BusyBees: Safe Controllers for Multi-Agent Swarms

Joshua Durham (jbdurham)

jbdurham@andrew.cmu.edu

1 Abstract

From fleets of industrial robots maneuvering autonomously through warehouses to flocks of drones coordinating their motions to perform large tasks, multi-agent swarms are on the cutting edge of advanced robotic systems. However, ensuring that a robotic swarm behaves in a safe and efficient manner is a particularly challenging task as individual agents must avoid collision while still remaining in a swarm-like formation. In this report, we initially define controllers for 2-agent systems in Differential Dynamic Logic (d$\mathcal{L}$) that ensure that the agents remain in close contact with one another in order to preserve the swarm-like behavior of the system while also ensuring that no two agents collide while maneuvering throughout their environment. We also present systems modeled in Quantified Differential Dynamic Logic (Qd$\mathcal{L}$) that illustrate how to model and control systems with an arbitrary number of agents. These controllers are shown to apply to a variety of swarm systems, particularly train-like swarms where each agent remains in close to their adjacent neighbors, and clustered swarms where the agents cluster around a single leader agent or some macroscopic feature of the swarm.

2 Introduction

Understanding the dynamics and control strategies of multi-agent systems is a key problem in the domains of robotics and biology. From a pack of wolves navigating together towards a goal by following the commands of the alpha male to swarms of bees clustering around the position of the queen bee, the dynamics and structure of various biological systems can often be closely modeled by mathematics and hybrid controllers that combine continuous dynamics with discrete controls. As a result, numerous strategies for controlling robotic systems with multiple agents have pulled inspiration from these biological models, and these robotic systems are analyzed to examine how individual agents have an impact on the larger group [Parpinelli, 2011]. By understanding how individual agents can control and direct the overall nature of multi-agent systems, we can develop a better understanding of how to control swarms of agents in a safe and efficient manner.

2.1 Overview of Hybrid System

The general hybrid system that we propose to study will be defined as an arbitrary swarm of n agents that traverse through $\mathbb{R}^2$ while constrained by two key distance relations. First, a swarm controller must ensure that no two agents move too close together as this would result in a collision, and the collision of two agents at a high velocity will almost certainly result in irreparable damage. This constraint is important since real-world swarms and multi-agent systems cannot be merely modeled as systems of point masses moving through space; every agent has some finite volume that only that agent can occupy.

The second constraint that we must consider is that agents cannot separate themselves from the swarm by a significant distance in order to ensure that the system still behaves in a swarm-like manner. This constraint is included to model the drawbacks of a single agent straying away from the swarm as a whole. For example, a 2015 study found that diseased wolves that lived on their own had a death rate five times higher than diseased wolves that remained with a pack of other wolves, suggesting that the support of the group as a whole is beneficial to the survival of each individual agent [Almberg et al., 2015]. A more physical implication of this maximum distance constraint would be to consider interconnected robotic agents, such as robots tethered to one another by electrical cables of fixed length or needing to be within a fixed distance of one another in order to communicate information through wireless signals, and violating the distance constraint would result in an agent losing its supply of power or wireless connection. Additionally, this maximum distance constraint forces the design of more nuanced and complex controllers. Without the maximum distance constraint, this would allow for the design of controllers that avoid collision by directing each agent along a path away from all other agents, resulting in relatively dull systems with no interesting swarm dynamics.

3 Modeling Decisions

3.1 Agent Dynamics

We will model our swarm of n agents as a system of n dynamically homogeneous agents where each of the n individual agents has the same physical dynamics. The hybrid programs presented in this report are for a fixed $n = 5$ agents, but inductive arguments are given for modeling systems with an arbitrary n agents. In order to focus on the industrial and robotic applications of swarm controllers, we will model each agent as a wheeled robot that can traverse freely through the plane in $\mathbb{R}^2$ with circular dynamics. Agents with circular dynamics are used rather than agents with holonomic dynamics (able to instantaneously accelerate in any direction) due to the fact that agents with circular dynamics such as cars and differential drive robots are much more commonly found in real-world environments rather than the idealized holonomic agent. We will model each robot as being able to accelerate at any value on the interval $[-B, A]$, where $A > 0$ and $B > 0$, and as a simplifying assumption we will assume that each agent is only able to move with a nonnegative velocity. This assumption of nonnegative velocities is justified since a wide variety of wheeled robotic agents such as autonomous cars and vehicles that cannot readily move backwards at all times. Additionally, we will impose that each agent also has the same maximum velocity v_{max} that it can attain, and this property will be crucial for designing and proving safe controllers. This assumption of a maximum velocity is justified as well since due to the physical constraints of friction, air resistance, and other mechanical inefficiencies, no robotic agent that can traverse the surface of the Earth is capable of an infinitely large velocity while complying with the laws of physics.

In order to model the circular dynamics of each individual agent, we assume that each agent moves along a circular path of radius r that it can change during each control cycle, and changing the sign of r allows for the agent to reverse the direction it moves along the circle from clockwise to counterclockwise or vice versa. We make the simplifying assumption that the agent can move in a nearly straight line by picking a sufficiently large radius. Additionally, we make the assumption that each robot has a minimum turning radius magnitude of $c_{min} > 0$ in order to simplify the complexity of our proofs while still properly modeling wheeled agents that cannot turn in place such as autonomous cars. Based upon the above modeling decisions and justified assumptions, we can model the dynamics of each individual agent as:

$$\{x' = v \cdot d_x, y' = v \cdot d_y, v' = a, d_x' = -\frac{v \cdot d_y}{r},$$

$$d_y' = \frac{v \cdot d_x}{r}, t' = 1 \& v \geq 0 \land v \leq v_{max} \land t \leq T\}$$

Where (x, y) is the current position of the robot, (d_x, d_y) is a unit vector pointing the the current direction of travel of the robot, r is the radius of the path that the robot is moving on, v is the robot's current linear velocity, a is the robot's currently selected linear acceleration, and t is a variable for monitoring the amount of time that has passed during the continuous evolution of the system.

While the dynamics of each individual agent in this system is relatively similar to the agents with circular dynamics modeled in previous classwork and labs, the addition of the maximum distance constraint results in agents with slightly more constrained dynamics. One particularly nuance of bounding the velocity of each agent to $0 \leq v \leq v_{max}$ is that whenever one agent hits either $v = 0$ or $v = v_{max}$, the continuous evolution of the system stops and a new loop iteration begins since the evolution domain constraint will no longer be satisfied if continuous evolution continues. This is a simplifying assumption that we must unfortunately make due to the fact that allowing for the continuous evolution to continue and letting the velocity constraint be violated results in significantly more complex models and proofs. Additionally, the modeling assumptions made here also make the assumption that all agents make their control decisions at the same time and are synced with the same central timekeeping system. This modeling assumption is justified since robotic agents in close proximity can synchronize their clocks and actions with relatively high precision, and the alternative of granting each agent its own clock and making decisions based upon its own internal timekeeping results in much more complex models.

3.2 Safety Properties

In order to fully define the two safety properties of minimum distances to avoid collision and maximum distances to preserve swarm-like behavior, we need to formally model these properties for each agent. Let us model our minimum distance safety property by saying that no two agents can move within radius r_{min} of each other, so for all pairs of agents i and j, $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \geq r_{min}$. Since we have assumed that each agent is modeled identically, this minimum distance assumption indicates that each real-world agent can be circumscribed by some minimum size circle with radius $\frac{r_{min}}{2}$ that is centered at the axis of rotation for each agent, so two agents will collide when their axes of rotation move within r_{min} units of one another. This circumscribed circle of radius r_{min} allows us to ensure that regardless of the orientation of our arbitrarily shaped agent, as long as no two agents move closer than r_{min} units of one another, no two agents will ever collide.

Our second safety property of maximum distances will be specific to each swarm model that we will study. In each swarm model, we will define some r_{max} as the maximum distance between agents in the swarm in order to ensure that agents do not become too dispersed. The details of this specific upper bound distance will be discussed in the context of each model in the subsequent section.

In future sections, we will refer to the first safety constraint as the "minimum distance safety constraint", and the second safety property will be the "maximum distance safety constraint" for consistency.

3.3 Swarm Models

The three primary swarm models that we will model and analyze controllers for are a train-like swarm, a heterogeneous clustered swarm, and a homogeneous clustered swarm. Each system closely represents biological swarms and has important applications to mechanical and robotic systems controls. The primary difference in each swarm model will be how we define the maximum distance safety constraint, and this variation in safe

In a train-like swarm, we ensure that no two agents collide while ensuring that neighboring agents are always within r_{max} units of one another. We will define the maximum distance constraint for agent i with its neighboring agents in a chain, so for agent i, the maximum distance to each agent $i-1$ and agent $i+1$ must be upper bounded by r_{max}. This system best models biological systems where agents move in a relatively sequential fashion, such as the migration patterns of spiny lobsters where a group of lobsters form a queued line and follow the motion of a leader while traversing across the ocean floor in order to minimize drag forces from ocean currents [Kanciruk and Herrnkind, 1978]. Additionally, the system can also model a chain of tethered robots that are connected sequentially together by electrical cables, and one agent cannot move too far away from either of its neighbors or else the entire system will lose power.

The second swarm system we would like to model is a cluster-based model around some central leader agent . Without loss of generality, we can define agent 1 to be the agent in which all of the other agents must cluster around. That is, for any agent i with position (x_i, y_i), the maximum distance safety constraint is $\sqrt{(x_1 - x_i)^2 + (y_1 - y_i)^2} \leq r_f$. This model best represents the biological system of a swarm of bees clustered around the queen bee, where each individual bee must not move too far away from the leader queen bee [Parpinelli, 2011]. Additionally, we could model this system as a group of robotic agents that all need to be within some fixed distance of a leader agent in order to ensure that they can wirelessly communicate with the leader.

The final swarm system we will focus upon will be a homogeneous swarm that utilizes the dynamically changing center of mass of the swarm as the clustering point of the system, where we make the assumption that each agent has the same mass. The goal will be to meet the same safety constraints and modeling considerations as in the previous system heterogeneous system clustered around a leader, except utilizing the center of mass of the system as the clustering point rather than some leader agent. This system is commonly studied in robotics and controls research due to the fact that the actions of individual agents can adjust the dynamics of the swarm as a whole by changing the macroscopic aspect of the center of mass through an individual agent's motion [Sartoretti et al., 2014].

4 Related Work

For the vast majority of multi-agent swarm controllers discussed in literature, probabilistic techniques are the general focus for modeling and controlling large swarms of agents. However, these control techniques are relatively difficult to model and understand within the KeYmaera X framework, an automated proof solving environment that allows for proving the validity of models presented in d$\mathcal{L}$. Additionally, the agents in these probabilistically modeled multi-agent swarms are generally assumed to have holonomic dynamics, which is an infeasible assumption for the agents with circular dynamics that we wish to model. Thus, we will look to these probabilistic models for inspiration, but we will ultimately have to make approximations and simplifications to each model and controller in order to ensure that we can safely control and prove each system within the context of d$\mathcal{L}$ and KeYmaera X. Additionally, d$\mathcal{L}$ allows for much more rigorous and precise proofs of correctness (rather than showing that probabilistic control models ensure properties of a swarm system with high probability [Sartoretti et al., 2014]), allowing us to move carefully through the control decisions made by each agent and illustrate that the actions of each agent are safe and valid.

4.1 Probabilistic Models and Vector Fields

As shown in prior research, a general control strategy for heterogeneous clustered swarms is to utilize a noisy input signal (usually a White Gaussian Noise source) to introduce variance in the position of the leader of the swarm, and then feed this noisy input position of the leader into a Kalman-Bucy filter that acts as a controller for each individual agent, and variations in the noisy input will result in the agents being clustered around the leader as desired [Sartoretti et al., 2014]. These probabilistic controllers also model swarms as if they were an arbitrary collection of point masses, and as such these controllers often fail to ensure that no two agents collide during any run of the system. As a result, while probabilistic models have their merits of being able to model and represent the large-scale dynamics of arbitrarily large swarms with millions of agents, the simplifying assumptions made to represent each agent as a point mass avoids the issue of ensuring that no two agents collide during a run of the real-world controller.

In regards to the homogeneously clustered swarm, previous literature [Eren and Açıkmeşe, 2017] has shown that the probability density distribution of homogeneous swarms can be controlled via velocity fields, so we can attempt to model how the center of mass of the swarm changes over time and thus use this as a clustering point that will drive agents towards the center of mass of the swarm (i.e. we know the center of mass M is $\frac{1}{\sum_{i=1}^{n} m_i} \sum_{i=1}^{n} m_i \mathbf{x_i}$, so $M' = \frac{1}{\sum_{i=1}^{n} m_i} \sum_{i=1}^{n} m_i \mathbf{x_i}'$, and we can use this term to model how the center of mass changes during the continuous evolution). This key insight of modeling the center of mass of a swarm in terms of the actions of the other agents allows us to apply the control strategies we generate for heterogeneous swarms to homogeneous swarms as well since the dynamics of the center of mass are constrained by the dynamics of each individual agent as well. Additionally, we are able to define differential invariants on the range of positions and velocities that each agent can obtain during the continuous evolution of the system, and these invariants are also extremely similar to the invariants we can apply to the center of mass of the system (i.e. if the maximum velocity of any agent in our swarm during the continuous evolution is v, we know that the maximum velocity of the center of mass is v as well since it is infeasible to have the center of mass move faster than the fastest agent in the system).

4.2 Barrier Certificates for Collision Avoidance

Recent studies have provably shown that localized control strategies utilizing control barrier certificates can ensure safety in multi-agent systems by having each agent only focus upon the dynamics of nearby agents that could potentially collide with the agent making control decisions [Borrmann et al., 2015]. A barrier function $\mathcal{B} : \mathcal{C} \to \mathbb{R}$ maps each element $x \in \mathcal{C}$, where $\mathcal{C} \subset \mathbb{R}^n$ to a real value where for each $x \in C$, if x is in the interior of $\mathcal{C}$, $\mathcal{B}(x) \geq 0$, and if x is on boundary of $\mathcal{C}$, $\mathcal{B}(x) = \infty$. Additionally, we can define barrier functions that are invariant level sets where $\dot{\mathcal{B}}(x) \leq 0$ for all $x \in \mathcal{C}$, and the goal of defining the barrier function $\mathcal{B}$ in this manner is it allows to define the invariant $\dot{\mathcal{B}} \leq \frac{\gamma}{\mathcal{B}}$ where $\gamma > 0$ and as long as this constraint is satisfied, it can be proved that the barrier function $\mathcal{B}$ that satisfies these properties will result in the set $\mathcal{C}$ being forward invariant, so the system will remain in the set $\mathcal{C}$ for all runs of the system. Thus, we will be able to prove that for a system with nonlinear control dynamics (i.e. the dynamics of a system with holonomic control dynamics), we can define barrier certificates that serve as invariants that preserve properties of the system. This concept of defining barrier certificates allows for a new method of conceptualizing the set of states that satisfy the safety conditions desired as $\mathcal{C}$, and showing that the existence of such a barrier function $\mathcal{B}$ for the set $\mathcal{C}$ will result in the system being provably safe.

This research models agents as having holonomic dynamics rather than the rotational dynamics we establish in our models, but the researchers suggest that utilizing holonomic double integrator dynamics can be used to approximate nonholonomic robot dynamics. The assumption of modeling the system in this manner results in the barrier certificate for avoiding collision between two agents as

$$\mathcal{B}_{ij} = (\frac{((x_i - x_j)(v_{x_i} - v_{x_j}) + (y_i - y_j)(v_{y_i} - v_{x_j}))}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}} + \sqrt{2(2A)(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} - r_{min})}^{-1}$$

for each pair of agents A_i and A_j where $i \neq j$. While the barrier certificate itself is relatively straightforward to define based upon the kinematics of the system, the actual implementation of a controller that ensures the invariant of these barrier certificates is significantly more complex. Additionally, modeling and proving controllers that utilize these barrier certificates in KeYmaera X with d$\mathcal{L}$ is extremely challenging. As a result, these barrier certificate controllers cannot be directly applied to control and ensure the safety of our systems despite these barrier certificate methods being the current state-of-the-art for swarm controllers that provably avoid collisions during all runs. However, a key result from this prior research is illustrating how agents can collaboratively work together in order to ensure the safety and correctness of the system. For example, the barrier certificates and controllers presented allow for two agents moving directly towards one another to cooperate in their acceleration decisions in order to avoid collision by having each agent base their control decisions on the states of neighboring agents as well, essentially doubling the maximum braking acceleration allowed to avoid a collision.

5 Proposed Solutions

For each of the above proposed swarm models, we present a series of increasingly complex controllers and models that represent the physical dynamics of each system as well as the safe control decisions that each individual agent can make.

5.1 Train-like Swarm

We will first focus upon the train-like swarm where a system of n agents will be associated to each other in a sequence, a multi-agent swarm with a defined leader that is heavily studied in literature [Krishnanand and Ghose, 2005]. That is, for agent i, the distance from agent i to both agent $i-1$ and agent $i+1$ will be bounded by some upper bound r_{max}. We will prove a series of simpler problems before moving towards the more difficult general case.

Two Agent Train Problem in $\mathbb{R}$

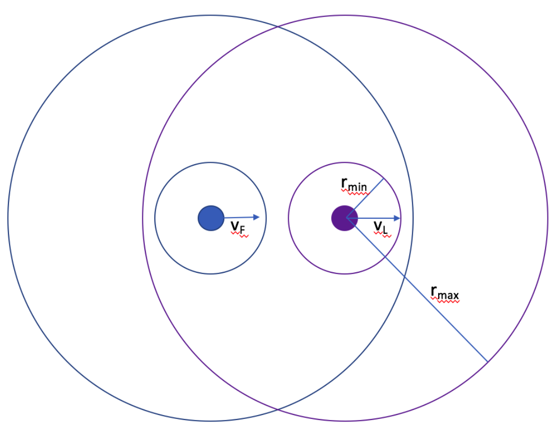

Figure 1: A leader and a follower agent traversing along a line. The inner circle for each agent represents the circle of collision that no two agents can cross, and the outer circle represents the maximum distance that two adjacent agents can be apart

The first system to model and prove is two agents maneuvering in a straight line while avoiding colliding with each other and also ensuring that the two agents do not move too far apart and violate the maximum distance safety constraint. Note that in this model we do not utilize the cooperative aspects of the follower knowing the leader's acceleration choice since it is not necessarily required to ensure the safety of the system when using the given controller. The general control strategy is that the follower agent is controlled based upon the worst-case scenario control decisions made by the leader agent (i.e. the leader accelerates with maximum acceleration A, endangering the maximum distance constraint, or the leader brakes with minimum acceleration $-B$, resulting in the two agents colliding if the incorrect control decisions are made). The dynamics of each agent are simplified in this case, and as a result we have the continuous evolution for the leader and follower agents as

$$\{x'_F = v_F, x'_L = v_L, v'_F = a_F, v'_L = a_L, t' = 1$$

$$\& v_F \geq 0 \land v_L \geq 0 \land v_F \leq v_{max} \land v_L \leq v_{max} \land t \leq T\}$$

This system has been fully modeled in KeYmaera X, and the resulting models and controllers bring forth several insights about the assumptions we must make in order to ensure that the desired safety constraints are satisfied. The first insight is about the necessity of each agent needing to have a maximum velocity v_{max} in order to ensure the compliance of each agent with the maximum distance constraint. Consider the leader and follower agents with position, velocity, acceleration x_L, v_L, a_L and x_F, v_F, a_F, respectively, where $v_L > v_F$ initially and both distance safety constraints are satisfied as well (so $(x_L - x_F) \geq r_{min} \land (x_L - x_F) \leq r_{max}$ initially). Since the leader's acceleration choices do not take into account the positions or velocities of the follower agents behind it, in the worst case scenario the leader agent can always choose to set $a_L := A$ and accelerate with the largest possible value. When the follower makes their control decision about their acceleration, the follower can accelerate with at most $a_F := A$ as well, so the velocity of the follower agent will never be able to increase enough in order to match the velocity of the leader agent, and thus the two agents will continue to move further apart over time. Thus, without the maximum velocity safety constraint, the two agents can become arbitrarily far apart and the maximum distance safety constraint will be violated.

In order to prevent the above case from occurring, we impose a maximum velocity v_{max} on each agent as discussed in the section on modeling decisions. The addition of this maximum velocity constraint allows for us to define our loop invariants and control decisions that will ensure the safety of the two agent system. First, we must ensure that there is sufficient distance between the two agents to allow for braking to a stop, which can be represented by

$$(x_L - x_F) + \frac{(v_L - v_F)v_L}{B} - \frac{(v_L - v_F)^2}{2B} \geq r_{min}$$

Where the first term is the initial distance between the two agents, the second term is change in distance as both agents brake with magnitude B and move closer together, and the third term represents the additional distance covered by the follower agent after the lead agent has reached velocity $v_L = 0$. Similarly, we can model the safety invariant for the maximum distance by

$$(x_L - x_F) + \frac{(v_L - v_F)(v_{max} - v_L)}{A} + \frac{(v_L - v_F)^2}{2A} \leq r_{max}$$

These two constraints will be shown to be the loop invariants for our system since they show that all of the safety constraints will always be satisfied and that the follower never enters some state where the leader agent has the opportunity to make control decisions that result in the safety constraints being violated in some later loop iteration. Based upon these loop invariants, we can now make control decisions for the follower agent's acceleration based upon the worst-case scenario acceleration choices made by the leader agent. If the follower agent chooses an arbitrary acceleration a_F on the interval $[-B, A]$, we must ensure that

$$((x_L + v_LT + \frac{-BT^2}{2}) - (x_F + v_FT + \frac{a_FT^2}{2})) + \frac{((v_L - BT) - (v_F + a_FT))(v_L - BT)}{B} - \frac{((v_L - BT) - (v_F + a_FT))^2}{2B} \geq r_{min}$$

and

$$((x_L + v_LT + \frac{AT^2}{2}) - (x_F + v_FT + \frac{a_FT^2}{2})) + \frac{((v_L + AT) - (v_F + a_FT))(v_{max} - (v_L + AT))}{A} + \frac{((v_L + AT) - (v_F + a_FT))^2}{2A} \leq r_{max}$$

are satisfied for the acceleration choice a_F. Note that these two above control guards are derived from the physical kinematics of the system running for at most time T, the maximum time between two control decisions made by the follower agent. These control guards are sufficient since they illustrate that even in the worst-case scenario of the leader agent making the worst case control decisions for time T, the loop invariants will still be satisfied after the continuous run of the system and thus the follower's control decision is safe.

The proof for this system is relatively straightforward since we are still in the realm of solvable ODE's and each agent's control decisions are based upon the solutions to these ODE's. A nearly complete proof for this system (a few cases are left as simple provable real arithmetic that takes KeYmaera X a long time to prove) is included in the deliverables.

Two Agent Train Problem in $\mathbb{R}^2$

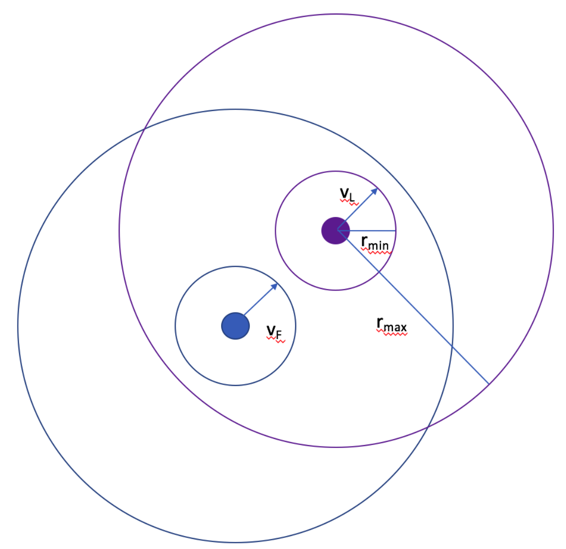

Figure 2: A leader and a follower agent traversing through the plane

We then increase the complexity of the problem to occur in $\mathbb{R}^2$ with two agents and develop a second time-triggered controller to model this system. While the addition of another physical dimension is relatively straightforward to model with circular dynamics, the control decisions for this system are significantly more challenging primarily due to maximum distance safety constraint. The complexities of this problem arise from a variety of different modeling decisions, and these problems and potential solutions are discussed below.

One potential issue that arose during the design of the controller for this system was the possibility of the leader agent choosing a sufficiently tight turning radius that allows for the leader to circle around and end up behind the follower agent, resulting in a significantly more complicated system since the leader agent does not generally consider collisions with its follower agents when making a control decision. There are a few different ways to overcome this challenge, such as requiring the minimum turning radius c_{min} to be large enough such that with respect to the maximum velocity v_{max} of each agent, we never have to worry about the leader agent being able to make a tight enough turn fast enough that results in the above case of the leader colliding with a follower agent. Another possible assumption would be to fix the direction of travel of the entire swarm, such as saying that the swarm only has the goal of traveling in the positive x direction, and ensure that for all agents i and j, $i < j$ (so i is closer to the front of the train and agent 1, the leader agent), $x_i - x_j \geq r_{min}$, and require that no agent can move with a velocity in the negative x direction at any time (so $d_x \geq 0$ for every agent).

Once we have overcome this issue of the leader agent being able to collide with the follower agent, we now need to resolve the grand issue of satisfying the maximum distance safety constraint in a collaborative control scheme. Consider a leader and follower agent that initially satisfy both the minimum and maximum distance safety constraints, and suppose the two agents have arbitrary orientations and velocities between 0 and v_{max}. In Lab 4 and other modeling problems, we utilized the infinity norm as a lower bound on the distance between an agent and the obstacle, and we supposed the worst case scenario that the robot is moving directly towards the obstacle. We proved that as long as the robot is safe in this worst-case scenario of moving directly towards the obstacle, the system as a whole will be provably safe. However, in the case where we have two agents with arbitrary orientations and velocities moving through a workspace while constrained by some maximum distance r_{max}, it appears that this method of ensuring safety in the worst case scenario is no longer feasible as identifying the worst-case scenario is significantly more complex.

Suppose that the two agents are facing directly opposite directions while moving with arbitrary velocities. If the minimum turning radius c_{min} of these two agents is sufficiently large or the agents have too fast of velocities, it will be impossible for the follower agent to turn around quickly enough in order to satisfy the minimum distance safety constraint while the leader agent makes arbitrary control decisions. Even if the leader agent announces their control decisions to the follower agent before the follower makes their control decisions, it becomes an extremely challenging task to correctly identify the turning radius and acceleration that the follower should choose in order to satisfy the maximum distance safety requirement. The immense amount of reasoning required about turning radii and how they affect the distances between two agents during the continuous evolution of the system makes this problem significantly more difficult than merely satisfying the safety property that no collisions should occur.

In order to overcome this issue of satisfying the maximum distance constraint with two agents, we will have to develop some relatively tight upper bounds for the distance that each agent covers while moving with rotational dynamics. For example, we know that if an agent at position (x, y) is moving with velocity v and accelerates with acceleration a, the agent can move anywhere in the region from $x - vT - \frac{aT^2}{2}$ to $x + vT + \frac{aT^2}{2}$ in the x direction and $y - vT - \frac{aT^2}{2}$ to $y + vT + \frac{aT^2}{2}$ in the y direction (assuming that we use the infinity norm as an approximation of distance). This approximation is relatively simple, but it is not sufficient for proving that the maximum safety distance is satisfied due to the fact that each agent's orientation plays a key role in satisfying the system's safety requirement. Suppose that each agent has an infinitely large minimum turning radius so they can only move in a straight line. In this case, if the two agents do not have the same initial orientation and do not operate cooperatively together, then after some arbitrarily long amount of time, the two agents will inevitably diverge unless they both come to a stop (since the agents will follow paths along two skew lines that will become further and further apart over time).

Thus, this further analysis of the problem has brought forth two key insights. First, the leader agent needs to cooperatively make their control decisions based upon the follower's position, velocity, and orientation in order to ensure the safety of the system. In the two robots moving along skew straight lines scenario discussed above, if the leader does not take the follower's position and velocity into account and makes acceleration decisions arbitrarily, the controller will not be safe since the agents will ultimately move too far away from each other. However, if the lead agent takes into account the follower's position and velocity and makes control decisions that benefit both agents, the issue of satisfying both safety constraints becomes more attainable.

Second, further analysis of the rotational dynamics of the system is ultimately required in order to ensure the safety of the system due to the fact that orientation and the minimum turning radius c_{min} have immense effects on the behavior and decisions that each agent can make. One potential modification that may grant further insights would be to modify the modeling constraints about minimum turning radius by making it proportional to the velocity of the agent, i.e. $c_{min} \sim v$ or $c_{min} \sim v^2$, so as an agent increases its velocity, the agent must move on straighter paths. This modeling assumption would still be justified, as real-world robotic agents cannot move along tight curves at high speeds due to centrifugal forces.

We have modeled the given system in $\mathbb{R}^2$ in KeYmaera X with a few additional simplifying assumptions in order to ensure the safety of the system. The primary assumption that we make is that every agent has the same orientation and velocity in the system's initial state. This assumption closely models the systems built and modeled by other researchers [Krishnanand and Ghose, 2005], and allows us to make collaboratively safe control decisions within the swarm that ensure that both safety constraints are satisfied at all times.

The general proof technique for proving the correctness of this controller is the use of differential cuts and differential invariants that insert as much information about the kinematics of the system into the evolution domain constraint as possible. For example, the presented control paradigm ensures that after each arbitrary iteration of the loop and throughout the continuous evolution, each agent has the same velocity and orientation, and this information can be added to the evolution domain constraint through simple differential cuts and subsequent proofs by differential invariants. The addition of these constraints about each agent having the same velocity and orientation at all times allows for finishing off more complex challenges such as illustrating that the two agents are never going to collide with one another (if two agents are moving with the same orientation and velocity at all times, the distance between the two agents will never differ) and similarly that the two agents will always satisfy the maximum distance constraint at all times.

While this controller is relatively simplistic as it makes coordinated assumptions about the leader driving the control decisions made by the follower agent, it allows for a provably safe system that allows for the swarm to move safely throughout the plane based upon the actions of the leader agent. Additionally, this controller grants insights into how collaborative actions between agents make for much more feasible control strategies in multi-agent swarms. The collaborative aspect of controlling multi-agent swarms in this fashion indicates how future controller design and research should focus upon the leader agent making control decisions that allows for the follower agents to make control choices that do not result in failures of the system's safety constraints.

n Agent Train Problem in $\mathbb{R}$

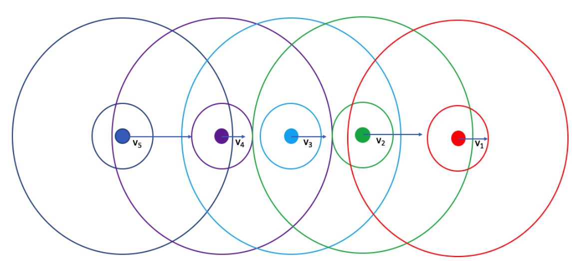

Figure 3: A subset of an n-agent system traversing along a line

This controller and model is an extension of the Two Agent Train Problem in $\mathbb{R}$, except now we will attempt to prove the controller for an arbitrary number of agents. The controller outlined in the two agent case is sufficient and safe for controlling the actions of an arbitrary n agents, except rather than considering the Leader and the Follower agent, we consider the control decisions for agent i based upon the possible acceleration decisions for agent $i-1$ (the agent directly in front of agent i). We know that the 2-agent controller is safe for an arbitrary number of agents due to the fact that agent i is able to make safe control decisions regardless of what choices the other agents in the chain make (the safety checks on the nondeterministic assignments ensure that even in the worst case scenarios of maximum or minimum accelerations by other agents, all control decisions made by agent i are safe).

The proposed system and controller is modeled in KeYmaera X with a system of $n = 5$ agents for simplicity. Since KeYmaera X currently does not have Quantified Differential Logic (QdL), we will work with a multi-agent system of finite size when illustrating and proving the properties of our controllers and systems.

With the use of Qd$\mathcal{L}$, it becomes feasible to model and control an arbitrary system of n agents. The two key differences in the hybrid program models presented in d$\mathcal{L}$ and the systems modeled in Qd$\mathcal{L}$ are the introduction of quantified ODE's to handle the continuous evolution of an arbitrary agent in the system, and the quantified assignment to express how an arbitrary agent i can make safe control decisions regardless of the control decisions made by agent $i-1$ directly in front of it. Thus, for each agent, the control decision can be written as

$$ctrl \equiv \forall i : Ca(i) := *; ?(a(i) \leq A \wedge a(i) \geq B$$

$$\wedge closeSafetyConstraint(i-1, i)$$

$$\wedge farSafetyConstraint(i-1, i))$$

and the continuous evolution can be written as

$$evol \equiv t := 0; \forall i : Cx(i)' = v(i), v(i)' = a(i),$$

$$t' = 1 \& v(i) \geq 0 \wedge v(i) \leq v_{max} \wedge t \leq T$$

while the overall hybrid program still has the same general structure of $safe \rightarrow [(ctrl; evol)^*](safeClose \wedge safeFar)$ (the same general structure of the models presented in d$\mathcal{L}$). The $closeSafetyConstraint(i-1, i)$ and $farSafetyConstraint(i-1, i)$ are the same control guards for the follower agent in the 2-agent controller previously discussed, except now for agent $i-1$ as the leader and agent i as the follower, and $safeClose$ and $safeFar$ are first order logic formulas that check that for all pairs of agents $(i-1, i)$, where $i \in C$, both the minimum distance safety constraint and the maximum distance safety constraint are satisfied.

The general proof for proving this system of an arbitrary n agents in Qd$\mathcal{L}$ proceeds in a similar manner to the 2-D case while also proving some more nuanced properties brought forth by prior research into provably safe systems modeled in Qd$\mathcal{L}$ [Loos, Platzer, Nistor 2011]. For example, since each agent is able to satisfy the safety constraint relative to the agent directly in front of it, a transitivity proof is necessary to show that each agent satisfies the collision-free property for the entire system while also ensuring that neighboring agents satisfy the maximum distance safety constraint. The proof would proceed with an initial application of Gödel's generalization rule to divide the proof into the local case between an arbitrary agent i and agent $i-1$, which has been shown to be true in our simpler proof of the 2-agent case, and the global case which ensures correctness of the entire system. The fact that the control decisions of agent i are only dependent on the state of agent $i-1$ makes the resulting proof very modular and similar to the 2-agent case, resulting in a very scalable proof that can easily be made more realistic by improved controllers and dynamics in the 2-agent case, which can then be quickly applied to the n agent case.

n Agent Train Problem in $\mathbb{R}^2$

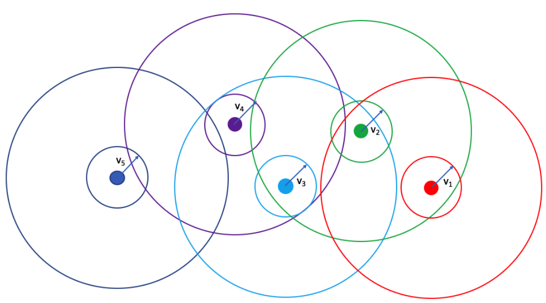

Figure 4: A subset of an n-agent system traversing through the plane

Finally, we can expand the results of the three simpler problems to modeling an arbitrary number of agents in $\mathbb{R}^2$. We have modeled the basics of the model in KeYmaera X for a system of $n = 5$ agents, but we have made the simplifying assumption that every agent has the same initial orientation and velocity in the system's initial state. However, the significant issues associated with modeling and proving correctness of simpler controllers has left this much more complex system in a modeled yet unverified state. Additionally, the arguments for proving this system for an arbitrary number of agents in Qd$\mathcal{L}$ are significantly more complex as additional assumptions must be made about the relations between agents in order to make the same inductive reasoning discussed in the n-agent system in $\mathbb{R}$. For example, the chain of agents can wrap around on itself, such as in a figure 8 formation, resulting in the fact that ensuring that agent i never collides with another agent can depend on more than just the actions of agent $i-1$, since the leader of the chain (agent 1) and other agents now have the potential to collide with agent i.

5.2 Heterogeneous Clustering

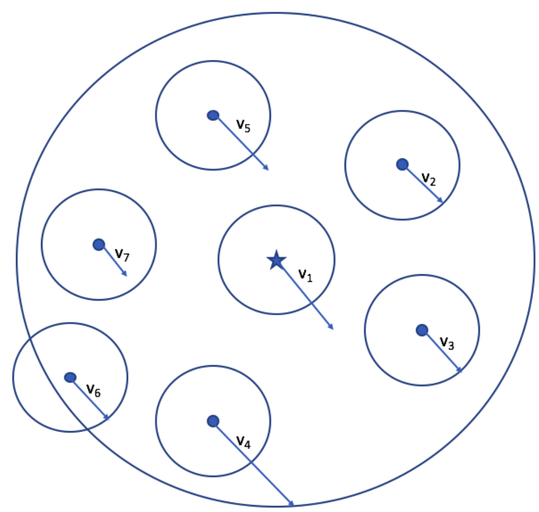

Figure 5: A subset of an n-agent system clustered around the central leader (denoted with a star), and the large outer circle around the central agent illustrates the maximum distance that any agent can be from the central leader

The heterogeneous swarm is relatively similar to the n-agent train in $\mathbb{R}^2$ except for the new maximum distance safety constraint around the central leader agent. The model and a simple controller have been created in KeYmaera X, and we have modeled this system in a very similar manner to the n Agent Train Problem in $\mathbb{R}^2$ except with the new maximum distance safety constraint. The model and controller currently has the same additional assumption as found in the train swarms that each agent has the same initial orientation and velocity. Additionally, a partial proof has been completed to show how to progress through the steps of applying the results from the proofs of the 2 agent systems to swarms with a larger number of agents.

5.3 Homogeneous Clustering

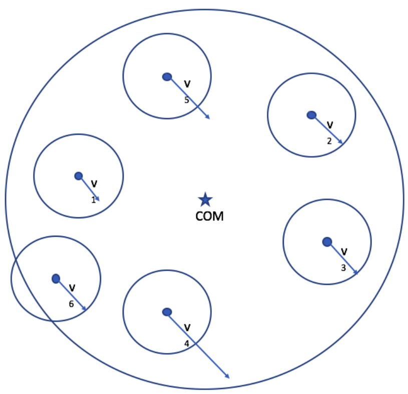

Figure 6: A subset of an n-agent system clustered around the center of mass of the system (denoted with a star), and the large outer circle around the center of mass illustrates the maximum distance that any agent can be from the center of mass

The homogeneous clustering problem follows similar modeling considerations as in the Heterogeneous Clustering system, except utilizing the center of mass of the system as the clustering point. The model and a simple controller for this system have been based closely off of the Heterogeneous Clustering model and controller, except we now model the center of mass of the swarm with position (M_x, M_y), and we have modeled the differential properties of the center of mass appropriately (the center of mass M is $\frac{1}{\sum_{i=1}^{n} m_i} \sum_{i=1}^{n} m_i \mathbf{x_i}$, so $M' = \frac{1}{\sum_{i=1}^{n} m_i} \sum_{i=1}^{n} m_i \mathbf{x_i}')$.

6 Goals and Deliverables

For each of the above hybrid systems, associated deliverable files are available that illustrate how to control and model each swarm system.

`.kyx` files have been written for each of the above proposed systems, and associated proof `.kya` files are included as well:

- `2_Agent_Train_In_1D.kyx` and `2_Agent_Train_In_1D.kya`
- `2_Agent_Train_In_2D.kyx` and `2_Agent_Train_In_2D.kya`
- `n_Agent_Train_In_1D.kyx`
- `n_Agent_Train_In_2D.kyx`
- `n_Agent_Heterogeneous_Cluster_In_2D.kyx` and `n_Agent_Heterogeneous_Cluster_In_2D.kya`
- `n_Agent_Homogeneous_Cluster_In_2D.kyx`

Note that some of the n agent systems do not have associated `.kya` files since the n agent proofs are extremely tedious and do not grant any significant insights that can concluded through the 2-agent cases and inductive arguments.

7 Discussion and Future Applications

The results of this study lay the groundwork for how to model, control, and prove correctness of a variety of multi-agent swarms. While the complexity of these systems and the extremely large dimensionality of these problems made it difficult to achieve significant results, we hope that the results of this study have brought forth significant insights into the challenges that need to be overcome in order to completely prove the safety and correctness of these systems. Moving forward, a few key steps need to be made in order to design provably safe controllers for multi-agent swarm systems.

First, a deeper understanding of circular dynamics or approximations of agents with circular dynamics needs to be made in order to ensure that agents with rotational dynamics have provably safe actions. While the majority of research in the area of provably safe multi-agent swarm controllers makes the simplifying assumption that agents have holonomic dynamics [Borrmann et al., 2015], this approximation severely restricts the possibility of implementing these controllers on many real-world systems such as autonomous cars and differential drive robots that do not possess holonomic dynamics. This is the most crucial step into proving the safety of multi-agent systems with circular dynamics as no significant progress can be made in the area of more complex controller design without a greater understanding of how to ensure the safety of systems during their continuous evolution.

Additionally, we have shown that Qd$\mathcal{L}$ is a valuable tool for proving the safety of controllers for arbitrarily large multi-agent systems, and further work needs to be done to illustrate how to prove the safety of multi-agent swarms in the plane. While we have outlined and showed how Qd$\mathcal{L}$ can be used to prove the safety and correctness of n-agent systems traversing through $\mathbb{R}$, introducing a second dimension into the problem makes the system significantly more challenging to prove inductively. While we showed that in $\mathbb{R}$, agent i in a system only has to worry about the control decisions of agent $i-1$ and thus the global problem becomes a simplification of proving safety in the local 2-agent problem, expanding the problem to maneuvering in $\mathbb{R}^2$ no longer allows us to make this assumption. As a result, much more complex inductive arguments need to be made that reason about how agent i is safe regardless of the actions of all other agents in the system. However, barrier functions discussed in prior research will serve as an immensely valuable tool when moving forward with these more complex Qd$\mathcal{L}$ proofs. As shown in prior works [Borrmann et al., 2015], these barrier functions allow for the decoupling of the problem in to local pairwise cases, and then further simplifying the control problem into only ensuring safety for agent i relative to the neighbors of this agent, where the neighbors are defined as the agents that have the possibility of colliding with agent i in the near future. This process of only having to worry about colliding with the neighbors closest to agent i makes the proof process significantly simpler, but further work needs to be done with Qd$\mathcal{L}$ to ensure that these swarm controllers are safe.

Based upon the results of our above studies and the controllers we have designed, these systems will have direct applications to research in robotics, multi-agent control theory, and biology. As swarm robotics is becoming an increasingly relevant field due to its industrial applications, it is imperative that controllers for these agents need to ensure collision avoidance while operating in constrained environments such as a factory. The results of n Agent Heterogeneous Clustering model and controller could be directly applied to this situation by defining an additional "dummy" agent as the leader with a fixed position that never changes, and defining r_{max} (the distance which every agent must operate within) as the size of the factory or workspace, thus ensuring that the agent is constrained to be within the factory workspace and the designed controller will ensure that no agent ever leaves the workspace or collides with another agent.

Another interesting application of our work would be to apply the n Agent Train model to navigation of microscopic medical robots injected into the bloodstream of patients. Some initial assumptions would have to be made about approximating the workspace of the agents to being either one or two dimensional (i.e. the robots are large enough that they can fit only one at a time through a blood vessel or artery, so we can model the system as if we are in $\mathbb{R}$ and the agents move sequentially through the bloodstream), but thereafter the existing train models controllers could be used to ensure the safe motion of agents within a patient's body. An extension of the models and controllers discussed in this report would be to modify the controllers to be safe in environments with static obstacles (i.e. the walls of a blood vessel) and show that the agents can still navigate appropriately while avoiding collision.

Finally, others that study cyber-physical systems and robotics could extend the results of our study to model more complex and general swarm systems. The controller models in $\mathbb{R}^2$ could likely be modified and extended to work in $\mathbb{R}^3$, allowing for proving the safety of controllers of flying swarms such as groups of drones. Additionally, further controllers can be based off of our existing swarm controllers for ensuring safety when encountering both static and dynamic obstacles (similar to the controllers and proofs in Lab 4, but ensuring that no agent collides with the obstacle as well as the minimum and maximum distance safety properties of the swarm are still preserved as well).

References

[Almberg et al., 2015] Almberg, E. S., Cross, P. C., Dobson, A. P., Smith, D.W., Metz, M. C., Stahler, D.R., and Hudson, P.J. (2015). Social living mitigates the costs of a chronic illness in a cooperative carnivore *Ecology Letters* 18:660-667.

[Borrmann et al., 2015] Borrmann, U., Wang, L., Ames, A., and Egerstedt, M. (2015). Control barrier certificates for safe swarm behavior *International Federation of Automatic Control* 48-27:68-73.

[Eren and Açıkmeşe, 2017] Eren, U. and Açıkmeşe, B. (2017). Velocity field generation for density control of swarms using heat equation and smoothing kernels *International Federation of Automatic Control*, 50:9405-9411.

[Kanciruk and Herrnkind, 1978] Kanciruk, P. and Herrnkind, W. (1978). Mass migration of spiny lobster, Panulirus Argus (Crustacea: Palinuridae): behavior and environmental correlates *Robotics and Autonomous Systems*, 53:194-213.

[Krishnanand and Ghose, 2005] Krishnanand, K.N. and Ghose, D. (2005). Formations of minimalist mobile robots using local-templates and spatially distributed interactions *Robotics and Autonomous Systems*, 53:194-213.

[Loos, Platzer, Nistor 2011] Loos, S., Platzer, A., and Nistor, L. (2011). Adaptive cruise control: Hybrid, distributed, and now formally verified *International Symposium on Formal Methods*, 6664:42-56.

[Parpinelli, 2011] Parpinelli, R.S. (2011). New inspirations in swarm intelligence: a survey *Bio-Inspired Computation*, 1-16.

[Platzer, 2010] Platzer, A. (2010). Quantified differential dynamic logic for distributed hybrid systems *Computer Science Logic*, 469-483.

[Sartoretti, 2016] Sartoretti, G. (2016). Leader-based versus soft control of multi-agent swarms *Artificial Life and Robotics*, 21:302-307.

[Sartoretti et al., 2014] Sartoretti, G., Hongler, M., and Filliger, R. (2014). The estimation problem and heterogeneous swarms of autonomous agents *Stochastic Modeling Techniques and Data Analysis International Conference*.