Bio-Inspired Non-Cooperative Multi-Robot Herding

Alyssa Pierson[1] and Mac Schwager[2]

Abstract— This paper presents a new control strategy to control a group of dog-like robots to drive a herd of non-cooperative sheep-like agents to a goal region in the environment. The sheep-like agents, which may be biological or robotic, respond to the presence of the dog-like robots with a repelling potential field common in biological models of the behavior of herding animals. Our key insight in designing control laws for the dog-like robots is to enforce geometrical relationships that allow for the combined dynamics of the dogs and sheep to be mapped to a simple unicycle robot model. We prove convergence of a single sheep to a desired goal region using two or more dogs, and we propose a control strategy for the case of any number of sheep driven by two or more dogs. Simulations in Matlab and hardware experiments with Pololu m3pi robots demonstrate the effectiveness of our control strategy.

I. Introduction

We consider the problem of non-cooperative herding, analogous to dogs coordinating their motions to drive a herd of sheep to a goal location. In this system, the "sheep" agents naturally run away from the "dog" robots, and by designing controllers for the dogs, we can drive the sheep to some desired region in the environment. We propose a feedback control strategy for the dogs to coordinate their positions with one another to so as to partially encircle the herd. The dogs use this partial encirclement to apply pressure to the herd to move it in a desired direction, and can thereby steer the herd towards the goal. First, we introduce control strategies for two dogs controlling a single sheep, and show that under certain geometrical constraints, the dynamics of this system can be reduced to the well-known unicycle kinematic robot. Using this insight, we map a simple linear control strategy for the unicycle robot back into a nonlinear feedback control law for the two dogs. We generalize this approach to the case of an arbitrary number of dogs driving a single sheep, and finally to the general case of multiple dogs driving multiple sheep. Performance of the control strategies is demonstrated in Matlab simulations and hardware experiments with Pololu m3pi robots in a motion capture environment.

Although we will use the dog-sheep analogy throughout this paper, in general the "dogs" are robotic agents under our control, while the sheep may be biological herding animals, or other robots that behave like herding animals. The herding agents respond to the dogs with a repulsive

[1]Alyssa Pierson is with the Department of Mechanical Engineering, Boston University, Boston, MA 02215 pierson@bu.edu
[2]Mac Schwager is with the Department of Mechanical Engineering and the Division of Systems Engineering, Boston University, Boston, MA 02215 schwager@bu.edu

potential field commonly used to model the response of herding animals to perceived threats. Our control strategy could be useful in wildlife management, as well as other applications. For example, in Australia, helicopters are used to muster cattle for large-scale relocation. This dangerous profession requires pilots to fly at low altitudes and perform quick maneuvers, which results in as many as 10 deaths per year [1]. Implementing our control strategy on teams of UAVs to autonomously muster cattle may reduce the human risks and fatalities. Another application is managing wildlife populations in national parks, where it is necessary to monitor animals, as well as steer them away from environmental dangers. Our controllers may also apply to the problem of micro-manipulation of bacteria with magnetic fields [2]. Human crowds may also be controlled and directed by robots in an emergency evacuation using our control strategy. The controller we describe may also offer a plausible biological hypothesis for how real dogs herd sheep.

We consider this as a non-cooperative multi-robot problem, since the objective of the dogs is to steer the sheep, but the sheep are not actively inclined nor opposed to being steered. This scenario lies somewhere between a fully a cooperative setting, in which all robots work towards the goal, and a fully adversarial setting, in which two teams of robots work against each other toward opposite goals.

A. Related Work

There has been surprisingly limited prior work on non-cooperative robotic herding. One exception is Vaughan's pioneering work [3], [4], in which a single robot is used to herd ducks in a specially design experiment arena. More recently, Lien et. al developed a set of behavior primitives for controlling a flock with multiple shepherds [5]. In contrast to both of these, our work takes a control theoretic approach to design feedback laws for an arbitrary number of dogs to drive an arbitrary number of sheep. Other authors have formulated the problem as a dynamic pursuit-evasion game to find optimal trajectories that allow the herder to drive the sheep to some goal position [6], [7]. In this work, the herder "catches" the sheep at the goal location, whereas in our setting the herders relocate the herd without the intent to "catch" it. Furthermore, in the area of multi-agent formation control, researchers have considered driving robots into a desired formation [8], [9]. In this setting, the robots typically have linear dynamics, and have cooperative control laws that are intended to move them into a formation, while in contrast, our herd of sheep are non-cooperative, and have a nonlinear response to the dogs.

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]. These models have been applied to multi-agent systems to simulate flocking [13], cooperative group control [14], [15], [16], and interaction with collision avoidance [17].

Our work proposes a reduction from the nonlinear dog-sheep system to the well-known unicycle model for a differential drive robot [18], [19]. This introduces a nonholonomic constraint, which limits the robot to only translate in the direction of its heading. Several techniques to drive the unicycle robot to the origin without violating Brockett's Theorem [20] include optimal control [21], [22], sliding mode control [23], or Lyapunov-like functions [24]. Our chosen strategy is to control a point that is offset from the center of mass of the robot, whose dynamics then become holonomic [25]. We call this control strategy a point-offset controller. By designing feedback controllers for the point offset, we obtain nonlinear feedback controllers for the dogs, which in turn drive the sheep to a goal region in the environment.

The remainder of the paper is organized as follows. In Section II we present our mathematical formulation of the problem. Section III builds the kinematic models for the various numbers of dogs and sheep and describes the reduction to a unicycle robot. We propose a two part control strategy in Section IV. Results of simulation and experiments are given in Sections V and VI, respectively, and we give our conclusions in Section VII.

II. Problem Formulation

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

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

Here, u_j is the control input moving d_j through the environment. Our main goal is to design u_j such that the dogs drive the sheep to some goal region. We will model the sheep's repulsion from the dogs using an artificial potential field [14], which is common in robotics and in models of biological herding animals. Using the potential field $W = \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{1}{\|d_j - s_i\|}$, we obtain

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

For now we do not consider the additional forces from flocking dynamics between members of the herd, although this will be introduced later in Section III.

Also consider a user-defined goal region

$$B_\ell(g) = \{q \in \mathbb{R}^2 \mid \|q - g\| \leq \ell\}$$

centered at a goal point $g \in \mathbb{R}^2$ with a desired radius $\ell > 0$. This goal region represents the set of allowable final configurations for the sheep to occupy. Without loss of generality, we can define our coordinate frame to be centered at the goal point, so that $g = 0$. We take the goal point to be the origin through the rest 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 $B_\ell(g)$.

III. Kinematic Modeling and Reduction to Unicycle

We propose a solution to Problem 1 that is both simple and scalable to m herders. 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 will utilize in our controller design. We will first introduce terminology and basic nomenclature to describe the unicycle model, then present our herding models that reduce to the unicycle-like system.

A. Ideal Unicycle Model

Consider the nonholonomic vehicle model shown in Figure 1. For a nonholonomic vehicle, we can define a local reference frame Q relative to the global base frame B. Its forward velocity v defines the local q_x direction, as shown on the right in Figure 1. The orientation ϕ relates the heading q_x to the global b_x, and the angular velocity is defined as $w = \dot{\phi}$.
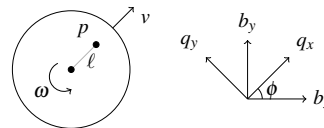

Fig. 1. Nonholonomic vehicle model

Consider also some point offset p a distance ℓ from the center of the vehicle. While the unicycle-like vehicle has nonholonomic dynamics, it turns out that p is holonomic. It can be shown that the dynamics of $\dot{p}$ can be related to v and w as [25]

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos(\phi) & \sin(\phi) \\ -\frac{\sin(\phi)}{\ell} & \frac{\cos(\phi)}{\ell} \end{bmatrix} \dot{p}. \tag{3}$$

In Section IV, we will introduce our control strategy for the point offset p, which relates to the unicycle using (3).

B. Single-Sheep Model

Instead of allowing the dogs to occupy any point in the environment, consider the case where all dogs are a fixed distance r from the sheep. We can show that when this occurs, the system dynamics reduce to a unicycle-like vehicle. First, we introduce basic concepts in our kinematic model with a single sheep and two dogs, then we generalize this to any m dogs.

1) Single Sheep with Two Dogs: We begin with the case of $n = 1$ sheep and $m = 2$ dogs, shown in Figure 2.
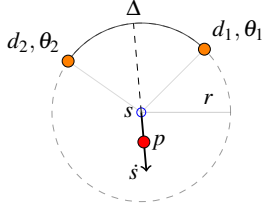

Fig. 2. Configuration of two dogs and a single sheep

Figure 2 illustrates the configuration where both dogs are located some distance r from the sheep. The position of the dogs d_j can then be written in terms of their angular orientation θ_j relative to the sheep as

$$d_j = s + r \begin{bmatrix} \cos(\theta_j) \\ \sin(\theta_j) \end{bmatrix}. \tag{4}$$

Furthermore, the dynamics of the herd introduced in (2) can be simplified as

$$\dot{s} = \frac{-1}{r^2} \begin{bmatrix} \sum \cos(\theta_j) \\ \sum \sin(\theta_j) \end{bmatrix}. \tag{5}$$

To maintain this kinematic relationship, the dynamics of the dogs in (1) must take the form

$$\dot{d}_j = \dot{s} + r\dot{\theta}_j \begin{bmatrix} -\sin(\theta_j) \\ \cos(\theta_j) \end{bmatrix}. \tag{6}$$

Similar to our unicycle model, we can define the heading ϕ as the direction of $\dot{s}$ relative to the base frame, where

$$\phi = \frac{1}{2}(\theta_1 + \theta_2) + \pi.$$

We can also see that d_1 and d_2 are always symmetric around the line formed by $\dot{s}$. Consider the angular separation between the two herders as $\Delta = \theta_2 - \theta_1$. Thus, we can re-write the angles in terms of ϕ and Δ as

$$\theta_1 = \phi + \pi - \frac{\Delta}{2}, \quad \theta_2 = \phi + \pi + \frac{\Delta}{2}. \tag{7}$$

These simplifications of the angle in (7) will allow us to distill the complex dynamics of the herders into two main state variables, ϕ and Δ, which makes it much simpler to describe the dynamics when considering m dogs.

2) Single Sheep with m Dogs: To generalize to m dogs, we will assume equal spacing of the dogs between d_1 and d_m along the desired radius, as shown in Figure 3. Thus, Δ becomes the total separation between the first dog d_1 and last dog d_m.
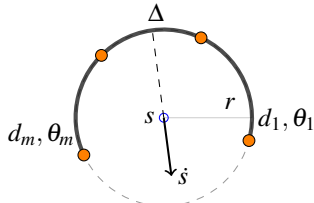

Fig. 3. Configuration of m dogs and one sheep.

The angular orientation of each dog with respect to ϕ can be expressed as

$$\theta_j = \phi + \pi + \Delta_j, \tag{8}$$

where

$$\Delta_j = \Delta \frac{(2j - m - 1)}{(2m - 2)}.$$

Substituting (8) into (5), the sheep dynamics become

$$\dot{s} = \frac{-1}{r^2} \begin{bmatrix} \sum \cos(\theta_j) \\ \sum \sin(\theta_j) \end{bmatrix} = \frac{-\sin\left(\frac{m\Delta}{2-2m}\right)}{r^2 \sin\left(\frac{\Delta}{2-2m}\right)} \begin{bmatrix} \cos(\phi) \\ \sin(\phi) \end{bmatrix} \tag{9}$$

which allows us to describe the dynamics of the sheep using only the two state variables, ϕ and Δ, despite having m dogs. Similarly, by substituting (8) in (6), the dynamics for the dogs become

$$\dot{d}_j = \dot{s} + r\left(\dot{\phi} + \dot{\Delta}_j\right) \begin{bmatrix} \sin(\phi + \Delta_j) \\ -\cos(\phi + \Delta_j) \end{bmatrix}. \tag{10}$$

By defining the orientation of the dogs in terms of ϕ and Δ along some radius in (8) and restricting the dogs' kinematics to obey (10), we can map these quantities to the angular and linear velocity of a unicycle-like vehicle.

Remark 1: Note that this model assumes the dogs are fixed on some circle of radius r relative to the herd, which limits the initial configurations of the dogs relative to the sheep. Later, we will introduce a tracking controller for the dogs that allows them to start anywhere in the environment and converge upon this configuration. We also present a radial controller in (14) to adjust the radius used by the dogs online when controlling multiple sheep.

Proposition 1: The herding dynamics in (5) and (6) can be reduced to an equivalent unicycle model with forward velocity v and orientation ϕ.

Proof: To see this mapping, note that the direction of the unicycle's velocity v and the direction of the herd's velocity $\dot{s}$ are both ϕ. As for the velocity, we can find

$$v = \|\dot{s}\| = \frac{\sin\left(\frac{m\Delta}{2-2m}\right)}{r^2 \sin\left(\frac{\Delta}{2-2m}\right)}. \tag{11}$$

Note that for (11), there are an infinite number of possible values of Δ for a given value of v. However, over the range of $\Delta = (0, 2\pi)$, this mapping is one-to-one. Thus, for a given velocity, we can find the corresponding Δ.

The remaining quantities in the mapping are $\dot{\phi}$ and $\dot{\Delta}$ in the herder's dynamics (10). We can directly map $\dot{\phi} = \omega$ from the unicycle dynamics. The dynamics for $\dot{\Delta}$ can also be found from the dynamics of $\dot{v}$. ∎

Ultimately, we will use Proposition 1 in our controller design of the system. Instead of trying to determine individual controllers for all of the dogs, we will instead design controllers for the ideal unicycle-like system. Based on the idealized system, we can find controllers for the dogs that will enforce this behavior.

C. Multi-Sheep Model

Now, consider the case of m dogs and n sheep. In the case of multiple sheep, we define the radius r from the mean of the herd, $\bar{s}$, as shown in Figure 4.
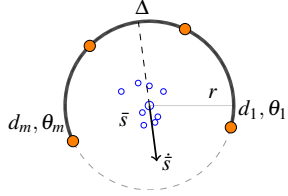

Fig. 4.   Configuration of m dogs and n sheep.

Here, the dynamics of the herd mean are

$$\dot{\bar{s}} \;=\; \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{-(d_j - s_i)}{\|d_j - s_i\|^3}. \tag{12}$$

Due to the varying nature of the extent of the herd, we also introduce an additional term $\dot{r}$ in the controller to regulate the radius r. The dynamics are then written

$$\dot{d}_j = \dot{\bar{s}} + r\dot{\theta}_j\begin{bmatrix} -\sin(\theta_j) \\ \cos(\theta_j) \end{bmatrix} + \dot{r}\begin{bmatrix} \cos(\theta_j) \\ \sin(\theta_j) \end{bmatrix}. \tag{13}$$

We design the radius controller $\dot{r}$ to maintain some desired radius r_0, as well as adjust for the standard deviation of the herd. Our proposed controller is

$$\dot{r} = (r_0 - r) + \frac{1}{n}\sum_{i=1}^{n}2(s_i - \bar{s})^T(\dot{s}_i - \dot{\bar{s}}). \tag{14}$$

where r_0 is the desired radius if the herd were a single sheep.

IV. Controller Design

Section III 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, in this section we propose a controller that drives a point-offset of the unicycle-like system to the origin. Given some desired velocity that controls the ideal system towards the goal region, we can calculate desired positions for the dogs along the circumference of the circle. We then employ a tracking controller to drive the dogs to these desired positions. For our analysis, we assume the dynamics of the ideal unicycle-like system are significantly slower than the dynamics to drive the dogs to their ideal positions.

To design the tracking controller, consider the desired position, d_j^*, for each dog. This ideal position occurs when the dogs lie on a circle of radius r around the sheep, with spacing Δ_j. Let the ideal orientation, ϕ^*, be the angle that points the herd's velocity towards the origin. To find the ideal velocity for the unicycle-like system, we will find a controller for a point-offset p from the sheep 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,

$$\dot{p} = -kp. \tag{15}$$

Plugging this into our mapping in (3), we find the ideal velocity becomes

$$v^* = [\cos(\phi)\quad \sin(\phi)](-kp).$$

Using the mapping in (11), we can then determine the desired separation Δ_j^* for the dogs. Overall, this yields the desired position of the dogs, written as

$$d_j^* = s + r\begin{bmatrix} \cos(\phi^* + \Delta_j^*) \\ -\sin(\phi^* + \Delta_j^*) \end{bmatrix}. \tag{16}$$

Finally, our tracking controller becomes

$$\dot{d}_j = -K_d(d_j^* - d_j), \tag{17}$$

Under the following mild assumption, we can analyze the performance of this control strategy.

Assumption 1: The desired dog positions d_j^* (16) evolve slowly enough compared to the speed of our dogs $\dot{d}_j$ (17) that we can approximate them as being fixed.

Under Assumption 1, the desired positions d_j^* are constant and the tracking controller (17), d_j converges exponentially to d_j^*. In practice, this allows us to the start the dogs from any point within the environment, and they will converge upon the ideal unicycle-like system. The following algorithm summarizes the steps in the controller. Note that when there are multiple sheep in the herd, the radius is continuously updated using our radial controller (14). In the case of a single sheep, the radius is always constant.

Algorithm 1 Herding Control

1: Calculate the controller for $\dot{p}$ (15)
2: Find ideal heading ϕ^* and velocity v^* (3)
3: Find Δ_j^* from v^* using (11)
4: Calculate desired dog positions d_j^* (16)
5: Calculate radial controller for $\dot{r}$ (14)
6: Calculate tracking controller for $\dot{d}_j$ (17)

Using our assumption that the tracking controller allows the dogs to converge upon their ideal positions, we can analyze the system as if it were the simple unicycle-like system. Before presenting our main proposition, we will first define some necessary quantities. The point-offset is defined relative to the herd in the direction of the herd's velocity. Let Q be the local coordinate frame of the herd, with q_x defined in the direction of the herd's velocity, shown in Figure 1, and written as

$$q_x = \begin{bmatrix} \cos(\phi) \\ \sin(\phi) \end{bmatrix} \qquad q_y \perp q_x. \tag{18}$$

Our point offset p and its derivative can then be written

$$p = s + \ell q_x \tag{19}$$
$$\dot{p} = \dot{s} + \ell\dot{\phi}q_y, \tag{20}$$

We can also write s and $\dot{s}$ in the Q frame as

$$s = s_x q_x + s_y q_y, \quad \dot{s} = v q_x$$

where v is the norm of the herd's velocity in the global reference frame. Combined, the expressions for p and $\dot{p}$ become

$$
\begin{aligned}
p &= (s_x + \ell)q_x + s_y q_y, \\
\dot{p} &= v q_x + \ell \dot{\phi} q_y.
\end{aligned}
\tag{21}
$$

Recall that we chose a simple feedback control for our point-offset in (15). Using our mapping in (3), we can write the velocity and angular velocity as

$$v = -k q_x^T p, \quad w = \dot{\phi} = \frac{-k}{\ell} q_y^T p. \tag{22}$$

We are now ready to state our main proposition on the behavior of a single sheep and m dogs.

Proposition 2: For the single sheep, m dog system described in (9) and (10), with the tracking controller in (17), the herd converges to the ball of radius l about the origin, B_ℓ.

Proof: For our unicycle-like system centered around the sheep, the point offset p is defined in (19). 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

$$V = \frac{1}{2} p^T p$$

with derivative

$$\dot{V} = p^T \dot{p}.$$

Substituting our expression for $\dot{p}$ (21), this becomes

$$\dot{V} = p^T (v q_x + \ell \dot{\phi} q_y).$$

For our unicycle system, we can then plug in the expressions for v and $\dot{\phi}$ chosen in (22), thus

$$
\begin{aligned}
\dot{V} &= p^T(-k(q_x^T p)q_x + -k(q_y^T p)q_y) \\
&= -k\|p\|^2 < 0.
\end{aligned}
$$

From Lyapunov theory [26], the equilibrium point $p^* = 0$ is asymptotically stable. Furthermore, when $p = 0$, the sheep are at most a distance ℓ away from the origin, thus proving Proposition 2. ∎

V. SIMULATIONS

The following simulations were performed in Matlab to demonstrate the capabilities of our herding algorithm. First, we present simulations illustrating the case of $n = 1$ sheep with m dogs. Despite starting from random configurations, our system converges to the dynamics of the ideal unicycle-like vehicle, and we can successfully relocate the herd to a ball around the origin. We also demonstrate our algorithm for multiple sheep, and investigate the effects of including additional inter-agent repelling and attracting forces among the sheep.

A. Herding with $n = 1$ Sheep

Our first simulation shows the case of $m = 4$ dogs and a single sheep. Figure 5 illustrates the configuration and trajectories of all agents over time. In the figure, the green x represents the goal point, and the green circle denotes the goal region $B_\ell(g)$. The blue squares denote the dogs, and black circle and x are the sheep and point offset, respectively.
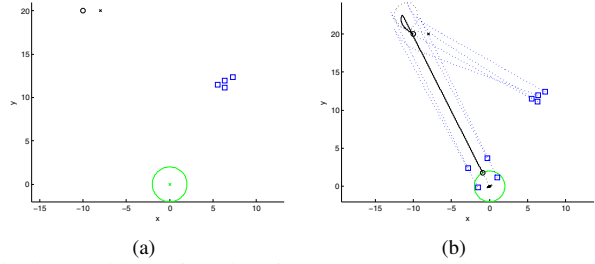

Fig. 5. (a) Initial configuration of $m = 4$ herders and single sheep moving towards goal region. (b) Trajectory of the agents and the point offset, ending within the goal region.

In Figure 5, the dogs do not start near the sheep, but 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, Figure 6 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.
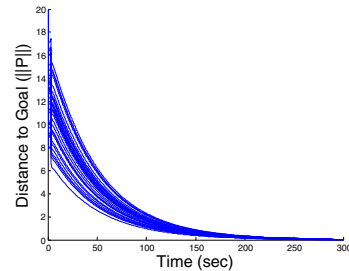

Fig. 6. Convergence of the point offset to the origin over 30 trials.

B. Herding with $n > 1$ Sheep

For the case of multiple sheep with m dogs, we add inter-agent forces between the sheep in the herd in addition to the repulsion forces the sheep experience from the dogs. For the purposes of these simulations, we use the flocking dynamics presented in Vaughan's work ([3], [4]) for inter-agent forces. Figure 7 shows two examples of controlling multiple sheep. The controllers use the herd mean $\bar{s}$, as well as the radial controller presented in (14). For the simulations presented in Figure 7, the inter-herd forces have low repulsion relative to the distances to the dogs, meaning the sheep act as a cohesive unit. With these properties, the dogs are still able to control the group to some goal region using our point-offset controller on the mean of the herd.
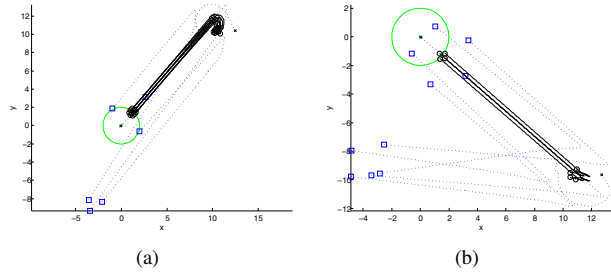
Fig. 7. (a) Trajectories of $m = 3$ dogs and $n = 10$ sheep moving towards goal region. (b) Trajectories of $m = 5$ dogs and $n = 4$ sheep moving towards goal region.

On the other hand, if we set the flocking dynamics to have higher repulsive forces between the herd members, the sheep have a greater tendency to disperse. Figure 8 shows the trajectories of two simulations where the sheep experience high repulsive forces.
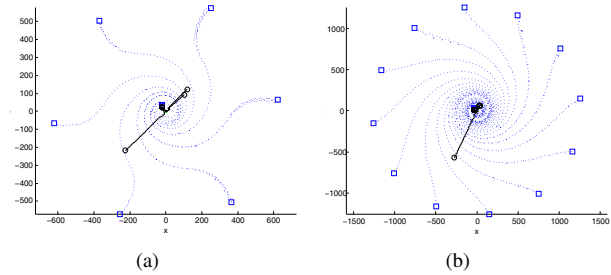

Fig. 8. Trajectories of (a) $m = 6$ dogs and single $n = 3$ sheep and (b) $m = 12$ dogs, $n = 10$ sheep with higher inter-herd repulsive forces relative to the distance to the dogs.

As seen in Figure 8, when the sheep have high repulsive forces between each other, the group spirals away from the mean. Surprisingly, the point offset from the mean still remains near the origin, as predicted by our controller. Note that the only metric in the radial controller is to adjust for the variance, but there is nothing in the current controller design to decrease the variance. As the sheep disperse under high repulsive forces, the dogs also disperse, but overall keep the mean of the herd near the origin. Future work will investigate modifications to the control strategy that gives guarantees on the final variance of the herd.

VI. Experiments

To demonstrate our algorithm, experiments were conducted at Boston University. Our lab utilizes an OptiTrack[1] system with IR cameras to track reflective markers and provide real-time localization. We use Pololu's m3pi[2] robot equipped with an mbed microcontroller and XBee[3] radio. Position data is obtained from OptiTrack and sent to Matlab, which is then used to compute control laws and send information to the m3pi robots via the XBee radio. Due to the limitations of the mbed microcontroller, computation is done on a central computer and only updated velocity information is sent to the m3pi robots.

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

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 nonholonomic vehicles with noisy, lossy actuation. In addition, the floor mats in the lab introduce a friction force on the robots, requiring them to travel at a minimum speed. These unmodeled behaviors are hard to predict or quantify in simulation. Despite these challenges, we were still able to perform successful a experiment with the m3pi robots in the loop.

For this experiment, we will use $n = 1$ m3pi "sheep", and $m = 3$ m3pi "dogs." Figure 9 illustrates the evolution of the system over time. The positions of the dogs (blue squares) and sheep (red circle) have been highlighted in each video frame. The goal region representing $B_\ell(g)$ is indicated by the green circle. Over the course of the experiment, we see the herders are able to successfully relocate the sheep to the desired goal region.

Figure 10 displays the time history of the sheep and dogs over the course of the experiment. Here, the trajectories are noisier than those seen in simulations. The additional noise comes from the unmodeled dynamics, communication delays, and a low-level nonholonomic controller within the experimental system. Despite the added noise, we still achieve our goal of relocating the sheep to some desired region. This demonstrates an inherent robustness in our feedback controllers to tolerate uncertainty in our system.
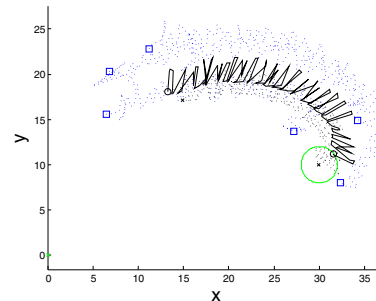

Fig. 10. Trajectories of the dogs (blue squares) and sheep (black circle).

We can also assess the performance by looking at the distance of the sheep's point-offset p from the goal, as shown in Figure 11. From this figure, we see the distance decreases over time, indicating the sheep was successfully relocated to the goal region by the herders. Although there is chatter present, it does not impact the overall performance of the controllers.

VII. Conclusion

We consider a scenario in which robotic dogs seek to control the location of a herd of non-cooperative sheep. The goal is for the herders to relocate the sheep to a region close to a goal point. Despite the highly nonlinear dynamics of the system, using the constraint that the dogs maintain some radius around the herd allows us to map these dynamics to unicycle-like dynamics, for which a simple feedback controller can be formulated. Unlike previous work in herding,

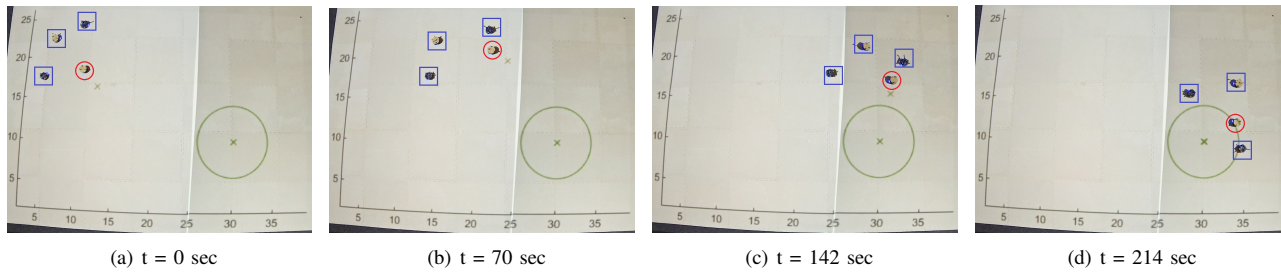(a) t = 0 sec   (b) t = 70 sec   (c) t = 142 sec   (d) t = 214 sec

Fig. 9. Images from the experimental video, illustrating the herding of $n = 1$ robot sheep (red circle) by $m = 3$ robot herders (blue squares).
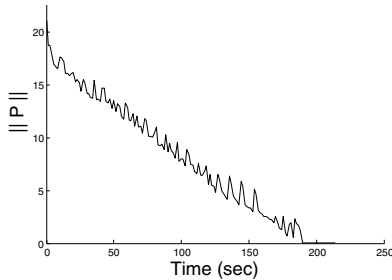

Fig. 11. Distance of the sheep's point offset p to the goal.

this is done with a single continuous control law and does not rely on switching or heuristic behaviors. For a single sheep with multiple dogs, we are able to prove with a Lyapunov-like proof that the sheep converge asymptotically to the goal region. We also propose a control strategy for the general case of multiple sheep and multiple dogs. The simulations and hardware experiments demonstrate the performance of these control strategies.

In the future, we plan to analyze the convergence of the multi-sheep, multi-dog case. Other areas to investigate would be more complex problems for the herders to solve. For example, the herders may have to move the sheep along some trajectory or path in the environment, such as navigating through a maze or maneuvering around obstacles. Beyond the relocation problems, there are also the herding problems of consolidating a sparse group of sheep, culling members from a group, or protecting the group from an external threat.

REFERENCES

[1] M. Lane. (2011, February) Helicopter cowboys of australia's outback. http://www.bbc.co.uk/news/world-asia-pacific-12408888. [Online]. Available: http://www.bbc.co.uk/news/world-asia-pacific-12408888

[2] A. Becker, C. Onyuksel, T. Bretl, and J. McLurkin, "Controlling many differential-drive robots with uniform control inputs," *The International Journal of Robotics Research*, 2014.

[3] R. Vaughan, "Experiments in automatic flock control," Ph.D. dissertation, University of Oxford, 1999.

[4] R. Vaughan, N. Sumpter, J. Henderson, A. Frost, and S. Cameron, "Experiments in automatic flock control," *Robotics and Autonomous Systems*, vol. 31, no. 1, pp. 109–117, 2000.

[5] J.-M. Lien, S. Rodriguez, J. Malric, and N. Amato, "Shepherding behaviors with multiple shepherds," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, April 2005, pp. 3402–3407.

[6] S. A. Shedied, "Optimal control for a two-player dynamic pursuit evasion game: The herding problem," Ph.D. dissertation, Virginia Polytechnic Institute and State University, 2002, aAI3040288.

[7] Z. Lu, "Cooperative optimal path planning for herding problems," Master's thesis, Texas A&M University, December 2006.

[8] M. Egerstedt and X. Hu, "Formation constrained multi-agent control," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 4, 2001, pp. 3961–3966 vol.4.

[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, J. Hespanha and A. Tiwari, Eds. Springer Berlin Heidelberg, 2006, vol. 3927, 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. ON SYSTEMS, MAN AND CYBERNETICS*, vol. 34, no. 1, pp. 539–557, 2004.

[12] J. Toner and Y. Tu, "Flocks, herds, and schools: A quantitative theory of flocking," *Phys. Rev. E*, vol. 58, no. 4, p. 4828, 1998.

[13] 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

[14] 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 5*, H. Asama, T. Arai, T. Fukuda, and T. Hasegawa, Eds. Springer Japan, 2002, pp. 299–308.

[15] H. Tanner, A. Jadbabaie, and G. Pappas, "Stable flocking of mobile agents, part i: fixed topology," in *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 2, Dec 2003, pp. 2010–2015 Vol.2.

[16] ——, "Flocking in fixed and switching networks," *Automatic Control, IEEE Transactions on*, vol. 52, no. 5, pp. 863–868, May 2007.

[17] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, pp. 401–420, 2006.

[18] D. DeVon and T. Bretl, "Kinematic and dynamic control of a wheeled mobile robot," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, Oct 2007, pp. 4065–4070.

[19] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 1994.

[20] R. W. Brockett, *Asymptotic Stability and Feedback Stabilization*. Defense Technical Information Center, 1983.

[21] J. Cameron and W. Book, "Optimal path planning for the motion of a wheel," in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, May 1994, pp. 1574–1580 vol.2.

[22] J. Laumond, S. Sekhavat, and F. Lamiraux, "Guidelines in nonholonomic motion planning for mobile robots," in *Robot Motion Planning and Control*, ser. Lecture Notes in Control and Information Sciences, J.-P. Laumond, Ed. Springer Berlin Heidelberg, 1998, vol. 229, pp. 1–53.

[23] D. Chwa, "Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates," *Control Systems Technology, IEEE Transactions on*, vol. 12, no. 4, pp. 637–644, July 2004.

[24] M. Aicardi, G. Casalino, A. Bicchi, and A. Balestrino, "Closed loop steering of unicycle like vehicles via lyapunov techniques," *Robotics Automation Magazine, IEEE*, vol. 2, no. 1, pp. 27–35, Mar 1995.

[25] N. Michael and V. Kumar, "Planning and control of ensembles of robots with non-holonomic constraints," *The International Journal of Robotics Research*, vol. 28, no. 8, pp. 962–975, 2009. [Online]. Available: http://ijr.sagepub.com/content/28/8/962.abstract

[26] H. Khalil, *Nonlinear Systems*. Prentice Hall PTR, 2002.