tracking system. *IEEE Trans. Robot.*, 21(3):704–712, June 2005.

[19] X. Pan, C. S. Han, K. Dauber, and K. H. Law. Human and social behavior in computational modeling and analysis of egress. *Automation in Construction*, 15(4):448–461, July 2006.

[20] J. H. Reif and H. Wang. Social potential fields: A distributed behavioral control for autonomous robots. *Robotics and Autonomous Systems*, 27:171–194, 1999.

[21] K. Takahashi, N. Ogawa, H. Oku, and K. Hashimoto. Organized motion control of a lot of microorganisms using visual feedback. In *IEEE Int. Conf. Rob. Aut.*, pages 1408–1413, Orlando, FL, May 2006.

[22] R. Vaughan, N. Sumpter, A. Frost, and S. Cameron. Robot sheepdog project achieves automatic flock control. In *Int. Conf. on the Simulation of Adaptive Behaviour*, 1998.