Multi Robot Control for Non-Cooperative Herding using Control Barrier Functions

Submitted in partial fulfillment
of requirements for the degree
Master of Science
in
Robotics

by
Nishant Mohanty

CMU-RI-TR-23-45

Thesis Committee:
Katia Sycara, *chair*
Maxim Likhachev
Swaminathan Gurumurthy

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

ABSTRACT

Control Barrier Functions (CBFs) have emerged as a powerful theoretical tool for designing controllers with provable safety guarantees. This work presents a novel methodology that leverages CBFs to synthesize controllers for the problem of protecting a high-value unit from inadvertent attack by a group of non-cooperative agents using defending robots. Specifically, we develop a control strategy for the defending agents that we call "dog robots" to prevent the non-cooperative agents, i.e., a flock of "sheep agents" from breaching a protected zone. The sheep agents have no knowledge about the presence of the high-value unit and follow flocking dynamics to reach their goal. We take recourse to CBFs to pose this problem and exploit the interaction dynamics between the sheep and dogs to find dogs' velocities that result in the sheep getting repelled from the zone.

Furthermore, we address a crucial limitation of existing CBF-based controllers that usually fail to respect the control input's limits, resulting in undesirable outcomes. Imposing these limits by capping the control input could compromise the safety guarantees offered by CBFs, and incorporating them as constraints in the optimization process often leads to infeasibility. To overcome these challenges, we propose a two-step cascaded optimization method. We parameterize the CBF and then compute their values that ensure that CBFs yield solutions, if they exist, within the control limits without compromising safety or feasibility. The performance and efficacy of our cascaded control approach are thoroughly evaluated through extensive simulations in the aforementioned multi-robot scenarios. We also experimentally demonstrate all the above algorithms using Khepera IV robots in a laboratory environment. Overall, this work contributes to advancing multi-robot coordination by providing a framework built on Control Barrier Functions, offering provable safety guarantees while addressing crucial challenges in controller design.

Keywords: Control Barrier Functions, Multi-Robot Coordination, Control Input Limits, Cascaded Optimization, Non-Cooperative Herding, Heterogeneous Swarm Control.

ACKNOWLEDGMENTS

I thank my advisor, Professor Katia Sycara, for her incredible support. Her guidance and insight have been indispensable in this project and my growth as a researcher.

Further, I would like to thank the other members of my committee: Professor Maxim Likhachev and Swaminathan Gurumurthy. Their feedback and varied perspectives on this work, as well as insights on how it connects to different fields has been invaluable.

I am incredibly grateful to Professor Wenhao Luo and Jaskaran Sigh Grover for their continued mentorship and support throughout my academic journey.

I also thank my Advanced Agent - Robotics Technology Lab colleagues for engaging and thought-provoking conversations, asking hard questions about my ideas, and creating an enriching work environment.

I thank my incredible friends and family for their unconditional support. And finally, to my mother and father for giving me the courage and strength to do any of this.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

The rapid advancements in multi-robot systems (MRS) have led to their widespread deployment in real-world scenarios to address various practical challenges. These systems offer advantages such as fault resilience and distributed information gathering, making them indispensable for a wide range of applications [1, 2, 3]. Within the domain of MRS, significant efforts have been dedicated to developing control algorithms that enable multiple robots to collaborate effectively to achieve collective objectives. These algorithms, characterized by local interactions, collision-free motions, and emergent collective behaviors, have proven to be highly successful in various team-level tasks [4, 5, 6].

In this thesis, we delve into an intriguing aspect of multi-robot systems, exploring the perspective of an external agent overseeing a group of robots engaged in a task. Specifically, we focus on scenarios where one group of robots serves as defenders, entrusted with safeguarding a high-value unit from an adversarial group referred to as the "sheep agents." The defenders face a formidable challenge as they cannot exert direct control over the adversarial agents. Instead, they must tactically leverage their interaction dynamics to influence the sheep agents' behavior and prevent them from breaching a designated protected zone.

The motivation behind this research stems from the need to address security and safety concerns in complex environments where high-value units require protection from potential threats posed by adversarial groups. Achieving successful defense in such scenarios is challenging and necessitates innovative control strategies. Key challenges to be addressed include the non-collocated nature of control, where defenders lack direct control over the adversarial agents, and the underactuated control problem that arises when the number of defenders is insufficient compared to the adversarial agents.

1.2 Contribution

This thesis aims to contribute novel solutions to the aforementioned challenges by proposing a control strategy based on control barrier functions to guide the defenders, referred to as "dog robots," in preventing the adversarial "sheep agents" from breaching the protected zone. The core of our approach involves formulating the control problem as a Quadratic Programming (QP) task to determine the optimal velocities for all dog robots, thereby orchestrating their collective actions strategically.

One notable feature of our proposed approach is its versatility, allowing for the simultaneous inclusion of multiple protected zones by augmenting constraints on the dog robots' velocities. This enables the defenders to safeguard several high-value units efficiently.

To establish the feasibility and effectiveness of our strategy, we provide theoretical proofs for the one dog/one sheep case, demonstrating the provably correct velocities for this scenario. Additionally, we present empirical results from extensive simulations, showcasing the success of our approach in scenarios involving multiple dog robots and numerous sheep agents. Furthermore, we validate our control strategy through real-world experiments, demonstrating the ability to protect multiple zones with multiple dog robots.

1.3 Summary

In summary, this thesis focuses on controlling a group of defending robots in a multi-robot system to safeguard high-value units from adversarial agents. We propose a control barrier approach, formulated as a Quadratic Programming problem, to guide the defenders in influencing the behavior of the adversarial agents. Our work addresses the challenges of non-collocated control and under-actuation, offering a versatile and effective defense strategy. Through extensive simulations and real-world experiments, we demonstrate the efficacy of our approach in preventing the breach of multiple protected zones, providing valuable insights for real-world applications of multi-robot systems in defense scenarios.

Chapter 2

Background

2.1 Control Barrier Functions

Control barrier functions (CBFs) have emerged as a powerful tool in the field of control theory, particularly in the context of safety-critical control systems. CBFs provide a systematic approach for enforcing safety constraints on the states of a dynamical system, ensuring that it remains within a predefined safe set while tracking desired trajectories. The concept of CBFs was initially introduced by Ames et al. [7], and since then, they have gained significant attention in various research areas, including robotics, autonomous vehicles, and cyber-physical systems.

One of the fundamental aspects of CBFs is that they enable the synthesis of control policies that guarantee forward invariance, i.e., the system's trajectories remain within a safe set for all time. This property is crucial in safety-critical applications where system states must be prevented from entering hazardous regions. The primary idea behind CBFs is to design control laws that leverage Lyapunov-like functions to ensure the satisfaction of safety constraints.

The application of CBFs in robotics has been extensively investigated. For instance, in [8], the author demonstrated the use of CBFs in the context of human-robot interaction, ensuring safe collaboration between humans and robots. CBFs have also been applied to collision avoidance in multi-robot systems, as shown in [9]. Moreover, the combination of CBFs with other control techniques has led to promising results. In [10], the author used Model Predictive Control combined with Control Barrier Function (RMPC-CBF) for a nonholonomic robot with obstacle avoidance

Despite their effectiveness, CBFs also pose some challenges. One such challenge is the computation of CBFs for complex systems, which may involve high-dimensional state spaces and non-trivial constraints. Several works have focused on addressing these challenges, such as using optimization techniques and leveraging machine learning methods as shown in [11].

In conclusion, control barrier functions have become a valuable tool in safety-critical control systems, offering a systematic approach to enforce safety constraints and guarantee forward invariance. Their application spans various fields, including robotics, autonomous vehicles, and cyber-physical systems. Despite some challenges, the versatility and efficacy of CBFs make them an increasingly popular choice for ensuring safety and stability in complex control systems. As research in this area continues to evolve, it is expected that CBFs will play an essential role in shaping the future of safe and reliable control systems.

2.2 Multi Robot Control: Non Cooperative herding

In the field of multi-robot systems (MRS), the concept of influencing group behavior extends beyond adversarial contexts and includes scenarios like shepherding behaviors. In the shepherding problem, external agents, known as shepherds, aim to control the motion of another group of agents, referred to as the flock, by exerting repulsive forces on them [12, 13]. Notably, the Robot Sheepdog Project [14, 15] successfully demonstrated the practical implementation of robotic herding, where autonomous wheeled mobile robots acted as shepherds to gather a flock of ducks and maneuver them towards a specified goal position.

Previous works have explored the problem of noncooperative shepherding using robots [13, 16, 17, 18, 19, 20]. In these approaches, the flock agents are not adversarial but exhibit repulsion from the robots. The works focus on developing feedback controllers to steer the flock agents towards a designated region. However, a common limitation among these studies is the omission of the self-motivated dynamics of the flock agents, neglecting their nominal behavior without any robot interaction. As a result, the flock agents' motions are solely driven by repulsion from the robots, overlooking potential influences of internal goals or natural behavior patterns. Additionally, these approaches often involve handcrafted solutions for generating specific flock behavior, such as herding the flock to a given location. Furthermore, scalability considerations regarding the number of agents are often not thoroughly explored.

In contrast, this thesis addresses the shepherding problem from a different perspective. Instead of assuming non-adversarial behavior, we consider scenarios where the sheep agents may or may not exhibit adversarial intentions. Adversarial sheep aim to breach the protected zone, while non-adversarial sheep have different objectives. We refer to the defending robots as "dog robots," and our control approach remains invariant regardless of the adversarial nature of the sheep. Our methodology considers various factors in the sheep's behavior, including cohesion, inter-sheep dynamics, attraction to their goal locations, and repulsion experienced from the dog robots. We adopt control barrier functions to formulate constraints on the dog robots' velocities, which allows for flexible representation of the sheep's dynamics as symbolic functions. This versatility enables the sheep to experience various attractive or repulsive forces. Moreover, our proposed approach demonstrates excellent scalability, as demonstrated through a Monte Carlo study with numerous sheep compared to the number of dog robots. The high success rates in this study provide empirical evidence of the approach's effectiveness in handling large-scale scenarios.

2.3 Summary

In summary, while prior works have primarily focused on noncooperative shepherding scenarios with fixed flock behavior and limited scalability considerations, our thesis addresses the shepherding problem with a broader perspective. We account for both adversarial and non-adversarial sheep, provide a safe control synthesis approach for the dog robots, and employ control barrier functions for flexible and scalable control strategies. By considering the self-motivated dynamics of the sheep and enabling effective multi-group interaction within MRS, our work contributes to a deeper understanding of the shepherding problem and its applications in various real-world scenarios.

Chapter 3

Single Agent Herding Using CBF

In this chapter, we discuss our methodology to derive the control barrier function and its constraints in the control space. We demonstrate it on a simple single-agent herding problem before moving on to the multi-agent scenario. First, we define the single agent herding problem. Following that, we discuss the derivation of CBFs followed by their application method.

3.1 Problem Statement

Suppose there is one "sheep" agent (the herd) and one dog agent (the defenders). We assume that the sheep dynamics comprise *i.e.* moving towards a common goal while staying close enough to each other and getting repelled by the dogs. Given these dynamics, it may end up breaching a high-value unit *i.e.* the protected zone while en route to its goal. Therefore, the objective of the dog robot is to steer it away from this zone.

To pose this requirement mathematically, denote the position of the sheep as $\boldsymbol{x}_S \in \mathbb{R}^2$ and the position of the dog as $\boldsymbol{x}_D \in \mathbb{R}^2$. We assume both sheep and dog have single-integrator dynamics *i.e.* they are velocity controlled. For the sheep, we have the dynamics as:

$$\begin{aligned}\dot{\boldsymbol{x}}_S &= \boldsymbol{u}_S \\ &= k_G\left(\boldsymbol{x}_G - \boldsymbol{x}_S\right) + k_D \frac{\boldsymbol{x}_S - \boldsymbol{x}_D}{\left\|\boldsymbol{x}_S - \boldsymbol{x}_D\right\|^3} \\ &:= \boldsymbol{f}(\boldsymbol{x}_S, \boldsymbol{x}_D)\end{aligned} \tag{3.1}$$

Here the first term represents attraction to the goal (x_G), and the second term represents repulsion from the dog robot. The attraction to the goal represents the self-motivated part of the dynamics of the sheep agents. Here k_G, k_D are proportional gains corresponding to forces in the dynamics. For the dog we have:

$$\dot{\boldsymbol{x}}_D = \boldsymbol{u}_D \tag{3.2}$$

We denote the protected zone as $\mathcal{P} \subset \mathbb{R}^2$ and here we assume that it is a disc centered at $\boldsymbol{x}_p$ and radius R_p:

$$\mathcal{P} := \{\boldsymbol{x} \in \mathbb{R}^2 | \; \|\boldsymbol{x} - \boldsymbol{x}_p\| \leq R_p\} \tag{3.3}$$

We denote the set excluding the protected zone as $\mathcal{P}^c := \mathbb{R}^2 \backslash \mathcal{P}$. The sheep are assumed to have no knowledge about the presence of $\mathcal{P}$. The dog robots need to ensure that the sheep remain in $\mathcal{P}^c$ if they are initially in $\mathcal{P}^c$ by finding suitable control inputs $\{\boldsymbol{u}_{D_1}, \cdots, \boldsymbol{u}_{D_m}\}$. We make the following assumption on the dog's knowledge before posing the problem:

Assumption 1. *The dog robots have knowledge about the sheep's dynamics i.e.* (3.1) *and can measure the sheep's positions accurately.*

This is not a stringent assumption because if the dynamics are unknown, the dog robots can learn the dynamics online using system identification algorithms like [21, 22] and use certainty equivalence to design the controllers. Based on these definitions, we can pose the dog robots' problem as follows:

Definition 1. *Assuming that the initial positions of the sheep $\boldsymbol{x}_S(0) \in \mathcal{P}^c$, the dog robot's problem is to synthesize controls $\boldsymbol{u}_D$ such that $\boldsymbol{x}_S(t) \in \mathcal{P}^c \; \forall t \geq 0$. If $\boldsymbol{x}_S(0) \notin \mathcal{P}^c$, the dog robots' problem is to synthesize controls $\boldsymbol{u}_D$ such that $\boldsymbol{x}_S(t) \rightsquigarrow \mathcal{P}^c$ in finite time.*

Additionally, we require that the dog robots never collide with the sheep. In the next section, we show how to address this problem using control barrier functions.

3.2 Controller Design using CBF

In this section, we discuss our controller design to solve the problem of defending the protected zone, as stated before. Given the protected zone as defined (3.3), we define a safety index $h(\cdot) : \mathbb{R}^2 \longrightarrow \mathbb{R}$ as follows:

$$h = \| \, \boldsymbol{x}_{S_i} - \boldsymbol{x}_p \, \|^2 - R_p^2 \tag{3.4}$$

By construction, $h \geq 0 \; \forall \boldsymbol{x}_S \in \mathcal{P}^c$ *i.e.* non-negative whenever i is on the boundary or outside the protected zone. Thus, assuming that at $t = 0$, $h(\boldsymbol{x}_S(0)) \geq 0$, we require $h(\boldsymbol{x}_S(t)) \geq 0 \; \forall t \geq 0$. Treating $h(\cdot)$ as a control barrier function [7], this can be achieved if the derivative of $h(\cdot)$ satisfies the following constraint:

$$\begin{aligned} &\dot{h}(\boldsymbol{x}_S, \boldsymbol{x}_D) + p_1 h(\boldsymbol{x}_S) \geq 0 \\ \implies &2(\boldsymbol{x}_S - \boldsymbol{x}_p)^T \dot{\boldsymbol{x}}_S + p_1 h(\boldsymbol{x}_S) \geq 0 \\ \implies &2(\boldsymbol{x}_S - \boldsymbol{x}_p)^T \boldsymbol{f}_i + p_1 h(\boldsymbol{x}_S) \geq 0. \end{aligned} \tag{3.5}$$

Define $\boldsymbol{x} = (\boldsymbol{x}_S, \boldsymbol{x}_D)$, we rewrite this as

$$2(\boldsymbol{x}_S - \boldsymbol{x}_p)^T \boldsymbol{f}(\boldsymbol{x}) + p_1 h(\boldsymbol{x}_S) \geq 0. \tag{3.6}$$

Here p_1 is a design parameter that we choose to ensure that

$$p_1 > 0 \quad \text{and} \quad p_1 > -\frac{\dot{h}(\boldsymbol{x}(0))}{h(\boldsymbol{x}(0))}. \tag{3.7}$$

The first condition on p_1 requires that the pole is real and negative. The second is to ensure forward invariance and depends on the initial positions $\boldsymbol{x}(0)$ of all the sheep and dogs relative to the protected zone. Now while (4.5) depends on the positions of the sheep and dog, it is the velocity of the dog that is directly controllable, not its position (4.3). Since $\boldsymbol{u}_D$ does not show up in (4.5), we define the LHS of (4.5) as another function $v(\cdot) : \mathbb{R}^{2(1+1)} \longrightarrow \mathbb{R}$:

$$v = \dot{h} + p_1 h. \tag{3.8}$$

Like before, in order to ensure $v \geq 0$ is always maintained, its derivative needs to satisfy

$$\dot{v}(\boldsymbol{x}) + p_2 v(\boldsymbol{x}) \geq 0. \tag{3.9}$$

Here p_2 is another design parameter which we choose p_2 to ensure that the following is satisfied at $t = 0$

$$p_2 > 0 \quad \text{and} \quad p_2 > -\frac{\ddot{h}(\boldsymbol{x}(0)) + p_1 \dot{h}(\boldsymbol{x}(0))}{\dot{h}(\boldsymbol{x}(0)) + p_1 h(\boldsymbol{x}(0))} \tag{3.10}$$

Using (4.7) in (4.8), we get:

$$\begin{aligned} &\ddot{h}(\boldsymbol{x}) + (p_1 + p_2)\dot{h}(\boldsymbol{x}) + p_1 p_2 h(\boldsymbol{x}) \geq 0 \\ \implies &\ddot{h}(\boldsymbol{x}) + \alpha \dot{h}(\boldsymbol{x}) + \beta h(\boldsymbol{x}) \geq 0. \end{aligned} \tag{3.11}$$

where we have defined $\alpha := p_1 + p_2$ and $\beta := p_1 p_2$. α and β are chosen by ensuring that p_1 and p_2 satisfy the requirements in (3.7) and (3.10). The time derivatives of $h(\cdot)$ required in (4.9) is

$$\dot{h}(\boldsymbol{x}) = 2(\boldsymbol{x}_S - \boldsymbol{x}_P)^T \dot{\boldsymbol{x}}_S = 2(\boldsymbol{x}_S - \boldsymbol{x}_P)^T \boldsymbol{f}(\boldsymbol{x}_S, \boldsymbol{x}_D) \tag{3.12}$$

$$\begin{aligned}\ddot{h}(\boldsymbol{x}) &= 2\dot{\boldsymbol{x}}_S^T\dot{\boldsymbol{x}}_S + 2(\boldsymbol{x}_S - \boldsymbol{x}_P)^T\left(\mathbb{J}^S\dot{\boldsymbol{x}}_S + \mathbb{J}^D\boldsymbol{u}_D\right)\\ &= 2\boldsymbol{f}_i^T\boldsymbol{f}_i + 2(\boldsymbol{x}_S - \boldsymbol{x}_P)^T\left(\mathbb{J}^S\boldsymbol{f}_i + \mathbb{J}^D\boldsymbol{u}_D\right) \end{aligned} \tag{3.13}$$

where $\mathbb{J}^S$ and $\mathbb{J}^D$ are

$$\mathbb{J}^S \coloneqq \nabla_{\boldsymbol{x}_S}\boldsymbol{f}(\boldsymbol{x}_S, \boldsymbol{x}_D); \quad \mathbb{J}^D \coloneqq \nabla_{\boldsymbol{x}_D}\boldsymbol{f}(\boldsymbol{x}_S, \boldsymbol{x}_D)$$

Note here that $\ddot{h}(\boldsymbol{x})$ contains the velocities of the dog robot as we wanted. Using (4.10) and (4.11) in (4.9), we get the following linear constraints on dog velocity to ensure that the sheep stays outside the protected zone $\mathcal{P}$:

$$A^H\boldsymbol{u}_D \leq b^H, \tag{3.14}$$

where,

$$\begin{aligned} A^H &\coloneqq (\boldsymbol{x}_P - \boldsymbol{x}_S)^T\mathbb{J}^D\\ b^H &\coloneqq \boldsymbol{f}^T\boldsymbol{f} + (\boldsymbol{x}_S - \boldsymbol{x}_P)^T(\mathbb{J}^S\boldsymbol{f} + \alpha\boldsymbol{f}) + \beta\frac{h}{2} \end{aligned}$$

Given these constraints on the dog robot's velocity, we compute the final velocity by solving a QP that searches for the min-norm velocities while satisfying these constraints.

$$\begin{aligned} \boldsymbol{u}_D^* &= \underset{\boldsymbol{u}_D}{\arg\min}\,\|\boldsymbol{u}_D\|^2\\ \text{subject to} \quad & \mathcal{A}^H\boldsymbol{u}_D \leq \boldsymbol{b}^H \end{aligned} \tag{3.15}$$

As long as the velocity of the dog satisfies this constraint, we can guarantee that it will herd the sheep away from the protected zone. Therefore the solution of this QP guarantees herding. The proof of this guarantee is provided in the following section.

3.3 Proof of CBF guarantee for herding

Theorem 1. *If there is one dog and one sheep, then* (3.15) *always has a solution.*

Proof. Given the sheep dynamics and CBF-based control for dog robot, the only case when (3.15) does not have a solution is when the herding constraint $A^H\boldsymbol{u}_D \leq b^H$ is infeasible. This can occur

- either when $A^H = \mathbf{0}$ and $b^H < 0$ (possibility 1)
- or when $b^H = -\infty$ (possibility 2).

For this case A^H is:

$$A^H = (\boldsymbol{x}_P - \boldsymbol{x}_S)^T \mathbb{J}^D_{11} \tag{3.16}$$

Thus, if $\mathbb{J}^D$ is non-singular, $(\boldsymbol{x}_P - \boldsymbol{x}_S)^T \mathbb{J}^D \neq \mathbf{0}$. From our calculations, we find that the determinant of $\mathbb{J}^D$ is

$$det(\mathbb{J}^D) = \frac{-2k_D^2}{\|\boldsymbol{x}_D - \boldsymbol{x}_S\|^3} \tag{3.17}$$

As long as the distance between the dog and the sheep is finite, $det(\mathbb{J}^D)$ is always non-zero. Thus, there exists no null space for the jacobian matrix $\mathbb{J}^D$. This implies $A^H \neq \mathbf{0}\ \forall \boldsymbol{x}_S \in \mathbb{R}^n, \boldsymbol{x}_D \in \mathbb{R}^2$. This rules out possibility 1 for infeasibility. For possibility 2, we need to examine when does $b^H \longrightarrow -\infty$. The expression for b^H is:

$$b^H = \boldsymbol{f}^T\boldsymbol{f} + (\boldsymbol{x}_S - \boldsymbol{x}_P)^T \mathbb{J}^S \boldsymbol{f} + \alpha(\boldsymbol{x}_S - \boldsymbol{x}_P)^T \boldsymbol{f} + \beta\frac{h}{2}$$

We want to find the worst case lower bound of b^H. Here $\boldsymbol{f}^T\boldsymbol{f} \geq 0$ always. We assume that at the current time step, the sheep is outside the $\mathcal{P}$, this ensures $\beta\frac{h}{2} \geq 0$.

Assumption 2. *Assume* $\|\boldsymbol{x}_S - \boldsymbol{x}_G\| \leq M_1, \|\boldsymbol{x}_S - \boldsymbol{x}_P\| \leq M_2$ *and* $\|\boldsymbol{x}_S - \boldsymbol{x}_D\| \geq M_3\ \forall t$.

With these assumptions, we can lower bound b^H as follows:

$$\begin{aligned} b^H &\geq (\boldsymbol{x}_S - \boldsymbol{x}_P)^T \mathbb{J}^S \boldsymbol{f} + \alpha(\boldsymbol{x}_S - \boldsymbol{x}_P)^T \boldsymbol{f} \\ &\geq -(\sigma_{max}(\mathbb{J}) + \alpha)\,\|\boldsymbol{f}\|\,\|\boldsymbol{x}_S - \boldsymbol{x}_P\| \\ &\geq -(\sigma_F(\mathbb{J}) + \alpha)\,\|\boldsymbol{f}\|\,\|\boldsymbol{x}_S - \boldsymbol{x}_P\| \end{aligned} \tag{3.18}$$

Here $\|\boldsymbol{f}\| \leq k_G\|\boldsymbol{x}_S - \boldsymbol{x}_G\| + \frac{k_D}{\|\boldsymbol{x}_S - \boldsymbol{x}_D\|^2}$ using triangle inequality on (3.1). This gives $\|\boldsymbol{x}_S - \boldsymbol{x}_P\|\,\|\boldsymbol{f}\| \leq k_G M_1 M_2 + \frac{k_D M_2}{M_3^2}$. We can show that $\sigma_F(\mathbb{J}_{11}) \leq \lambda_M := \sqrt{2k_G^2 + 5\frac{k_D^2}{M_3^6} + \frac{2k_G k_D}{M_3^3}}$. Thus, using this, we obtain the following lower bound for b^H

$$b^H \geq -(\lambda_M + \alpha)\left(k_G M_1 M_2 + \frac{k_D M_2}{M_3^2}\right) \tag{3.19}$$

This shows that b^H is lower bounded and thus does not reach $-\infty$. Hence possibility 2 is also ruled out. Thus, (3.15) is always feasible. □

3.4 Cascaded CBF Design

While CBF provides us with a provable method of non-cooperative herding, there is an inherent issue with this type of controller (eqn. 3.15). That is, there is nothing preventing it from outputting a velocity that is beyond the physical limits of the robots being used. In order to address this issue, we consider a two-stage optimization. First, we consider a norm velocity constraint on the dog robots of the form:

$$\|\boldsymbol{u}_D\|_2 \leq v_{max} \tag{3.20}$$

Then we consider the parameterized form of the CBF. That is the herding equation before decomposing the CBF into a linear constraint on the velocity of the dog robot, as in eqn. 4.9:

$$\ddot{h}(\boldsymbol{x}) + (p_1 + p_2)\dot{h}(\boldsymbol{x}) + p_1 p_2 h(\boldsymbol{x}) \geq 0$$

Here p_1 and p_2 are tuning parameters that we handpicked within their constraints to get the velocities in eqn. 3.15. In this subsection, our aim is to choose p_1 and p_2 such that the resulting herding constraint has an overlapping region with the velocity constraint mentioned in eqn. 3.20. This region is called a feasible region. To have a simplified calculation, we first represent the herding constraint as $A^H u_D \leq b^H$. Following that, there are two cases to consider.
Case 1: when $u_D = [0, 0]$ is a possible solution of herding constraint. i.e., $b \geq 0$. In this case, there always exists a feasible region.
Case 2: when $u_D = [0, 0]$ is not a possible solution in herding constraint. i.e, $b \leq 0$. In this case, a feasible region will only exist only when the distance of the line given by the herding constraint ($A^H u_D - b^H = 0$) is less than the value corresponding to the maximum possible velocity, i.e.

$$\frac{\left|-b^H\right|}{\left\|A^H\right\|_2} \leq v_{max} \implies \left\|A^H\right\|_2 v_{max} - \left|-b^H\right| \geq 0$$

as we are considering the case where $-b \geq 0$, the above equation reduces to

$$\left\|A^H\right\|_2 v_{max} + b^H \geq 0 \tag{3.21}$$

Note that this equation also covers the case 1 scenario as well. That is when $b \geq 0$, the LHS of eqn. 3.21 is always positive. Thus we can say that as long as the designed herding constraint of the form $A^H u_D \leq b^H$ has A^H and b^H satisfying eqn. 3.21 as well, the resulting velocity output of eqn. 3.15 will always stay within the velocity limits.

Now to reiterate our requirement, our aim is to choose p_1 and p_2 such that A^H and b^H satisfy eqn. 3.21. In order to achieve that, we expand the above constraint using A^H and b^H as:

$$A^H := (\boldsymbol{x}_P - \boldsymbol{x}_S)^T \mathbb{J}^D$$
$$b^H := \boldsymbol{f}^T\boldsymbol{f} + (\boldsymbol{x}_S - \boldsymbol{x}_P)^T(\mathbb{J}^S\boldsymbol{f} + \alpha\boldsymbol{f}) + \beta\frac{h}{2}$$

Using this the constraint (eqn. 3.21) reduces to:

$$\left\|(\boldsymbol{x}_P - \boldsymbol{x}_S)^T \mathbb{J}^D\right\| v_{max} + \boldsymbol{f}^T\boldsymbol{f} + (\boldsymbol{x}_S - \boldsymbol{x}_P)^T(\mathbb{J}^S\boldsymbol{f} + \alpha\boldsymbol{f}) + \beta\frac{h}{2} \geq 0 \tag{3.22}$$

We can rewrite it as:

$$(\boldsymbol{x}_S - \boldsymbol{x}_P)^T\alpha\boldsymbol{f} + \beta\frac{h}{2} \geq -\left\|(\boldsymbol{x}_P - \boldsymbol{x}_S)^T \mathbb{J}^D\right\| v_{max} - \boldsymbol{f}^T\boldsymbol{f} - (\boldsymbol{x}_S - \boldsymbol{x}_P)^T\mathbb{J}^S\boldsymbol{f} \tag{3.23}$$

By define the RHS of this equation as b^C, α as $p_1 + p_2$ and β as p_1p_2, we finally get the constraint of p_1 and p_2 as:

$$(\boldsymbol{x}_S - \boldsymbol{x}_P)^T\boldsymbol{f}(p_1 + p_2) + p_1p_2\frac{h}{2} \geq b^C \tag{3.24}$$

As long as this nonlinear constraint on the constants p_1 and p_2 is met, we can guarantee that the resulting velocity by eqn. 3.15 is within the velocity limits v_{max}. In order to obtain the velocity of dog robots for herding while maintaining the velocity limits, the problem can be solved as a two-step optimization that is described below:
Step 1: Optimize for p_1 and p_2 and obtain the constants as follows:

$$[p_1^*, p_2^*] = \underset{p_1,p_2}{\arg\min}\, \dot{h}(p_1 + p_2) + hp_1p_2 \tag{3.25}$$
$$\text{subject to} \quad (\boldsymbol{x}_S - \boldsymbol{x}_P)^T\boldsymbol{f}(p_1 + p_2) + p_1p_2\frac{h}{2} \geq b^C$$
$$p_1^{max} > p_1 > \max(0, p_1'); \quad p_2^{max} > p_2 > \max(0, p_2')$$

Here p_1^{max} and p_1^{max} are constants that we need to add such that the resulting value does not go to infinity. where p_1' and p_2' are calculated in the first iteration as,

$$p_1' = -\frac{\dot{h}(\boldsymbol{x}(0))}{h(\boldsymbol{x}(0))}; \quad p_2' = -\frac{\ddot{h}(\boldsymbol{x}(0)) + p_1'\dot{h}(\boldsymbol{x}(0))}{\dot{h}(\boldsymbol{x}(0)) + p_1'h(\boldsymbol{x}(0))} \tag{3.26}$$

Here, p_1' and p_2' stay constant in every iteration. The objective function here is chosen after experimenting with various others. This provides us with the most stable output, i.e., less oscillation in the dog robot's velocity. However, the intuition behind this can be when $\dot{h} < 0$ and $h < 0$, that means the sheep robot is already inside the protected zone, and hence both α and β should be increased to increase the reactivity of CBF. And when $\dot{h} > 0$ and $h > 0$, the sheep robot is outside and is moving away from it as well, then both α and β can be reduced.

Step 2: Use the p_1^* and p_2^* to compute the herding constraint and solve the following optimization problem to get the velocity of the dog robots:

$$\begin{aligned} \boldsymbol{u}_D^* &= \arg\min_{\boldsymbol{u}_D} \|\boldsymbol{u}_D\|^2 \\ \text{subject to} \quad & A^H \boldsymbol{u}_D \leq \boldsymbol{b}^{H*} \end{aligned} \tag{3.27}$$

Here $\boldsymbol{b}^{H*}$ is computed using p_1^* and p_2^*. In order to implement this, both steps are performed in each time step. And the resulting velocity ($\boldsymbol{u}_D^*$) will always stay within the velocity limits of the robot while also herding the sheep away from the protected zone. However, this method comes with a disadvantage,i.e., the velocity limits are required to be of the form mentioned above ($\|\boldsymbol{u}_D\|_2 \leq v_{max}$). In a general formulation, velocity limits could be of any form, and in that case, the cascaded CBF method cannot be implemented.

3.5 Slack CBF Design

In order to address the problems of Cascaded CBF, we propose another method called Slack CBF Design. In this method, we treat the constants of the herding constant as slack variables and optimize for the parameters p_1 and p_2 as a single optimization problem. Consider the herding constraint before decomposing it into a linear constraint on the velocity of the dog robot, as in eqn. 4.9:

$$\ddot{h}(\boldsymbol{x}) + (p_1 + p_2)\dot{h}(\boldsymbol{x}) + p_1 p_2 h(\boldsymbol{x}) \geq 0$$

This equation can be decomposed into a constraint on u_D, p_1, and p_2 as:

$$A^H \boldsymbol{u}_D + (p_1 + p_2)\dot{h} + p_1 p_2 h \leq b$$

Now we can solve for the dog velocity while optimizing for p_1 and p_2 while sticking to the velocity limits as a single optimization problem:

$$\begin{aligned} \boldsymbol{u}_D^* = \underset{\boldsymbol{u}_D, p_1, p_2}{\arg\min} \, \|\boldsymbol{u}_D\|^2 + \dot{h}(p_1 + p_2) + h p_1 p_2 \\ \text{subject to} \quad A^H \boldsymbol{u}_D + (p_1 + p_2)\dot{h} + p_1 p_2 h \leq b \\ p_1^{max} > p_1 > \max(0, p_1'), p_2^{max} > p_2 > \max(0, p_2') \\ \|\boldsymbol{u}_D\|_2 \leq v_{max} \end{aligned} \tag{3.28}$$

This formulation won't lead to an infeasible solution due to the presence of slack variables. And we can incorporate any kind of constraints on the velocity. The resulting velocity ($\boldsymbol{u}_D^*$) will ensure herding happens while staying within the velocity limits. The objective here is a combination of the objectives considered in the cascaded controller and holds the same intuitive reasoning. However, this problem is a nonlinear problem and hence requires more time to solve per iteration. Note that this is a general formulation and in order to simplify the calculation one can represent $p_1 + p_2$ as α and $p_1 p_2$ as β to convert the herding constraint into a linear constraint of the velocity of dog robots along with α and β.

3.6 Summary

In this chapter, we derived the constraints required for herding a single sheep using a single-dog robot by utilizing the control barrier functions. Following that, we proved that by utilizing the CBF-based controller, there is a theoretical guarantee that the dog will be able to herd the sheep. Note that the guarantee is theoretical and is often broken in practical scenarios where there exist other constraints. Following that, we discuss the drawback of this method of producing velocity beyond physical limits. Cascaded CBF and Slack CBF designs are presented to be able to provide velocities within physical limits. Along with that, we also discuss the limitations of the formulations.

Chapter 4

Multi Agent Herding Using CBF

In this chapter, we discuss our methodology to derive the control barrier function and its constraints to achieve multi-robot coordination. We demonstrate it on the multi-agent herding problem. First, we define the multi-agent herding problem. Following that, we discuss the derivation of CBFs for two different cases, namely, centralized control and decentralized control.

4.1 Problem Statement

Now instead if one each, suppose there are n "sheep" agents (the herd) and m dog robots (the defenders). This time we assume that the sheep are exhibiting flocking dynamics *i.e.* moving towards a common goal while staying close enough to each other and getting repelled by the dogs. Similar to the single-agent case, given these dynamics, they may end up breaching a high-value unit *i.e.* the protected zone $\mathcal{P} \subset \mathbb{R}^2$ while en route to their goal. Therefore, the objective of the dog robots is to steer them away from this zone.

In this case we denote the position of the i^{th} sheep as $\boldsymbol{x}_{S_i} \in \mathbb{R}^2$ and the collective positions of the herd as $\boldsymbol{x}_S^{all} \coloneqq (\boldsymbol{x}_{S_1}, \cdots, \boldsymbol{x}_{S_n})$. Likewise, we denote the position of the k^{th} dog as $\boldsymbol{x}_{D_k} \in \mathbb{R}^2$ and the collective positions of the defending robots' group as $\boldsymbol{x}_D^{all} \coloneqq (\boldsymbol{x}_{D_1}, \cdots, \boldsymbol{x}_{D_m})$. We assume both sheep and dogs have single-integrator dynamics *i.e.* they are velocity controlled. For the

i^{th} sheep, we have:

$$\begin{aligned}\dot{\boldsymbol{x}}_{S_i} &= \boldsymbol{u}_{S_i} \qquad (4.1)\\ &= k_S \sum_{j\in\mathcal{S}} \left(1 - \frac{R_S^3}{\left\|\boldsymbol{x}_{S_j} - \boldsymbol{x}_{S_i}\right\|^3}\right)(\boldsymbol{x}_{S_j} - \boldsymbol{x}_{S_i})\\ &+ k_G\left(\boldsymbol{x}_G - \boldsymbol{x}_{S_i}\right) + k_D \sum_{k\in\mathcal{D}} \frac{\boldsymbol{x}_{S_i} - \boldsymbol{x}_{D_k}}{\left\|\boldsymbol{x}_{S_i} - \boldsymbol{x}_{D_k}\right\|^3}\\ &:= \boldsymbol{f}_i(\boldsymbol{x}_{S_1}, ..., \boldsymbol{x}_{S_n}, \boldsymbol{x}_{D_1}, ..., \boldsymbol{x}_{D_n}) \qquad (4.2)\end{aligned}$$

Here the first term represents cohesion of the flock, the second represents attraction to goal and the third represents repulsion from dog robots. The attraction to the goal represents the self-motivated part of the dynamics of the sheep agents. This term is often neglected in prior work. R_S is the safety radius for sheep i to avoid collisions with the other sheep, x_G is its desired goal position (common for all sheep) and k_S, k_G, k_D are proportional gains corresponding to forces in the dynamics. For each dog we have:

$$\dot{\boldsymbol{x}}_{D_k} = \boldsymbol{u}_{D_k} \ \forall k \in \{1, 2, \cdots, m\} \qquad (4.3)$$

We make similar assumption as before, as mentioned in Assumption 1. Now We can pose the dog robots' problem as follows:

Definition 2. *Assuming that the initial positions of the sheep* $\boldsymbol{x}_{S_i}(0) \in \mathcal{P}^c\ \forall i \in \{1, \cdots, n\}$*, the dog robots' problem is to synthesize controls* $\{\boldsymbol{u}_{D_1}, \cdots, \boldsymbol{u}_{D_m}\}$ *such that* $\boldsymbol{x}_{S_i}(t) \in \mathcal{P}^c\ \forall t \geq 0\ \forall i \in \{1, \cdots, n\}$*. If* $\boldsymbol{x}_{S_i}(0) \notin \mathcal{P}^c\ \forall i \in \{1, \cdots, n\}$*, the dog robots' problem is to synthesize controls* $\{\boldsymbol{u}_{D_1}, \cdots, \boldsymbol{u}_{D_m}\}$ *such that* $\boldsymbol{x}_{S_i}(t) \rightsquigarrow \mathcal{P}^c\ \forall i \in \{1, \cdots, n\}$ *in finite time.*

In the next section, we show how to address the multi-robot coordination problem among dog robots to herd the sheep robots using control barrier functions collectively.

4.2 Centralized Multi-Robot Controller using CBF

In this section, we discuss our proposed approach to solve the problem of defending the protected zone as stated before. Given the protected zone as defined (3.3), we first pose the requirement for defending against one sheep, say sheep i located at $\boldsymbol{x}_{S_i}$. Subsequently, we will generalize this to the rest of the sheep in the herd. For this sheep, define a safety index $h(\cdot) : \mathbb{R}^2 \longrightarrow \mathbb{R}$ as follows:

$$h = \|\boldsymbol{x}_{S_i} - \boldsymbol{x}_p\|^2 - R_p^2 \tag{4.4}$$

By construction, $h \geq 0 \; \forall \boldsymbol{x}_{S_i} \in \mathcal{P}^c$ *i.e.* non-negative whenever i is on the boundary or outside the protected zone. Thus, assuming that at $t = 0$, $h(\boldsymbol{x}_{S_i}(0)) \geq 0$, we require $h(\boldsymbol{x}_{S_i}(t)) \geq 0 \; \forall t \geq 0$. Treating $h(\cdot)$ as a control barrier function [7], this can be achieved if the derivative of $h(\cdot)$ satisfies the following constraint:

$$\begin{aligned}
&\dot{h}(\boldsymbol{x}_{S_1}, \cdots, \boldsymbol{x}_{S_n}, \boldsymbol{x}_{D_1}, \cdots, \boldsymbol{x}_{D_m}) + p_1 h(\boldsymbol{x}_{S_i}) \geq 0 \\
\implies &2(\boldsymbol{x}_{S_i} - \boldsymbol{x}_p)^T \dot{\boldsymbol{x}}_{S_i} + p_1 h(\boldsymbol{x}_{S_i}) \geq 0 \\
\implies &2(\boldsymbol{x}_{S_i} - \boldsymbol{x}_p)^T \boldsymbol{f}_i + p_1 h(\boldsymbol{x}_{S_i}) \geq 0.
\end{aligned} \tag{4.5}$$

Define $\boldsymbol{x} = (\boldsymbol{x}_S^{all}, \boldsymbol{x}_D^{all})$, we rewrite this as

$$2(\boldsymbol{x}_{S_i} - \boldsymbol{x}_p)^T \boldsymbol{f}_i(\boldsymbol{x}) + p_1 h(\boldsymbol{x}_{S_i}) \geq 0. \tag{4.6}$$

with the constraints on p_1 being similar to the single agent case. Since $\boldsymbol{u}_D^{all}$ does not show up in (4.5), we define another function $v(\cdot) : \mathbb{R}^{2(m+n)} \longrightarrow \mathbb{R}$:

$$v = \dot{h} + p_1 h. \tag{4.7}$$

Like before, in order to ensure $v \geq 0$ is always maintained, its derivative needs to satisfy

$$\dot{v}(\boldsymbol{x}) + p_2 v(\boldsymbol{x}) \geq 0. \tag{4.8}$$

Here p_2 is another design parameter for which we choose p_2 to ensure that the constraint mentioned in the single agent case is satisfied at $t = 0$. Finally using (4.7) in (4.8), we get:

$$\begin{aligned}
&\ddot{h}(\boldsymbol{x}) + (p_1 + p_2)\dot{h}(\boldsymbol{x}) + p_1 p_2 h(\boldsymbol{x}) \geq 0 \\
&\qquad \implies \ddot{h}(\boldsymbol{x}) + \alpha \dot{h}(\boldsymbol{x}) + \beta h(\boldsymbol{x}) \geq 0.
\end{aligned} \tag{4.9}$$

where we have defined $\alpha \coloneqq p_1 + p_2$ and $\beta \coloneqq p_1 p_2$. α and β are chosen by ensuring that p_1 and p_2 satisfy the requirements in (3.7) and (3.10). The time derivative of $h(\cdot)$ required in (4.9) is

$$\begin{aligned}
\dot{h}(\boldsymbol{x}) &= 2(\boldsymbol{x}_{S_i} - \boldsymbol{x}_P)^T \dot{\boldsymbol{x}}_{S_i} \\
&= 2(\boldsymbol{x}_{S_i} - \boldsymbol{x}_P)^T \boldsymbol{f}_i(\boldsymbol{x}_{S_1}, \cdots, \boldsymbol{x}_{S_n}, \boldsymbol{x}_{D_1}, \cdots, \boldsymbol{x}_{D_m})
\end{aligned} \tag{4.10}$$

$$\begin{aligned}\ddot{h}(\boldsymbol{x}) &= 2\dot{\boldsymbol{x}}_{S_i}^T\dot{\boldsymbol{x}}_{S_i}\\ &\quad + 2(\boldsymbol{x}_{S_i}-\boldsymbol{x}_P)^T\left(\sum_{j=1}^{n}\mathbb{J}_{ji}^{S}\dot{\boldsymbol{x}}_{S_i}+\sum_{k=1}^{m}\mathbb{J}_{ki}^{D}\boldsymbol{u}_{D_k}\right)\\ &= 2\boldsymbol{f}_i^T\boldsymbol{f}_i\\ &\quad + 2(\boldsymbol{x}_{S_i}-\boldsymbol{x}_P)^T\left(\sum_{j=1}^{n}\mathbb{J}_{ji}^{S}\boldsymbol{f}_i+\sum_{k=1}^{m}\mathbb{J}_{ki}^{D}\boldsymbol{u}_{D_k}\right),\end{aligned} \tag{4.11}$$

where $\mathbb{J}_{ji}^S$ and $\mathbb{J}_{ki}^D$ are

$$\begin{aligned}\mathbb{J}_{ji}^{S} &:= \nabla_{\boldsymbol{x}_{S_j}}\boldsymbol{f}_i(\boldsymbol{x}_{S_1},\cdots,\boldsymbol{x}_{S_n},\boldsymbol{x}_{D_1},\cdots,\boldsymbol{x}_{D_m})\\ \mathbb{J}_{ki}^{D} &:= \nabla_{\boldsymbol{x}_{D_k}}\boldsymbol{f}_i(\boldsymbol{x}_{S_1},\cdots,\boldsymbol{x}_{S_n},\boldsymbol{x}_{D_1},\cdots,\boldsymbol{x}_{D_m})\end{aligned}$$

Note here that $\ddot{h}(\boldsymbol{x})$ contains the velocities of dogs as we wanted. Using (4.10) and (4.11) in (4.9), we get the following linear constraints on dog velocities to ensure that the i^{th} sheep stays outside the protected zone $\mathcal{P}$:

$$A_i^H\boldsymbol{u}_D^{all} \leq b_i^H, \tag{4.12}$$

where,

$$\begin{aligned}A_i^H &:= (\boldsymbol{x}_P-\boldsymbol{x}_{S_i})^T\begin{bmatrix}\mathbb{J}_{1i}^D & \mathbb{J}_{2i}^D & ..... & \mathbb{J}_{mi}^D\end{bmatrix}\\ b_i^H &:= \boldsymbol{f}_i^T\boldsymbol{f}_i+(\boldsymbol{x}_{S_i}-\boldsymbol{x}_P)^T\sum_{j=1}^{n}\mathbb{J}_{ji}^S\boldsymbol{f}_j\\ &\quad + \alpha(\boldsymbol{x}_{S_i}-\boldsymbol{x}_P)^T\boldsymbol{f}_i+\beta\frac{h}{2}\end{aligned}$$

To ensure all n sheep stay away from $\mathcal{P}$, we augment constraints (4.25) for all the herd as follows:

$$\begin{bmatrix}A_1^H\\ \vdots\\ A_n^H\end{bmatrix}\boldsymbol{u}_D^{all} \leq \begin{bmatrix}b_1^H\\ \vdots\\ b_n^H\end{bmatrix} \implies \boldsymbol{\mathcal{A}}^H\boldsymbol{u}_D^{all} \leq \boldsymbol{b}^H \tag{4.13}$$

Here $\boldsymbol{\mathcal{A}}^H \in \mathbb{R}^{n\times 2m}$ and $\boldsymbol{b}^H \in \mathbb{R}^n$. Given these constraints on the dogs' velocities, we pose the following QP that searches for the min-norm velocities that satisfies these constraints

$$\begin{aligned}\boldsymbol{u}_D^{*all} &= \underset{\boldsymbol{u}_D^{all}}{\arg\min}\left\|\boldsymbol{u}_D^{all}\right\|^2\\ \text{subject to}\quad & \boldsymbol{\mathcal{A}}^H\boldsymbol{u}_D^{all} \leq \boldsymbol{b}^H\end{aligned} \tag{4.14}$$

By construction, our approach is centralized *i.e.* it computes velocities of all dog robots together. Future work will consider ways to decentralize this approach.
Considering multiple protected zones: While in the above derivation, we considered preventing the sheep from breaching only one protected zone, we can just as easily consider another protected zone by formulating similar constraints $\mathcal{A}_2^H \boldsymbol{u}_D^{all} \leq \boldsymbol{b}_2^H$ on the dogs' velocities. By augmenting (3.15) with these constraints for the other zone, we will be able to defend both zones from all sheep simultaneously. This is a benefit offered by our constraint-based framework.

4.3 Centralized Multi-Robot Control using Slack CBF

In this section, we discuss how to use the methods we designed earlier to keep the output velocities within the limits. Similar to the single agent case, the constraints on the velocities that we are dealing with can be represented as:

$$\|\boldsymbol{u}_{D_k}\|_2 \leq v_{max}, \quad \forall k \in \mathcal{D} \tag{4.15}$$

And in the previous section, we derived the herding constraint on the centralized system to be:

$$\mathcal{A}^H \boldsymbol{u}_D^{all} \leq \boldsymbol{b}^H \tag{4.16}$$

To be able to use Cascaded CBF formulation, we would require the constraints on the velocity to be of the form:

$$\|\boldsymbol{u}_{D_{all}}\|_2 \leq v_{max} \tag{4.17}$$

However, that is not the case. Hence, the Cascaded CBF formulation cannot be applied here. Thus to address the problem, we resort to Slack CBF design. Similar to the one dog against one sheep case we first decompose the herding constraint of the i^{th} sheep into a constraint on u_D, p_1, and p_2 as:

$$A_i^H \boldsymbol{u}_D + (p_{1i} + p_{2i})\dot{h}_i + p_{1i}p_{2i}h_i \leq b_i$$

Now we can solve for the velocity of all the dog robots while optimizing for p_1 and p_2 corresponding to all the sheep robots. This can be done as a single optimization problem while also sticking to the velocity limits as:

$$\begin{aligned} \boldsymbol{u}_D^{*all} = \underset{\boldsymbol{u}_D^{all}}{\arg\min} & \left\|\boldsymbol{u}_D^{all}\right\|^2 + \dot{h}_{all}^T(p_1^{all} + p_2^{all}) + h_{all}^T p_1^{all} p_2^{all} \\ \text{subject to} \quad & \mathcal{A}^H \boldsymbol{u}_D + \dot{h}_{all}^T(p_1^{all} + p_2^{all}) + h_{all}^T p_1^{all} p_2^{all} \leq \boldsymbol{b}^H \\ p_1^{all,max} > p_1^{all} > \max(0, p_1'^{all}), & p_2^{all,max} > p_2^{all} > \max(0, p_2'^{all}) \\ & \|\boldsymbol{u}_{D_k}\|_2 \leq v_{max}, \quad \forall k \in \mathcal{D} \end{aligned} \tag{4.18}$$

Here $\mathcal{S}$ represents all the sheep robots and $\mathcal{S}$ represents all the dog robots. Also, $\dot{h}_{all}^T$ represents the transposed column vector of $\dot{h}$ corresponding to all the sheep robots. Similarly, h_{all}^T represents the transposed column vector of h. $\mathcal{A}^H$ and $\boldsymbol{b}^H$ are the matrices and the vector respectively mentioned in the previous section centralized formulation as herding constraint.

4.4 Decentralized Multi-Robot Controller using CBF

In this approach, we assume that we have an equal number of dogs and sheep. By exploiting this equality, we assign a unique sheep S_i for $i \in \{1, \cdots, n\}$ to a unique dog D_k for $k \in \{1, \cdots, n\}$ and make D_k responsible for herding S_i away from $\mathcal{P}$. In other words, D_k computes a velocity $\boldsymbol{u}_{D_k}$ that repels S_i from $\mathcal{P}$ thereby ensuring that $\boldsymbol{x}_{S_i}(t) \notin \mathcal{P}\ \forall t \geq 0$. The premise is that owing to the equality, each sheep will end up being herded by a unique dog, therefore, no sheep will breach the protected zone [1]. Now while this strategy necessitates having an equal number of dogs and sheep, the benefit of this approach stems from the feasibility guarantee (that we prove shortly), which the centralized approach lacks. Simple algebraic manipulation of constraint (4.12) yields a constraint on the velocity of D_k as follows

$$A_i^H \boldsymbol{u}_{D_k} \leq b_i^H, \quad \text{where} \tag{4.19}$$

$$\begin{aligned} A_i^H &:= (\boldsymbol{x}_P - \boldsymbol{x}_{S_i})^T \mathbb{J}_{ki}^D \\ b_i^H &:= \boldsymbol{f}_i^T \boldsymbol{f}_i + (\boldsymbol{x}_{S_i} - \boldsymbol{x}_P)^T \Big\{ \sum_{j \in \mathcal{S}} \mathbb{J}_{ji}^S \boldsymbol{f}_j + \alpha \boldsymbol{f}_i + \beta \frac{h}{2} + \sum_{l \in \mathcal{D} \backslash k} \mathbb{J}_{li}^D \boldsymbol{u}_{D_l} \Big\} \end{aligned}$$

Here $A_i^H \in \mathbb{R}^{1\times 2}$ and $\boldsymbol{b}_i^H \in \mathbb{R}$. The term u_{D_l} in the expression of b_i^H is computed by using numerical differentiation of the positions $\boldsymbol{x}_{D_l}$. We pose a QP to obtain

[1] Note that although S_i is assigned to D_k, the position of the remaining dogs $\{1, \cdots, n\}\backslash k$ and the remaining sheep $\{1, \cdots, n\}\backslash i$ do influence D_k's constraint parameters (A_i^H, b_i^H), and in turn, its computed velocity $\boldsymbol{u}_{D_k}^*$.

the min-norm velocity for D_k as follows

$$\boldsymbol{u}^*_{D_k} = \arg\min_{\boldsymbol{u}_{D_k}} \|\boldsymbol{u}_{D_k}\|^2$$
$$\text{subject to} \quad A_i^H \boldsymbol{u}_{D_k} \leq b_i^H \tag{4.20}$$

The obtained velocity $\boldsymbol{u}^*_{D_k}$ guarantees that the protected zone $\mathcal{P}$ will not be breached by sheep S_i by ensuring that $h(\boldsymbol{x}_{S_i}(t)) \geq 0 \ \forall t \geq 0$. Since each dog in $\mathcal{D}$ is in-charge of herding exactly one sheep in $\mathcal{S}$, feasibility of (4.25) $\forall k \in \mathcal{D}$ would ensure no sheep breaches $\mathcal{P}$.

4.5 Proof of feasibility: Decentralized Approach

Next, we show the conditions under which (3.15) remains feasible but first state some assumptions.

Assumption 3. *We make the following assumptions on the distances between pairs of agents:*

1. *There exists a lower bound and upper bound on the distance between any pair of sheep, i.e,* $L_S \leqslant \left\|\boldsymbol{x}_{S_i} - \boldsymbol{x}_{S_j}\right\| \leqslant M_S$, $\forall i, j \in \mathcal{S}$ *and* $i \neq j$.
2. *There exists a lower bound on the distance between every sheep and dog, i.e.,* $\|\boldsymbol{x}_{S_i} - \boldsymbol{x}_{D_k}\| \geq L_D \ \forall i \in \mathcal{S}$ *and* $k \in \mathcal{D}$.
3. *There exists a upper bound on the distance between each sheep and its goal i.e.,* $\|\boldsymbol{x}_{S_i} - \boldsymbol{x}_G\| \leqslant M_G$ *and between the sheep and the center of the protected zone i.e.,* $\|\boldsymbol{x}_{S_i} - \boldsymbol{x}_P\| \leqslant M_P$.

Theorem 2. *In a scenario with 'n' dogs and 'n' sheep, with each dog assigned a unique sheep, the herding constraint* (4.25) *for a given dog is always feasible, provided assumptions 3 are met.*

Proof. Our strategy to guarantee feasibility of constraint (4.25) relies on ruling out situations in which it is infeasible. (4.25) can become infeasible

- either when $A_i^H = \mathbf{0}$ and $b_i^H < 0$ (possibility 1)
- or when $b_i^H = -\infty$ (possibility 2).

To determine the conditions in which possibility 1 occurs, we calculate the determinant of $\mathbb{J}^D_{ki}$ as

$$det(\mathbb{J}_{ki}^{D}) = \frac{-2k_D^2}{\|\boldsymbol{x}_{D_k} - \boldsymbol{x}_{S_i}\|^3}$$

The determinant $det(\mathbb{J}_{ki}^{D})$ is non-zero as long as the distance between dog D_k and sheep S_i is finite. Therefore, $\mathbb{J}_{ki}^{D}$ will have no null space, implying that $A_i^H \neq 0 \ \forall \boldsymbol{x}_{S_i} \in \mathbb{R}^2, \boldsymbol{x}_{D_k} \in \mathbb{R}^2$. This rules out possibility 1 for infeasibility. To rule out possibility 2, we need to check for condition when $b_i^H \longrightarrow -\infty$. Given b_i^H in (4.25), we find its worst case lower bound. Here $\boldsymbol{f}_i^T \boldsymbol{f}_i \geq 0$ and as we assume that at the current time step, the sheep is outside $\mathcal{P}$, this ensures $\beta \frac{h}{2} \geq 0$. By removing these terms, the lower bound of b_i^H can be given as

$$\begin{aligned} b_i^H \geq \sum_{j \in \mathcal{S} \setminus i} (\boldsymbol{x}_{S_i} - \boldsymbol{x}_P)^T \mathbb{J}_{ji}^S \boldsymbol{f}_j + (\boldsymbol{x}_{S_i} - \boldsymbol{x}_P)^T \mathbb{J}_{ii}^S \boldsymbol{f}_i + \sum_{l \in \mathcal{D} \setminus k} (\boldsymbol{x}_{S_i} - \boldsymbol{x}_P)^T \mathbb{J}_{li}^D \boldsymbol{u}_{D_l} \\ + \alpha (\boldsymbol{x}_{S_i} - \boldsymbol{x}_P)^T \boldsymbol{f}_i \end{aligned} \tag{4.21}$$

Using the triangle inequality on the RHS and Cauchy-Schwarz inequality on individual terms, we get

$$\begin{aligned} b_i^H \geq \sum_{j \in \mathcal{S} \setminus i} \left(-\sigma_{max} \left(\mathbb{J}_{ji}^S \right) \|\boldsymbol{x}_{S_i} - \boldsymbol{x}_P\| \, \|\boldsymbol{f}_j\| \right) - \sigma_{max} \left(\mathbb{J}_{ii}^S \right) \|\boldsymbol{x}_{S_i} - \boldsymbol{x}_P\| \, \|\boldsymbol{f}_i\| \quad (4.22) \\ + \sum_{l \in \mathcal{D} \setminus k} \left(-\sigma_{max} \left(\mathbb{J}_{li}^D \right) \|\boldsymbol{x}_{S_i} - \boldsymbol{x}_P\| \, \|\boldsymbol{u}_{D_l}\| \right) - \alpha \|\boldsymbol{x}_{S_i} - \boldsymbol{x}_P\| \|\boldsymbol{f}_i\| \end{aligned}$$

where σ_{max} is the largest singular value of a matrix. Further, using the fact that the largest singular value of a matrix (σ_{max}) is upper bounded by its Frobenius norm (σ_F), we obtain

$$\begin{aligned} b_i^H \geq \sum_{j \in \mathcal{S} \setminus i} \left(-\sigma_F \left(\mathbb{J}_{ji}^S \right) \|\boldsymbol{x}_{S_i} - \boldsymbol{x}_P\| \, \|\boldsymbol{f}_j\| \right) - \sigma_F \left(\mathbb{J}_{ii}^S \right) \|\boldsymbol{x}_{S_i} - \boldsymbol{x}_P\| \, \|\boldsymbol{f}_i\| \quad (4.23) \\ \sum_{l \in \mathcal{D} \setminus k} \left(-\sigma_F \left(\mathbb{J}_{ki}^D \right) \|\boldsymbol{x}_{S_i} - \boldsymbol{x}_P\| \, \|\boldsymbol{u}_{D_l}\| \right) - \alpha \|\boldsymbol{x}_{S_i} - \boldsymbol{x}_P\| \|\boldsymbol{f}_i\| \end{aligned}$$

Now to compute this lower bound we make use of assumption 3. We use the dynamics in (3.1) to compute $\mathbb{J}_{ii}^S$ and obtain the upper bound on $\sigma_F \left(\mathbb{J}_{ii}^S \right)$ and use the bounds on distances from assumption 3 to get following upper bound:

$$\begin{aligned} \sigma_F \left(\mathbb{J}_{ii}^S \right) &\leqslant \sum_{j \in \mathcal{S} \setminus i} k_S \left(\sqrt{2} + \frac{(3+\sqrt{2}) R^3}{\|\boldsymbol{x}_{S_i} - \boldsymbol{x}_{S_j}\|^3} \right) + \sqrt{2} k_G + \sum_{l \in \mathcal{D} \setminus k} \frac{\left(3+\sqrt{2}\right) k_D}{\|\boldsymbol{x}_{S_i} - \boldsymbol{x}_{D_l}\|^3} \\ &\leqslant (n-1) \left(\sqrt{2} k_S + \frac{(3+\sqrt{2}) k_S R^3}{L_S^3} \right) + \sqrt{2} k_G + n \left(\frac{\left(3+\sqrt{2}\right) k_D}{L_D^3} \right) := \lambda_M \end{aligned}$$

We omit the proof of this computation in the interest of space. Similarly, using the dynamics in (3.1), we compute an expression for $\mathbb{J}^S_{ji}$ and obtain an upper bound on $\sigma_F\left(\mathbb{J}^S_{ji}\right)$ as follows:

$$\sigma_F\left(\mathbb{J}^S_{ji}\right) \leqslant \sqrt{2}k_S + \frac{(3+\sqrt{2})k_S R^3}{\|\boldsymbol{x}_{S_1} - \boldsymbol{x}_{S_j}\|^3} \leqslant \sqrt{2}k_S + \frac{(3+\sqrt{2})k_S R^3}{L_S^3} := \lambda_S$$

Likewise, an upper bound of $\sigma_F\left(\mathbb{J}^S_{li}\right)$, is given by

$$\sigma_F\left(\mathbb{J}^S_{li}\right) \leqslant \frac{(3+\sqrt{2})k_S R^3}{\|\boldsymbol{x}_{S_1} - \boldsymbol{x}_{D_l}\|^3} \leqslant \frac{(3+\sqrt{2})k_S R^3}{L_D^3} := \lambda_D$$

Lastly, we use obtain an upper bound on the dynamics of each sheep $\boldsymbol{f}_i$ as:

$$\begin{aligned}\|\boldsymbol{f}_i\| \leqslant &\sum_{j\in\mathcal{S}\backslash i} k_S\left(\|\boldsymbol{x}_{S_i} - \boldsymbol{x}_{S_j}\| + \frac{R^3}{\|\boldsymbol{x}_{S_i} - \boldsymbol{x}_{S_j}\|^2}\right) + k_G\|\boldsymbol{x}_G - \boldsymbol{x}_{S_i}\| \\ &+ \sum_{l\in\mathcal{D}} k_D \frac{\|\boldsymbol{x}_{S_i} - \boldsymbol{x}_{D_l}\|}{\|\boldsymbol{x}_{S_i} - \boldsymbol{x}_{D_l}\|^3} \end{aligned} \tag{4.24}$$

Now we need to compute the maximum possible value of the RHS to get the upper bound of the sheep dynamics. The first term has a local minima at $\|\boldsymbol{x}_{S_i} - \boldsymbol{x}_{S_j}\| = (2)^{1/3}R$. Therefore the maximum value can occur at either the lower bound or upper bound of $\|\boldsymbol{x}_{S_i} - \boldsymbol{x}_{S_j}\|$. Thus the maximum value of the first term can be given as $F_{max} := \max(k_S L_S + k_S\frac{R^3}{L_S^2}, k_S M_S + k_S\frac{R^3}{M_S^2})$. Second term is maximum when $\|\boldsymbol{x}_G - \boldsymbol{x}_{S_i}\| = M_G$. The last term is maximum when distance of the sheep to the dogs are minimum, $\|\boldsymbol{x}_{S_i} - \boldsymbol{x}_{D_k}\| = L_D$. Using these the upper bound on the sheep dynamics is computed as:

$$\|\boldsymbol{f}_i\| \leqslant (n-1)F_{max} + k_G M_G + nk_D\left(\frac{1}{L_D^2}\right)$$

Assuming that the velocity of the dog robots have an upper bound, and by taking the upper bound on the dynamics of all the sheep to be equal, the lower bound on b_i^H from 4.23 is (taking $\gamma = -(\alpha + \lambda_M + (n-1)\lambda_S)M_p$)

$$b_i^H \geqslant \gamma\left\{(n-1)F_{max} + k_G M_G + \frac{nk_D}{L_D^2}\right\} - (n-1)\lambda_D M_P\,\|\boldsymbol{u}_D\|_{\max}$$

This shows that b_i^H has a finite lower bound, thus ruling out possibility 2. Thus, the herding constraint (4.25) for a one dog to repel one sheep from the protected zone is always feasible. Since each sheep in $\mathcal{S}$ is allocated to one unique dog in $\mathcal{D}$, the extension of this feasibility result to all sheep ensures that none of them will breach the protected zone. $\square$

4.6 Decentralized Multi-Robot Controller using Cascaded CBF

In this section, we discuss formulation to keep the output velocities within the limits while using Decentralized Control. The herding constraint on the k^{th} dog robot is:

$$\mathcal{A}^H \boldsymbol{u}_{D_k} \leq \boldsymbol{b}^H, \quad k \in \mathcal{D} \tag{4.25}$$

The constraints on the velocities that we are dealing with can be represented as:

$$\|\boldsymbol{u}_{D_k}\|_2 \leq v_{max}, \quad k \in \mathcal{D} \tag{4.26}$$

In this form, we can use Cascaded CBF formulation as well as Slack CBF to get the velocity within limits. In practice, Cascaded CBF has lower computation, so we show the formulation for the same.

As we have shown before in one dog case, the constraint of p_1 and p_2 for the i^{th} sheep is:

$$(\boldsymbol{x}_{S_i} - \boldsymbol{x}_P)^T \boldsymbol{f_i}(p_{1i} + p_{2i}) + p_{1i}p_{2i}\frac{h}{2} \geq b_i^C \tag{4.27}$$

As long as this nonlinear constraint on the constants p_{1i} and p_{2i} is met, we can guarantee that the resulting velocity is within the velocity limits v_{max}. In order to obtain the velocity of dog robots for herding while maintaining the velocity limits, the problem can be solved as a two-step optimization that is described below:

Step 1: Optimize for all p_1 and p_2 every sheep and obtain the constants as follows:

$$[p_1^{all*}, p_2^{all*}] = \underset{p_1^{all}, p_2^{all}}{\arg\min} \quad \dot{h}_{all}^T(p_1^{all} + p_2^{all}) + h_{all}^T p_1^{all} p_2^{all} \tag{4.28}$$

$$\text{subject to} \quad (\boldsymbol{x}_{S_i} - \boldsymbol{x}_P)^T \boldsymbol{f_i}(p_1^{all} + p_2^{all}) + p_1^{all}.p_2^{all}\frac{h_i}{2} \geq b_i^C \quad \forall i \in \mathcal{S}$$
$$p_1^{max,all} > p_1^{all} > \max(0, p_1'^{all}); \quad p_2^{max,all} > p_2 > \max(0, p_2'^{all})$$

Here $p_1^{all}.p_2^{all}$ is an element-wise multiplication that results in a vector. Also, $p_1'^{all}$ and $p_2'^{all}$ are calculated in the first iteration as,

$$p_1'^{all} = -\frac{\dot{h}(\boldsymbol{x}(0))}{h(\boldsymbol{x}(0))}; \quad p_2'^{all} = -\frac{\ddot{h}(\boldsymbol{x}(0)) + p_1'^{all}.\dot{h}(\boldsymbol{x}(0))}{\dot{h}(\boldsymbol{x}(0)) + p_1'^{all}.h(\boldsymbol{x}(0))} \tag{4.29}$$

Here, $p_1'^{all}$ and $p_2'^{all}$ stay constant in every iteration. Also, (.) represents element-wise multiplication.

Step 2: Use the p_1^{*all} and p_2^{*all} to compute the herding constraint and solve the following optimization problem to get the velocity of the k^{th} dog robots as:

$$\begin{aligned} \boldsymbol{u}_{D_k}^* &= \underset{\boldsymbol{u}_{D_k}}{\arg\min} \left\|\boldsymbol{u}_{D_k}\right\|^2 \\ \text{subject to} \quad & \mathcal{A}^H \boldsymbol{u}_{D_k} \leq \boldsymbol{b}^{H*} \end{aligned} \tag{4.30}$$

Here $\boldsymbol{b}^{H*}$ is computed using p_1^* and p_2^*. In order to implement this, both steps are performed in each time step. And the resulting velocity ($\boldsymbol{u}_{D_k}^*$) will always stay within the velocity limits of the robot while also herding the sheep away from the protected zone.

4.7 Summary

In this chapter, we derived the constraints required for herding multiple sheep using multiple dog robots by utilizing the control barrier functions. We produced a way to achieve centralized control and a way to achieve decentralized control. The limitation of decentralized control being it requires an equal number of dog and sheep robots. However, the advantage is it preserves the guarantees of a CBF. Intuitively if there exists a solution for a Centralized case it preserves the guarantee of CBF. However, there is no guarantee that there exists a solution. In a decentralized case, there is a guarantee that a solution exists. Note that the guarantee is theoretical and is often broken in practical scenarios where there exist other constraints.

Chapter 5

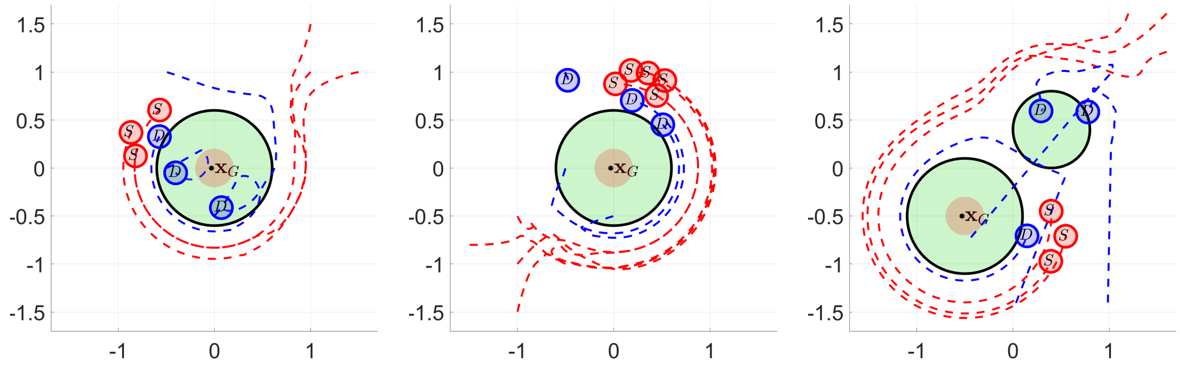
Experimental Results and Simulations

In this section, we show results of our approach by testing it on different scenarios consisting of varying numbers of sheep and dog and varying their initial positions. Additionally, we also run validate these results experimentally. We perform several experiments with nonholonomic Khepera robots and demonstrate how our algorithm find velocities for one dog to simultaneously defend multiple protected zones from multiple sheep.

5.1 Numerical Simulation

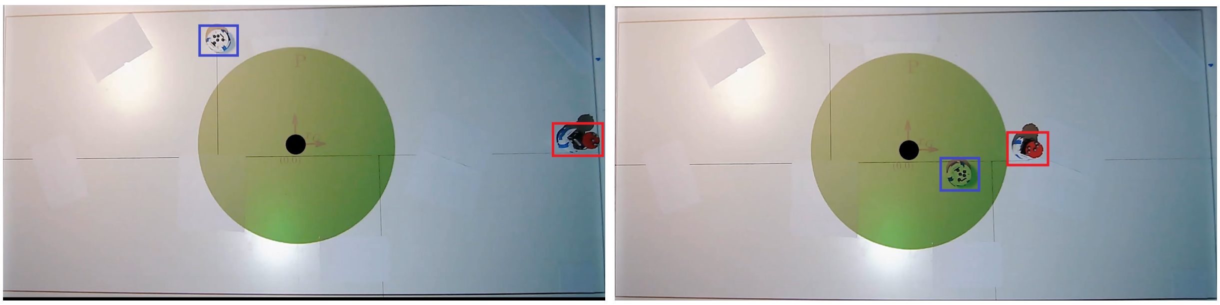
We represent the protected zone using a circular disc with radius R_p and its center at the origin *i.e.* $\boldsymbol{x}_P = \mathbf{0}$. In our simulations, we purposefully choose the agent's goal $\boldsymbol{x}_G = \boldsymbol{x}_P$ so that the sheep are motivated to breach the protected zone should the dog robots not interfere. Thus, this is an adversarial scenario. The initial position $\boldsymbol{x}_{S_i}(0)$ of all the sheep is chosen such that they are all close to each other. This is done to ensure that the sheep have enough time to stabilize as a flock before interacting with the dog robots. The initial position $\boldsymbol{x}_D^{all}(0)$ of the dog robots is chosen randomly within the area of operation. The sheep's velocities are calculated using (3.1). The values of the gains in the sheep dynamics were taken as $k_G = 1$, $k_S = 0.3$ and $k_D = 0.08$.

The velocities of the dog robot were obtained using eqn. (4.14). The hyperparameters α, β, γ are tuned satisfy the conditions on the design parameters (3.7, 3.10) . Figure 5.1 shows three simulation results for this behavior. In these simulations, we varied the initial position of the sheep (blue), the dog (red), the number of sheep, and the number of dogs. It can be noticed from the figure that in all three scenarios, the dog robots are able to successfully intercept the sheep and prevent them from entering the protected zone while also avoiding collision with the sheep.

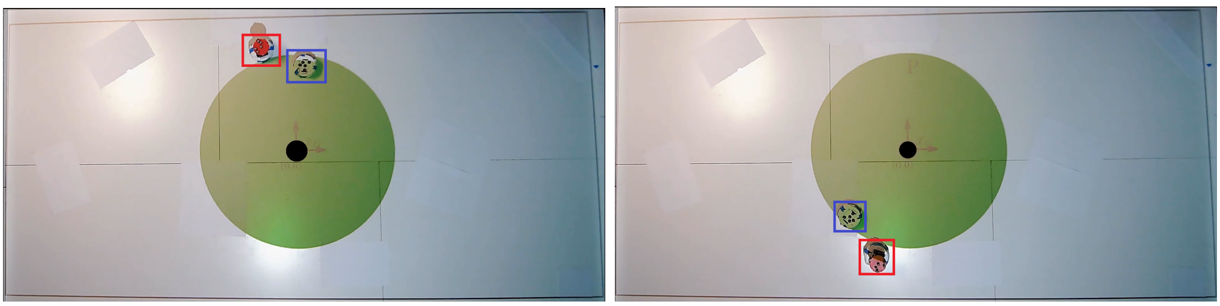


(a) Three dog robots v/s three sheep robots.
(b) Three dog robots v/s five sheep.
(c) Three dog robots v/s three sheep robots.

Figure 5.1: Centralized Control: Preventing the breaching of the protected zone. In these simulations, the dog is shown in blue and the sheep is shown in red. The green disc represents the protected zone. The nominal task of the red agent is to go straight towards its goal $\boldsymbol{x}_G$. However, since this would result in infiltration of the protected zone, the dog intervenes using the control algorithm presented in (4.14). In 5.1(c), we defend two protected zones from three sheep.

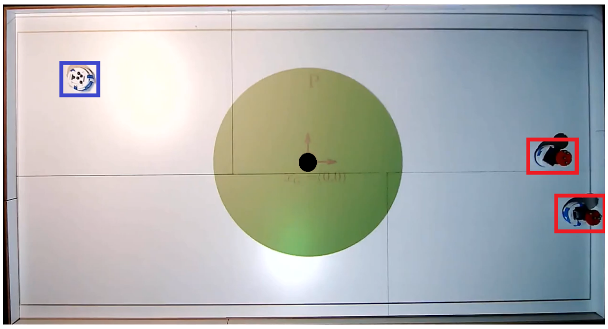


(a) $t = 0s$
(b) $t = 3s$

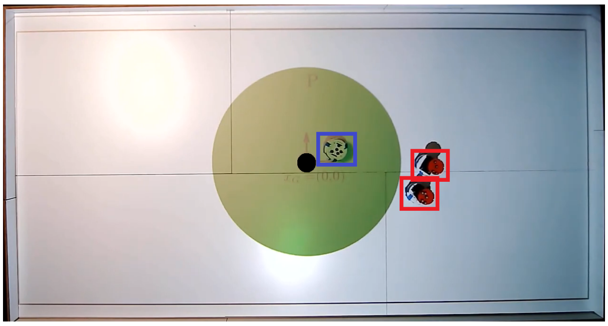


(c) $t = 20s$
(d) $t = 55s$

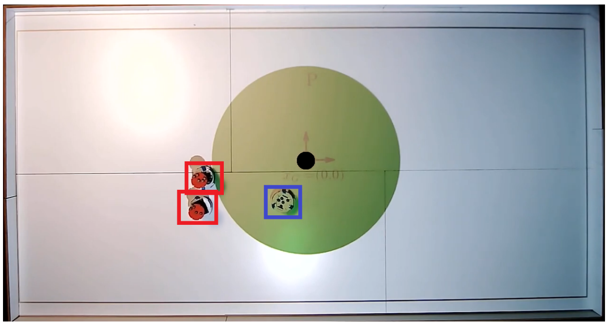
Figure 5.2: Experiments for Centralized Control: One dog robot preventing one sheep from the breaching of the protected zone. The dog robot is highlighted in blue and the sheep in red. The goal position x_G is at the center of the protected zone and given as a black solid circle. The nominal task of the sheep is to go straight towards its goal $\boldsymbol{x}_G$. However, since this would result in infiltration of the protected zone, the dog intervenes using the control algorithm presented in (4.14).



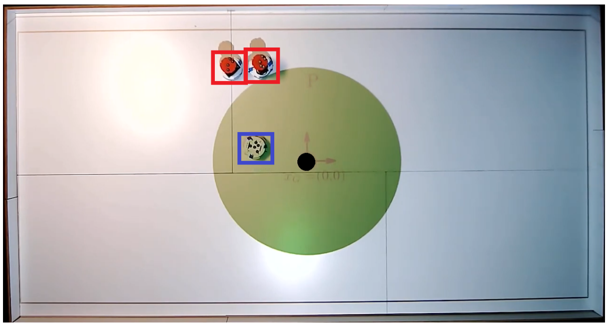
(a) $t = 0s$



(b) $t = 6s$

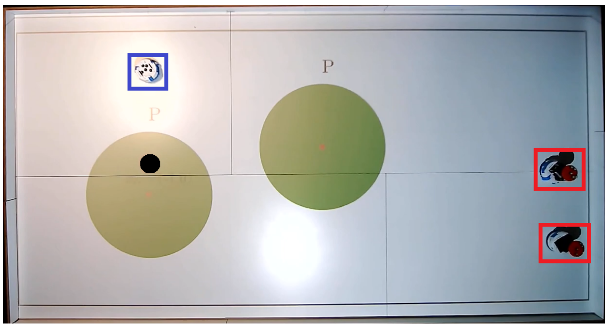


(c) $t = 26s$

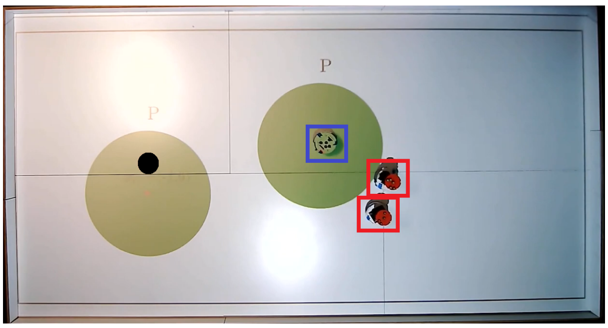


(d) $t = 35s$

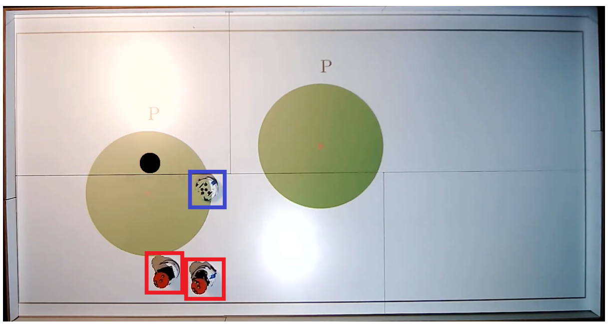
Figure 5.3: Experiments for Centralized Control: Hardware experiment with one dog robot preventing two sheep from breaching of the protected zone.



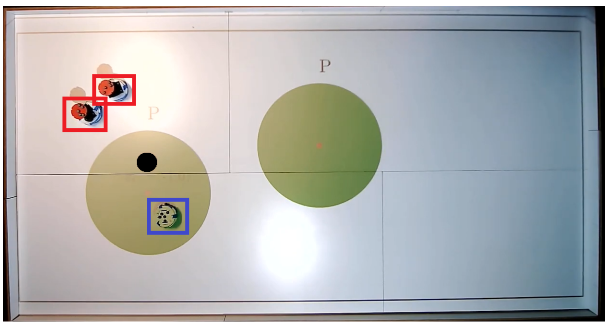
(a) $t = 0s$



(b) $t = 7s$

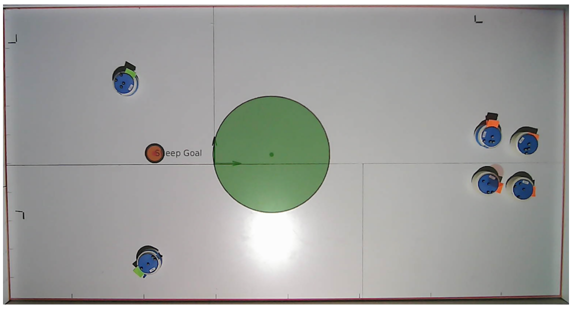


(c) $t = 10s$

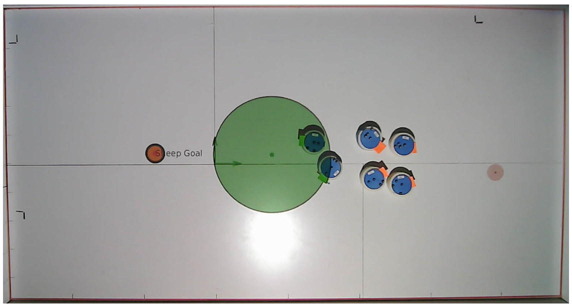


(d) $t = 14s$

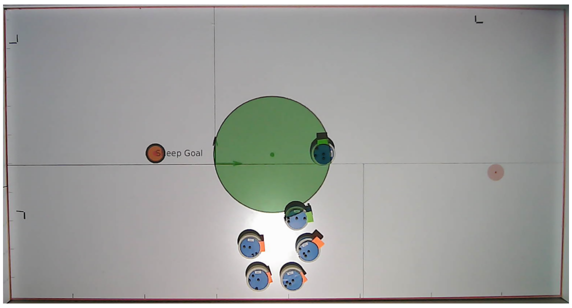
Figure 5.4: Experiments for Centralized Control: Hardware experiment with one dog robot preventing two sheep from the breaching of two protected zones. The goal lies in the left most protected zone.



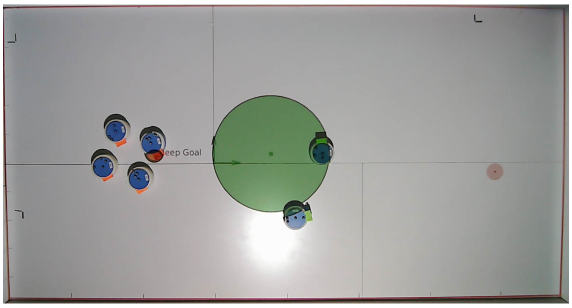
(a) $t = 0s$



(b) $t = 5s$

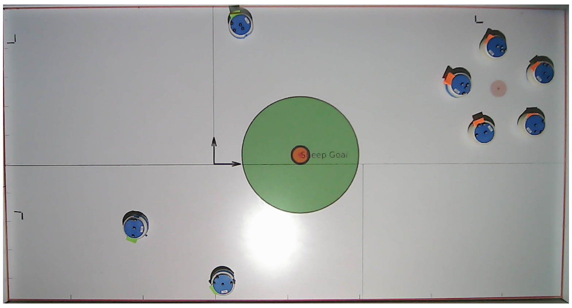


(c) $t = 12s$

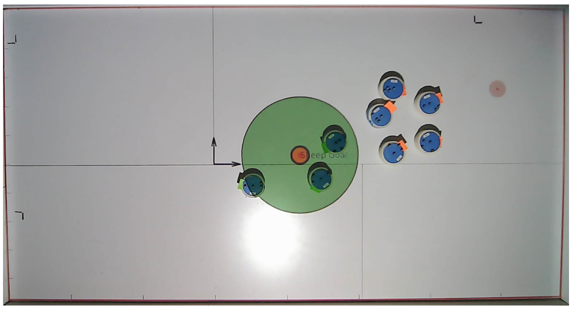


(d) $t = 30s$

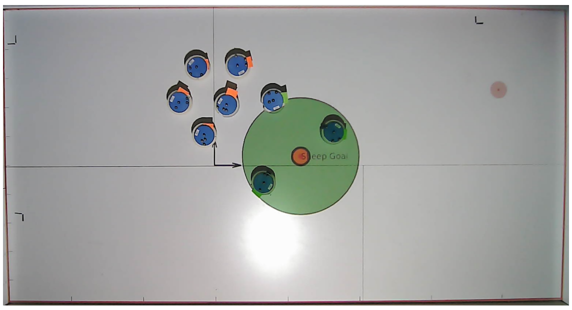
Figure 5.5: Experiments for Centralized Control: Two dogs defending the protected zone from four sheep using centralized control algorithm (4.14).



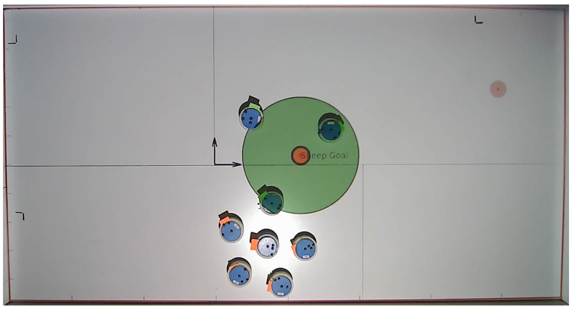
(a) $t = 0s$



(b) $t = 5s$

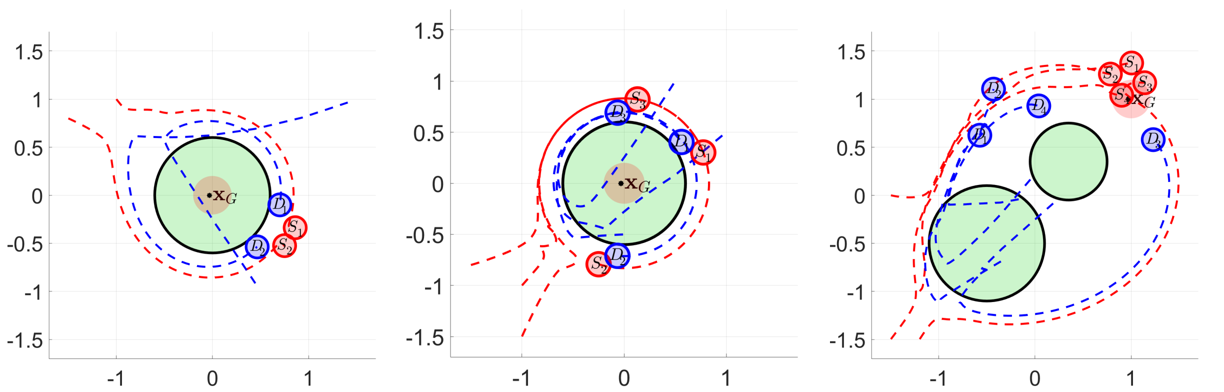


(c) $t = 30s$



(d) $t = 50s$

Figure 5.6: Experiment for Centralized Control: Three dogs (green-tailed robots) defending a protected zone from five sheep (orange-tailed robots) using centralized control (4.14).



(a) Two dogs v. two sheep.

(b) Three dogs v. three sheep

(c) Four dogs v. four sheep.

Figure 5.7: Decentralized Control Simulation: Preventing the breaching of the protected zone using our proposed distributed algorithm 4.20. Here dogs are shown in blue and sheep in red. The green disc represents the protected zone. The nominal task of the sheep is to go straight towards goal $\boldsymbol{x}_G$. In Fig. 5.7(c), we defend two protected zones from four sheep.

5.1.1 Decentralized Control

We first validate the first distributed algorithm and the feasibility proof given in section 4.4. For this, we model the sheep with the Reynolds-Boids dynamics (3.1) with gains $k_S = 0.5$, $k_G = 1$ and $k_D = 0.1$. The dogs use (3.15) to compute their velocities, where hyperparameters α and β are computed following (3.7) and (3.10). We chose a circular protected zone of radius $R_p = 0.6$m and center $\boldsymbol{x}_P$ at origin. The sheep are initialized outside of the protected zone, and their goal location $\boldsymbol{x}_G$ is chosen such that their nominal trajectory would make them breach the zone, thus necessitating intervention from dogs. The positions of dogs are initialized randomly within a certain range of the protected zone. In figures 5.7(a) and 5.7(b), we show two examples involving a) two dog robots vs. two sheep robots and b) three dog robots vs. three sheep robots. To demonstrate the compositionality of our approach, we consider two protected zones in figure 5.7(c) where we have four dogs defending both zones from four sheep. In all these simulations, none of the sheep breach any zone, thus demonstrating the correctness of our approach.

5.2 Monte Carlo Simulations

We further study the performance of the proposed control strategy by using Monte Carlo simulations with varying initial configurations and varying number of sheep n and dog robots m. The values of the constants in sheep dynamics are $k_G = 1$, $k_S = 0.3$ and $k_D = 0.08$. We vary n and m from 1 to 10 and for a given pair of (n, m) we run the simulation for a hundred times with a random initialization of $\boldsymbol{x}_0 = (\boldsymbol{x}_{S_1}(0), \cdots, \boldsymbol{x}_{S_n}(0), \boldsymbol{x}_{D_1}(0), \cdots, \boldsymbol{x}_{D_m}(0))$ in every run. Table 5.1 reports these results. Each entry of this table reports the percentage success rate *i.e.* in how many cases the sheep got diverted away from the protected zone. As can be seen, almost all entries are 100, which proves the success of our algorithm. The failure cases correspond to scenarios when during the transition from the initial configuration to the final configuration, the QP becomes infeasible in certain cases and hence leads to breaching of the protected zone. When the QP becomes infeasible, we assign the dog robots to have zero velocity. Further, we considered the impact of including collision avoidance constraints. These results are reported in Table 5.2. Because of additional constraints, it is possible that collision avoidance conflicts with the defending constraint. As a result, we do not observe as good successes in this case compared to when there are no collision avoidance constraints.

Table 5.1: Performance of the proposed strategy with varying number of sheep and dog robots. Here, we did not consider collision avoidance constraints *i.e.* the dogs were allowed to run into the sheep.

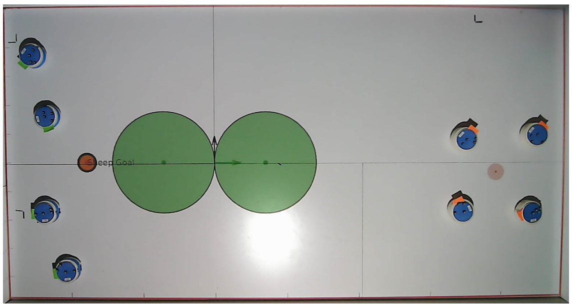
$N_S N_D$	**2**	**4**	**6**	**8**	**10**
2	100	100	100	100	100
4	100	100	100	100	100
6	100	98	100	100	100
8	100	98	100	100	98
10	100	98	98	100	96

Table 5.2: Performance of the proposed strategy with varying number of sheep and dog robots. Here we considered collision avoidance constraints in the dynamics of the dogs.

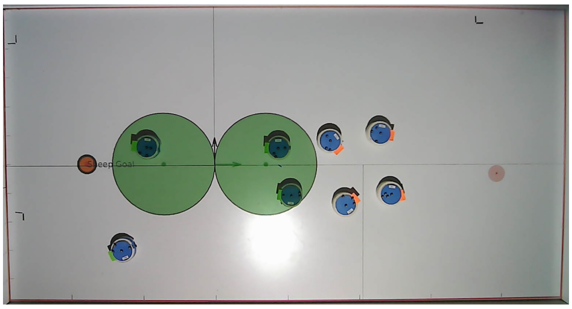
$N_S N_D$	**2**	**4**	**6**	**8**	**10**
2	72	99	99	100	100
4	62	74	90	97	100
6	28	83	99	99	100
8	63	82	100	100	100
10	70	79	90	91	94

5.3 Hardware Experiments

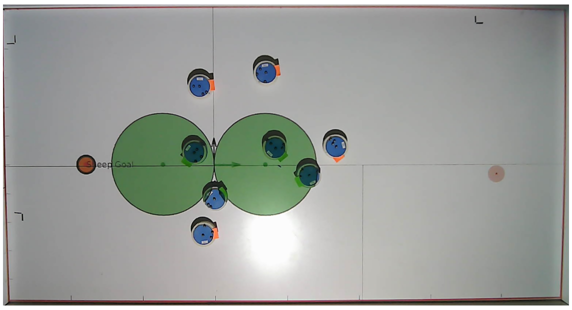
Finally, we tested our algorithm in robots in the multirobot test arena in our lab. It consists of a 14ft $\times$ 7ft platform, several Khepera IV robots and addition-



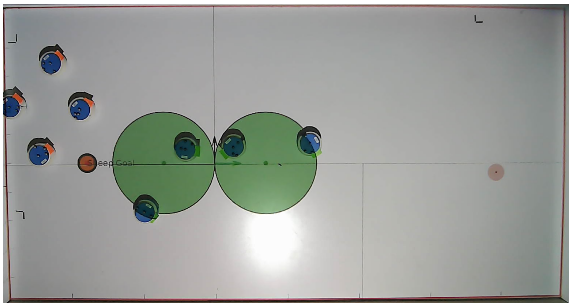
(a) $t = 0s$



(b) $t = 6s$

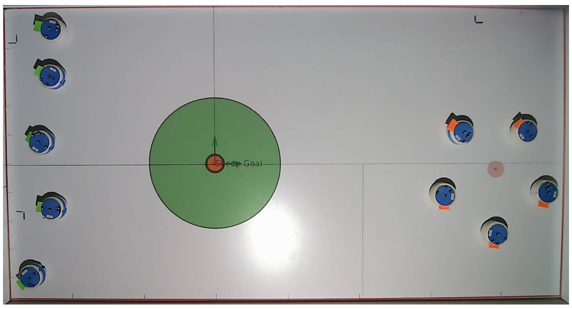


(c) $t = 12s$

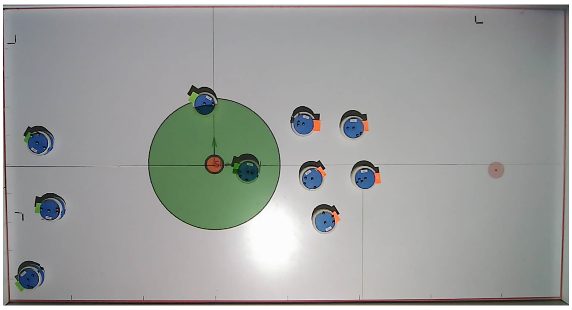


(d) $t = 20s$

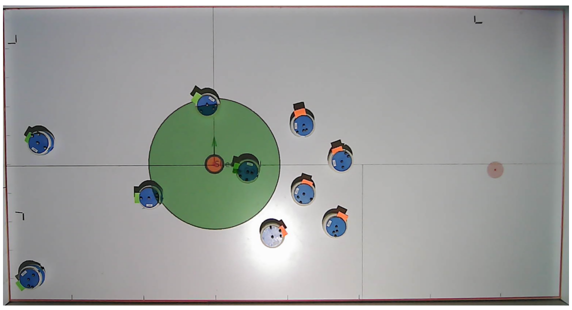
Figure 5.8: Experiment for the distributed algorithm: Four dogs (green-tailed robots) defending two protected zone from four sheep (orange-tailed robots). The goal position x_G (red disc) is in extreme left that would encourage sheep to breach both zones. However, our proposed algorithm moves the dogs so that none of the zones get breached.



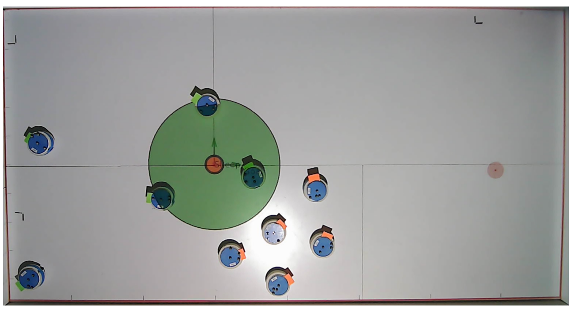
(a) $t = 0s$



(b) $t = 12s$

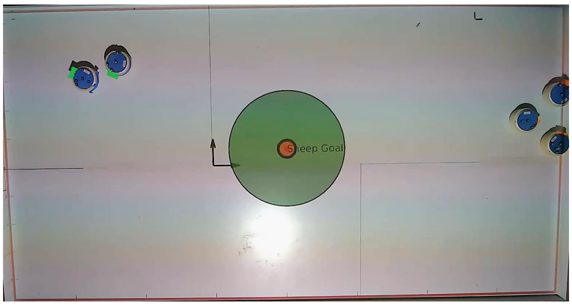


(c) $t = 25s$

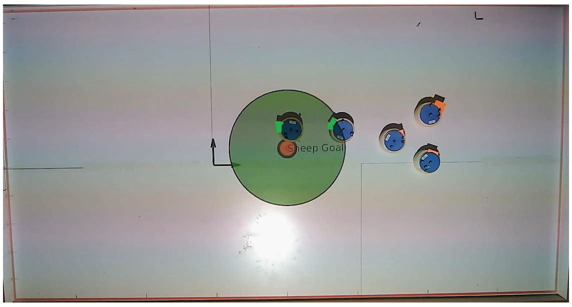


(d) $t = 40s$

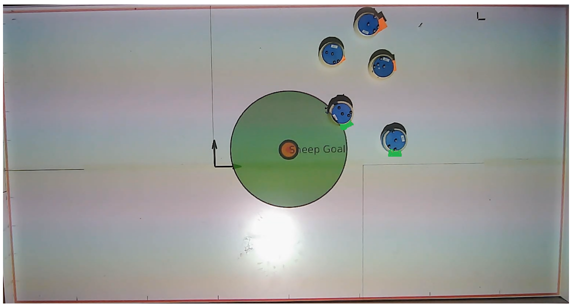
Figure 5.9: Experiment for the distributed algorithm: Five dogs (green-tailed robots) defending the protected zone from five sheep (orange-tailed robots). The sheep's goal (red disc) is in the center of the protected zone. Eventually, in this scenario a deadlock occurs where all sheep come to a stop outside the protected zone.



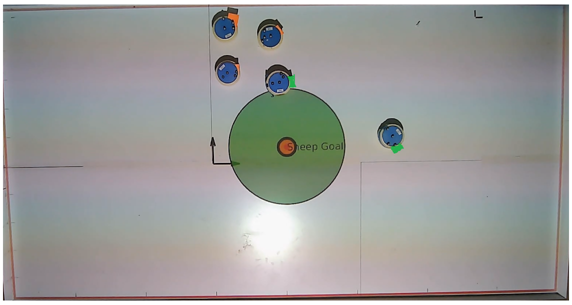
(a) $t = 0s$



(b) $t = 4s$

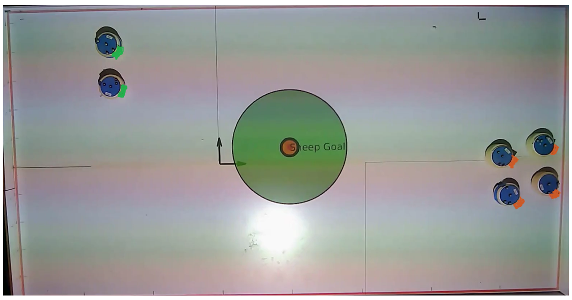


(c) $t = 15s$

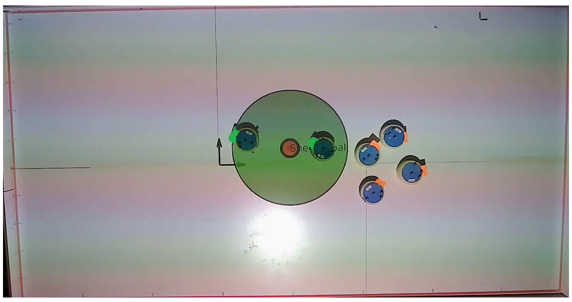


(d) $t = 30s$

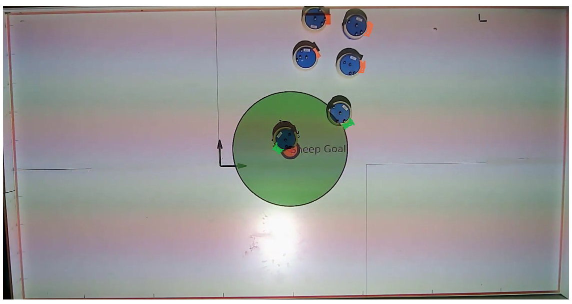
Figure 5.10: Experiment for distributed algorithm: Two dogs (green-tailed robots) defending the protected zone from three sheep (orange-tailed robots). The goal position x_G (red disc) is at the center of the zone.



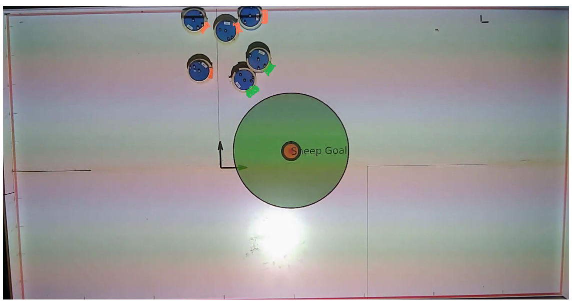
(a) $t = 0s$



(b) $t = 4s$



(c) $t = 15s$



(d) $t = 30s$

Figure 5.11: Experiment for distributed algorithm: Two dogs (green-tailed robots) defending the protected zone from four sheep (orange-tailed robots). This case is similar to the one shown in fig. 5.5.

ally eight Vicon cameras for motion tracking. All control inputs are computed on a desktop and conveyed to the robots over WiFi. While we developed our algorithms assuming that the dynamics of all agents are single-integrator based, the robots have unicycle dynamics given by

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} v\cos\theta \\ v\sin\theta \\ \omega \end{pmatrix} \tag{5.1}$$

Thus, we do a minor adjustment to map the inputs computed from our algorithms to the angular speed and forward translational speed of these robots. This is done by considering a point at a distance d on the x_b axis of the body frame of the robot:

$$\begin{aligned} \boldsymbol{x} &= \begin{pmatrix} x + d\cos\theta \\ y + d\sin\theta \end{pmatrix} \\ \implies \dot{\boldsymbol{x}} &= \underbrace{\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & d \end{pmatrix}}_{M} \begin{pmatrix} v \\ \omega \end{pmatrix} = \tilde{\boldsymbol{u}} \\ \implies \begin{pmatrix} v \\ \omega \end{pmatrix} &= M^{-1}\tilde{\boldsymbol{u}} \end{aligned} \tag{5.2}$$

For the robots representing the sheep, $\tilde{\boldsymbol{u}}$ is obtained from (3.1) while for the robots representing the dog, $\tilde{\boldsymbol{u}}$ is obtained from (3.15). In Fig. 5.2, we have one sheep (in red box) and one dog robot (in blue box). The protected zone is highlighted in green and the goal and center of the protected zone are the black dot. We use (3.15) to compute the velocity of the dog robot and convert it to angular speed and forward translational speed using (5.2). As can be noted from the snapshots, the dog robot is able successfully defend the zone from the sheep. Next we consider multiple sheep in Fig. 5.3. As can be seen from the snapshots, in this case, the dog is able to defend the zone from both sheep. Finally, in Fig. 5.4 we demonstrate that our approach is can deal with multiple protected zones simultaneously. In this figure, we purposefully kept the goal of the sheep in the left most protected zone. This way, the sheep would be incentivized to breach both the protected zones. Figure 5.5 shows a case with 2 dog and 4 sheep robots. This figure shows the performance in the case of an underactuated system, i.e, there are more sheep against less number of dogs. Another example is shown in figure 5.6 where 3 dogs successfully prevent breaching against 5 sheep robots. Yet still, our algorithm is able to find velocities for dogs to defend both zones from both sheep.

5.3.1 Decentralized Control

In this section, we show the results obtained by performing robot experiments by implementing the distributed algorithms. Additionally, we also present more experimental results for our prior centralized algorithm. We conduct these experiments in our lab's multi-robot arena, which consists of a 14ft $\times$ 7ft platform with multiple Khepera IV robots and eight Vicon cameras for motion tracking.

Multiple experiments were conducted using the distributed algorithm, which requires equal numbers of dogs and sheep. Figure 5.8 shows 4 dog robots against 4 sheep robots scenario. Here we take two protected zones and show that the dogs can protect both of them. This highlights the compositional nature of our proposed algorithm. We conducted experiments with 5 dog robots and 5 sheep robots, as shown in Figure 5.9. Here we can see some dog robots did not require to move as the assigned sheep were being prevented from entering the protected zone due to the configuration of the flock itself. Finally, we test our distributed algorithm. Figure 5.10 shows a case where 2 dogs prevent the breaching of protected zone against three dogs. This highlights that our distributed approach can handle under-actuated scenarios. Figure 5.11 and figure 5.5 can be compared to see both centralized and distributed algorithm handling a similar scenario of 2 dogs against 4 sheep.

References

[1] R. D'Andrea, "Guest editorial: A revolution in the warehouse: A retrospective on kiva systems and the grand challenges ahead," *IEEE Transactions on Automation Science and Engineering*, vol. 9, no. 4, pp. 638–639, 2012.

[2] R. D'Andrea and G. E. Dullerud, "Distributed control design for spatially interconnected systems," *IEEE Transactions on automatic control*, vol. 48, no. 9, pp. 1478–1495, 2003.

[3] W. Kazmi, M. Bisgaard, F. Garcia-Ruiz, K. D. Hansen, and A. la Cour-Harbo, "Adaptive surveying and early treatment of crops with a team of autonomous vehicles," in *Proceedings of the 5th European Conference on Mobile Robots ECMR 2011*, 2011, pp. 253–258.

[4] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 693–703, 2007.

[5] J. Lin, A. S. Morse, and B. D. Anderson, "The multi-agent rendezvous problem-the asynchronous case," in *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, vol. 2. IEEE, 2004, pp. 1926–1931.

[6] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, 1987, pp. 25–34.

[7] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 3420–3431.

[8] F. Ferraguti, C. Talignani Landi, S. Costi, M. Bonfè, S. Farsoni, C. Secchi, and C. Fantuzzi, "Safety barrier functions and multi-camera tracking for human–robot shared environment," *Robotics and*

Autonomous Systems, vol. 124, p. 103388, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889019306426

[9] M. Jankovic and M. Santillo, "Collision avoidance and liveness of multi-agent systems with cbf-based controllers," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 6822–6828.

[10] Y. S. Quan, J. S. Kim, and C. C. Chung, "Robust model predictive control with control barrier function for nonholonomic robots with obstacle avoidance," in *2021 21st International Conference on Control, Automation and Systems (ICCAS)*, 2021, pp. 1377–1382.

[11] A. E. Chriat and C. Sun, "On the optimality, stability, and feasibility of control barrier functions: An adaptive learning-based approach," 2023.

[12] J.-M. Lien, O. B. Bayazit, R. T. Sowell, S. Rodriguez, and N. M. Amato, "Shepherding behaviors," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 4. IEEE, 2004, pp. 4159–4164.

[13] A. Pierson and M. Schwager, "Controlling noncooperative herds with robotic herders," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 517–525, 2017.

[14] R. Vaughan, N. Sumpter, J. Henderson, A. Frost, and S. Cameron, "Robot control of animal flocks," in *Proceedings of the 1998 IEEE International Symposium on Intelligent Control (ISIC) held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA) Intell.* IEEE, 1998, pp. 277–282.

[15] ——, "Experiments in automatic flock control," *Robotics and autonomous systems*, vol. 31, no. 1-2, pp. 109–117, 2000.

[16] A. Pierson and M. Schwager, "Bio-inspired non-cooperative multi-robot herding," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1843–1849.

[17] R. A. Licitra, Z. I. Bell, E. A. Doucette, and W. E. Dixon, "Single agent indirect herding of multiple targets: A switched adaptive control approach," *IEEE Control Systems Letters*, vol. 2, no. 1, pp. 127–132, 2017.

[18] R. A. Licitra, Z. D. Hutcheson, E. A. Doucette, and W. E. Dixon, "Single agent herding of n-agents: A switched systems approach," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 14 374–14 379, 2017.

[19] E. Sebastián and E. Montijano, "Multi-robot implicit control of herds," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 1601–1607.

[20] M. Bacon and N. Olgac, "Swarm herding using a region holding sliding mode controller," *Journal of Vibration and Control*, vol. 18, no. 7, pp. 1056–1066, 2012.

[21] J. S. Grover, C. Liu, and K. Sycara, "Parameter identification for multirobot systems using optimization based controllers (extended version)," *arXiv preprint arXiv:2009.13817*, 2020.

[22] J. Grover, C. Liu, and K. Sycara, "Feasible region-based identification using duality (extended version)," *arXiv preprint arXiv:2011.04904*, 2020.