15424-Flock on the March

BY ZIHAN ZHOU

zihanz@andrew.cmu.edu
Class of 2017 , Qatar CS

Table of contents

1 introduction

1.1 What does this paper model

This paper models many 2 dimensional boids(vechicles that can accelerate to any direction in a 2D plane)'s movement and their avoiding many obstacles.

1.2 Why does this paper model boids

Swarm-like robots have been becoming popular these days. Sometimes, we want 100 or 1000 robots to go to some where through some path-find algorithm or human-remote-control. However, maybe we do not want to remote-control each of the 1000 robots or let all of them run a A-star algirhtm. Rather , we want to control one, and let the rest follow without colliding on each other.

1.3 Why is this project awesome

1.3.1 Arbitrary many boids and obstacles without distributed DL!

This model can incoprate the scenario of arbitrary many boids and obstacles. Usually we need distributed DL to model such a scenario. However, in this paper, I use just DL to build such a model that ensures safty.

1.3.2 Abstraction and Safty Duty Seperation

Safety is the key idea in CPS, but to prove safety is not a easy job. However, in this paper, though i did not prove much by my own due to time limits, but I set a frame for modularity and layer-seperated-safety-reponsibility to allow more efficient safety proof in the futre. I would discuss this matter further in the later section.

1.4 About the model

Basically, this model have a leader and a follower, and prove various properties between a leader and a follower.However, since I let the follower always maintain same distance to the leader , I can have different followers at different distances without harm the safety of others. Further more, we can regard a bunch of boids together as one follower or one leader, and combine them freely as long as we give them enough radius for safety. I would also define the action such as split and merge at last.

1.5 Brief description of layers

In order for readers to understand the paper better, I would first explain some nouns this paper later refers to.

I use the word layer to denote control level. Lower layer to higher layer is like assembly to python.

1.5.1 locomotion Layer: moving in circle and follow the leader

I usually use "locomotion layer" to refer to the control of the follower that is reponsible for making it always maintain within a range of distance from the leader. This layer is also responsible for making the follower's circular motion around the leader. In a word, locomation layer should be able to prove the distance range property , and should give higher layer an API to control the angular acceleration as well as detect current angle(approximately, within a range).

Here are some of my thoughts on locomiotion layer, but this is not the focus of the paper. The code in Lab3 folder is also a very primitive and incomplete locomotion layer, and give kind of bad approximation on angles.

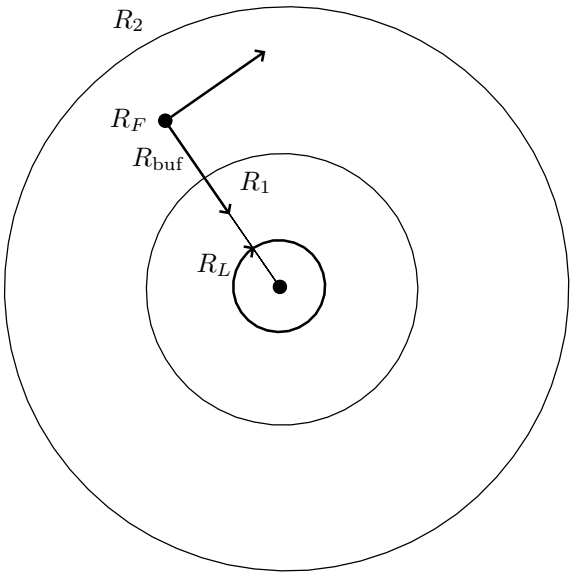


Figure 1.

In figure 1, the innermost small circle is the leader have a radius R_L(could be zero if we use virtual leader, ie. we do not really a leader in the middle, but just let the robot regard there is a signal in the middle, and let one robot track or compute the movement of the leader and tell other robots), and around the leader there is a ring runway, defined by R_1 R_2 so it has width $R_2 - R_1$. The follower can move arbitrarily inside . For example, he can accelaerate in a straight line and then turn to achieve a kind of circular motion. What it gives higher layer is actually R_F, which is a ideal circle whose center is always R_F away from the leader, and has radius R_{buf}. If the desired shape is not circle, then simply make a larger circle, to cover the desired shape.

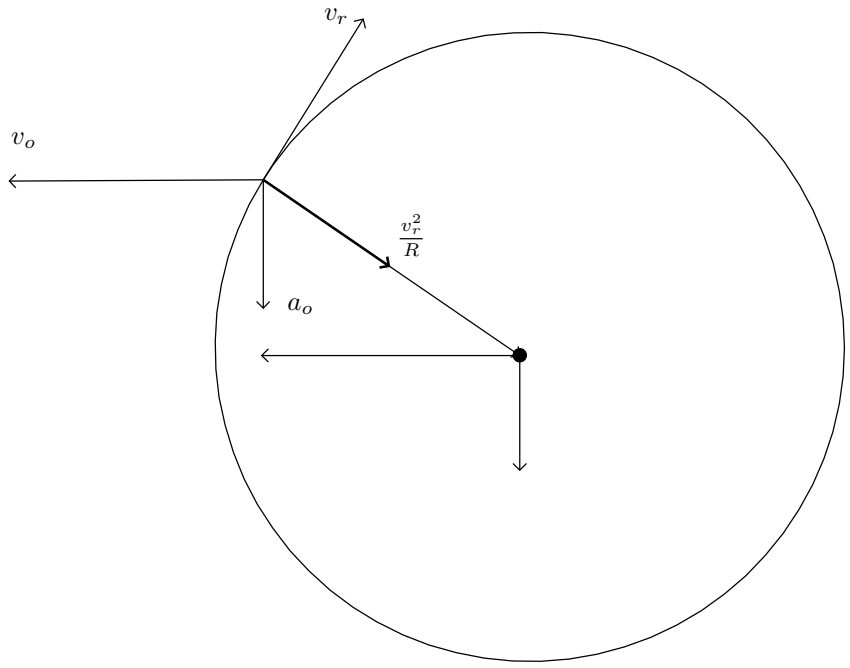


Figure 2.

Because the higher layer would have the follower move not just in circle, implmentation of locomotion layer need to ensure the follower's circular motion relative to the leader can be combined with leader's own velocity and accleration.

1.5.2 buffer layer

The buffer layer is the focus of the paper. This layer is the key of the framework, because it can not only accept different locomotion layer as long as they satisfy the safety, but also give pathing layer much freedom, it has very little requirement for pathing layer to follow, just leader never itself run into an obstacle, and this layer would ensure the safety of all followers.

This is possible because the limitation is incoprated into "boundary of obstacles", for any obstacle, we normally model it a circle that can cover it, and then we make the circle larger to ensure safety and provde a space for special movemnt implemented in buffer layer. The larger circle is the boundary, and the pathing layer only need to aware of the boundary and just ensure the projection of the leader's velcoity on the direction toward the center of obstacles is zero when reach the boundary:

$$(x_{\text{leader}} - x_{\text{ob}})^2 + (y_{\text{leader}} - y_{\text{ob}})^2 = R^2_{\text{boudary}} \rightarrow v_x \times (x_{\text{ob}} - x_{\text{leader}}) + v_y \times (y_{\text{ob}} - y_{\text{leader}}) \leqslant 0$$

This layer also provides nodes and special movement control for A-star path finding algorithm.

Lastly, we could implement many different buffer layer to adapt to different terrians or obstacle situations, and verify their safety seperately.

This paper suggest several ways for buffer layer and proved the safety of one . Then this paper developed some special buffer layer based on the proven one,

1.5.3 pathing layer

This is not the focus of the paper either. It could be A-star algirthm, or any other way of travel the leader to a destinity without crashing it into any boundary.

2 Buffer Layers Discussion

2.1 What is the safety property?

Let us start from a single obstacle. Since we are gurranteed that the leader would never run into the obstacle, then no matter how large the obstacle is, it can cover at most half of the track as shown in the below figure 3.

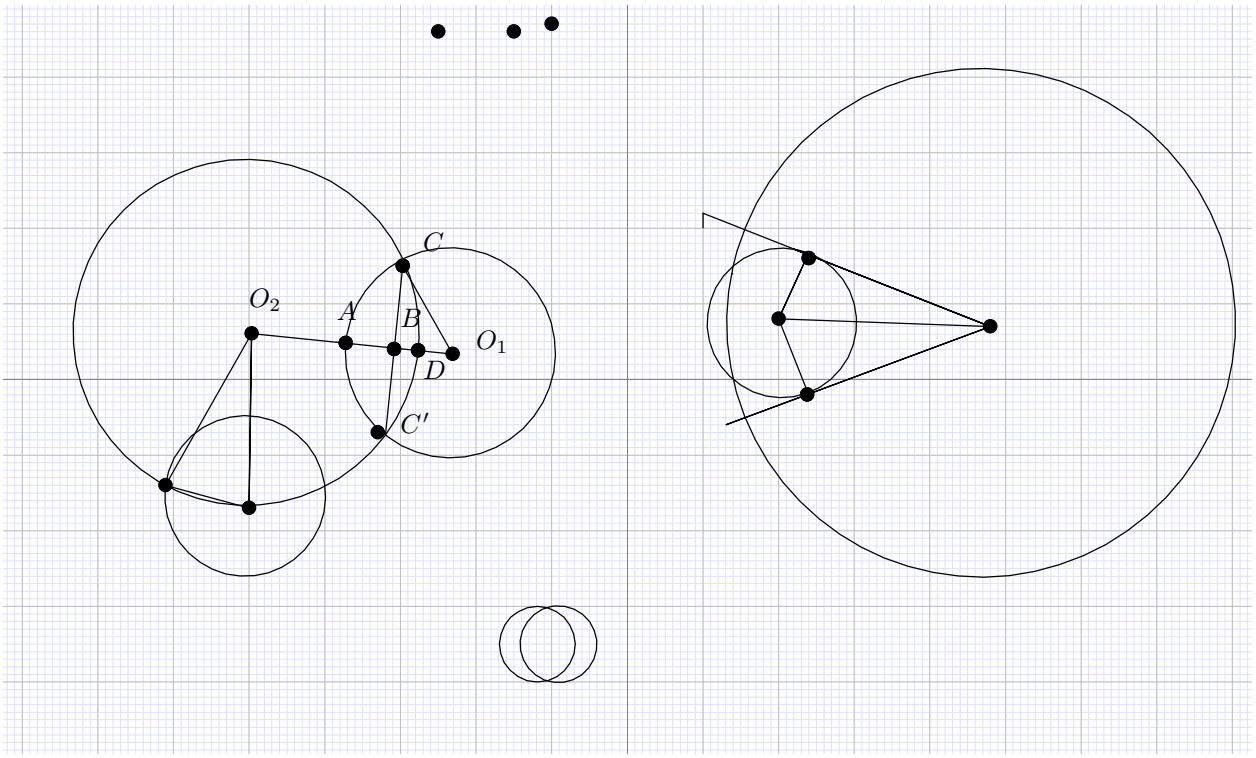


Figure 3.

Figure 3 shows two scenarios, one the left : O_2 is an obstacle, O_1 is the leader. O_1C is the outmost of the some follower. It has safe region CO_1C'. The max safe region would be when $O_1O_2 = O_2C$, ie. leader on the obstacle boudary.As shown in the circle on the bottom, but we gurrantee this is less than 180 degree or π centered at O_1O_2. On the right, the small circle is obstacle. It is obvious that ,the obstacle can get in side the ring track of the follower, and follower would be always safe in that case. We can still use 60 degree or $\pi/3$ to bound this region, because when the small circle just fits inside, its radius is half of the big circle and have cover region at most 60.

For any control strategy, stay out of the danger region(the overlapping arc), then it would be safe.

Notice here we haven't talked about the extended boudary, just the true raduis of the obstacle.

2.2 Contrl Strategies

2.2.1 Relative Speed direction

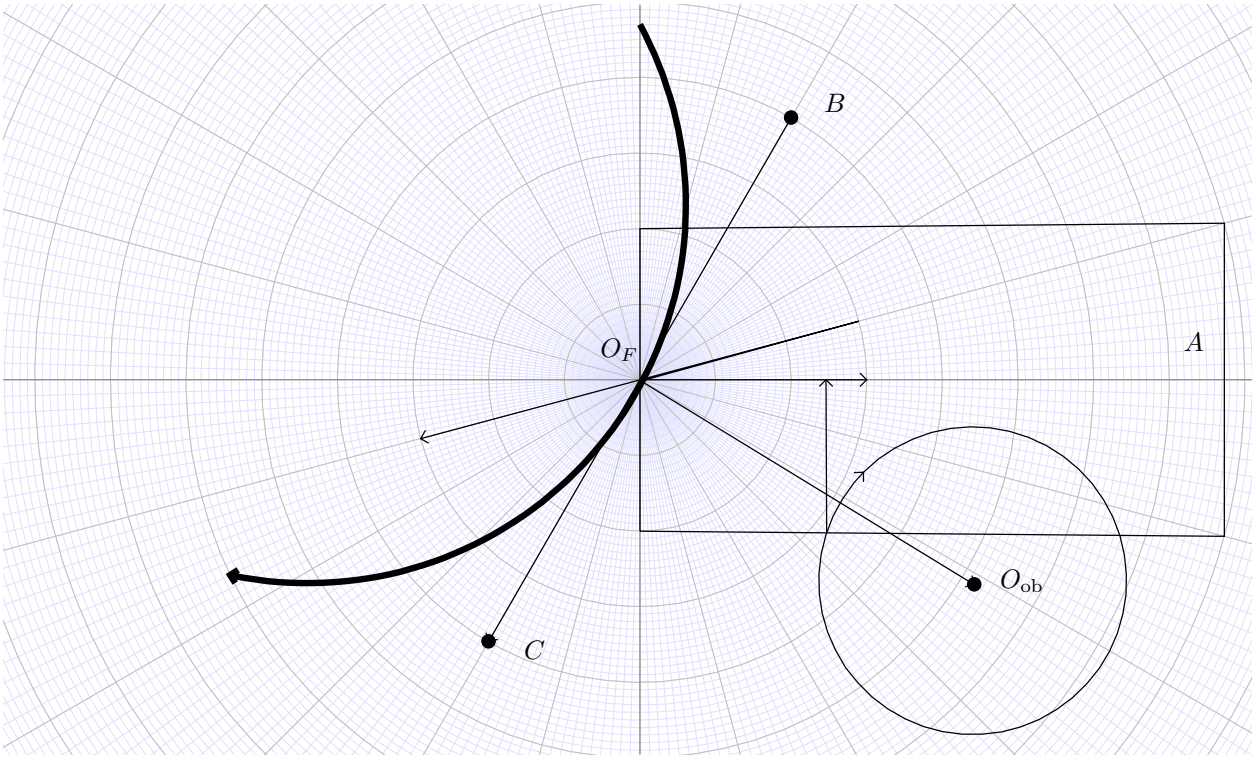
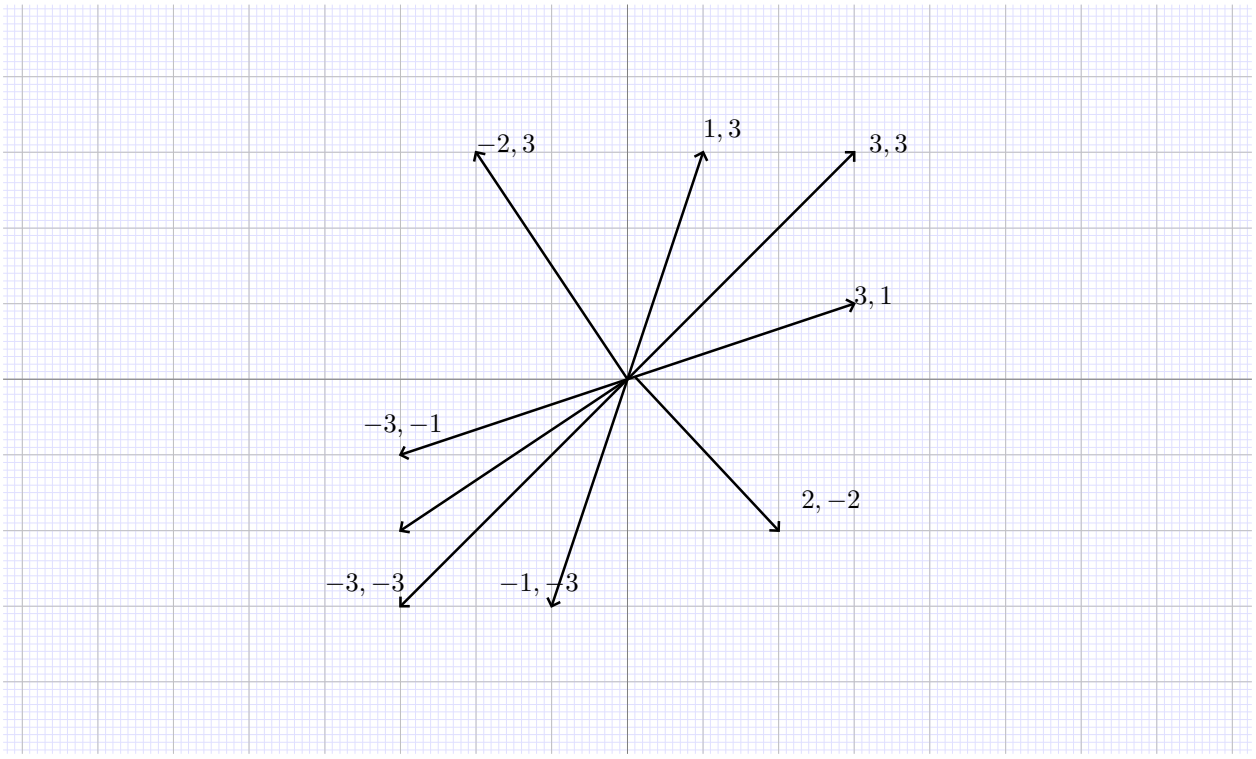


Figure 4.

the center of the figure4 is the follower, the circle is the obstacle, and the square is the detection square area, such that it start to control after it detect an obstacle in the square. This isn't using an extended boudary, or say we extend the boundary by zero. This Strategy is implmented in "oneObstacle.kyx" it decides whether to accelaerate clockwise or anti-clockwise based on the angular direction of the relative velocity of the follower to the obstalces and the relative position direction of the obstacle to the follower. In the figure 4, suppose O_FA is the direction of the relative speed, and OB OC are two choice of acceleration , we would choose OB, because OB is at the same direction of O_FO_{ob} as O_FA, and will make the follower turn away from the obstacle. However, this model is extremely hard to proven to be safe.

The math to decide the angular direction



if $y_1\, x_2 < y_2\, x_1$

(x1, y1) is at clock wise of (x2, y2)

2.2.2 Safe position prediction

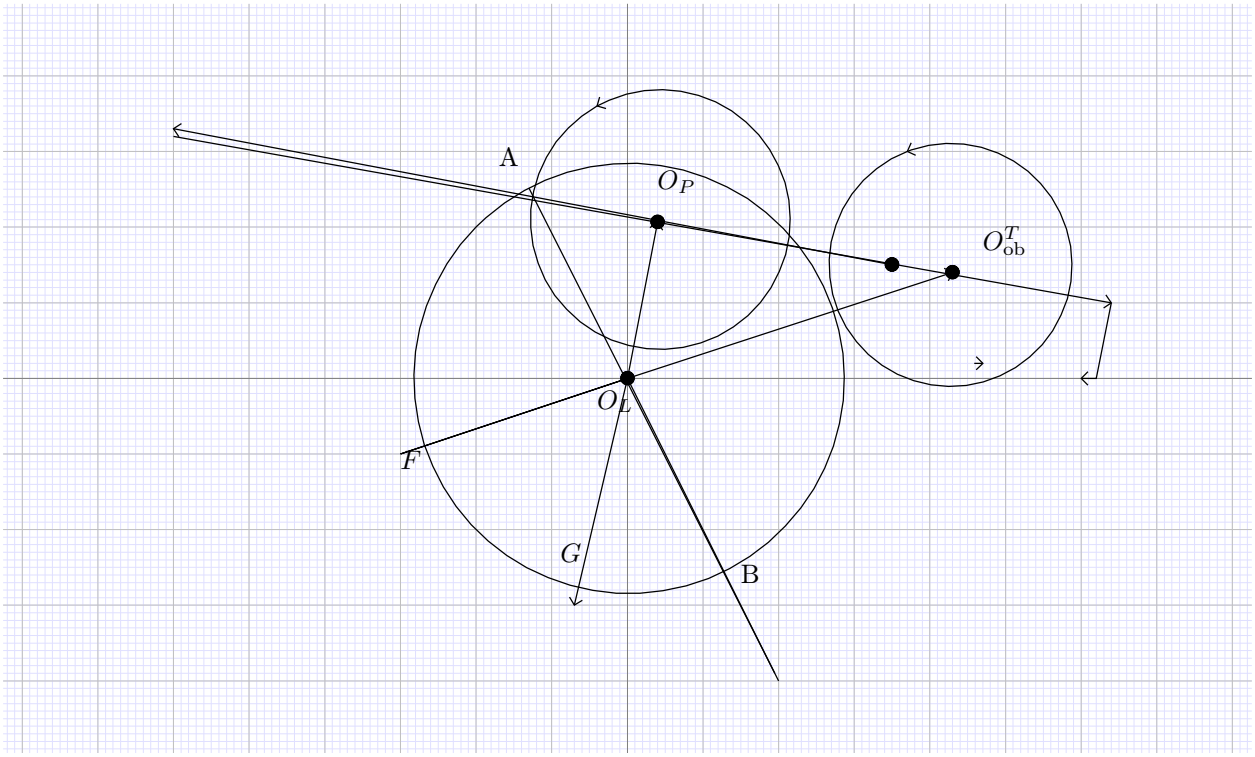


Figure 5.

This figure 5 use leader as reference frame, so it is the obstacle that moves and accelerates.

This one predict the touching area between the true obstacle and the leader-follower circle. The strategy let the follower just go to the opposite of the center of the touching area, because we know the area is at most 180 degree, so the opposite side would always be safe. However, it is somehow trciky to do this in angular accelaration because π 180 , is an trancedental, and can not be accurately decided in DL. As a result, we would approxmiate a safe region. As shown on the figuire 5, the O^T_{ob} denotes the center of the obstalce circle that is just tangent to the follower-leader circle, and it gives an 180 opposite safe region(notice the 180 oppsite of middle in inside the safe region), intersected with the safe region when it goes out and tangent, we can have a region, for our follower to stay.

We can also regard the region as dynamic rather than fix, it can be regarded as a chase problem, where there are a two end of safe region A B in the figure 5, and both are moving,

we can assume A moving the fastest(when O_P), and the chased side B moving slowest (O^T_{ob}) both in terms of angular velocity. This only take into account of the velocicy, but the leader may also acceleartes, which would make the apprixmation much harder. So it is a good idea to forbid the leader to accelerate around the obstacle, and here comes the concept of extended boundary, we could make a larger radius of the obstacle, such that leader decides its velocity at that boundary and does acceralte inside to make the model work.

This one also assumes the distance between the leader to the direction of the velocity(O_LO_P) is always greater than R_{ob}, which would also be taken care of at the extended boundary.

It would be much easier to prove safety, if we give a specific behavior of the leader instead of arbirarily , but that would lost the power of abstraction. In order to find a balance, we can assume the specific behavior after inside a larger radius.

It could be improved by concerning the size of the obstacles, and adujust the angles intead of always 180,

It could be improved by dealing with the situation when the size is really small, it can go in the big circle, and would be safe for the arc above.

Due to the complexity of mathematical apprixmation(probably need taylor series), I did not implement this.

2.2.3 Just Relativa postion

In file “oneObstacle2.kyx”, I implmented another strategy. it always accelerate toward the direction from the obstacles to the leader , If we using figure 5 as reference, it would be F, and G. This safety is also hard to prove.

2.2.4 Safe Proved Strategy

This strategy is implemented fully in "oneObstacle4.kyx". Though it is named one Obstacle, it actually works for arbitrary many. I won't write too much detail about model itself in the paper, since I described it pretty carefully in the comment of the code.

Figure 6 shows the safe region.

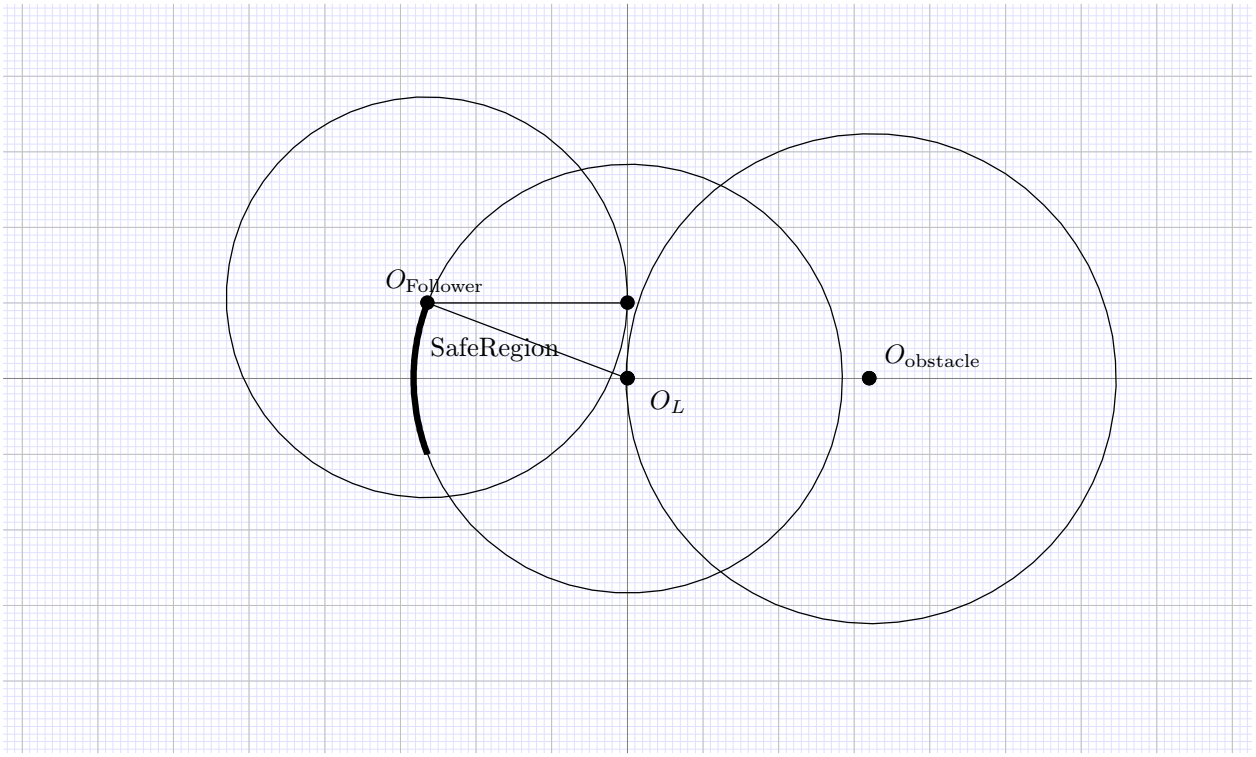


Figure 6.

$$\text{safe rad} \approx \pi \pm \sqrt{\frac{R^2_{\text{leader}} - R^2_{\text{follower}}}{R^2_{\text{leader}}}}$$

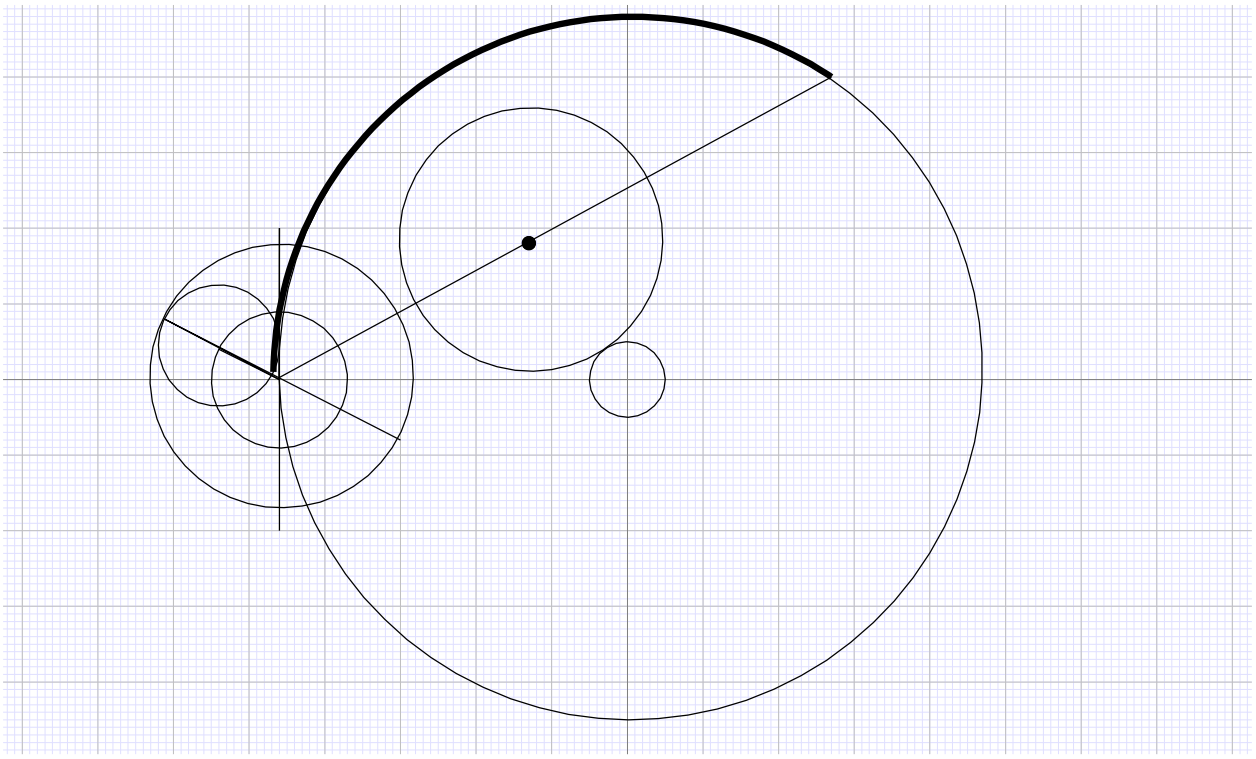


Figure 7.

As in figure 7, the strategy , in general, is to stop at the extended boundary,then spin the follower to the opposite side of the obstacles, then move toward the obstacle, stop at its true boundary, and the leader spin around the obstacle(together with follower) to any direction the leader wants, and then move straight toward that direction untill reach the extended boudany again. However, that is not always efficient, above figure seven show region, when leader do not need to go straight to the obstacle, but rather direction go to another part of extended boudary.

The good thing about spin is that, it is easy to express in polar coordiniates, and easier to model and prove

safety.

This strategy has requirement for the extended boundary.

$R_{\text{ex}} = R_{\text{ob}} + R_{\text{follow}} + R_{\text{Leader2follower}}$

The advantange is that this can apply to more than one obstacles. since, as long as the leader is outside extended boundary, the follower is gurrantee to be safe.

Then the most direct inference on the distance between two obstacles would be as shown in figure 8

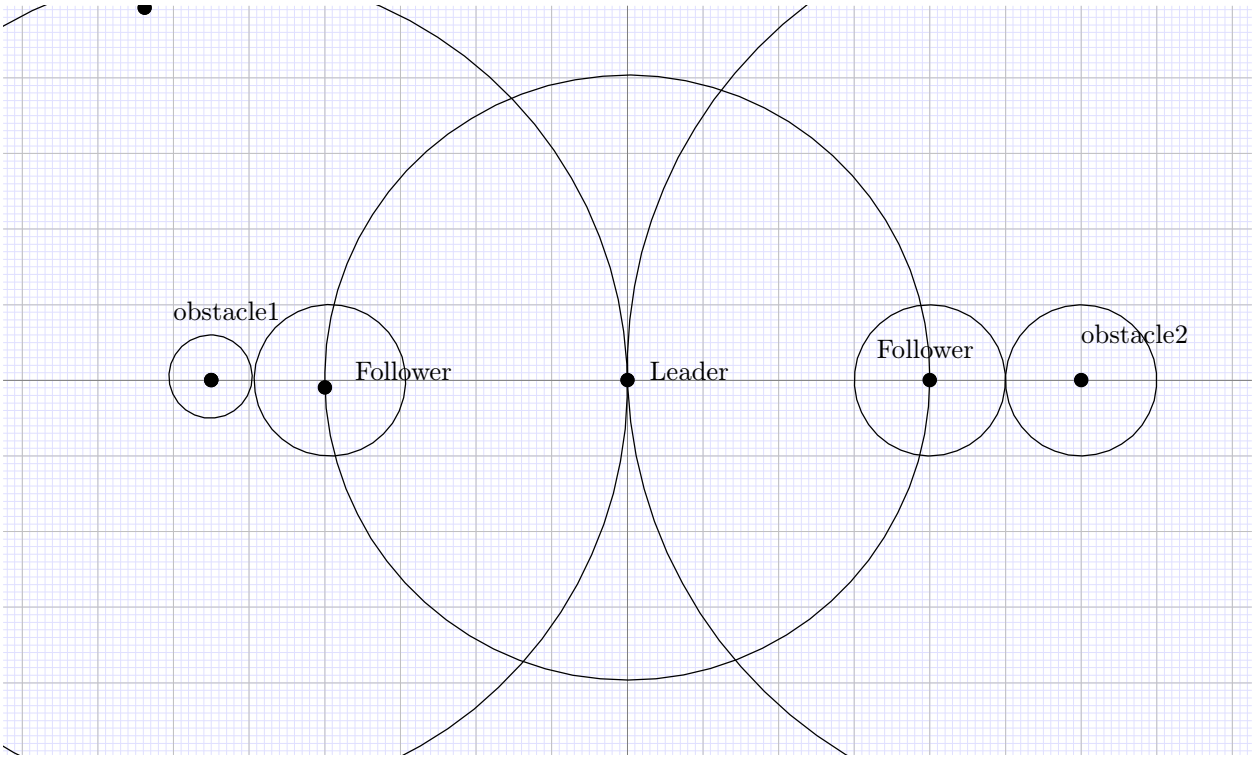


Figure 8.

Obstcales r1 r2 need to be r1+r2+(rL+rS)*2 away. This would make it safe for any high level algorithm followes property

Note 1. Potentially,when making A star nodes, one can make any point on the ext boundary circle as a node, and then to access the other node on same circle, one can call my control strategy.

Advanced case, model a lane,efficiency ,

Since our model normally models circular obstacles , it is actually enoguh to have only r1+r2+(rL+rS), distance between two obstacles. Below Figure 9 is a special case, for r1+r2+(rL+rS), but not jut for circles.

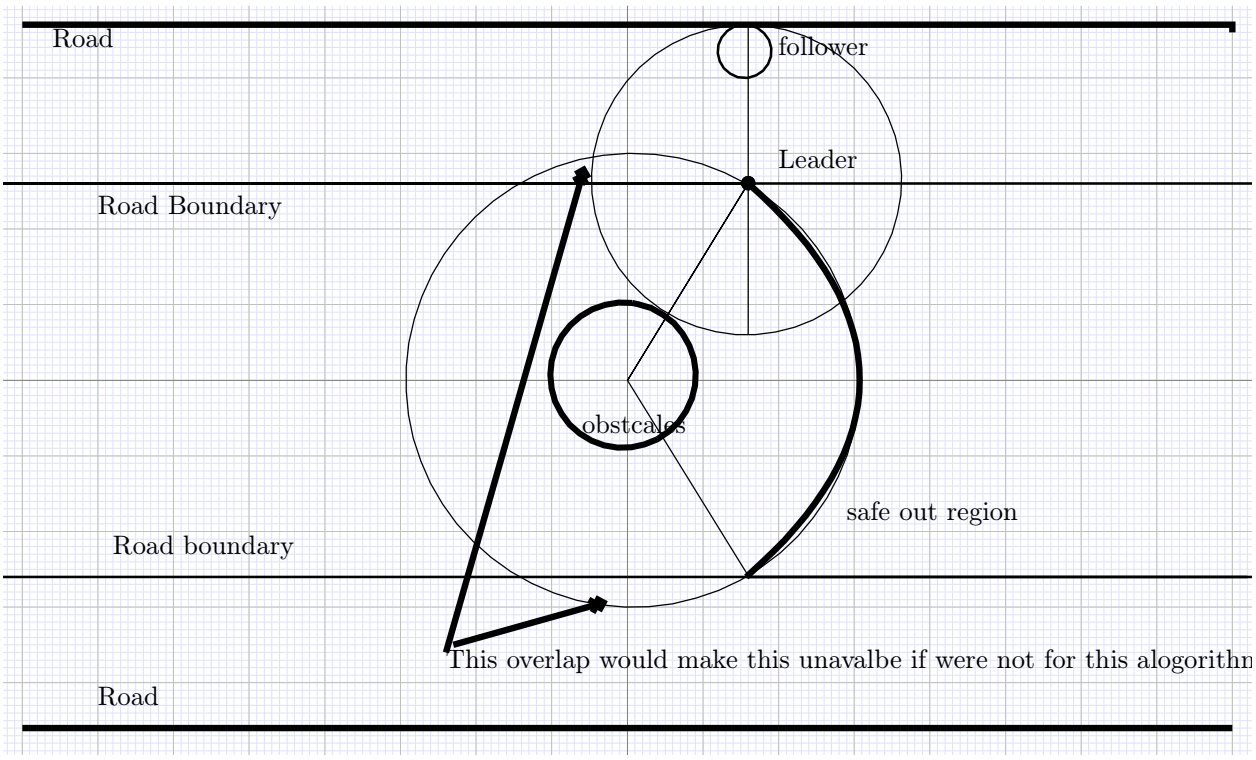
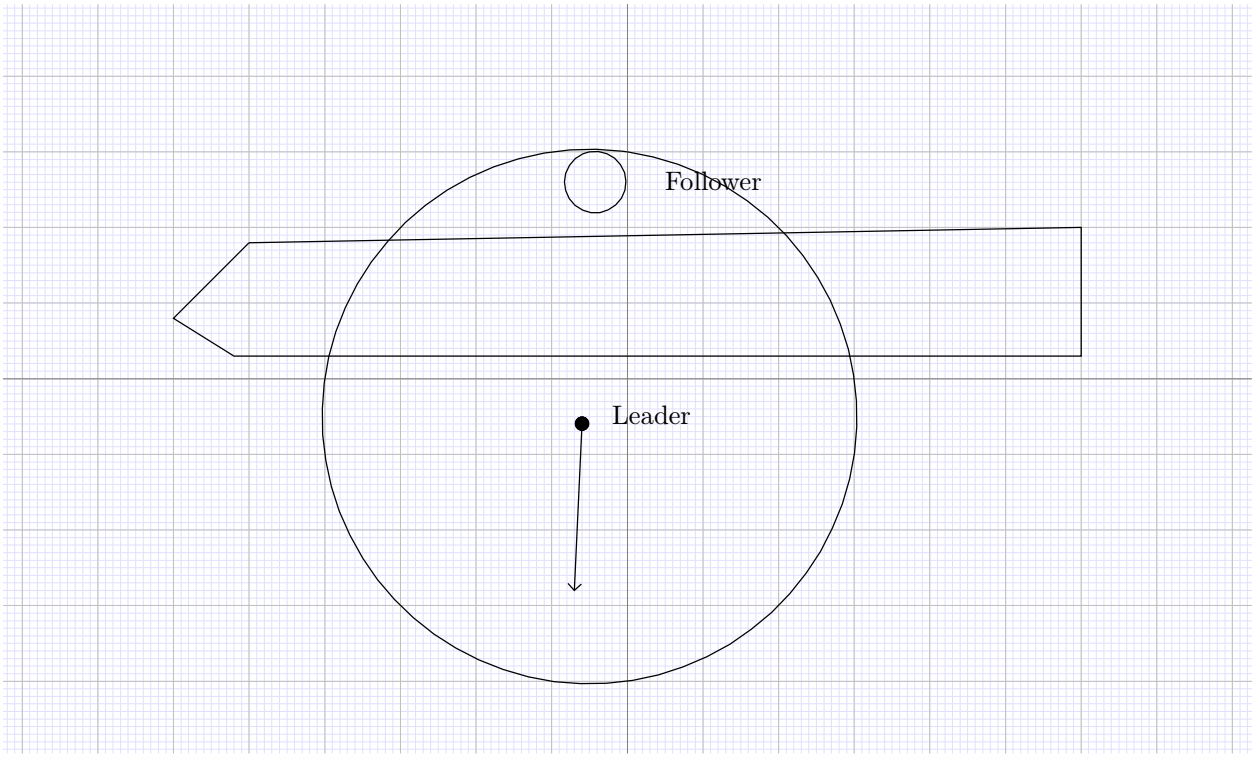


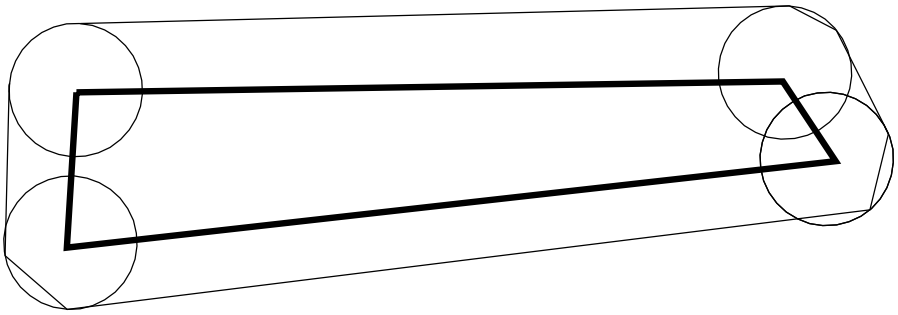
Figure 9.

No kyx for this. Notice i made a safe region there, so that it can not get out arbitratly but only in safe region. This is important in later application.

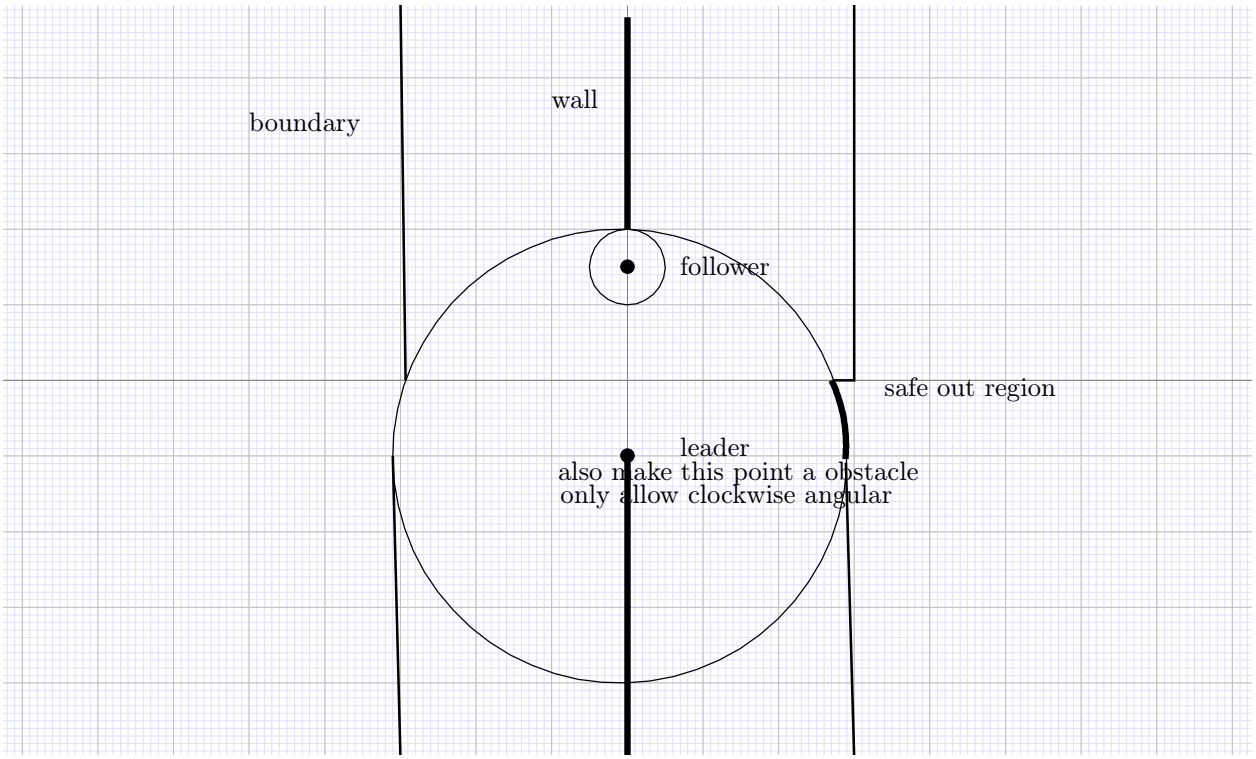
This is model of a lane, which is special case , otherwise it could cause problem.



We could solve this by regarding this wall a large cirlce, but if we have a wall so long, but we do not want a large circle, We can add circles at each corner; we make the circle as large as the leader - follower circle, so it is not possible to let leader go around without follower



we could also do something special for small entries



The above figure passes a small entry, which is not possible without this modification.

2.2.5 more speical midlayer

We can made more speical modified buffer layers based on my control to adapt to more situations.

Potentially, this allow more follower on the same track as long as they syncronize their angular accelaeration.

However,that made it not possible to have a way across that small entry.

2.2.6 Split safety

It just satisfies our buffer layer, just let the leader stop and split the outside, the splited follower become an obstacle and create an extended boudanry such that our leader lies just at the boundary. One need be carefully checking whether the new obstacles is too close to other obstacles.

2.2.7 Merge safety

This is same as enter a buffer layer. Stop at the boundary, then send control to the follower, that follower would at the exact right track according to our constraint stated above.Then we can remove that extended boundary made by the follower , we need to update all other extended layer accordingly.