Proceedings of the 2007 IEEE/RSJ International
Conference on Intelligent Robots and Systems
San Diego, CA, USA, Oct 29 - Nov 2, 2007

Controlling a Team of Ground Robots via an Aerial Robot

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

Abstract— **We consider the task of controlling a large team of nonholonomic ground robots with an unmanned aerial vehicle in a decentralized manner that is invariant to the number of ground robots. The central idea is the development of an abstraction for the team of ground robots that allows the aerial platform to control the team without any knowledge of the specificity of individual vehicles. This happens in much the same way as a human operator can control a single robot vehicle by simply commanding the forward and turning velocities without a detailed knowledge of the specifics of the robot. The abstraction includes a gross model of the shape of the formation of the team and information about the position and orientation of the team in the plane. We derive controllers that allow the team of robots to move in formation while avoiding collisions and respecting the abstraction commanded by the aerial platform. We provide simulation and experimental results using a team of indoor mobile robots and a three-dimensional, cable-controlled, parallel robot which serves as our indoor unmanned aerial platform.**

I. Introduction

In the past few years, the shift in the paradigm of single robot systems to multi-agent systems has been widespread in the area of robotics research. The practical arguments for studying control algorithms for large teams of robots are numerous. However, as the teams scale in size, the challenge of solving the path and trajectory planning problems scale in computational complexity. It is well known from examples in nature that local interaction rules lead to an amazing repertoire of group behaviors [1], [2]. However, there still remain many fundamental questions to be answered. What is the best approach to modeling a team of ground and aerial platforms? How should we specify group level behaviors? And most importantly, how can group behaviors be translated to local robot behaviors?

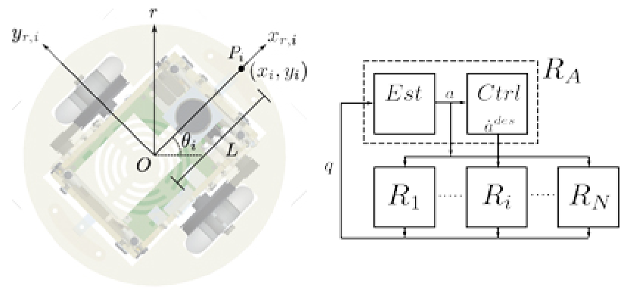
One way of reducing the dimension of the control problem for large numbers of robots is to require them to conform to a rigid virtual structure [3], [4]. In this case, the motion planning problem is reduced to a left invariant control system on $SE(3)$ (or $SE(2)$ in the planar case), and the individual trajectories are $SE(3)$ ($SE(2)$) - orbits. Most of the recent works on stabilization and control of virtual structures model formations using *formation graphs* [5], [6], [7], [8]. The controllers guaranteeing local asymptotic stability of a given rigid formation can be derived using standard techniques such as input-output linearization [5], input-to-state stability [9], Lyapunov energy-type functions [10], [8], and biologically-inspired artificial potential functions [11]. Virtual structures unnecessarily constrain the problem, making this approach inappropriate for tasks such as obstacle avoidance or the passing of narrow tunnels. Also, graph formulations and leader-follower architectures require identification and ordering of robots, which makes the overall architecture sensitive to failures.

In this paper, we are particularly interested in approaches and solution methodologies that are independent of the number of ground robots and do not rely on identifying individual robots in the team. We are also interested, as in the papers above, in decentralized controllers or local behaviors.

In our own previous work, the problem of controlling a large team of point robots in a distributed and decentralized manner was studied in [12], [13]. We 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 shape is also invariant to left actions of the group. This description allows one to define and control the behavior of the team at a high level, with automatic generation of individual robot control laws. The key limitation of this approach was the need for a global observer to provide estimates of the abstract state. While it is possible to design decentralized estimators that provide estimates of the abstract state [14], this approach requires communication between ground robots and it is difficult to predict the convergence rates in practical situations, thus slowing down the controllers to unacceptably low speeds.

In this paper, we overcome this limitation by considering an aerial vehicle that is able to track ground robots and provide an estimate of an abstract shape. We require that the aerial vehicle is able to communicate using a broadcast protocol with the ground robots without requiring inter-robot communications. We also require an invariance to the number of ground robots, anonymity, and communication bandwidth which is independent of the size of the team. We also extend the work in [12], [13] in two ways. First, we apply the methodology to real wheeled platforms with an aerial robot, providing experimental results on systems with nonholonomic constraints. Second, we relax the assumption of point robots and provide guarantees for collision avoidance.

1-4244-0912-8/07/$25.00 ©2007 IEEE.



(a) (b)

Fig. 1. Figure 1(a) shows a top view of a generic ground robot showing the body-fixed coordinate system. P_i is a reference point on the i^{th} robot whose position is regulated by our controller. Figure 1(b) depicts the control architecture. The aerial vehicle, R_A observes the team members and estimates the abstract state, a, of the team. The aerial vehicle computes a desired abstract state velocity $\dot{a}^{des}(t)$ which serves as a feed-forward control for the agents.

II. Modeling

A. Control

1) Ground robots: Consider a set of N robots described by position vectors $q_i = [x_i, y_i]^T$, $i = 1, \ldots, N$ in the world frame, $\{W\}$, of a two-dimensional Euclidean space $\mathbb{R}^2$. We collect all the robot states in $q = [q_1^T, \ldots, q_N^T]^T \in Q \subset \mathbb{R}^{2N}$ and the robot controls in $u = [u_1^T, \ldots, u_N^T]^T \in \mathbb{R}^{2N}$. We will control each robot to emulate a point robot with Euclidean dynamics given by:

$$\dot{q} = u. \tag{1}$$

In the differential-drive robot in Figure 1(a), these are the coordinates of a reference point P_i on the robot which is offset from the axle by a distance L. We consider a simple kinematic model for this point robot:

$$\begin{aligned} \dot{x}_i &= u_{i,1} \\ \dot{y}_i &= u_{i,2} \end{aligned} \tag{2}$$

with the understanding that velocities of the reference point can be translated to commanded linear and angular velocities for the robot through the equations below:

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} = \begin{bmatrix} \cos\theta_i & -L\sin\theta_i \\ \sin\theta_i & L\cos\theta_i \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{3}$$

Let r be the radius of the circle circumscribing the robot. If the robot's reference point is at point P_i, all points on the robot lie within a circle of radius is $L + r$ centered at P_i. In other words, if the reference point tracks a trajectory $\left(x_i^{des}(t), y_i^{des}(t)\right)$, the physical confines of the robot are within a circle of radius $L + r$ of this trajectory. In what follows, we will use the simple kinematic model of Equation (2) to design our controllers relying on our ability to invert the model in Equation (3) for $L \neq 0$ to implement controllers on the real robot.

2) Aerial Robot: In a similar manner to the ground robots, we assume that the aerial robot is kinematically controlled, or simply,

$$\dot{q}_A = u_A, \tag{4}$$

where $q_A \in \mathbb{R}^3$ is the position of the aerial robot in the world frame. The notation in Section II-A.1 is easily adapted by replacing $\mathbb{R}^2$ with $\mathbb{R}^3$.

B. Abstraction

We now define our model of the formation which allows us to abstract away the details of individual robots. Instead of modeling the precise shape, which requires $2N$ variables, we consider an ellipsoidal approximation of the formation analogous to [12], [13], called the *spanning ellipsoid*. Our *abstract state*, $a = (g,\ s) \in \mathcal{A}$, consists of the pose, $g \in SE(2)$, and shape, $s \in \mathcal{S}$, of the formation, where $\mathcal{S}$ is the shape space. In this paper, since we only consider ellipsoids in the plane, $s \in \mathcal{S} \subset \mathbb{R}^2$.

1) Pose: Define the mean position of the team of ground robots as

$$\mu = \frac{1}{N} \sum_{i=1}^{N} q_i. \tag{5}$$

We associate with μ the translation from $\{W\}$ to the body frame of the robots $\{B\}$.

Allowing p_i to represent the position of the i^{th} robot in $\{B\}$, we write the local body frame coordinates, $p_i = [x_i,\ y_i] = R_B{}^{\mathrm{T}}(q_i - \mu)$.

The orientation of the body frame (and thus spanning ellipse), θ, is constrained such that

$$\sum_{i=1}^{N} x_i y_i = 0. \tag{6}$$

2) Shape: The general shape and span of the team of robots are modeled by the semi-axes of the spanning ellipse:

$$s_1 = \sum_{i=1}^{N} \frac{x_i^2}{N-1} \text{ and } s_2 = \sum_{i=1}^{N} \frac{y_i^2}{N-1}. \tag{7}$$

3) Abstraction Map: From the above equations, we can define a surjection, ϕ (from the system state $q \in Q$ to an abstraction $a \in A$) given by:

$$\phi : Q \to \mathcal{A},\ \phi(q) = a \tag{8}$$

where in coordinates, $a = [\mu_1, \mu_2, \theta, s_1, s_2]$.

III. Control Algorithm

A. Abstract State Control

By differentiating Equation (8), we formulate the linear system,

$$\dot{a} = d\phi\dot{q}, \tag{9}$$

As in [12], [13], we use the Moore-Penrose inverse of $d\phi$ to derive controls

$$\tilde{u} = (d\phi)^{\dagger}\dot{a}, \tag{10}$$

which are a function of $\{q,\ a,\ \dot{a}\}$. This inverse, $(d\phi)^{\dagger}$ yields vector fields that are decoupled (and orthogonal with respect to the standard Euclidean metric on $\mathbb{R}^2$) and allow independent control of each component of the abstract state

a. Thus we can command $\{\dot{\mu},\ \dot{\theta},\ \dot{s}_1,\ \dot{s}_2\}$ independently. The control law defined in Equation (10) gives us:

$$\begin{aligned}\tilde{u}_i = \dot{\mu} + \frac{s_1 - s_2}{s_1 + s_2} H_3(q_i - \mu)\dot{\theta} \\ + \frac{1}{4s_1} H_1(q_i - \mu)\dot{s}_1 + \frac{1}{4s_2} H_2(q_i - \mu)\dot{s}_2,\end{aligned} \tag{11}$$

where H_1, H_2, and H_3 are defined in [12], [13].

B. Collision Avoidance

The control law presented in Section III-A does not consider the local neighbor interactions and the possibility for collisions between team members. In this section, we derive the controller for each agent, $\hat{u}_i$, that is orthogonal to the controls $\tilde{u}_i$ and does not affect the evolution of the abstract state. Further, $\hat{u}_i$ is designed to avoid collisions.

We assume that the shape description, s_1 and s_2, are chosen such that lengths of the semi-axes of the spanning ellipse are geometrically larger than the physical span of the team of robots in a packed configuration. From our model in Section II-A.1, each robot can be modeled by a circular disk of radius $\epsilon = r + L$ which reflects the characteristic radius of the robot (treating the robot as a disc). Therefore, it is reasonable to require that the robots perform collision avoidance when the distance between two robots

$$\|q_{ji}\| = \|q_i - q_j\| < 2\epsilon + \delta, \tag{12}$$

where δ is chosen as a safety factor to ensure adequate clearance between the robots.

The idea underlying the collision avoidance controller is simple. We find a control $\hat{u}$ that lies in the null space of $d\phi$, $\hat{u} \in \mathcal{N}(d\phi)$. However, finding $\hat{u}$ for avoiding collisions requires knowledge of the complete state of the system, a restriction that eliminates the distributed properties of the algorithm. Additionally, the symbolic computation of the null space for large N shows that the spanning vector for $\mathcal{N}(d\phi)$ can be chosen so that the controls can easily be made zero for $N-3$ robots. Thus we design our collision avoidance controller for triads of neighboring robots, indexed by $\{i,\ j,\ k\}$. For such a system of three robots, $\mathcal{N}(d\phi)$ is one-dimensional and given by:

$$\mathcal{N}(d\phi) = \begin{bmatrix} \cos(\theta)(x_j - x_k) - \sin(\theta)(y_j - y_k) \\ \sin(\theta)(x_j - x_k) + \cos(\theta)(y_j - y_k) \\ \cos(\theta)(x_k - x_i) - \sin(\theta)(y_k - y_i) \\ \sin(\theta)(x_k - x_i) + \cos(\theta)(y_k - y_i) \\ \cos(\theta)(x_i - x_j) - \sin(\theta)(y_i - y_j) \\ \sin(\theta)(x_i - x_j) + \cos(\theta)(y_i - y_j) \end{bmatrix}. \tag{13}$$

Thus, in the world frame, the resulting local control law is

$$\hat{u}_i = \lambda_{ijk}(q_j - q_k), \tag{14}$$

where λ_{ijk} is a scaling on the magnitude of the control input yielding controls for the i^{th} robot that are along the direction of the vector going from robot k to robot j. Similarly the controls for the j^{th} (k^{th}) robot are along the direction of the vector going from robot i (j) to robot k (i) as shown in Figure 2. By superpositioning Equations (11) and (14), we get the control input for robot i:

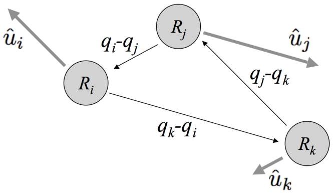


Fig. 2. The null space for designing collision avoidance controls, $\hat{u}$, for a triad of three robots, $\{i,\ j,\ k\}$.

$$u_i = \tilde{u}_i + \hat{u}_i, \tag{15}$$

or, more generally,

$$u = \tilde{u} + \hat{u}. \tag{16}$$

The collision avoidance for the triad of robots results in a motion that does not change the abstract state. Each robot triad must determine the appropriate scaling λ_{ijk} to ensure that the control, $u_i = \tilde{u}_i + \hat{u}_i$, results in a safe motion. In the worst case scenario when all three pairs of robots are within danger of colliding, λ_{ijk} must be selected so that for the triad of robots, as shown in Figure 2,

$$(u_n - u_m) \cdot (q_n - q_m) \geq 0, \tag{17}$$

letting $\{m,\ n\} = \{\{i,\ j\}, \{j,\ k\}, \{k,\ i\}\}$. If a pair of robots, say i and j, are in danger of colliding, the robots must select λ_{ijk} so that:

$$\text{if } \|q_{ji}\| \leq 2\epsilon + \delta,\ (u_i - u_j) \cdot (q_i - q_j) \geq 0. \tag{18}$$

The magnitude of λ_{ijk} is chosen to be between 0 and α depending on the distance, $\|q_{ji}\|$:

$$\lambda_{ijk} = \pm\alpha \frac{2\epsilon}{2\epsilon + \delta} \left[\frac{2\epsilon + \delta}{\|q_{ji}\|} - 1 \right]. \tag{19}$$

The sign is chosen to ensure the inequality (18) is satisfied.

For larger N, one can develop schemes for automatically decomposing the team into sets of overlapping triads and develop decentralized approaches for calculating the scaling coefficients λ_{ijk} for each triad. For this paper, it suffices to say that it is possible to come up with the coefficients λ_{ijk} that satisfy all inequalities involving pairs of potentially colliding robots, except when the volume of the ellipsoid becomes small relative to the number and size of robots.

C. Estimation and control of the abstract state

The estimation of the abstract state, a, the trajectory planning or the generation of the desired abstract state trajectory, $a^{des}(t)$, and the computation of the feed-forward control, $\dot{a}^{des}$, can be best performed by an aerial platform that has a “bird’s-eye” perspective of the world. Here, we will assume that the desired abstract state trajectory is available from a human operator and discuss the other two tasks below.

1) Abstract State Estimation: We will use a camera attached to the aerial platform in Section IV-B.2 to estimate the pose and shape of the system in its local frame. As shown in the schematic in Figure 1(b), the resulting state estimate is then transformed from the frame of the aerial robot to the world frame before it is broadcast to the ground robots.

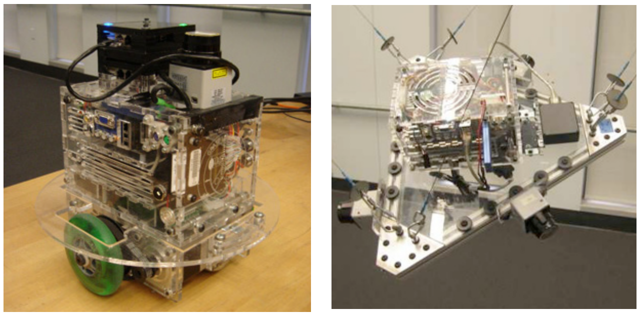

(a) (b)

Fig. 3. Experimental Hardware. The 20 × 13.5 × 22.2 cm^3 Scarab platform is shown in Figure 3(a). The Khepri robot shown in Figure 3(b) is controlled by six cables and has a suite of sensing and computational abilities making it well suited for emulation of an UAV in indoor environments.

2) Control of the Abstract State: We wish to control the system to a desired abstract state a^{des} or $a^{des}(t)$, for waypoint control or trajectory tracking, respectively. Proportional or proportional-derivative control laws are suitable since the abstract state remains bounded (see [12], [13]), guaranteeing convergence. Given a trajectory, $a^{des}(t)$, we want to command the abstract state velocity with a simple proportional control law:

$$\dot{a} = \dot{a}^{des} + K\left(a^{des} - a\right) \tag{20}$$

where K is a suitable positive-definite matrix. This equation combined with Equations (11) and (15) gives us the controllers for individual robots for achieving asymptotic convergence to $a^{des}(t)$. Each ground robot requires estimates of its own state, estimates of the abstract state, a, from the aerial platform, and the trajectory, $a^{des}(t)$. Further, it requires estimates of its neighbors' positions for collision avoidance.

IV. Implementation and Results

A. Simulation Environment

The 3D environment Gazebo, part of the Player/Stage/Gazebo project [15], was used to verify the correctness of the control laws presented for both the ground and aerial robots. Models of the environment of the local laboratory and hardware (discussed in Section IV-B) were reproduced in a simulated world. The robot models accurately reflect the geometric, kinematic, and dynamic descriptions of the robots used in the hardware implementation.

B. Experimental Testbed

1) Ground Robots: Our small form-factor robot is called the Scarab. The Scarab is a $20 \times 13.5 \times 22.2$ cm^3 indoor ground platform with $r_i = 15$ cm. Each Scarab is equipped with a differential drive axle placed at the center of the length of the robot with a 21 cm wheel base.

2) Aerial Robot: Khepri [16] is a six degree of freedom cable-controlled robot shown in Figure 3(b). It is equipped with three Hokuyo URG laser range finders, a three axis IMU, and a color firewire camera. While the Khepri has an interesting actuation system with a complex workspace and forward kinematics equations, it also serves as a platform on which to develop distributed algorithms to deal with air-ground coordination and sensing in a laboratory environment.

3) Ground Truth: A locally developed ground-truth verification system permits the tracking of LED markers (visible in Figure 3(a)) with a position error of approximately 1 cm and an orientation error of 1°. The tracking system consists of LED markers and eight overhead cameras.

C. Software

Every robot is running identical modularized software with well defined abstract interfaces connecting modules via the Player robot architecture system. We process global overhead tracking information but hide the global state of the system from each robot, providing only the current abstract state (a) of the system as well as the positions of each robot's set of neighbors during collision avoidance. In this way, we use the tracking system in lieu of an inter-robot sensor implementation.

The abstract state was estimated in simulation and experimentation using the LED tracking system or from the camera on Khepri. The resulting state estimation was broadcast to the ground robots. In addition, the abstract state was used to control the aerial robot, Khepri, using the control law

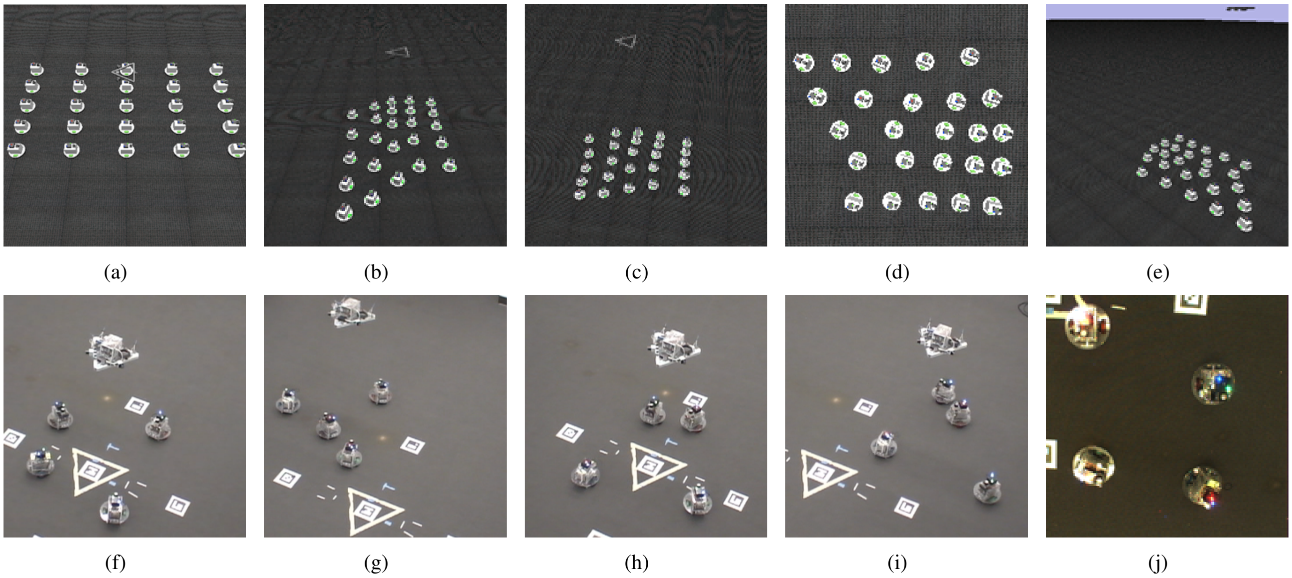
$$\begin{aligned} \dot{x} &= k_x(\mu_x - x), \\ \dot{y} &= k_y(\mu_y - y), \\ \dot{z} &= k_z(z^{des} - z), \end{aligned}$$

where $k_x,\ k_y,\ k_z > 0$ and z^{des} was either a constant height or a function of the abstract state and the intrinsic parameters of the camera.

D. Simulation Results

Simulations were created to verify the effectiveness of the control algorithm. The simulations were designed to realistically emulate a team of robots (e.g. parallel computation, networked messaging between agents, etc.). Each simulation consisted of twenty-five ground robots and an aerial robot.

A series of trials were designed to test the stability and convergence properties of the abstract controller and the effectiveness of the local collision avoidance. The mean squared error (MSE) of the abstract state from the desired abstract state was computed for each trial and averaged over ten runs. Additionally, the tests were performed by specifying a^{des} or $a^{des}(t)$ to the aerial robot. The resulting $\dot{a}$ was computed by the aerial robot and broadcast to the ground robots. The aerial robot also tracked the mean of the abstraction. Figures 4(a)–4(e) depict images from a trial run where $a^{des}(t)$ was specified to be a sinusoidal trajectory of the form $\mu_y(t) = \sin(\frac{2\pi}{10}\mu_x(t))$, $\theta(t) = \arctan(\cos(\frac{2\pi}{10}\mu_x(t)))$, $s_1 = 1.5$, and $s_2 = 1$.


(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

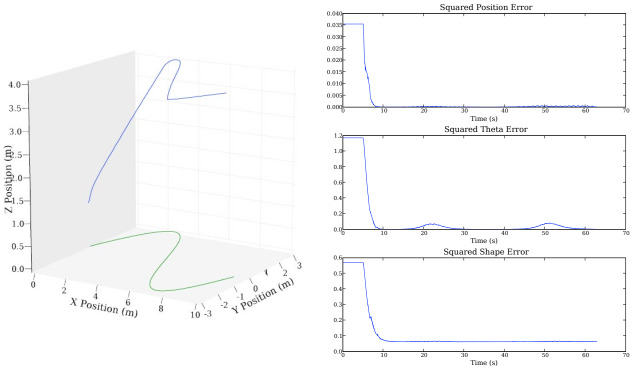
Fig. 4. Comparable Simulation and Experimentation Scenarios. Figures 4(a)–4(e) present a representative trial run simulated in GAZEBO. Figure 4(a) shows the starting formation of twenty five robots. Figures 4(b)–4(c) present the motion of the group given a sinusoidal abstract trajectory. Figures 4(c) and 4(d) show the position of the system and the corresponding view from the aerial robot's camera Figure 4(e) shows the final position of the system. Figures 4(f)–4(j) depict a similar scenario to Figures 4(a)–4(e) but with four SCARAB robots. Figure 4(f) shows the start configuration. Figure 4(g) depicts the convergence of the ground robots to $a^{des} = \{1, 1, 0.5, 1, 0.5\}$ (where $a^{des} = \{\mu_x,\ \mu_y,\ \theta,\ s_1,\ s_2\}$). Figures 4(h)–4(i) depict the motion of the system to $a^{des} = \{1, -1, -0.5, 0.5, 1\}$. Figure 4(j) shows an image from the camera on the aerial robot. Note that the aerial robot is controlling to $x = \mu_x, y = \mu_y$, and $z = 3.0\ m$ or $z = 1.5\ m$, in simulation or experimentation, respectively.

The results of the MSE for these trials are shown in Table I. The errors shown for the runs are likely due to a significant but practical implementation consideration. As the robots upon which the algorithms were run are driven by stepper motors, a minimum velocity ($0.03\ m/s$) is required to prevent overheating and motor stall. This limitation is captured in the simulation. As such, an additional series of trials was performed without this restriction. The results are shown in Table I.

a^{des}, v_{min}	XY MSE (m)	θ MSE (rad)	s_1,s_2 MSE
Constant, 0	0.0000	0.0000	0.0000
Constant, 0.03	0.0002	0.0001	0.0001
Time-varying, 0.03	0.0023	0.0894	0.0980

TABLE I

SIMULATION RESULTS OF AVERAGE MSE OVER TEN TRIALS WITH AND WITHOUT A LOWER VELOCITY BOUND.



(a) (b)

Fig. 5. Analysis of simulated system following a sinusoidal trajectory. Figure 5(a) shows the 3D position of the aerial robot and abstraction. Figure 5(b) shows the squared error of μ, θ, s_1, and s_2.

The MSE for the trials in simulation demonstrate the effectiveness of the control algorithm for a team of nonholonomic robots. The difference in MSE values for the two trials with waypoints demonstrates the effects of the minimum velocity bound on the convergence of the algorithm.

It should be noted that an additional modification was required during experimentation. When $s_1 \approxeq s_2$, the numerical computation of the shape parameters may result in a rapid change in the orientation θ of the ellipse. During such a situation, the scaling of $\dot{\theta}$ in Equation (11) becomes zero (or nearly zero). Thus, these rapid changes have little impact on the control law in the direction of θ. However, the symmetry creates a situation in which the directions of s_1 and s_2 are not consistently defined. These issues were resolved by placing a low-pass filter on the estimation of the abstract state that removed large fluctuations of θ. This addition is reasonable under the bounded velocity assumption on the ground robots.

E. Experimental Results

The abstract controller was implemented on a team of four SCARABS and KHEPRI in scenarios similar to the simulations. The algorithm was tested for convergence and stability but not for collision avoidance (this is an avenue of future research).

A series of waypoints (a^{des}) were specified over ten trials (allowing convergence of the system to twenty specified de-

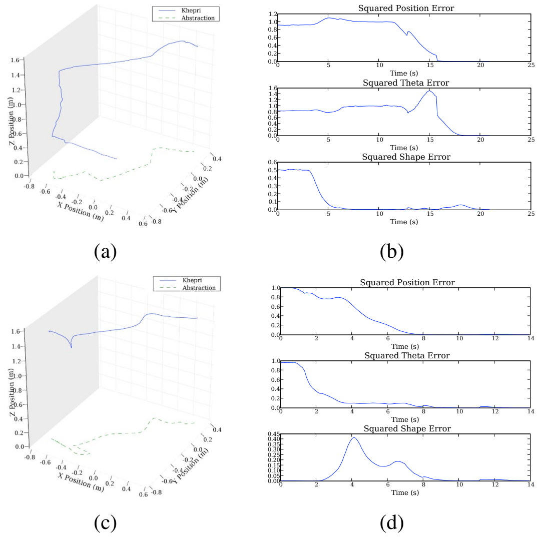


Fig. 6. Experimental results for a sample trial run following two waypoints. Figures 6(a) and 6(c) show the trajectories of KHEPRI while tracking the μ values of the abstraction. Figures 6(b) and 6(d) present the squared error for the abstract description of the system corresponding to the trajectories shown in Figures 6(a) and 6(c), respectively.

sired abstract states). The resulting average MSE for twenty trials on the hardware are presented in Table II. Figures 4(f)–4(j) show images from one of the trial runs.

a^{des}, v_{min}	XY MSE (m)	θ MSE (rad)	s_1,s_2 MSE
Constant, 0.03	0.001	0.008	0.017

TABLE II

EXPERIMENTAL RESULTS OF AVERAGE MSE OVER TWENTY TRIALS.

The MSE over twenty trials is indicative of the effectiveness of the algorithm on real hardware. Under the assumptions of the accuracy of our ground truth system (as discussed in Section IV-B.3), these results support the convergence properties for the control algorithm on a team of robots.

V. DISCUSSION

We presented an architecture and decentralized controllers for controlling a large team of nonholonomic robots. The key idea is the development of an abstraction for the team of ground robots that allows the aerial platform to anonymously control the team without requiring specific knowledge of the physical system. The abstraction includes a gross model of the shape of the formation of the team and information about the position and orientation of the team in the plane. In our previous work [12], [13], we established the geometric framework and developed control algorithms for idealized point robots. In this paper, we demonstrated the feasibility of this approach by reducing the theoretical concepts and algorithms to practice with real nonholonomic ground robots and a three-dimensional, cable-controlled, aerial robot. We also derived controllers that allow the team of robots to move in formation while avoiding collisions.

There are two main directions of current research. First, we want to enrich the set of abstract models for our ground robots and aerial robot. The simple five-dimensional abstract state does not allow the team to split into subteams or merge subteams into a single team. Further, we want to develop a unified three-dimensional abstract model of aerial and ground robots using generalized cones. Developing decentralized estimators and controllers to realize such abstractions is a focus of ongoing work.

VI. ACKNOWLEDGMENT

We gratefully acknowledge the support of ARO Grants DAAD19-02-01-0383, W911NF-05-1-0219, and W911NF-04-1-0148, and NSF Grant IIS-0427313.

REFERENCES

[1] C. Reynolds, "Flocks, birds, and schools: a distributed behavioral model," *Computer Graphics*, vol. 21, pp. 25–37, 1987.

[2] A. Jadbabaie, J. Lin, , and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transaction on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[3] T. Eren, P. N. Belhumeur, and A. S. Morse, "Closing ranks in vehicle formations based rigidity," in *Proc. of the IEEE Conference on Decision and Control*, vol. 3, Las Vegas, NV, 2002, pp. 2959–2961.

[4] C. Belta and V. Kumar, "Optimal motion generation for groups of robots: a geometric approach," *Journal of Mechanical Design*, vol. 126, pp. 63–70, 2004.

[5] J. P. Desai, J. P. Ostrowsky, and V. Kumar, "Controlling formations of multiple mobile robots," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Leuven, Belgium, 1998, pp. 2863–2869.

[6] T. Sugar and V. Kumar, "Decentralized control of cooperating mobile manipulators," *IEEE Transactions on Robotics*, vol. 18, no. 1, pp. 94–103, 2002.

[7] P. Tabuada, G. J. Pappas, and P. Lima, "Feasible formations of multi-agent systems," in *Proc. of the American Control Conf.*, Arlington, VA, June 2001.

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

[9] H. Tanner, G. J. Pappas, and V. Kumar, "Input-state stability in teams of robots," in *Proc. of the IEEE Conference on Decision and Control*, vol. 3, 2002, pp. 2439–2444.

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

[11] P. Ogren, E. Fiorelli, and N. E. Leonard, "Formations with a mission: stable coordination of vehicle group maneuvers," in *Symp. on Math. Theory of Networks and Sys.*, Notre Dame, IN, Aug. 2002.

[12] 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.

[13] 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, Florida, May 2006, pp. 964–969.

[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, Minnesota, June 2006, pp. 749–755.

[15] B. 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.

[16] D. Theodorakatos, "Cable actuated parallel manipulators," Master's thesis, University of Pennsylvania, Philadelphia, PA, Apr. 2007.