Jackal Robot Moving Obstacle Avoidance Strategy with Lidar

Zhe Luo, Yuan Liang, and Puhan Zhang
Department of Computer Science, University of Virginia, Charlottesville, VA 22904, USA
(Dated: November 4, 2024)

The emergence of autonomous vehicle driving techniques has greatly promoted a safer and more convenient driving experiences. With technologies such as emergency breaking, lane keep assist and adaptive cruise control, vehicles have become more intelligent. On the other hand, with the fast deployment of these technologies, The accuracy and effectiveness of the strategies becomes an essential question. On road, pedestrians can approach vehicles in a variety of situations which makes the efficiency and accuracy of pedestrian detection is lower than other driver's assistance technologies. In this project, we propose a strategy that simulates the situation when the moving obstacle is approaching our Jackal robot in various of different scenarios. Only with the detection data from Lidar on board, our strategy shows that the Jackal robot is able to avoid moving obstacles with confidence and largely reduce the possibility of collision. This strategy shows potential of real world implementation that can increase the level of autonomous driving.

INTRODUCTION

In this project, we created a closed environment including Jackal robot and multiple fixed and mobile obstacles. Then we implement an obstacle avoidance behavior for Jackal robot. The closed environment is a square region. And we try to navigate Jackal robot to move along one diagonal axis of the square. The fixed obstacles include five solid spheres placed in the square region. The mobile obstacles include another ten Jackal robots. These ten Jackals are initially aligned along the vertical direction and move towards the horizontal direction to ensure that there is a chance they can collide with the Jackal robot we use to realize a go-to-goal behavior. When these mobile obstacles hit the environment boundaries, static obstacles or other moving obstacles, They will reverse their linear velocities and continue moving inside the closed environment until our navigated Jackal robot reaches its destination. The Jackal robot's behavior strategy combines not only what we learnt from previous labs but also an newly implemented method to detect and avoid multiple obstacles. Its moving behavior composes three methods to deal with normal go-to-goal, static obstacle avoidance and mobile obstacle avoidance respectively.

In order to determine which situation Jackal robot is facing with, we calculate the difference of the direction angle between the goal and its own pose at first. When the goal is in the front of Jackal robot, we use the laser data to obtain the distance between Jackal robot and the closest obstacle. If the distance is not small enough, Jackal robot will follow the normal go-to-goal behavior to adjust its orientation according to the the direction angle difference. When the distance is below a certain threshold, we need to decide whether the closest obstacle is static or mobile. To do that, we compare the relative velocity between the obstacle and Jackal with Jackal robot's own linear velocity. If the relative velocity is larger, Jackal robot assumes the obstacle is mobile. It will decide which direction the obstacle is moving and adjust its own angular velocity to move towards the opposite way. If the relative velocity is not larger than Jackal robot's own linear velocity, Jackal robot assumes the obstacle is static. It separates all lidar data in its front direction into six groups and obtains the smallest distance within each group. Among these smallest distances we can find out which group has the largest one. Then Jackal robot adjust its angular velocity towards the corresponding direction.

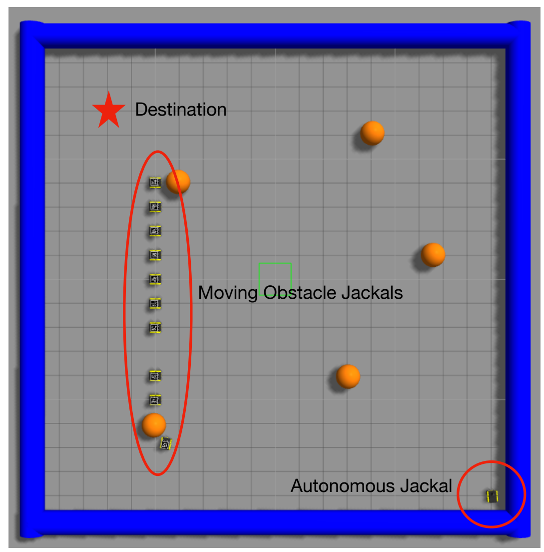

FIG. 1. The test Gazebo world schematic plot. There are ten moving obstacles Jackals. The Moving obstacles Jackals follow the moving-hit-moving motion strategy. The obstacle jackal will move in the world until hit something and moving backwards until next collision. The Autonomous Jackal starts from the right down corner and navigates to the top left corner destination. In this process, our autonomous Jackal needs to avoid moving obstacle Jackals and still obstacle along the way.

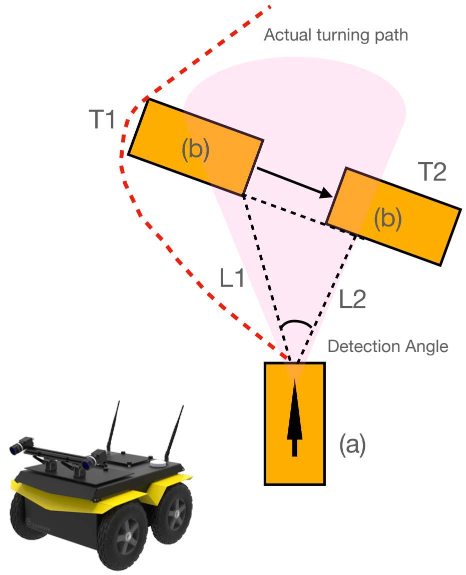

FIG. 2. Schematic plot of the moving obstacle avoidance. The Jackal (a) is the fully autonomous jackal which is navigating to the target coordinate. Another Jackal (b) is moving from (a)'s left upper corner to (a)'s right upper corner. The Jackal (b) is approaching (b). Without any maneuver of autonomous (a), the two jackals are going to collide. Our autonomous Jackal (a) calculate the relative speed of the moving obstacle based on the detective angle and the relative distance L1 and L2. These data are collected from Lidar detection only. The Jackal (a) compares the relative speed of the obstacle Jackal (b) and decide whether the obstacle is a moving object or a still object. In the case of the above figure, the Jackal (a) is able to identify that the relative speed of between (a) and (b) is larger than the forwarding speed of (a) which indicates the jackal (b) is an approaching moving obstacle. Also with the L1 and L2 difference that reflected on the Lidar, the Jackal (a) also recognize the moving direction of the obstacle (b). In which case, the autonomous Jackal (a) will turn left and follow the path in red dash line avoid the Jackal (b).

METHODOLOGY AND IMPLEMENTATION

The framework of this project includes 3 stages: construct the map; design obstacle jackals behaviors; design the strategy for the main jackal. Firstly we need an closed world map, surrounded by wall objects. We firstly tried the competition map, however, the wall objects used in that map are not compatible with the moving patterns we designed for the obstacle jackals. We then decided to put our jackals in the square map surrounded by 'cylinder-shaped' walls, which is used in the lab 1. When those obstacle jackals hit the boundary of the map, it'll be less likely for them to rollover due to the slope of wall. With the boundary set, we also decided to keep several static sphere obstacles on the map, to increase randomness. Given the fact that this map we are using is rather small(20x20), we set the start point at bottom right corner and the goal at top left corner, so the target jackal could move along diagonal for furthest distance. Secondly, we need an automatic motion pattern that has enough randomness and some mechanism to avoid getting stuck for the obstacle jackals. The reasons are that: (1)by increasing the randomness of obstacle jackals motion, we hope to create more variation in their routes, so more ground of this map could be covered and there will be a high probability for the target jackal to cross over with obstacle jackals; (2)sooner or later after the demo starts, some obstacle cars would inevitably hit each other or collide with the wall. So we use IMU sensor to monitor each obstacle car's acceleration. If we see that one of the acceleration exceeds our threshold value, we consider that jackal is hitting into something, so we flip its linear velocity. With this mechanism in mind, one could imagine that all the obstacle cars will follow their own fixed route and 'bounce' between two end points of that route. However, since we have many obstacle jackals and there are also several static sphere objects placed on the map, small deviation from one jackal's original path would lead to a 'chain-effect' that dramatically randomize the whole scene, which is exactly we want. The next thing is how and where to place these obstacle jackals. After discussion and experiments, we eventually decided to line up all the 10 jackals along x-axis. The reason we chose this formation is stated before: this formation of jackals would 'scan' the entire map along y direction, thus cover as many as space and increase the chance of that one of the obstacle jackal would come into a cross with the target jackal and clearly show the avoidance capability of our target jackal. In terms of implementation, we modified the corresponding file in the 'world' folder to add and remove static obstacles. We had also changed file in the 'launch' folder to add 10 more jackals as our moving obstacles, each has a QR code cube object planted on the top, to differentiate them from our main target jackal. We also came across a problem in the codes that the 10 obstacle jackals cannot move simultaneously, we fixed that by only allow one spinOnce() and sleep per loop, while using 10 independent flag variables to control when the corresponding jackal inverses its direction. After stage 1 and stage 2, the initial setup should look like figure 1. The third and most important stage of our project is to design the strategy for the target jackal. We recalled what we have learned through this entire semester and finalized the behavior into 3 pattern modes. Mode 0 is for static obstacle avoidance, which utilizes the data retrieved from lidar unit. The lidar has a scan range of $\pm 135°$, however, for the purpose of avoiding static obstacles in the front of the target jackal's path, we only need the data from $\pm 90°$ semicircle region. Notice that the raw data from lidar has some noises, we then divide

this semicircle region into 6 sectors and each sector covers 30°. For each sector we find the smallest distance. Then we sort these 6 values and find the largest one. If the target jackal turns toward the corresponding sector, it has the lowest chance of running into an obstacle. Next is mode 1, which we called 'go-to-goal'. This mode usually has the highest priority, especially when the target jackal accidentally runs oppositely to the direction of goal, so whenever the goal direction is not within the ±135° region of the target jackal, the mode 1 will be switched on to correct its path angle. For angle correction, we use this formula:
$\theta_{error} = arctan(\frac{y_{goal}-y_{jackal}}{x_{goal}-x_{jackal}}) - yaw$ where the y_{jackal}, x_{jackal}, yaw are updated in odometry sensor. In order to avoid over-turning, the correction angle θ_{error} is always converted back within ±180°. Finally we have mode 2 for the moving obstacle avoidance. The crucial part is to get the relative velocity and compare it with the current velocity of our target jackal. Refer to figure 2, with jackal(b) as an obstacle and moving from left to right, at time T1, the lidar detects a distance L1, we record it and its corresponding angle θ_1. At time T2, the shortest length in the lidar data is found to be L2 and its angle is θ_2. Thus the relative velocity can be calculated as following:

$$\theta_{diff} = \theta_1 - \theta_2 \tag{1}$$

$$v_{rel} = \frac{\sqrt{L1^2 + L2^2 - 2L1 * L2 * cos(\theta_{diff})}}{T2 - T1} \tag{2}$$

If v_{rel} is obviously outside the range we set around target jackal velocity, then it means this object (b) is moving. Also notice that from the sign of θ_{diff} we can find the direction of this moving obstacle jackal, in this case, $\theta_{diff} < 0$, which means the jackal(b) is moving from left-side of target jackal to its right-side. With this information, we let the target jackal to turn left and slightly lower its velocity, to give its more time to adjust its movement.

CONCLUSION

In this project, we successfully proposed and demonstrated our strategy of avoiding moving obstacle by Jackal robot which has the potential to implement as a real world pedestrian detection methodology. We created a closed environment including Jackal robot and multiple fixed and mobile obstacles. Then we create an obstacle avoidance behavior for Jackal robot. The closed environment is a square region. And we try to navigate Jackal robot to move along one diagonal axis of the square. The fixed obstacles include five solid spheres placed in the square region. The mobile obstacles include another ten Jackal robots. These ten Jackals are initially aligned along the vertical direction and moving towards

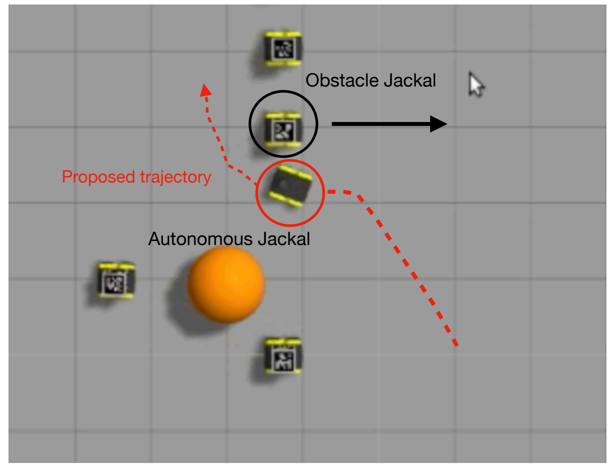

FIG. 3. Autonomous Jackal moving obstacle avoidance simulation result example.

the horizontal direction to ensure that there is a chance they can collide with the Jackal robot we use to realize a go-to-goal behavior. When these mobile obstacles hit the environment boundaries, static obstacles or other moving obstacles, They will reverse their linear velocities and continue moving inside the closed environment until our navigated Jackal reaches its destination. The Jackal robot's behavior strategy combines not only what we learnt from previous labs but also newly an implemented method to detect and avoid multiple obstacles. Its moving behavior composes three methods to deal with normal go-to-goal, static obstacle avoidance and mobile obstacle avoidance respectively. Only with the detection data from Lidar on board, our strategy shows that the Jackal robot is able to avoid moving obstacles with confidence and largely reduce the possibility of collision. This strategy shows potential of real world implementation that can increase the level of autonomous driving.