New Approach to navigation with obstacles identification

*Note: Sub-titles are not captured in Xplore and should not be used

1st Lijing Zhou
University of Detroit Mercy
University of Detroit Mercy
BeiJing,China
zhouli4@udmercy.edu

2nd Ran Duan
University of Detroit Mercy
University of Detroit Mercy
BeiJing,China
duanran@udmercy.edu

3rd Jing Hanyi
University of Detroit Mercy
University of Detroit Mercy
BeiJing,China
jingha@udmercy.edu

4th Zifan Zhao
University of Detroit Mercy
University of Detroit Mercy
BeiJing,China
zhaozi1@udmercy.edu

***Abstract*—Nowadays CNC robotics have developed very perfect and have been applied to many factories' production lines. Robots that perform autonomous path planning and maneuvering in an unknown environment always need to identify the most obvious distinction like the changing environment and planing the suitable route to catch the robot . The robot is continuous searching for the object based on the specialized requirements of the surface color and shape before catching the object. In our research, the input variable is the environment, the coordinates of the start point and end point. On the one hand. for the recognition, a critical aspect of VFH is build a vectogram to choose the possible section to define the angular and linear speed . On the other hand , we need to create the route planning algorithm that is same as VFH but not VFH to enforce the robot to move the object to the goal fast after searching the position of the goal. In addition, we consider the obstacle avoiding during the whole program . In conclusion, the algorithm enables the robot to avoid the obstacles the and to move it to the goal .**

I. INTRODUCTION

Nowadays, the increasing numbers of mobile robots have become common in our daily life. More and more people realize the importance of the mobile robots in completing complex tasks. Furthermore, engineers starts to focus on the objects color and shape recognition and maneuvering in an unknown environment. The objective of the task is to do the correct navigation and find a suitable path plan to get the destination when avoiding obstacles. The algorithm we created is based on the VFH and it can give the precise obstacle avoiding in the gazebo world environment and really world. The VFH algorithm created histogram to divide the part that the robot can run to guide the robot to the goal. We use the similar theory to divide the parts based on the lidar. The lidar is like a radar to launch the light and accept the feedback to identity the obstacles and return the distance. The different angle between the start point and the end point also can influence the linear and angular speeds that we want to give the robot

II. MODELLING

In the section, the models during the project will be introduced. It mainly includes the main software and hardware parts. The theory of controlling the really robot is shortly introduced.

A. GAZEBO Software Modeling

This section introduced the models and environments used in this paper. Environment Our model is built in Gazebo and controlled by MATLAB. Gazebo provides the robot model (turtlebot), LIDAR (Hokuyo) ,ground and obstacles with different size and shape. MATLAB is used to get data from odometry and LIDAR, then calculate the desired moving direction. Finally it is converted into angular velocity and linear velocity before sending to the robot model.

B. LIDAR Hardware Modeling

The Hokuyo sensor can detect obstacles and gives the feedback of the distance. Gazebo can download the lidar topic and the angle range is form -135 degrees to 135 degrees when we assume the front direction is relative 0 degree. The distance range is 0 to 10 meters. The 720 datapoints created by LIDAR are divided with 0.375 datapoints in each sector. In really robot modeling we use the Hokuyo UST-10LX Scanning Laser to return the necessary data. The scan angle is 270 with the angular resolution of 0.25. The detecting distance is from 0.06m to 10m. The different thing is the Hokuyo UST-10LX Scanning Laser divide the angle range into 1080 datapoints based on its angular resolution.

C. HUSKY

Husky is a robot designed by the clearpath company.By providing the speed, the robot can be guided to move with special speed. The robot can be supported by the ROS systhem and by open source code. Husky has very high resolution encoders that can deliver improved state estimation and return precise odometry data. In fact, with the speed provided, the robot itself can incredibly provide smooth motion profiles with vary slow speeds that is less than 1m/s and with reject some rather large disturbance

III. PROPOSED ALGORITHM

The whole process of the algorithm are shorten below. We firstly collected the data from the lidar and give the coordinates of start point and end point in the gazebo world. Then three factors are calculated by the data. Using the three factors to define the final data, the suitable velocity will be chosen til the robot reach the end point.First of all, LIDAR data is divided into several groups (no. 1-45), then eliminate too dangerous group, and then based on three criteria for each group (safety degree-avoidpo, path-pathpo, robot direction-headingpo) according to the different weighted scores(a,b and c). In the next step we sum the score and get the number of group with the highest score, according to the subtraction of number of this group and the number of the middle number (23). Finally the angular velocity and linear velocity is calculated.

$$choice(i) = a*avoidpo(i)+b*headingpo(i)+c*pathpo(i); \tag{1}$$

A. Obstacles Index

Avoiding obstacles is the most basic and important requirement of path planning in this paper. After separating the distance data getting from LIDAR, We firstly get the distance data through the radar, then divide the data into 45 groups in order, and then sum up the data of each group.Exclusion of groups with data below the safety line and groups adjacent to these groups is directly considered a risk area.

$$impossibleSector = find(hm < 180); //safetylineis180 \tag{2}$$

However, if there are less than three groups meeting the safety line, our algorithm will automatically supplement the three highest data as an alternative. After that, the data is smoothed according to the method provided by VFH,

$$h(k) = (h_(k-2)+h_(k-1)+2*h_(k)+h_(k+1)+h_(k+2))/5; \tag{3}$$

and the current values of each group are subtracted from the maximum possible value.

$$avoidpo(i) = 240 - h(k); \tag{4}$$

Finally the score of the group with obstacles will be lower than that of the group without obstacles, and the closer the obstacle is to the robot (Lidar), the lower the score will be.

B. Path Index

Firstly, the head orientation, theta, and the angular of the line from the present position and end point position ,degree,aplha, can be calculated based on the current data from the odometry. Then the index of all possible sectors in the robot frame can be transformed to the count sector index, C.

$$C = ((allpossiblesector.*270/45) - 135 - 270/45/2) \tag{5}$$

when we assume the mid sector (23) is equal to 0 degree. The index of all possible sectors in the gazebo world frame can be transformed to the count sector index, DC. And translate it into range of 0 to 2 *pi.

$$DC = \theta * 180/pi + C; \tag{6}$$

$$DC(DC < -180) = DC(DC < -180) + 360; \tag{7}$$

$$DC(DC > 180) = DC(DC > 180) - 360; \tag{8}$$

Then we calculate the absolute value of the deviation between the DC and aplha.

$$deviation = abs(DC - \alpha); \tag{9}$$

For any value that is more than 180 degree we make it less than 180 degree. It means it will rotate from the opposite direction

$$deviation(deviation > 180) = 360 - deviation(deviation > 180); \tag{10}$$

Next, we sum the deviation up

$$sumdeviation = sum(final); \tag{11}$$

In order to make the suitable proportion index. By the below equation, we make the value of the diviation into range of 0 to 100;

$$mark(k) = (2000*(sumdeviation - final(k))/sumdeviation) - 1900; \tag{12}$$

Finally the path index,path-po , is equal to the rate of the sum of the marks:

$$pathpo = (mark/max(mark)).*100; \tag{13}$$

C. Heading Index

This score has the least weight.The idea is to make the robot think about its direction and turn as little as possible.As shown in the figure below, the robot will be more inclined to drive along path A after adding this criterion.

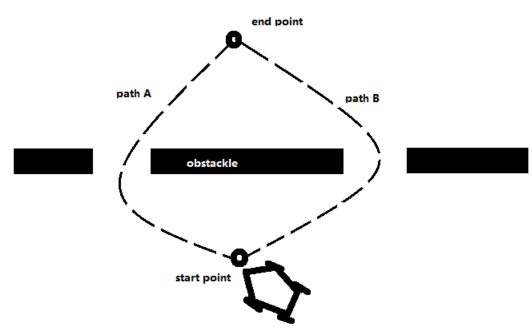

Fig. 1. heading figure

D. brief result expectation

Assumed that the special environment is given and several obstacles are existed in the gazebo world. The initial position can be thought as [0 0]. The end position can be assumed as [10 10]. The algorithm will enforce the robot to drive to the end position with avoiding obstacles and planing a suitable path. The path will show the advantages in smooth and faster aspects.

IV. RESULTS

The special environment built in simulation aims to create a simple and reliable simulation of the algorithm in order to achieve the goal in the real world. The tool named Gazebo can create a 3-D environment with Robot Operating System (ROS). Firstly, the initial environment are infinite. And the scale can be seen as the gridding with the 1m x 1m size. The first map called mapa can be seen as the start map to record the start point of the robot, the obstacles and the end point of the robot. The second map named mapb is made to show the path of the movement. The clear path will illustrate how the robot avoids the robot and gets to the end point.

A. Results in Mapa

- During this map , there are 3 obstacles and a robot.
- The robot is located in the [10 10] when it moved from the center of green square [0 0].
- In addition, each grid with size 1m x 1m and we design the three obstacles between the straight line from the start point to the end point.

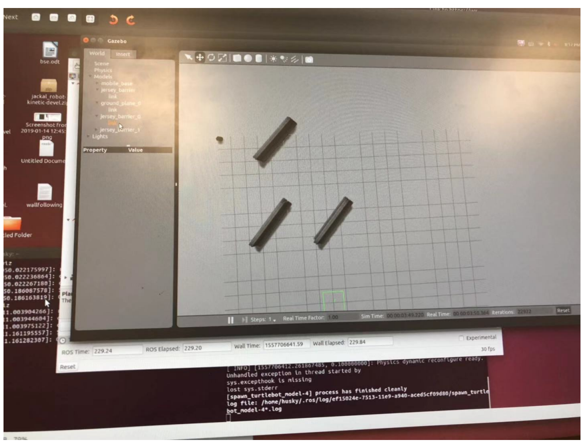

Fig. 2. Mapa

B. Results in Mapb

- It is clear that the robot moved to the end point without touching any obstacles.
- According our algorithm, the robot just turns its direction when it detects the obstacles within the detection of the ladar.
- The arrow shows the direction of the robot, which means the path is made by the movement of the arrows.

.

C. The direction

The direction of the robot in the rviz is shown as red arrow. The software record the change of the arrow to record the path of the robot. From the path, we can see how the robot change its direction when it meets the obtacles.

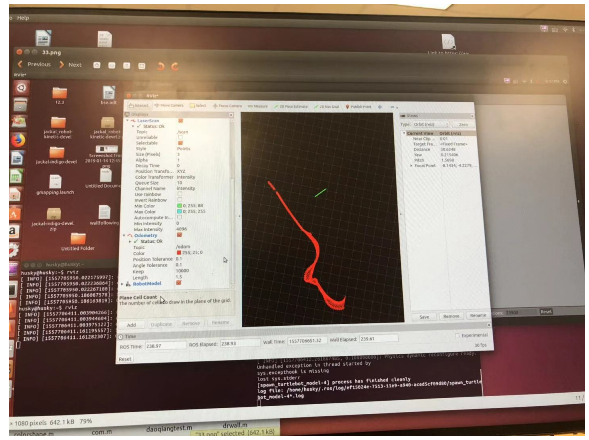

Fig. 3. Mapb

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] https://www.roscomponents.com/en/lidar-laser-scanner/85-ust-10lx.html.

[4] https://www.roscomponents.com/en/lidar-laser-scanner/85-ust-10lx.html.