Short Papers

Controlling Noncooperative Herds with Robotic Herders

Alyssa Pierson and Mac Schwager

***Abstract*—We present control strategies for robotic herders to drive noncooperative herds. Our key insight enforces geometrical relationships that map the combined dynamics to simple two-dimensional or three-dimensional nonholonomic vehicle models. We prove convergence of single-agent herds to a goal and propose strategies for multi-agent herds, verified in simulations and experiments.**

***Index Terms*—Biologically inspired robots, kinematics, mobile robots, multirobot systems, robotic herding.**

I. Introduction

We consider the problem of noncooperative herding, analogous to shepherding, wherein dogs drive a herd of sheep to a goal location. In this system, the "sheep" agents naturally run away from the "dog" robots. We design a feedback control strategy for the dogs to relocate the sheep to a goal region in the environment. The dogs coordinate their positions to partially encircle the herd, which steers the herd in a desired direction. To design these controllers, we show, under certain geometrical constraints, the dynamics of the system reduce to common nonholonomic vehicles in two-dimension (2-D) and three-dimension (3-D) models. Using this insight, we map a simple linear control strategy back to the dogs' control laws. Simulations and experiments with Pololu m3pi robots demonstrate the performance.

Although we use the dog–sheep analogy to describe our system, this is only for illustrative purposes. In general, the "dogs" are robotic agents under our control. The "sheep" agents are not under our control, but assumed to behave with herd dynamics.

One application of our control strategy is in wildlife management. For example, in Australia, helicopters are used to muster cattle, requiring pilots to fly at low altitudes and perform dangerous maneuvers, resulting in as many as ten deaths annually [1]. Implementing our control strategy on teams of unmanned aerial vehicles (UAVs) to autonomously muster cattle may reduce human risk. Another application is managing wildlife populations in national parks, where it is necessary to monitor animals and steer them away from potential environmental dangers. In 3-D, this may include directing schools of fish or flocks of birds. In the case of an emergency evacuation, human crowds could be directed by robots using our control strategy. We consider this a noncooperative multirobot problem, since the objective of the dogs is to steer the sheep, but the sheep are not actively inclined nor opposed to being steered.

Related Work: There has been surprisingly limited prior work on noncooperative robotic herding. One exception is Vaughan's pioneering work [2], [3], in which a single robot is used to herd ducks. In Vaughan's work, the robot communicates with a centralized computer vision system to choose controllers to drive the herd to the goal. Lien *et al.* developed a set of behavior primitives for controlling a flock with multiple shepherds [4]. The herders are placed at a set of "steering points" around the flock, and choose their behavior primitive based on environmental properties. In [5], the authors use a sliding model controller to place herders around a single evader to drive that evader along a desired trajectory. The evader is repulsed from the herders with a linear force within some sensing range, otherwise does not react to the herders. In contrast, our work takes a control theoretic approach to design feedback laws for multiple dogs to drive an arbitrary number of sheep. Other authors have formulated the problem as a dynamic pursuit-evasion game [6], [7]. Using formation control, agents can be relocated by driving a formation to a goal [8], [9]; however, this assumes the herd cooperates with the herders. In contrast, our herd is noncooperative and have a nonlinear repulsion to the herders. To model the herd dynamics, we use potential fields, which is common in animal aggregation modeling for schools of fish [10], birds, slime molds, mammal herds, and other swarms [11], [12], as well as human crowd dynamics [13]. These models have been applied to multi-agent systems to simulate flocking [14], cooperative group control [15], [16], and interaction with collision avoidance [17].

Our work proposes a reduction from the nonlinear dog–sheep system to well-known nonholonomic vehicle models. In 2-D, our system reduces to a unicycle model for a differential drive robot [18]. While there are many techniques to maneuver a unicycle-like vehicle, we choose a "point-offset controller" that controls a point offset from the center of the robot, whose dynamics are holonomic [19].

Similarly, in 3-D, we show that our system reduces to a common nonholonomic vehicle model used in underwater autonomous vehicle modeling [20], [21] and aerial vehicles [22], [23]. We also derive a 3-D extension of the point-offset controller for this nonholonomic vehicle model. By designing feedback controllers for the point offset, we obtain the controllers for the herders that drive the herd to a goal region in 3-D.

A preliminary version of the 2-D problem appeared in our conference publication [24]. This paper includes the extension to 3-D, as well as expanded simulations that include trajectory following for obstacle avoidance and experimental results. The remainder of the paper is organized as follows. In Section II, we formulate the 2-D problem. Section III presents the extensions to 3-D. Simulation results are pre-

Manuscript received February 25, 2017; revised August 14, 2017; accepted October 27, 2017. This paper was recommended for publication by Associate Editor H. Kress-Gazit and Editor T. Murphy upon evaluation of the reviewers' comments. This work was supported in part by National Science Foundation (NSF) CAREER award IIS-1350904, in part by the NSF under Grant CNS-1330036, and in part by the Office of Naval Research under Grant N00014-12-1000. *(Corresponding author: Alyssa Pierson.)*

A. Pierson is with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: apierson@mit.edu).

M. Schwager is with the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305 USA (e-mail: schwager@stanford.edu).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2017.2776308

Algorithm 1: Herding Control in 2-D.

1: Calculate the controller for $\dot{p}$ (10)
2: Find ideal heading ψ^* and velocity v^* (5)
3: Find Δ_j^* from v^* using (9)
4: Calculate desired dog positions d_j^* (11)
5: Calculate tracking controller for $\dot{d}_j$ (13)

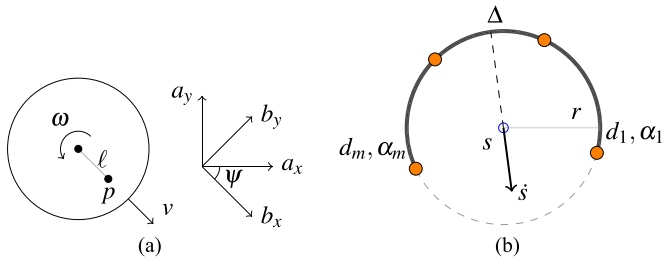


Fig. 1. (a) Nonholonomic vehicle model and (b) placement of m dogs around a single sheep s with overall angular spacing Δ.

sented in Sections IV and V, with experimental results in Section VI. We give our conclusions in Section VII.

II. 2-D Formulation and Controller Design

Consider m herders (or "dogs") with positions $d_j \in \mathbb{R}^2$, where $j \in \{1, \dots, m\}$, and n herd members (or "sheep") with positions $s_i \in \mathbb{R}^2$, where $i \in \{1, \dots, n\}$. The "dogs" are robots under our control, while the herd members can be robots, herding animals, or even humans. For the purposes of this paper, we use the shepherding analogy throughout. We assume the dogs have integrator dynamics

$$\dot{d}_j = u_j \tag{1}$$

for control input u_j. Our main goal is to design u_j such that the dogs drive the sheep to some goal region. We model the sheep's repulsion from the dogs using an artificial potential field [15]. Using the potential field $W(d, s) = \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{1}{\|d_j - s_i\|}$, we obtain

$$\dot{s}_i = \frac{\partial W}{\partial s_i} = \sum_{j=1}^{m} \frac{-(d_j - s_i)}{\|d_j - s_i\|^3}. \tag{2}$$

We define the set of desired final configurations of the sheep as $B_\ell(g) = \{q \in \mathbb{R}^2 \mid \|q - g\| \leq \ell\}$, centered at goal point $g \in \mathbb{R}^2$ with a desired radius $\ell > 0$. Without loss of generality, we define $g = 0$ for the remainder of the paper.

Problem 1: **(Multi-Agent Herding)** Given the dynamics of the herd (2), find control laws $u_j = f(d, s)$ for d_j herders with dynamics (1) to relocate the herd from arbitrary initial conditions to the desired region in the environment $B_\ell(g)$.

We propose a solution to Problem 1 that is both simple and scalable to $m \geq 2$ herders, summarized in Algorithm 1. The key insight of our approach lies in enforcing geometrical relationships that map the complex, nonlinear dog and sheep dynamics to a simple unicycle model. This creates an ideal unicycle-like system, which we utilize in our controller design. We first introduce terminology and basic nomenclature to describe the unicycle model, then present our herding models that reduce to the unicycle-like system.

A. Modeling of a 2-D Unicycle Vehicle

Consider the unicycle-like vehicle shown in Fig. 1(a). For a unicycle-like vehicle with position s, we define a local reference frame B relative to the global frame A, with the rotation matrix from B to A denoted ${}^A R^B$, and ψ defined as the angle of rotation, written as

$${}^A R^B = \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix}.$$

We express the B frame in global coordinates as

$$b_x = {}^A R^B \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \text{ and } b_y = {}^A R^B \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The vehicle moves with forward velocity v in the local b_x direction, and angular velocity $\omega = \dot{\psi}$, as shown in Fig. 1(a). Overall, the kinematic constraints are

$$\dot{s} = v b_x, \quad \dot{\psi} = \omega. \tag{3}$$

B. Point-Offset Control

Consider a point p offset a distance ℓ from s, written $p = s + \ell b_x$. While the unicycle-like vehicle has nonholonomic dynamics, it turns out that p is holonomic [19], which allows us to design controllers for p then transform them back to the vehicle dynamics v and ω. The derivative of p reduces to

$$\dot{p} = {}^A R^B \begin{bmatrix} v \\ \ell\omega \end{bmatrix}. \tag{4}$$

We can rearrange (4) to solve for the velocity and angular velocity in the local basis vectors as

$$v = b_x^T \dot{p}, \quad w = \dot{\psi} = \frac{1}{\ell} b_y^T \dot{p}. \tag{5}$$

This relationship allows us to find some desired control $\dot{p} = u$ and map it back to the vehicle controls v and ω.

C. Kinematic Reduction

Instead of allowing the herders to occupy any point in the environment, consider the case where all herders are a fixed distance r from the sheep. In this section, we show that under this constraint, the system dynamics reduce to a unicycle-like vehicle. The position of each dog is written as

$$d_j = s + r \begin{bmatrix} \cos(\alpha_j) \\ \sin(\alpha_j) \end{bmatrix} \tag{6}$$

where α_j is the angular orientation with respect to ψ, found as

$$\alpha_j = \psi + \pi + \Delta_j, \quad \Delta_j = \Delta \frac{(2j - m - 1)}{(2m - 2)}.$$

Substituting (6) into (2), the sheep dynamics become

$$\dot{s} = \frac{-\sin\left(\frac{m\Delta}{2-2m}\right)}{r^2 \sin\left(\frac{\Delta}{2-2m}\right)} \begin{bmatrix} \cos(\psi) \\ \sin(\psi) \end{bmatrix} \tag{7}$$

which allows us to describe the dynamics of the sheep using only the two state variables ψ and Δ, despite having m dogs. Similarly, the dynamics for the dogs becomes

$$\dot{d}_j = \dot{s} + r\left(\dot{\psi} + \dot{\Delta}_j\right) \begin{bmatrix} \sin(\psi + \Delta_j) \\ -\cos(\psi + \Delta_j) \end{bmatrix}. \tag{8}$$

By defining the orientation of the dogs in terms of ψ and Δ along some radius in (6) and restricting the dogs' kinematics to obey (8), we can map these quantities to the angular and linear velocity of a unicycle-like vehicle.

Remark 1: Note this model assumes the dogs are fixed on some circle of radius r relative to the herd. Later, we introduce a tracking controller that allows the dogs to start anywhere and converge upon this configuration.

Proposition 1: The herding dynamics in (7) and (8) can be reduced to an equivalent unicycle model with forward velocity v and orientation ψ, described by (3).

Proof: We can solve this mapping by equating (7) to (3) and solving for v. To see this mapping, note that by (7), the direction of the herd's velocity is only in the local x-direction, similar to the unicycle model in (3). The direction in global coordinates is defined by ψ. The magnitude of the velocity is

$$v = \|\dot{s}\| = \frac{\sin\left(\frac{m\Delta}{2-2m}\right)}{r^2 \sin\left(\frac{\Delta}{2-2m}\right)}. \tag{9}$$

For (9), there exist an infinite number of possible values of Δ for a given value of v, but over the range of $\Delta = (0, 2\pi)$, this mapping is one-to-one. For a given velocity, we can find the corresponding Δ. It remains to map $\dot{\psi}$ in the herder's dynamics (8). We directly map $\dot{\psi} = \omega$ from the unicycle dynamics. ■

Instead of determining individual controllers for all dogs, Proposition 1 allows us to design controllers for the ideal unicycle-like system. Based on this idealized system, we find controllers for the dogs that enforce this behavior. This approach can be extended to other repelling potential field models by recalculating $\|\dot{s}\|$ for the mapping in (9).

D. Controller Design

Section II-C introduced geometric constraints on the system, which allow us to map the kinematics of the herding system to a unicycle-like vehicle. Our goal, as stated in Problem 1, is to drive the herd to some ball around the origin. To control this system, we propose a controller that drives a point offset of the ideal unicycle-like system to the origin. Given the velocity and angular velocity controls of the ideal system, we then calculate the positions for the dogs along the circumference of the circle.

The desired position for each dog lies on a circle of radius r around the sheep, with spacing Δ_j between dogs. We assume the dogs are able to perfectly track their desired positions, a common assumption in the multirobot literature [8], [25]. The ideal orientation ψ^* is the angle that points the herd's velocity toward the origin. To find the ideal velocity for the unicycle-like system, we find a controller for the point offset p that drives the point offset to the origin. While there exist many possible choices for controlling the point offset p, we opt for a simple proportional feedback controller

$$\dot{p} = -kp. \tag{10}$$

Plugging this into (5), the ideal velocity becomes $v^* = [\cos(\psi^*) \;\; \sin(\psi^*)](-kp)$. Using (9), the desired separation is Δ_j^*. Overall, this yields the desired position of the dogs

$$d_j = s + r \begin{bmatrix} \cos(\psi^* + \Delta_j^*) \\ -\sin(\psi^* + \Delta_j^*) \end{bmatrix}. \tag{11}$$

We can now analyze the system as if it were the simple unicycle-like system, and are ready to state our main proposition on the behavior of a single sheep and m dogs.

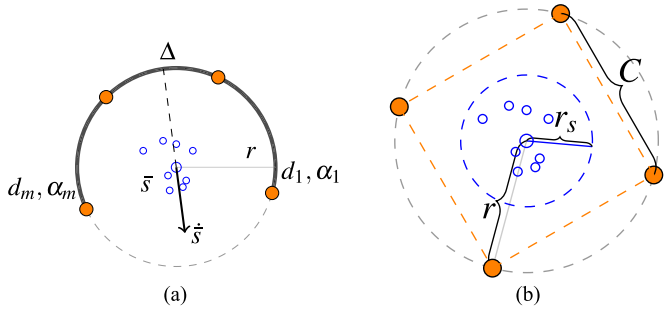


Fig. 2. (a) Model of m dogs n sheep, (b) defining the control radius r for m dogs based on herd footprint r_s and dog spacing C.

Proposition 2: For the single sheep, m dog system described in (7) and (8), using the controllers

$$v = -kb_x^T\,(s + \ell b_x)\,, \quad w = \dot{\psi} = \frac{-k}{\ell} b_y^T\,(s + \ell b_x) \tag{12}$$

the herd converges to the circle of radius ℓ about the origin.

Proof: It is equivalent to say that if the point offset converges to the origin, the herd converges to the ball B_ℓ about the origin. Consider the Lyapunov candidate function $V = \frac{1}{2}\,(s + \ell b_x)^T\,(s + \ell b_x)$. Substituting our expressions for $\dot{p}$ defined in (4), its derivative becomes $\dot{V} = p^T\,(v b_x + \ell\dot{\psi} b_y)$. Plugging in the expressions for v and $\dot{\psi}$ chosen in (12), this becomes

$$\dot{V} = p^T\,(-k(b_x^T p)b_x + -k(b_y^T p)b_y) = -k\|p\|^2 < 0.$$

By Lyapunov's direct method [26], the equilibrium point $p^* = 0$ is asymptotically stable. Furthermore, by the form of $\dot{V}$, it is exponentially stable. When $p = 0$, the sheep is a distance ℓ away from the origin, thus proving Proposition 2. ■

Proposition 2 proves that m dogs spaced on a circle can relocate a single sheep to a goal. To extend this to n sheep, we calculate the dogs' controller from the herd's mean $\bar{s}$ and choose a radius r that contains the herd's footprint, as shown in Fig. 2(c). Let r_s be the radius of herd's footprint. We choose a desired control radius r such that $r\cos\left(\frac{\pi}{m}\right) - r_s < \frac{1}{2}C$, where C is the distance between dogs when $\Delta = 2\pi$. Under this expression, the repulsive force experienced by any herd member will not drive it outside the footprint of the herders. If the herd's extent grows, additional controllers to modify the radius may be added to the design.[1]

For Proposition 2, we enforce that the dogs remain on a circle of radius r around the herd. In this next section, we introduce a tracking controller for the dogs, which allows them to start from any point in the environment. This additional modification is summarized in the overall control algorithm, presented in Algorithm 1. The proposed tracking controller is

$$\dot{d}_j = -K_d(d_j^* - d_j) \tag{13}$$

where d_j^* is the desired location expressed in (11). Under the following mild assumption, we can analyze the performance of this control strategy.

Assumption 1: The desired dog positions d_j^* (11) evolve slowly enough compared to the speed of our dogs $\dot{d}_j$ (13) that we can assume perfect tracking, $d_j^* = d_j$.

[1] A rigorous analysis of this complex phenomenon requires additional modeling on the flocking dynamics of the herd as well as the dynamics of r_s. In practice, for a static r_s, our algorithm can control multiple sheep.

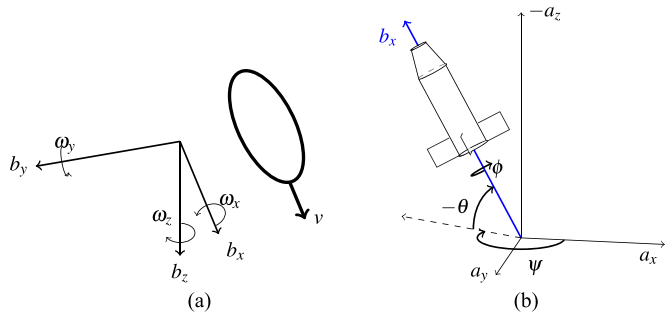


Fig. 3. Forward-moving 3-D nonholonomic vehicle with (a) local frame B and (b) ZYX rotation from global frame A.

In practice, this assumption means the dynamics of the ideal system are significantly slower than the dynamics to drive the dogs to their ideal positions. Under Assumption 1, the desired positions d_j^* are constant and the tracking controller (13) d_j converges exponentially to d_j^*. This allows us to the start the dogs from any point within the environment, and they will converge upon the ideal unicycle-like system. We validated this assumption in MATLAB simulations and hardware experiments.

III. 3-D Formulation and Controller Design

In this section, we present the herding problem formulation for a 3-D system. For consistency with the previous section, we denote the m herders as "dogs" with positions $d_j \in \mathbb{R}^3$, and n herd members as "sheep" with positions $s_i \in \mathbb{R}^3$. Of course, real dogs and sheep do not move freely in three dimensions, but we keep the analogy for consistency. In practice, 3-D herding may be applied to situations with schools of fish, flocks of birds, spacecraft swarms, or UAV swarms. Here, we drive the dogs to a configuration wherein the herd is controlled like a nonholonomic vehicle. First, we formulate a kinematic model for a nonholonomic vehicle, introduce geometric constraints that reduce the herd dynamics to a nonholonomic vehicle, and later present our controllers for the herd. As before, we model the sheep's repulsion from the dogs using an artificial potential field (2). Here, our goal region $B_\ell(g)$ is a ball defined around a goal point $g \in \mathbb{R}^3$ with radius $\ell > 0$. Without loss of generality, we define our coordinate frame to be centered at the goal point, such that $g = 0$.

A. Modeling of a 3-D Nonholonomic Vehicle

Consider a 3-D nonholonomic vehicle with center point $s = [x\ y\ z]^T$ relative to the global reference frame, as shown in Fig. 3(a). We define a local reference frame B relative to some global frame A. Its forward velocity v defines the local b_x direction. The vehicle is able to rotate around its x-, y-, and z-axis with angular velocities ω_x, ω_y, and ω_z, respectively. This is a common model in both underwater autonomous vehicle modeling [20], [21], [27]–[29] and aerial vehicles [22], [23], [30]. Using the North-East-Down notation, the rotation matrix between the local frame B to the global frame A is

$$ {}^A R^B = \begin{bmatrix} c_\psi c_\theta & -s_\psi c_\phi + c_\psi s_\theta s_\phi & s_\psi s_\phi + c_\psi c_\phi s_\theta \\ s_\psi c_\theta & c_\psi c_\phi + s_\psi s_\theta s_\phi & -c_\psi s_\phi + s_\theta s_\psi c_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} $$

where ϕ, θ, and ψ are the ZYX Euler angles, as described in Fig. 3(b). We express the B frame in global coordinates as

$$ b_x = {}^A R^B \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},\ b_y = {}^A R^B \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix},\ b_z = {}^A R^B \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. $$

Overall, the kinematic constraints of the vehicle dynamics are

$$ \begin{aligned} \dot{s} &= v b_x, \\ \begin{bmatrix} \dot\phi \\ \dot\theta \\ \dot\psi \end{bmatrix} &= \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \end{aligned} \tag{14} $$

where v is the forward velocity of the vehicle, and ω_x, ω_y, and ω_z are the angular velocities of the vehicle [20].

B. Point-Offset Control

Similar to the 2-D case, we control the nonholonomic vehicle by controlling a point offset along the b_x axis, and it turns out that the point p is holonomic. For a vehicle with position s, the point $p = s + \ell b_x$ has time derivative

$$ \dot p = \dot s + \frac{d}{dt}\left[{}^A R^B\right] \begin{bmatrix} \ell \\ 0 \\ 0 \end{bmatrix}. \tag{15} $$

By definition, $\dot s = v b_x$, since our vehicle can only move forward in its local x-axis. It is well known from 3-D kinematics that the derivative of a rotation matrix reduces to $\frac{d}{dt}\left[{}^A R^B\right] = {}^A R^B \Omega$, where Ω is a skew-symmetric matrix of local angular velocities [18]. Overall, (15) reduces to

$$ \dot p = {}^A R^B \begin{bmatrix} v & -\ell\omega_z & \ell\omega_y \\ \ell\omega_z & v & -\ell\omega_x \\ -\ell\omega_y & \ell\omega_x & v \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. \tag{16} $$

To solve for $[v, \omega_x, \omega_y, \omega_z]$, we rearrange (16) as

$$ {}^B R^A \dot p = \begin{bmatrix} v & \ell\omega_z & -\ell\omega_y \end{bmatrix}^T \tag{17} $$

with $\omega_x = 0$. This allows us to find a desired control $\dot p = u$ and map it back to vehicle controls v, ω_z, and ω_y.

Lemma 1 (Point-Offset Control in 3-D): For a nonholonomic vehicle located at s with forward velocity v and angular velocities $(\omega_x, \omega_y, \omega_z)$, using the controller

$$ v = -k b_x^T p,\ \omega_x = 0,\ \omega_y = \frac{k}{\ell} b_z^T p,\ \omega_z = \frac{-k}{\ell} b_y^T p \tag{18} $$

the vehicle converges to a ball of radius ℓ around the origin.

Proof: It is equivalent to say that if the point offset converges to the origin, the herd converges to the ball B_ℓ about the origin. Consider the Lyapunov candidate function

$$ V = \frac{1}{2} p^T p = \frac{1}{2} (s + \ell b_x)^T (s + \ell b_x). $$

Its derivative is

$$ \dot V = \left(s + {}^A R^B \begin{bmatrix} \ell \\ 0 \\ 0 \end{bmatrix} \right)^T \left({}^A R^B \begin{bmatrix} v \\ \ell\omega_z \\ -\ell\omega_y \end{bmatrix} \right). $$

Substituting (18) yields

$$ \dot V = -k p^T \left(b_x^T p b_x + b_y^T p b_y + b_z^T p b_z \right) = -k\|p\|^2 \le 0. $$

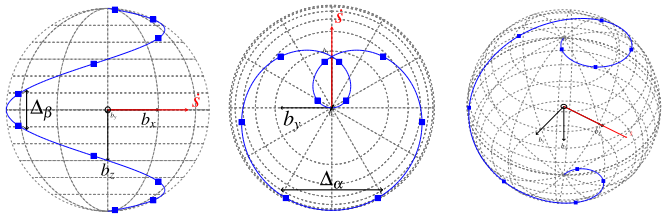


Fig. 4. Herder placement (blue squares) on sphere centered about herd (black circle). The direction of the herd $\dot{s}$ is given by the red arrow. The herders space themselves along a spiral parameterized by Δ_α and Δ_β, which allows the system to scale to any number of herders. (a) Side view. (b) Top view. (c) Isometric view.

By Lyapunov's direct method [26], the equilibrium $p^* = 0$ is asymptotically stable. Furthermore, by $\dot{V}$, $p^* = 0$ is exponentially stable. When $p = 0$, s is a distance ℓ away from the origin, thus completing the proof of Lemma 1. ■

C. Kinematic Reduction

Instead of allowing the herders to occupy any point in the environment, consider when all herders are a fixed distance r from the sheep. We show that when this occurs, the system dynamics reduce to a 3-D nonholonomic vehicle, as described in Section III-A. For the case of $n = 1$ sheep and $m \geq 3$ dogs, let α_j and β_j be the azimuthal and polar angles relative to the herd. The relative location of a dog becomes

$$d_j = s + {}^A R^B \begin{bmatrix} r\cos(\alpha_j)\sin(\beta_j) \\ r\sin(\alpha_j)\sin(\beta_j) \\ r\cos(\beta_j) \end{bmatrix}. \tag{19}$$

Previously, we introduced Δ that described the angular separation between d_1 and d_m, and evenly distributed all other dogs within Δ. Extending these principles to this spherical case, we similarly distribute the herders across their azimuthal and polar angles. Let Δ_α and Δ_β describe the separation in the azimuthal and polar angles. We then define the individual angles

$$\alpha_j = \Delta_{\alpha j} - \pi, \quad \beta_j = \Delta_{\beta j} + \frac{\pi}{2}$$

where

$$\Delta_{\alpha j} = \Delta_\alpha \frac{(2j - m - 1)}{(2m - 2)} \quad \text{and} \quad \Delta_{\beta j} = \Delta_\beta \frac{(2j - m - 1)}{(2m - 2)}.$$

In our implementation, we assign $\Delta_\beta = \pi$ and vary Δ_α. Fig. 4 illustrates how the herders distribute themselves along a spiral wrapping around the herd's sphere. The spiral wrapping is one method of distributing the herders around the sphere, and what we use in our subsequent controller design. Other methods may place the herders on fixed latitude and longitude "tracks," or a mesh grid that scales with the desired velocity. We choose the spiral design due to its ability to scale to any number of herders while only requiring two parameters Δ_α and Δ_β.

By distributing the herders around the parameterized spiral, we find that the dynamics reduce to a 3-D nonholonomic vehicle similar to a fixed-wing aircraft or an autonomous underwater vehicle. Given the desired angular separations Δ_α and Δ_β, the herd's dynamics in (2) become

$$\dot{s} = \frac{-1}{2r^2}\left(\frac{\sin\left(\frac{m(\Delta_\alpha - \Delta_\beta)}{2-2m}\right)}{\sin\left(\frac{\Delta_\alpha - \Delta_\beta}{2-2m}\right)} + \frac{\sin\left(\frac{m(\Delta_\alpha + \Delta_\beta)}{2-2m}\right)}{\sin\left(\frac{\Delta_\alpha + \Delta_\beta}{2-2m}\right)}\right) b_x \tag{20}$$

which implies that the herd only moves in its local b_x direction. To maintain the kinematic relationship, the dogs' dynamics are

$$\begin{aligned} \dot{d}_j = \dot{s} &+ {}^A R^B \Omega \begin{bmatrix} -r\sin(\Delta_{\alpha j})\cos(\Delta_{\beta j}) \\ -r\cos(\Delta_{\alpha j})\cos(\Delta_{\beta j}) \\ -r\sin(\Delta_{\beta j}) \end{bmatrix} \\ &+ {}^A R^B \dot{\Delta}_{\alpha j} \begin{bmatrix} -r\cos(\Delta_{\alpha j})\cos(\Delta_{\beta j}) \\ r\sin(\Delta_{\alpha j})\cos(\Delta_{\beta j}) \\ 0 \end{bmatrix} \end{aligned} \tag{21}$$

where Ω is the skew-symmetric matrix of local angular velocities and $\dot{\Delta}_{\alpha j}$ is the derivative of the local azimuthal angle. Note that since Δ_β is constant, $\dot{\Delta}_{\beta j} = 0$. By defining the dogs in terms of the rotation matrix ${}^A R^B$ and angular separations Δ_α and Δ_β along some radius and restricting the kinematics to obey (21), we can map these quantities to the linear and angular velocities of a 3-D nonholonmic vehicle.

Proposition 3: The herding dynamics in (20) and (21) can be reduced to an equivalent three-dimensional nonholonomic vehicle with forward velocity v and angular velocities $(\omega_x, \omega_y, \omega_z)$ described by (14).

Proof: To see this mapping, note that the direction of herd is determined by its velocity, which moves solely in the local x-direction. For the velocity, we find from (20)

$$v = \|\dot{s}\| = \frac{1}{2r^2}\left(\frac{\sin\left(\frac{m(\Delta_\alpha - \Delta_\beta)}{2-2m}\right)}{\sin\left(\frac{\Delta_\alpha - \Delta_\beta}{2-2m}\right)} + \frac{\sin\left(\frac{m(\Delta_\alpha + \Delta_\beta)}{2-2m}\right)}{\sin\left(\frac{\Delta_\alpha + \Delta_\beta}{2-2m}\right)}\right). \tag{22}$$

Note that for (22), there are an infinite number of possible values of Δ_α and Δ_β for a given value of v. However, when $\Delta_\beta = \pi$, over the range of $\Delta_\alpha = (0, 3\pi)$, this mapping is one-to-one. Thus, for given velocity, we can find the corresponding Δ_α. The dynamics for $\dot{\Delta}_\alpha$ are also be found from the dynamics of $\dot{v}$. It remains to map the angular rotation rates to the dogs dynamics. From (21), the dogs' kinematics directly incorporate Ω, the skew-symmetric matrix of angular velocities. Thus, given a linear velocity v and angular velocities $(\omega_x, \omega_y, \omega_z)$, we can control the herd dynamics to that of a 3-D nonholonomic vehicle. ■

We use Proposition 3 in our controller design. Instead of determining individual controllers for each of the herding agents, we design controllers for the ideal nonholonomic-like system. Based on this idealized system, we then find the controllers for the herders to enforce this behavior.

D. Controller Design

Section III-C introduced geometric constraints on the system, which map the kinematics of the herding system to a 3-D nonholonomic vehicle. Our goal in Problem 1 is to relocate the herd to some goal region. To control this system, we use a point-offset controller, as described in Section III-B.

To find the desired dog positions, we first find a controller for a point offset p from the herd. While there exist many possible choices for controlling p, we opt for a simple proportional feedback controller:

$$\dot{p} = -kp. \tag{23}$$

Algorithm 2: Herding Control in 3-D.

1: Calculate the controller for $\dot{p}$ (23)
2: Calculate ideal velocities $(v^*, \omega_x^*, \omega_y^*, \omega_z^*)$ (17)
3: Find ideal rotation matrix ${}^A R^{B*}$ from Ω^*
4: Find Δ_α^* from v^* (22)
5: Calculate desired dog positions d_j^* (24)
6: Calculate tracking controller for $\dot{d}_j$ (25)

Substituting (23) into (17) yields the controllers for (v, w_y, w_z) from (18). Using the mapping in (22), we calculate the desired separation Δ_α^* for the herders. The desired angular velocities yield Ω, which we use to calculate a desired rotation matrix ${}^A R^{B*}$. Combined, we write the herder locations as

$$d_j = s + {}^A R^{B*} \begin{bmatrix} -r\sin(\Delta_{\alpha j}^*)\cos(\Delta_{\beta j}) \\ -r\cos(\Delta_{\alpha j}^*)\cos(\Delta_{\beta j}) \\ -r\sin(\Delta_{\beta j}) \end{bmatrix}. \tag{24}$$

Proposition 4: For the single herd, multiherder system in (20) and (21), the herd converges to the ball of radius ℓ about the origin B_ℓ.

Proof: Consider the point offset $p = s + \ell b_x$. It is equivalent to say that is the point offset converges to the origin, the herd converges to the ball B_ℓ about the origin. By Proposition 3, our mapping in (22) reduces the system to a 3-D nonholonomic-like system. By Lemma 1, we show the controllers in (18) drive the nonholonomic vehicle to a ball B_ℓ. Thus, we apply these results and conclude that our multiherder system converges to a ball of radius ℓ about the origin. ■

In practice, we may want the herders to start at any point. Similar to the 2-D case, we can use Assumption 1 that states the desired positions of the dogs evolve slowly enough compared to the speed of our dogs $\dot{d}_j$ that we can start the dogs anywhere and they will converge upon the ideal nonholonomic vehicle system. The tracking controller is

$$\dot{d}_j = -K_d\left(d_j^* - d_j\right) \tag{25}$$

where d_j^* is the desired position given in (24). Algorithm 2 summarizes the overall controller for herding in 3-D.

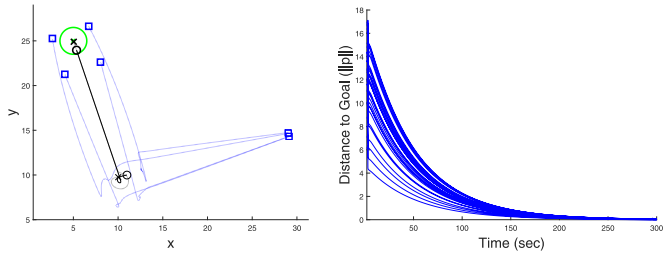
IV. 2-D Simulations

We demonstrate the capabilities of our algorithm in MATLAB simulations. First, we present simulations illustrating the case of $n = 1$ sheep with m dogs in 2-D. Despite starting from random configurations, our system converges to the dynamics of the ideal unicycle-like vehicle, and successfully relocates the herd to the goal. We also demonstrate our algorithm for multiple sheep with flocking dynamics and navigating around obstacles to reach the goal.

A. Herding With $n = 1$ Sheep

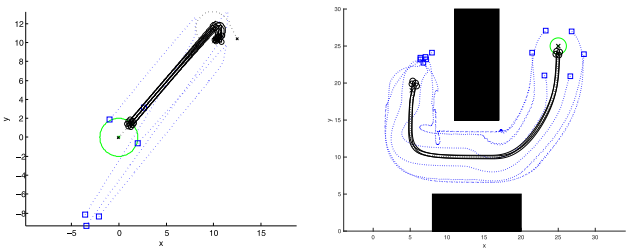
Our first simulation shows the case of $m = 4$ dogs and a single sheep. Fig. 5(a) illustrates the trajectory of one simulation. The goal $B_\ell(g)$ is shown in green. The blue squares denote the dogs, and the black circle and x are the sheep and point offset, respectively. The dogs start at random locations and converge to a circle around the sheep, which then drives the point offset to the origin.

To illustrate the performance for a variety of initial conditions, Fig. 5(b) compares the distance between the point offset ($\|p\|$) and the goal over 30 trials. The initial starting locations were randomized for each agent in each of the trials, yet we see in all simulations the point offset converges to the origin, validating our claims in Proposition 2.



(a) (b)

Fig. 5. (a) Trajectory of the herders (blue square) and herd (black circle), ending within the goal region (green). Note the point offset from the herd (black x) is at the center of the goal region. (b) Distance of point offset to goal over 30 trials with randomized initial conditions.



(a) (b)

Fig. 6. (a) Trajectories of $m = 3$ dogs and $n = 10$ sheep moving toward goal region. (b) Trajectories of $m = 6$ dogs and $n = 4$ sheep moving around obstacles toward goal. To avoid the obstacles, the herders generate a trajectory as a series of waypoints for the point-offset control.

B. Herding With $n > 1$ Sheep

For the case of multiple sheep with m dogs, we add interagent herd forces in addition to the repulsion from the dogs. For the purposes of these simulations, we use the flocking dynamics presented in Vaughan's work [2], [3]. Fig. 6 shows two examples of controlling multiple sheep. The controllers use the herd mean $\bar{s}$. Here, the dogs are still able to control the group to some goal region using the point-offset controller on the mean of the herd. In Fig. 6(b), the herders navigate the herd around obstacles in the environment. Here, the herders generate the herd trajectory as a series of intermediate goal waypoints, such that each waypoint avoids the obstacle and the final waypoint is concurrent with the desired goal.

V. 3-D Simulations

This section presents MATLAB simulations that demonstrate our herding algorithm in 3-D. First, we present a simulation of a single herd agent and m herders. Despite starting from random configurations, our system converges to the dynamics of a nonholonomic vehicle and successfully relocates the group to the goal. Next, we present simulations with multiple herd members in both an obstacle-free environment as well as an environment with obstacles. To navigate around obstacles, the herders generate a series of intermediate goal waypoints to create a trajectory around the obstacles and demonstrate the flexibility of our controllers. Finally, for the case of multiple herd members, we examine the system's robustness to noise. We add noise to both the herd movement as well as the measurement of the herd's position, and examine the efficacy of our herding control policy.

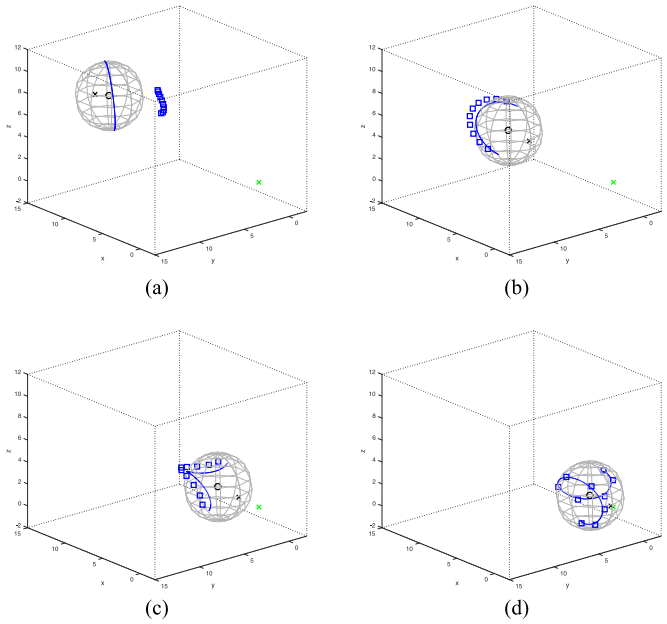


Fig. 7. Positions of the $m = 10$ herders (blue squares) and herd (black circle) over time. The goal is denoted by the green x. Over time, the herd is relocated to the region around the goal. (a) $t = 0$. (b) $t = 10$. (c) $t = 20$. (d) $t = 30$.

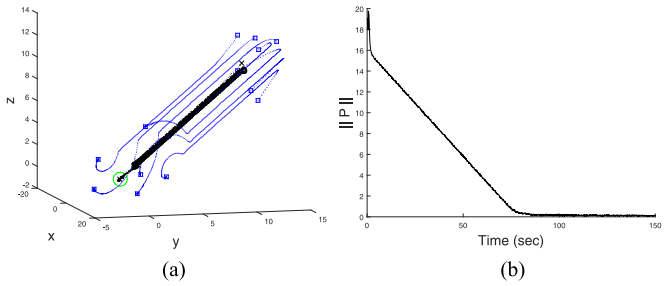


Fig. 8. (a) Trajectories of $n = 3$ herd members and $m = 7$ herders over time. (b) Distance of the point offset from the mean of the herd p to the goal. Note that over time, the distance decreases until reaching the goal, at which point the herd is contained at the goal region.

A. Relocation to Goal

Our first simulation shows the case of $m = 10$ herders and a single herd agent. Fig. 7 shows the system evolving over time. In the figure, the green x denotes the goal point. The blue squares denote the herders, and the blue circle and x are the herd and the point offset, respectively.

Similar to 2-D, we extend our algorithm to a flocking herd. Fig. 8(a) shows a simulation with $n = 3$ herd members and $m = 7$ herders. Despite controlling multiple herd members, the herders successfully relocate the herd to a ball about the origin, shown in Fig. 8(b).

B. Obstacle Avoidance

Our algorithm allows for the herders to move the herd along some desired trajectory, which is useful for avoiding obstacles or obstructions en route to the goal. Fig. 9 demonstrates $m = 6$ herders moving the herd around an obstacle. The herders generate and follow a trajectory to avoid collision. Here, we generate the trajectory as a series of waypoints for the herd and herders, such that the herd and herders will never intersect the obstacle at each waypoint or moving between the waypoints.

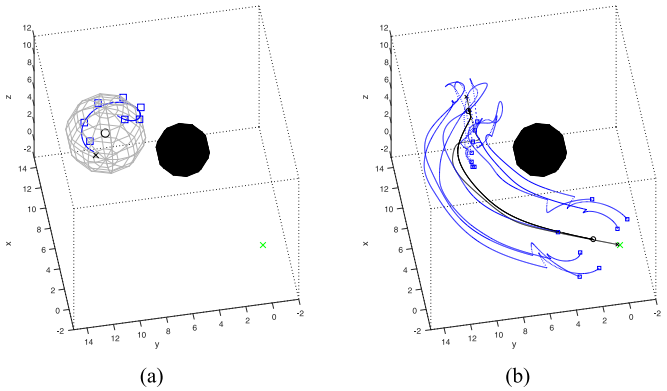


Fig. 9. (a) Herders must avoid an obstacle between the herd and the goal region. (b) Trajectories of the herders (blue squares) and herd (black circle) avoiding the obstacle while relocating to the goal by choosing a trajectory that navigates around the obstacle.

TABLE I
SUCCESS RATE OF TRIALS AS COMPARED TO THE RATIO $\frac{K_n}{v_{\max}}$

Noise ratio	$\frac{K_n}{v_{\max}} < 1$	$1 \geq \frac{K_n}{v_{\max}} \leq 1.2$	$\frac{K_n}{v_{\max}} > 1.2$
Success rate	99.7%	42.9%	0%

C. Robustness to Noise

In this section, we investigate the robustness to noise in the 3-D system. We look at two main sources of noise: noise added to the movement of the herd, and noise in dogs' measurement of the herd's position. For each case, we add varying levels of noise to examine the effect on the dogs' ability to relocate the herd to the goal.

To model noise in movement, we consider a group of $m = 7$ dogs and $n = 1$ sheep. Noise was added to the sheep's dynamics (2) as

$$\dot{s} = \sum_{j=1}^{m} \frac{-(d_j - s)}{\|d_j - s\|^3} + K_n X$$

where X is a random number drawn from a uniform distribution and K_n varies the magnitude of X. The noise is not directly seen by the dogs during their calculation for $\dot{d}_j$. To study the effect of noise, we varied K_n and examined the distance of the point offset from the goal at time $t = 800$ s, well beyond the time required to relocate the herd. For each value of K_n, we performed 70 trials with randomized initial configurations. Table I summarizes the success rate of the herders relative to the noise. Note that the transition between successful herding and failed attempts occurs around $\frac{K_n}{v_{\max}} > 1$, once the dynamics of $\dot{s}$ are dominated by the noise instead of the repulsion forces induced by the dogs.

To study the effect of noise in measurement, we add noise to the herder's estimate of the herd's position, with each herder using a different estimate of the herd. We use

$$\hat{h}_j = h + K_n X_j \tag{26}$$

where h is the true herd position, X_j is a random number drawn from a uniform distribution, and K_n is the magnitude of random noise. For each value of K_n, we ran 70 trials with randomized initial configurations and examined the distance of the point offset to the goal at 800 s. From Table II, we notice a sharp transition in the success rate around

TABLE II
SUCCESS RATE OF TRIALS AS COMPARED TO THE RATIO $\frac{K_n}{r}$

Noise ratio	$\frac{K_n}{r} < 0.9$	$0.9 \geq \frac{K_n}{r} \leq 1$	$\frac{K_n}{r} > 1$
Success rate	97.8%	36.9%	0%

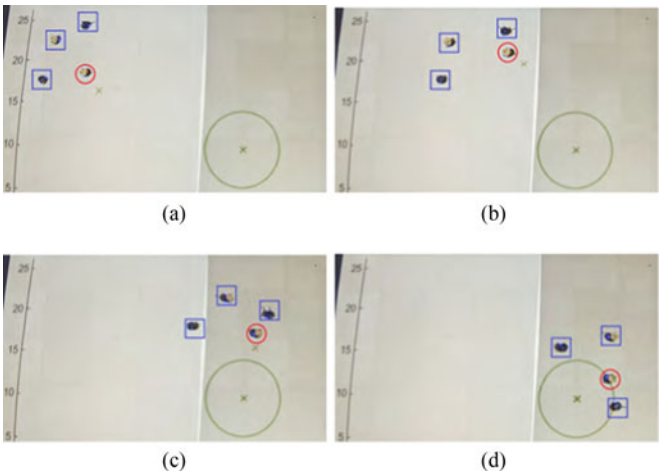


Fig. 10. Video stills illustrating the herding of $n = 1$ robot sheep (red circle) by $m = 3$ robot herders (blue squares). (a) $t = 0$ s. (b) $t = 70$ s. (c) $t = 142$ s. (d) $t = 214$ s.

$\frac{K_n}{r} = 1$. For $\frac{K_n}{r} < 1$, this implies the estimate of the herd's position is within the control radius.

Overall, these simulation results show the system is inherently robust to noise. For noise on the herd's movement, this implies that if the herd had unmodeled dynamics, but was reasonably close to a potential field repulsion model, our algorithm would be able to account for these changes. Similarly, the noise on the herder's measurement of the herd implies that the herders do not need to be perfectly aligned on the control radius to still effectively relocate the herd.

VI. EXPERIMENTS

To demonstrate our algorithm, experiments were conducted in the Multi-Robot Systems Lab at Boston University. OptiTrack[2] was used to provide real-time localization. We use Pololu's m3pi[3] robot equipped with an mbed microcontroller and XBee[4] radio. Control laws were computed in MATLAB and sent to the m3pi robots via XBee radios. The biggest challenge during implementation was the culmination in system inefficiencies not present in simulation. While our simulations assume that all robots have holonomic dynamics, the m3pis are small (10 cm) nonholonomic vehicles with noisy, lossy actuation. These unmodeled behaviors are hard to predict or quantify in simulation. Despite these challenges, we performed repeated successful experiments with the m3pi robots. For this experiment, we use $n = 1$ m3pi "sheep," and $m = 3$ m3pi "dogs" in a 4 m $\times$ 3 m environment. Fig. 10 and the video attachment illustrate the herders successfully relocating the herd. The positions of the dogs (blue squares) and sheep (red circle) have been highlighted. Fig. 11 shows the results of five different trials with random initial conditions. The additional noise on the trajectories comes from the unmodeled dynamics, communication delays, and a low-level nonholonomic controller within the experimental system. In each trial, the herd is relocated to the goal, demonstrating an inherent robustness in our system.

[2]www.naturalpoint.com/optitrack
[3]www.pololu.com
[4]www.digi.com/xbee

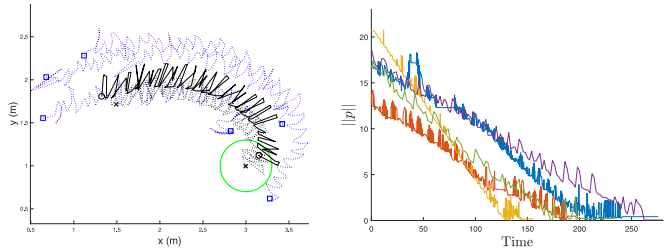


Fig. 11. (a) Trajectories of the dogs (blue squares) and sheep (black circle). (b) Distance of the sheep's point offset p to the goal over five trials with varying initial conditions.

VII. CONCLUSION

We consider the problem of herders controlling a noncooperative herd, analogous to sheep herding. Despite the nonlinear dynamics of the system, by utilizing geometric constraints, we can map these dynamics to common nonholonomic vehicles in 2-D and 3-D, for which a simple feedback controller can be formulated. Unlike prior work, we use a single continuous control law without relying on switching or heuristic behaviors. For a single sheep, we use a Lyapunov-like proof to show the sheep converge exponentially to the goal. We also propose strategies for multiple sheep. Simulations and hardware experiments demonstrate the performance of these strategies. Future extensions may analyze the convergence of the multisheep case. Additional problems may also include navigation through environment with dynamic obstacles.

REFERENCES

[1] M. Lane, "Helicopter cowboys of Australia's outback," Feb. 2011. [Online]. Available: http://www.bbc.co.uk/news/world-asia-pacific-12408888
[2] R. Vaughan, "Experiments in automatic flock control," Ph.D. dissertation, Univ. Oxford, Oxford, U.K., 1999.
[3] R. Vaughan, N. Sumpter, J. Henderson, A. Frost, and S. Cameron, "Experiments in automatic flock control," *Robot. Auton. Syst.*, vol. 31, no. 1, pp. 109–117, 2000.
[4] J.-M. Lien, S. Rodriguez, J. Malric, and N. Amato, "Shepherding behaviors with multiple shepherds," in *Proc. 2005 IEEE Int. Conf. Robot. Autom.*, Apr. 2005, pp. 3402–3407.
[5] M. Bacon and N. Olgac, "Swarm herding using a region holding sliding mode controller," *J. Vib. Control*, vol. 18, no. 7, pp. 1056–1066, 2012. [Online]. Available: https://doi.org/10.1177/1077546311411346
[6] S. A. Shedied, "Optimal control for a two-player dynamic pursuit evasion game: The herding problem," Ph.D. dissertation, Virginia Polytech. Inst. State Univ., Blacksburg, VA, USA, 2002.
[7] Z. Lu, "Cooperative optimal path planning for herding problems," Master's thesis, Texas A&M Univ., College Station, TX, USA, Dec. 2006.
[8] M. Egerstedt and X. Hu, "Formation constrained multi-agent control," in *Proc. 2001 IEEE Int. Conf. Robot. Autom.*, vol. 4, 2001, pp. 3961–3966.
[9] G. Ferrari-Trecate, M. Egerstedt, A. Buffa, and M. Ji, "Laplacian sheep: A hybrid, stop-go policy for leader-based containment control," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, vol. 3927, J. Hespanha and A. Tiwari, Eds. Berlin, Germany: Springer, 2006, pp. 212–226.
[10] C. M. Breder, "Equations descriptive of fish schools and other animal aggregations," *Ecology*, vol. 35, no. 3, pp. 361–370, Jul. 1954.

[11] V. Gazi and K. M. Passino, "Stability analysis of social foraging swarms," *IEEE Trans. Syst. Man Cybern. B*, vol. 34, no. 1, pp. 539–557, Feb. 2004.
[12] J. Toner and Y. Tu, "Flocks, herds, and schools: A quantitative theory of flocking," *Phys. Rev. E*, vol. 58, no. 4, 1998, Art. no. 4828.
[13] S. Patil, J. van den Berg, S. Curtis, M. C. Lin, and D. Manocha, "Directing crowd simulations using navigation fields," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 2, pp. 244–254, Feb. 2011.
[14] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 25–34, Aug. 1987. [Online]. Available: http://doi.acm.org/10.1145/37402.37406
[15] A. Howard, M. Matari, and G. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *Distributed Autonomous Robotic Systems*, vol. 5, H. Asama, T. Arai, T. Fukuda, and T. Hasegawa, Eds. Tokyo, Japan: Springer, 2002, pp. 299–308.
[16] H. Tanner, A. Jadbabaie, and G. Pappas, "Flocking in fixed and switching networks," *IEEE Trans. Automat. Control*, vol. 52, no. 5, pp. 863–868, May 2007.
[17] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. Automat. Control*, vol. 51, no. 3, pp. 401–420, Mar. 2006.
[18] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*, 1st ed. Boca Raton, FL, USA: CRC Press, 1994.
[19] N. Michael and V. Kumar, "Planning and control of ensembles of robots with non-holonomic constraints," *Int. J. Robot. Res.*, vol. 28, no. 8, pp. 962–975, 2009. [Online]. Available: http://ijr.sagepub.com/content/28/8/962.abstract
[20] Y. Nakamura and S. Savant, "Nonholonomic motion control of an autonomous underwater vehicle," in *Proc. IEEE/RSJ Int. Workshop Intell. Robots Syst. Intell. Mech. Syst.*, vol. 3, Nov. 1991, pp. 1254–1259.
[21] O. Egeland, M. Dalsmo, and O. Srdalen, "Feedback control of a nonholonomic underwater vehicle with constant desired configuration," *Int. J. Robot. Res.*, vol. 15, pp. 24–35, 1994.
[22] G. Roussos, D. V. Dimarogonas, and K. J. Kyriakopoulos, "3-D navigation and collision avoidance for nonholonomic aircraft-like vehicles," *Int. J. Adapt. Control Signal Process.*, vol. 24, no. 10, pp. 900–920, 2010.
[23] G. Ambrosino, M. Ariola, U. Ciniglio, F. Corraro, A. Pironti, and M. Virgilio, "Algorithms for 3-D UAV path generation and tracking," in *Proc. 45th IEEE Conf. Decis. Control*, Dec. 2006, pp. 5275–5280.
[24] A. Pierson and M. Schwager, "Bio-inspired non-cooperative multirobot herding," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2015, pp. 1843–1849.
[25] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton, NJ, USA: Princeton Univ. Press, 2010.
[26] H. Khalil, *Nonlinear Systems*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2002.
[27] M. Aicardi, G. Cannata, G. Casalino, and G. Indiveri, "Guidance of 3-D underwater non-holonomic vehicle via projection on holonomic solutions," in *Proc. Symp. Underwater Robot. Technol. World Autom. Congr.*, Maui, 2000, pp. 11–16.
[28] M. Aicardi, G. Casalino, and G. Indiveri, "Closed loop control of 3-D underactuated vehicles via velocity field tracking," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatron.*, vol. 1, 2001, pp. 355–360.
[29] C. Canudas de Wit and O. Sordalen, "Exponential stabilization of mobile robots with nonholonomic constraints," in *Proc. 30th IEEE Conf. Decis. Control*, vol. 1, Dec. 1991, pp. 692–697.
[30] G. Roussos, D. Dimarogonas, and K. Kyriakopoulos, "3-D navigation and collision avoidance for a non-holonomic vehicle," in *Proc. Amer. Control Conf.*, Jun. 2008, pp. 3512–3517.