Safe Robot Follow-the-Leader in the Plane
15-424 Foundations of Cyber-Physical Systems, Fall 2014

Austin Davis
Carnegie Mellon University

David Wise
Carnegie Mellon University

Abstract

Our project focuses on keeping a follower robot safe in the presence of obstacles by having it stay close to a leader robot that stays away from the obstacles. We precisely define the problem and possible applications. We also discuss related work that has been done previously. We create a number of models with various assumptions, some of which we were able to verify and some of which are quite difficult to work with. We outline the proofs for the models we verified, and we discuss the inherent difficulties with the models that we did not provide control decisions or verification for. Finally, we discuss further applications of our work.

1 Introduction

The study of multi-robot systems often becomes difficult and complicated when many robots and environmental factors come into play. Yet there is much to be gained from having multiple robots be able to interact and work together rather than having them be limited to independent control and decision-making. The specific application of one robot traveling behind another is commonly known as follow the leader. There are several presentations and papers that mention this application within the context of multi-robot cyber-physical systems, such as those given below. Some commonly desired properties of such systems involve following another robot closely and avoiding collisions.

1.1 Related work

The sophistication and performance of autonomous multi-robot systems continues to develop [4]. Current state-of-the-art robots employ various means of working together, such as tracking one another, communicating state data, or executing a global plan that involves multiple agents performing respective tasks or acting collectively. Flocking, the coordinated movements of multiple agents while avoiding collisions, is a relevant topic to our project and will be discussed more at the end of this paper.

1.2 Problem space

We plan to contribute to this space by proposing models and verifying safety for two-robot systems with enforced minimum and maximum follow distances. This contributes to the study of flocking by isolating the challenges associated with a single follower in a two-robot systems to avoid collisions. We also explore the variety of assumptions that can be made to simplify the problem space and their affect on the ease with which the safety properties can be verified. We chose this topic and area because of the wide variety of useful applications and extensions this model supports, as well as its relevane to current research.

2 System description

At a high level, our project addresses trying to keep a follower robot safe in the presence of obstacles by staying close to a leader robot. As long as the

leader can stay sufficiently far away from the obstacles, they can be stationary, moving, or even adversarially moving. Our work focused on keeping the follower safe assuming there was already a safe controller for the leader that stayed far from obstacles. The We looked specifically at the case of a single leader and follower, but extensions of our work could deal with more complex cases.

2.1 Scenarios

The scenario that we are considering involves two robots, a leader robot and a follower robot. We assume the robots are traveling in the 2D plane, and that there may be obstacles they need to avoid. We restrict the velocity and acceleration of the leader in our models to make its motion reasonably predictable by the follower, so that it will not stray too far away from the leader between control decisions.

2.2 Applications

We consider several possible applications and scenarios in which our models and safety properties would be useful.

Robots traveling in a group. If robots need to travel together, or one robot wants to escort another, the following distance properties would ensure smooth travel for both robots.

Path following. By maintaining the following distance properties, a second robot can roughly follow the leader along a path without sensing the path directly.

Robots with differing capabilities. Our model is ideal when the robots have different sensors or characteristics. For example, the leader robot can sense obstacles on behalf of the follower. The leader need not receive updates of the follower's status. The follower can execute its specialized purpose without regard for planning paths or avoiding obstacles.

Search and rescue. In light of the previous point, the model works well for robot pairs performing search and rescue applications, such as a scout robot and a rescuer robot.

Wireless inter-robot communication. When two robots are communicating wirelessly over a limited-range connection, it would be desirable to ensure a maximum following distance to avoid losing the connection and thus stranding one of the robots. This model is ideal for ensuring this property, while avoiding collisions. The wireless communication is also helpful for transmitting state data in real-time from the leader to the follower so the follower can make intelligent control decisions.

Robots tethered together. If one robot is tied or tethered to another robot via a flexible cable, the following distance properties from our model would be helpful and necessary to ensure safety. In this case, the minimum following distance safety property would guarantee that the two robots do not collide and could help prevent tangling of the cord. The maximum following distance safety property would help guarantee that the cord will not stretch or break, and that the cable's attachment points to each robot will not be damaged. This concept is not farfetched, as NASA has already developed a robot with two independently mobile components connected by a tether [1].

3 Modeling

Our models include a leader robot and a follower robot moving in the plane. We assume that we can draw a circle around each robot that contains its perimeter, so that if no obstacle or other robot interesects this circle then the robot is not colliding with another object. We represent the robots as points, and require that the distance between them is at least the sum of the radii of their bounding circles. We could easily inflate these radii as well to account for and avoid circular obstacles in the plane. Our various models have different restrictions on the movement of the two robots, which will be described more detail as each model is discussed.

3.1 Desired safety properties

The two important safety proporties for our models are that the follower does not collide with the leader,

but yet it stays close enough to it. We model this by setting constants $minfd$ and $maxfd$ (minimum follow distance and maximum follow distance) representing the minimum and maximum distances respectively that we allow the follower to be from the leader. If the radii of the bounding circles of the follower and the leader are $radF$ and $radL$ respectively, we would want $minfd > radF + radL$ to prevent collisions. If there is a controller for the leader that ensures that its center is at least $bufferL$ away from any point on an obstacle, then the closest any point on the follower can get to any point on an obstacle is $bufferL - maxfd - radF$. As long as

$$bufferL > maxfd + radF,$$

then the follower cannot intersect an obstacle if it succesfully stays within $maxfd$ of the leader. See Figure 1.

What makes this elegant is that the follower stays safe *regardless* of how the leader is able to stay away from the obstacles and what they are: there could be a lot of obstacles, and they could be moving. They could represent hazardous areas, other robots to avoid, or even missiles. As long as the leader can stay sufficiently far away, our work provides a way for the follower to stay safe.

3.2 Model restrictions

There are some restrictions that apply generally to all of our models. We assume that the follower has a time-trigger; i.e. it will make a control decision and then stick with that decision until its timer goes off again and tells it to make new observations and choose a new decision. The maximum amount of time between these updates is T. We also require the follower to be able to observe the leader's position (and, depending on the model, perhaps velocity) at each update time.

Additionally, we impose a maximum speed on the leader robot: this restriction allows us to bound how far away the robot will move in a certain amount of time from where we last observe it. In our most complete model, we also require that the leader cannot turn very sharply in the space of one time-triggered interval, to ensure that its behavior is reasonably predictable to the follower.

If we were to add obstacles to any of our models, we would of course require that the leader's center stays at least $bufferL$ away from any points on obstacles, where $bufferL > maxfd + radF$, as mentioned in the previous section. This ensures that neither the leader nor the follower will collide with an obstacle.

3.3 Model motivations

There are many useful properties of our model when the two safety properties are verified. These motivated our design of the model and our selection of the properties. The properties are simple to understand and express, but relatively versatile in practice.

Collision avoidance. We can apply the minimum following distance safety property to ensure the two robots do not collide with each other. Specifically, we can choose values of r_l and r_f such that the leader robot fits within a planar bounding circle of radius r_l and the follower within a circle of radius r_f. Then we can enforce a minimum following distance value at least the sum of the radii, $r_l + r_f$, which guarantees that the leader and the follower never collide with each other.

Obstacle avoidance. The maximum following distance property ensures that the follower stays close to the leader while the pair travels. If the leader uses a controller that maintains an appropriate minimum clearance away from obstacles, this would guarantee the safety of the follower from colliding with the obstacle as well. In a sense, the leader could protect the follower from obstacles without knowing anything about the follower's state over time.

Few planning and sensing requirements. Only the leader robot needs to sense obstacles or plan the route to travel, and the follower can benefit from this sensing and planning indirectly via our model.

No shepherding. In the model we present, the leader robot does not need to make many adjustments in order to be followed, nor does it

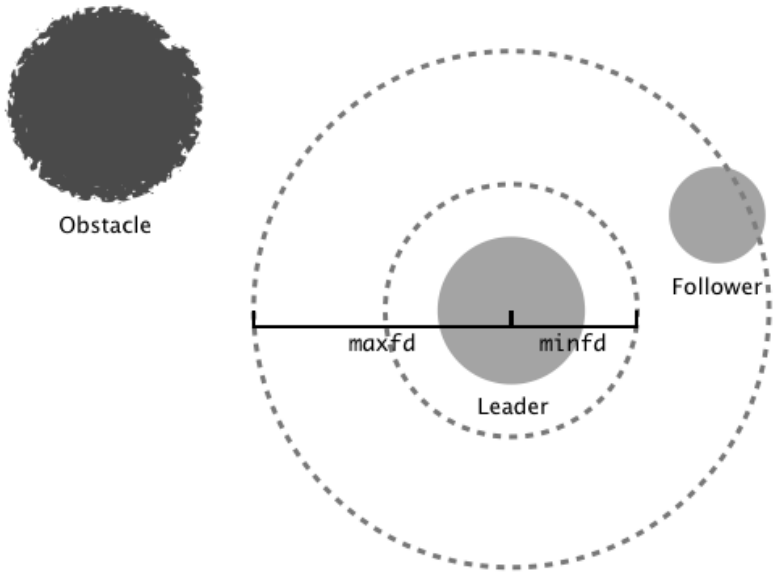

Figure 1: If the center of the follower stays between the two circles, it will stay safe.

need to worry about the state of the follower over time. Its behavior can be identical and reasonable whether or not it is being followed by a follower robot. This is a significant advantage for simplicity and elegance of the model, and also is helpful for further extensions of the model.

Robot teamwork. A number of robot "teamwork" scenarios motivate the maximum following distance, since robots that are close together can often collaborate more effectively than when they are far apart[1].

3.4 Background work

Some of the lab assignements we worked on in this class provided basic stepping stones for the work presented here. Lab 2 was about two cars moving in the positive x-direction of the x-axis, which is like our complete 1D model. However, even though the cars were not allowed to crash (which is like our minimum follow distance), there was no strict efficieny requirement corresponding to our maximum follow distance, which made control significantly easier.

Additionally, the Lab 4 controller would be a great starting point for the controller of a leader robot to be used by our system. This controller had the same type of motion as the leader in our intermedate and complete models, and could provably stay an arbitrary distance away from a single circular obstacle.

4 Simple models

To start off with something concrete that was easy to understand and verify, we created a couple of very simple models. The two models are essentially the same, although their assumptions are marginally different, and they have different controller implementations and associated proofs.

In the first model, we assume that the leader has a constant velocity, and that the follower can instanteously change its velocity. Whenever the follower makes a control decision, it sets is velocity to the leaders velocity. (Of course, this will not make a difference except for the initial control decision if it is not initially moving in the same direction as the leader). This way, the relative velocity between the follower and the leader is always 0. As long as the distance between the follower and the leader is initially between $minfd$ and $maxfd$, this will ob-

viously continue to hold, and so the model is safe. See Figure 2. This model was in fact so simple that it proved automatically in Keymaera with no interactive steps.

In our second model, we removed the assumption that the follower can instantaneously change its velocity, and replaced it with the assumption that it starts with the same initial velocity as the leader. This version would not automatically prove in Keymaera, but by creating a loop invariant using several of the initial conditions, it was easily verifiable.

5 Intermediate model

Our intermediate model is a stepping stone between our very simple, easy-to-verify models and our complete and unrestrictive but very-difficult-to-verify model. We make some assumptions that are not necessarily realistic, but which are designed to make the model reasonable to prove. By making these assumptions, we are able to prove safety, and the idea is that the techniques used in this proof might aid us in a more complicated proof that makes more realistic assumptions. Even with these assumptions, the proof of this model is fairly involved: it is more complicated than the proofs for any of the labs previously in this course.

5.1 Description

The intermediate model assumes that there is one leader robot and one follower robot in a 2D plane. For simplicity we assume no obstacles: however, if the leader's controller were replaced with one which could guarantee that it stayed a certain buffer distance away from all obstacles, then the follower would be safe because it stays close to the leader. (We would need the buffer distance to be at least the maximum allowed distance from the leader to the follower, plus corrections for the radii of the obstacle and the robots.)

We use a time-triggered model. Note that we are using the same time-trigger for the leader and follower, which while not realistic is not concerning, for reasons described in the next section.

The leader robot moves in circular arcs, so that its velocity is continuous. It has a maximum speed of $(maxfd - minfd)/T$, which we enforce as a domain constraint in the differential equation for simplicity but which could be easily enforced by not allowing any acceleration that could allow the robot to exceed this speed. The leader may choose any acceleration from $-B$ to A. However, we also assume that the leader may only move in a direction such that its y-velocity is nonnegative. The follower robot's motion is more restricted: at the start of every time-triggered interval, it is allowed to instantaneously adjust its speed and direction.

5.2 Limitations and possible solutions

The most unrealistic assumption in this model is that the follower can instantaneously adjust its velocity. As mentioned before, we made this assumption to make the problem more tractable so that we could better understand the type of arguments that might be necessary for a more complicated model. However, if the follower could adjust its velocity very quickly this might not be too unrealistic (e.g. if the robots are moving very slowly it could make the required velocity change very quickly). Additionally, even if it were not appropriate to assume that the follower could change velocity instantaneously, it might not be difficult to adjust this model to one where the leader and follower move in synchronized bursts, where they are stopped at the beginning and end of each burst. During the time between the bursts, the follower could adjust its direction to the desired direction for the next burst, and it would not have to instanteously change speed either.

An additional limitation is that the leader is only allowed to move in a direction such that its y-velocity is nonnegative. The problem that this is designed to address is that if the leader can turn around during the time-triggered interval, it could conceivably chase the follower. Then the follower would need a strategy that would guarantee that it is not too close to the leader, while at the same time not also being too far from the leader if it runs away from the follower, since the follower has no knowledge of the leader's current location during the interval. Ideally, as André suggested, we would rather assume that the leader may only make wide turns, which is a more realistic way of ensuring that the follower does not need to be afraid of moving to-

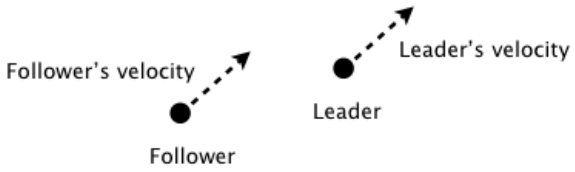

Figure 2: Motion in our simple models, where the follower and leader have the same velocity

wards the leader. In fact, if T and the minimum turn radius *minr* were related in such a way that the leader could turn by say at most 90° during an interval of length T, then we could probably modify the follower's current control decision without too much trouble so that it would still be safe in this model. If the leader can turn at most 90°, then the follower does not need to be afraid of moving towards a location just behind where it was at the start of the interval. However, in the interests of time and simplicity, we chose to keep the nonnegative y-velocity assumption for this model, because the proof was already fairly involved and we wanted to be sure our assumptions were simple enough for us to make reasonable progress.

Another simplication is that we allow both the leader and the follower to make a decision at the start of every time-triggered interval. This is not a realistic assumption unless the robots tried to synchronized their decision-making: if the leader's timer goes off and tells it to adjust its steering and acceleration, the follower would not also know to adjust its velocity at the same time. However, this is not a major concern because our proof does not rely on the fact that the leader does not change direction or acceleration during the time-triggered interval: we only need its maximum velocity to be restricted. If desired we could have separate time-triggers for the leader and the follower, which would greatly complicate the .key file and the proof, but without significantly altering its overall flow.

5.3 Description of follower's behavior

The follower must make a decision at the beginning of each time-triggered interval. Let (Fx, Fy) and (Lx, Ly) be the locations of the follower and leader respectively at the time it makes the control decision. The follower will adjust its velocity so that it moves at a constant velocity from (Fx, Fy) to $(Lx, Ly - minfd)$, at a speed such that it will reach $(Lx, Ly - minfd)$ exactly at time T is it is allowed to continue for the maximum allowed time T before its next control decision. See Figure 3

Essentially, the follower is chasing the point that is $minfd$ below the point where it last observed the leader. By chasing the point $minfd$ below the leader's last observed location instead of the last observed location itself, the follower avoids colliding with the leader. Since the leader cannot have a negative y-velocity, if we move towards this point then we cannot be closer than $minfd$ to the leader's location at any time.

Intuitively, this decision also ensures we stay close enough to the leader. To analyze this, consider the point $p = (lx, ly - minfd)$ which is always $minfd$ below the leader's current location. We want to stay within $maxfd - minfd$ of this point. We know inductively that this holds at the start of the loop iteration. It's reasonable that this will continue to hold during the continuous evolution because we move directly towards p so that we reach it after time T, and p cannot move faster than the leader's maximum velocity $(maxfd - minfd)/T$. For a rigorous proof, see the next section.

5.4 Proof sketch

In this section we present a sketch of the proof of safety for this model. Note that we successfully verified most of this proof in Keymaera, except for one last case which is not too hard to reason through but would be tedious to verify. The first condition we wish to show is

$$(lx - fx)^2 + (ly - fy)^2 \geq minfd^2.$$

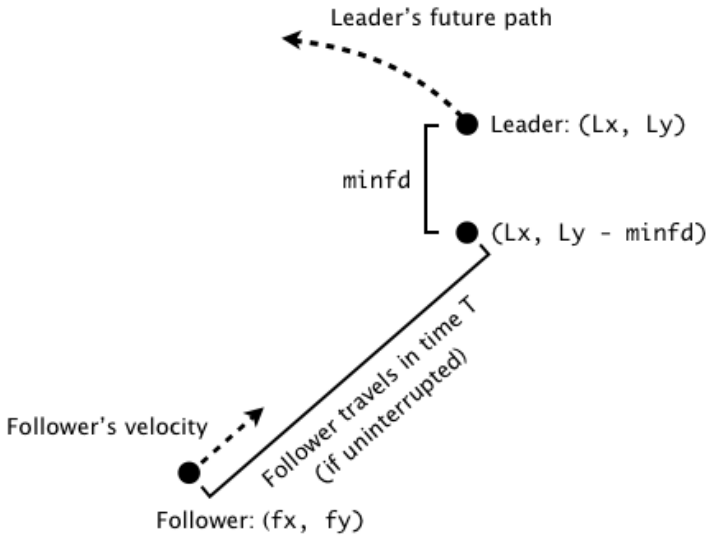

Figure 3: The follower's control decision in the intermediate model

We will instead prove the stronger statement $fy \leq ly - minfd$. We can show this with the loop invariant

$$A > 0 \;\&\; B > 0 \;\&\; T > 0 \;\&\; minr > 0 \;\&\; minfd > 0 \;\&\; maxfd > minfd \;\&\; ldy \geq 0 \;\&\; lv \geq 0 \;\&\; ldx^2 + ldy^2 = 1 \;\&\; ly - fy \geq minfd.$$

All but the last two of these conditions are trivial, since they involve only constants or they are guaranteed by the domain constraint of the differential equation. We can show $ldx^2 + ldy^2 = 1$ easily by the Differential Invariant proof rule. We would like to use a differential invariant to show $ly - fy \geq minfd$, but unfortunately the left-hand side might be decreasing and the left-hand side is constant. However, we know that the relationship is true because in the worst case, ly is constant (otherwise it is increasing), and fy is increasing at a constant rate until it reaches $ly - minfd$ at time T. Therefore, $fy + (T-t) \cdot fvy$ is is nonincreasing, so $fy + (T-t) \cdot fvy \leq ly - minfd$ is a valid differential invariant that implies $ly - fy \geq minfd$. This means we use $fy + (T-t) \cdot fvy \leq ly - minfd$ as a differential cut to prove $ly - fy \geq minfd$, and $fy + (T-t) \cdot fvy \leq ly - minfd$ is a differential invariant.

The other condition we wish to show is

$$(lx - fx)^2 + (ly - fy)^2 \leq maxfd^2.$$

As for the previous condition, we will instead show something stronger:

$$(lx - fx)^2 + (ly - minfd - fy)^2 \leq (maxfd - minfd)^2.$$

(This can be seen to be stronger by algebra, or by noting that the distance from (fx, fy) to $(lx, ly - minfd)$ is at most the distance from (fx, fy) to $(lx, ly - minfd)$ plus $minfd$.) Also as with the previous condition, we will use a loop invariant:

$$A > 0 \;\&\; B > 0 \;\&\; T > 0 \;\&\; minr > 0 \;\&\; minfd > 0 \;\&\; maxfd > minfd \;\&\; ldy \geq 0 \;\&\; lv \geq 0 \;\&\; ldx^2 + ldy^2 = 1 \;\&\; (lx - fx)^2 + (ly - minfd - fy)^2 \leq (maxfd - minfd)^2.$$

Again, all but the last two of these are trivial, and $ldx^2 + ldy^2 = 1$ is just a differential invariant.

We have shown these steps in the proof file for our lab. The remaining part of the box split left to show,

$$(lx - fx)^2 + (ly - minfd - fy)^2 \leq (maxfd - minfd)^2,$$

can be argued to hold without too much effort, but a formal proof would be very tedious. We want to bound the distance from (fx, fy) to $(lx, ly - minfd)$. Let (Lx, Ly) be the position of the leader at the start of this iteration of the loop. We will bound the

desired distance by the sum of the distance from (fx, fy) to $(Lx, Ly - minfd)$ and the distance from $(Lx, Ly - minfd)$ to $(lx, ly - minfd)$.

First, let us examine the distance from (fx, fy) to $(Lx, Ly - minfd)$. The second point is constant, and the first one is changing at a constant rate of $((Lx - fx)/T, (Ly - fy - minfd)/T)$. However, this means that the first point is moving straight towards the second point at a constant rate, and will reach it at time T. Therefore, if d_t is the distance from (fx, fy) to $(Lx, Ly - minfd)$ at time t, then

$$d_t \leq d_0 \left(1 - \frac{t}{T}\right) \leq (maxfd - minfd) \left(1 - \frac{t}{T}\right),$$

since at the start of the loop the distance is at most $maxfd - minfd$.

The second distance to analyze is the distance form $(Lx, Ly - minfd)$ to $(lx, ly - minfd)$. The first point is constant, and this distance is initially 0. The second point moves with rate lv, but by the domain constraint, $lv \leq (maxfd - minfd)/T$. Therefore, at time t this distance is at most $(maxfd - minfd) \cdot \frac{t}{T}$.

Combining, the sum of the distance from (fx, fy) to $(Lx, Ly - minfd)$ and the distance from $(Lx, Ly - minfd)$ to $(lx, ly - minfd)$ is at most

$$(maxfd - minfd) \left(1 - \frac{t}{T}\right) + (maxfd - minfd) \cdot \frac{t}{T}$$
$$= maxfd - minfd,$$

so the distance from (fx, fy) to $(lx, ly - minfd)$ is at most $maxfd - minfd$ for all $t \in [0, T]$, as desired.

If we wanted to prove this in Keymaera, one approach would be to introduce new variables Lx, Ly, $d0$, $d1$, and $d2$. Before running the differential equation, (Lx, Ly) would be assigned the value of (lx, ly), $d1$ would be assigned d (the distance from (fx, fy) to $(lx, ly - minfd)$), and $d2$ would be assigned 0. We would continously update $d1$ and $d2$ in the differential equation so that they represent the two respective distances we analyzed in the proof sketch above. We would assign $d0$ to be the distance between (fx, fy) and $(Lx, Ly - minfd)$ at the start of the iteration but would not update it during the differential equation.

To use these variables, we would apply the generalization

$$d \leq d1 + d2 \;\&\; d1 + d2 \leq maxfd - minfd,$$

which is strong enough to show $(lx - fx)^2 + (ly - minfd - fy)^2 \leq (maxfd - minfd)^2$ because at all times we have $d \geq 0$ and $d^2 = (lx - fx)^2 + (ly - fy - minfd)^2$. To show $d \leq d1 + d2$, we just need the triangle inequality, so this part could probably just be done using Quantifer Elimination, cutting in helpful intermediate steps if necessary. To show $d1 + d2 \leq maxfd - minfd$, we would generalize to

$$d1 \leq (maxfd - minfd) \left(1 - \frac{t}{T}\right) \;\&$$
$$d2 \leq (maxfd - minfd) \cdot \frac{t}{T}.$$

The second inequality is very easy to prove: at the start of the differential equation, $d2$ is 0, and its rate of change is at most $(maxfd - minfd)/T$. To show the first inequality, we can prove

$$d1 = d0 \left(1 - \frac{t}{T}\right).$$

We could show this using a differential invariant: both sides have the same constant derivative. This equality would then generalize to

$$d1 \leq (maxfd - minfd)(1 - t/T),$$

as desired.

6 Complete 1D model

Our complete 1D model is designed to be a very unrestrictive model of follower-leader behavior when movement is restricted to a line. It seems that finding a safe control decision for the follower is quite challenging, but we were able to make some key observations.

6.1 Description

In this model, the follower and leader move in the positive x-direction on the x-axis. They each have a constant acceleration, which they can change every time they make a control decision. Accelerations must be in the interval $[-B, A]$, where B is the maximum amount of braking and A is the maximum acceleration. As before, we use a single time trigger. The distance from the follower to the leader must be between $minfd$ and $maxfd$. See Figure 4 for an illustration of this model.

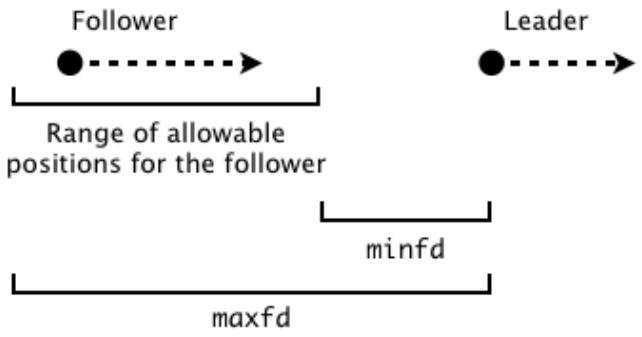

Figure 4: Movement in the complete 1D model

6.2 Limitations

This model assumes that neither robot can move backwards. This is a reasonable assumption, but it is conceivable that there might be scenarios where we would not want to assume this. Additionally, as before we are only using a single time-trigger for simplicity but it does not seem that this would affect that validity of a reasonable control decision for the follower.

6.3 Update interval requirements

It is difficult to find conditions on the positions, velocities, and accelerations of the robots that guarantee that the safety conditions can be preseved, which is perhaps the main difficulty of this model. However, we were able to find a necessary relationship between the range of possible accelerations, the following distances, and the length of the maximum time-triggered interval T.

Suppose that when the follower makes its control decision, the position, velocity and acceleration of the leader are L, v_L, and a_L respectively. Then after time t the position of the leader is

$$L + v_L + \frac{1}{2} a_L t^2.$$

However, the follower cannot observe the new acceleration that the leader chooses at the start of this time-triggered interval. Therefore, from its perspective, the minimum possible position of the leader after time t is

$$L + v_L - \frac{1}{2} B \cdot t^2$$

and its maximum possible position after time t is

$$L + v_L + \frac{1}{2} A \cdot t^2.$$

Let F_1 be the final position of the follower after time t. To stay safe, the follower's position after time t must be at least $minfd$ behind the minimum possible position of the leader after that time, and at most $maxfd$ behind the maximum possible position of the leader after that time. Therefore,

$$F_1 \geq L + v_L + \frac{1}{2} A \cdot t^2 - maxfd$$
$$F_1 \leq L + v_L T - \frac{1}{2} B \cdot t^2 - minfd.$$

Subtracting yields

$$0 \geq \frac{1}{2}(A+B)t^2 - (maxfd - minfd).$$

Since this must hold for all $t \in [0, T]$, we get

$$maxfd - minfd \geq \frac{1}{2}(A+B)T^2.$$

This is an interesting necessary condition, and it shows the flavor of the type of analysis we would have to do to make more progress with this model.

7 Complete model

Our complete model is our most realistic model. While designed to allow a verifable solution without an unreasonable amount of effort, it is still quite a difficult model to work with. Note that there still parts of this model that are unspecified, such as the

follower's control choice: this model is sufficiently complicated that even finding a safe choice is difficult, much less proving it. Filling in the unspecified parts of this model would be a good first step for future work.

7.1 Description

As in the previous model, we use a single time-trigger, and the leader moves in circular arcs during each interval so that its velocity is continuous. This time, the follower is also restricted to this type of movement: it cannot instantaneously change its velocity. Both robots may choose accelerations in $[-B,A]$, where B represents the magnitude of the maximum braking allowed and A that of the maximum acceleration allowed. The follower has a minimum turn radius of $minrF$, to ensure it doesn't make unreasonably tight turns.

To ensure that the leader's motion is somewhat predictable to the follower, the leader's speed may not exceed $maxLV$, and it has its own minimum turning radius of $minrL$. To make the model provable, an assumption should be added to the initial conditions specifying how large $minrL$ must be. For example, if we could ensure that the leader could never change direction by more than 90° during a time-triggered interval, then the follower could advance in the leader's general direction without worrying about hitting it if it moves towards a location just behind the position in which it observed the leader. We can ensure that the leader's direction changes by at most 90° in the interval by ensuring that ldx changes by at most 1 during the interval, since ldx is the x-component of the normalized velocity of the leader. Therefore, we would like $|ldx'|$ to be bounded by $1/T$, or

$$|-ldy \cdot lv/ltrackr| \leq 1/T.$$

Since $|ldy| \leq 1$, we can ensure this if

$$|ltrackr| \geq T \cdot lv,$$

so a reasonable condition on $minrL$ would be

$$minrL \geq T \cdot maxLV.$$

We would probably also want to add a restriction on the leader's maximum speed $maxLV$. The previous bound shows that $maxLV \leq minrL/T$, but we want something that guarantees that if we move to a location just behind where we last saw the leader, he will not move so far in the time-triggered interval that we will violate the maximum follow distance. In the previous model we required the leader to have speed at most $(maxfd - minfd)/T$. The condition for this model would probably be similar in spirit but perhaps significantly more complicated.

7.2 Limitations and possible solutions

This model has the fewest limitations of all of our models. One aspect of this model that would need to be modified to prove safety in a more complicated scenario would be its lack of obstacles. However, as long as there is a behavior for the leader which provably maintains a large enough distance from all obstacles, the follower will stay safe by being close enough to the leader, as mentioned previously.

Another aspect of this model that may need to be modified is its time-trigger. As before, for simplicity we only included one time-trigger, but this is not realistic because unless the robots communicate they would not synchronize the times at which they make their control decisions. It would be desirable to add another time-trigger so that each robot has its own, although this would make the model more difficult to work with without changing the overall idea's of its safety proof. (As before, we should be able to construct an algorithm for the follower that does not care if the leader changes its control decision during the follower's time-triggered interval.) Additionally, it might be simplest to think of the follower's control decisions as happening in phases: maybe when it wants to move closer to the leader without running into it, it will accelerate quickly for a certain period and then decelerate quickly when it gets close. If we wanted to use such an algorithm for the follower we would want to change the model to keep track of this state. In such a case we would also want to let the follower set the value of its time trigger Tf. For example, if it only wants to accelerate for a short period of time, it would assign Tf a small value so that it will be guaranteed to be "woken up" from its acceleration after a period of time of length at most Tf.

7.3 Discussion of follower behavior

Writing a control decision for the follower in this model is a difficult task. In the previous model, even with the assumption that the follower's velocity could have isolated points of discontinuity, it was still somewhat challenging to find a control decision that guaranteed safety, and the proof was fairly complicated. As mentioned in the previous section, it may be most natural to think of the follower moving in phases. However, each phase could be thought of as its own control decision, so it is probably simpler for the analysis of the follower's behavior to eliminate phases once we can come up with a valid algorithm. (That is, instead of having our algorithm decide to accelerate and decelerate, it will decide to accelerate, and then the next time it makes a decision it will decide to decelerate.)

The advantage of our intermediate model is that, even while is assumptions are not completely realistic, it gives us a concrete, safe control decision that can inspire us when creating a control decision for this more realistic but much more difficult model. In the intermediate model, the follower always chases the point that is $minfd$ below that last location we observed the leader. The follower avoids a collision with the leader because we chase a point below where was observed the leader, rather than chasing the point we observed the leader, and the follower ensures that it stays close to the leader by moving towards this ponit because in that model, that was the closest point to the leader which was guaranteed to be safe (because for all we know, the leader might be stationary, and then we couldn not get closer than the point $minfd$ below it).

We can think about similar behavior for the follower that might apply to this more complex model. We might want to chase the point $(Lx,Ly) - minfd * (ldx,ldy)$, which is the point $minfd$ away from the leader, directly in back of its current direction of movement at the time we observed it. (Note that in this case, we would require the follower to be able to observe or be told what the leader's velocity is, which was unnecessary in the previous model.) See Figure 5. However, there are more factors to consider now that were not issues in the previous model. What if the follower is moving too slowly, so that even if it is currently close enough to the leader, the leader will outrun it no matter what it does? What if the follower is currently heading in a different direction than it would prefer to? Since the follower's velocity must be continuous, we might need to think about what we want its velocity to be at the end of the time-triggered interval (assuming it lasts for exactly time T) so that it is not stuck with a velocity that will make things difficult or impossible for it to follow in the future. To verify that the follower's behavior is safe, it would be important to think about an invariant in terms of the positions and velocities of the robots that would ensure safety and that we could inductively satisfy after one more iteration of the loop. Determining a safe control decision for the follower in this model is one of the most interesting areas of future work for this project.

8 Results and discussion

Our work resulted in creation of a number of models of follower and leader robot movement with a variety of assumptions. We were successful in our efforts to use Keymaera to verify safety of our simple models and complete most of the proof of safety of our intermediate model[2]. Our simple and intermediate models are more restrictive and have assumptions that are probably not realistic, but they were still very helpful in guiding us towards relevant techniques that could be used to prove safety on more realistic but complex models. While we did not construct and verify safety of a follower control decision for the complete models (both the 1D and 2D versions), we described the models precisely in a .key file and discussed some of the difficulties inherent in making progress with them.

9 Further Applications

As mentioned before, our model is easily extended for other useful and relevant applications in cyber-physical systems research. One logical next step would be to generalize the models from two into three dimensions. It is not difficult to see how, with additional variables and slightly modified properties, the one-dimensional and two-dimensional models could be adapted into a model for the third dimension. This would be crucial in writing con-

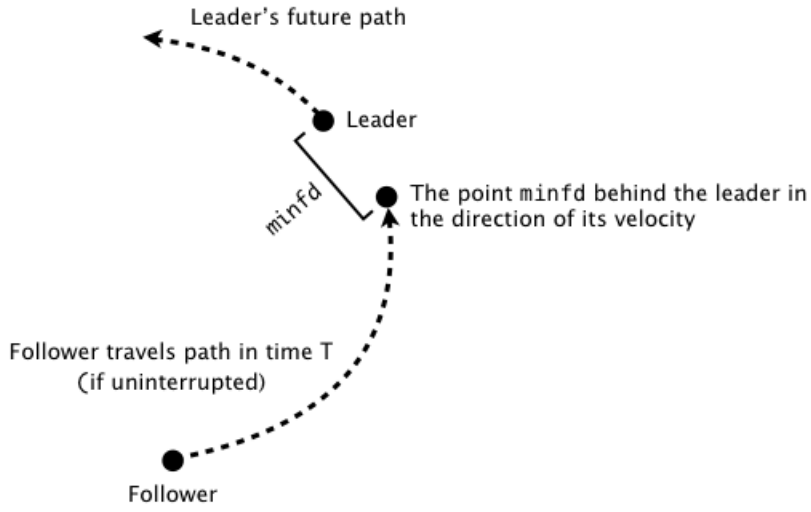

Figure 5: A possible path for the follower in the complete model

trollers and verifying safety for flying vehicles, such as quadrocopters and unmanned aerial vehicles (UAVs). Indeed, work has already been done for such an application of one quadrocopter tracking and following another [6].

Another useful application involves multiple followers following a single leader. This is often known as flocking, and is an area of active research. Useful and safe controllers avoid collisions [2], typically between both the followers and the leader as well as among the followers[3]. This flocking concept can also be combined with the three-dimensional models to study flocking of birds [3] or unmanned aerial vehicles [5].

References

[1] CALIFORNIA INSTITUTE OF TECHNOLOGY / JET PROPULSION LABORATORY. JPL Robotics: System: The Axel Rover. https://www-robotics.jpl.nasa.gov/systems/system.cfm?System=16. (Visited on 12/09/2014).

[2] CUCKER, F., AND DONG, J.-G. A general collision-avoiding flocking framework. *IEEE Trans. Automat. Contr. 56*, 5 (2011), 1124–1129.

[3] DUTTA, K. How birds fly together: The dynamics of flocking. *Resonance 15*, 12 (2010), 1097–1110.

[4] GIAMPAPA, J. A. Test and evaluation of autonomous multi-robot systems. https://resources.sei.cmu.edu/asset_files/ConferencePaper/2013_021_001_296965.pdf, 10 2013. (Visited on 12/09/2014).

[5] GUENARD, A., AND CIARLETTA, L. The {AETOURNOS} project: Using a flock of {UAVs} as a cyber physical system and platform for application-driven research. *Procedia Computer Science 10*, 0 (2012), 939 – 945. {ANT} 2012 and MobiWIS 2012.

[6] WENZEL, K. E., MASSELLI, A., AND ZELL, A. Visual tracking and following of a quadrocopter by another quadrocopter. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012, Vilamoura, Algarve, Portugal, October 7-12, 2012* (2012), IEEE, pp. 4993–4998.

Notes

[1] Some examples of such a teamwork scenario include the search and rescue application and wireless connectivity scenario, both briefly discussed previously.

[2] The remaining case of the proof is tedious and not feasible with software limitations of KeYmaera, but details of how the remaining branch of the proof can be verified can be found in the proof sketch.

[3] Sometimes in flocking systems there may be no designated leader, but rather many peer robots or agents cooperating to form a dynamical system.