**Introduction**

Good afternoon everyone, my name is Zixiang Lu and I’m going to present my final year project – the design, implementation and evaluation of an autonomous vehicle with 2D mapping.

As you may have noticed that autonomous vehicle has become one of the **hottest topic** among recent scientific research, and a large number of companies have invested into this emerging domain.

So this is the **motivation** that I select this research topic.

However, when talking about autonomous vehicle, I believe the **first thing that emerges** in front of your eyes is that – **Google self-driving car** (now they call it Waymo).

But in fact there are also **other more applications** in autonomous vehicles when looking at a broader field, for example, **home robotics and planet exploration rover**, all of them could **travel autonomously** without human intervention and simultaneously **accomplish their specific tasks**.

Although they are designed **for different application scenarios** and they might require **different levels of intelligence, efficiency and reliability**, the common fundamental challenge an autonomous vehicle have to face is **self-navigation**.

Just think about driving. To be honest, driving is **not a very complicated task** and I’m sure all of you have a driving license. Driving is just actions like **accelerating, braking and turning**, right? But what makes self-driving so difficult is the **complexity** and **uncertainty** of the environment to deal with.

In terms of navigation, we have the most popular **GPS navigation** but unfortunately GPS cannot provide a precision less than 10 meters that makes **indoor applications impossible** and **on road vehicle unreliable** to use this technology. And absolutely, it’s impossible to use **GPS on Mars**.

So this means that autonomous vehicles have to rely on its **internal sensors** to collect the information of nearby surroundings to provide references for self-navigation. Lidar or stereo cameras are the typical sensors used on autonomous vehicles.

So I can say mapping and localizing technique is the **premise** of autonomous driving. On the basis of mapping and localizing, autonomous vehicle can perform more advanced tasks like **decision making and path planning** to achieve self-driving.

**Aims and Objectives**

Therefore, the **aim of my project** is to develop an autonomous mapping vehicle. As it’s impractical to develop a real on road vehicle, this project focuses on **small-scale application** in an indoor environment based on a mini robotic platform.

These are the 5 objectives of my project:

1. Do relevant research
2. Construct a robotic platform
3. Integrated on sensors
4. Develop a mapping algorithm
5. [Advanced] Integrate on Raspberry Pi for more advanced tasks

**Ideas**

Since most of the indoor environment is surrounded with walls, a **widely proposed** mapping method is to **find and track the walls** then record the trajectory of the vehicle to form a map.

This method is **quite simple and easy to implement**, but it is obvious that this is not a suitable solution in the practical world, as it is so **ridiculous** for a self-driving car to follow the walls to obtain a map.

So this method was not taken into my consideration because of its **low application prospect**.

My basic idea of autonomous mapping was to **scan around the surroundings** using a low cost ultrasonic ranger then **plot the locations** of the detected obstacles on a map.

This method is more similar to the mapping methodology of a real autonomous vehicle that uses Lidar scanning. Taking advantage of the **flexible movement** of my 2 wheel robot platform, the initial mapping procedure I have attempted is like this. The vehicle **spin 360 degree** and **simultaneously measuring the range** to the nearby obstacles than convert the collected data on a single map.

There is a **maximum range and a maximum angle limit** so the vehicle need to **switch the position of scanning**. In this method, it first turns to the direction **towards an empty area**, and **moves straight** to another location then **perform another scanning** to collect more data from the environment.

**Repeating the scanning and moving**, the vehicle is able to map out roughly a layout of the environment.

This idea was carried out during the **first half of the project** period and it generally works well, except for some **inaccuracy at corners** cause by the limitation of ultrasonic sensor.

However, as you may notice that this method **requires a very accurate measurement** for the turning angle of the entire vehicle but the gyroscope used in my project **fails to provide very** **fast and accurate readings**, it takes a 10ms delay for one reading and only provide an angle precision up to 5 degree. So it significantly increases the error of the map over a long period of time and make the angle control very difficult.

Additionally, it’s apparently too **slow and inefficient** to turn the vehicle an **entire cycle**.

So after the gyroscope on my robot **broke down** at some point of the project, which means that I **cannot use the gyroscope sensor** to measure the angles any more, I finally decided to **abandon this idea** and change to another method like this and it is now my final version of my autonomous mapping robot.

**Instead of spinning the whole vehicle**, I switch to **spin the ultrasonic sensor** mounted on a stepper motor just like a **radar** (I call it ultrasonic radar). This time the rotation angle can be obtained very accurately thanks to the high precision of stepping angle the stepper motor provides (which is up to 0.085 degree that is almost 60 times more than before).

In addition, the vehicle can **move freely along a curved path** rather than turning to the direction then moves straight, so I do not need to spend too much effort on **forcing the vehicle to turn very accurately and move very straightly**. This method can totally **get rid of the gyroscope** and it is more **similar to a real autonomous vehicle**.

**Design**

With this general idea, a dedicated design was developed **step by step**

1. First, in order to formulate the map, **a rectangular coordinate system** is initially created and every **obstacle locations and vehicle positions** are expressed as **point coordinates** being plotted on this coordinate system. I call this coordinate system the **Global coordinates** because this is the standard coordinate system that later collected data points referred to. For convenience, the **initial orientation** of the vehicle is regarded as the positive x direction of the coordinate system.
2. The second step is to obtain the **location of the nearby obstacles** and **plot them on the Global coordinate system**. The **range R** from the obstacles to the vehicle can be measured by the ultrasonic sensor, and the **corresponding angle theta** is a known value from the stepper motor. After a very simple trigonometry calculation, point coordinates of the obstacles can be converted to the Global coordinate system. Spinning the ultrasonic sensor by **360 degree**, an **array of point coordinates** can be obtained and after plotting them on the Global coordinates a **general map** **of the surroundings** can be drawn. Express in matrix form is like these.
3. The third step is to obtain the **real-time vehicle position** with respect to the initial vehicle position. After suffering from the failure of gyroscope, I have attempted many other ways to replace the function of gyroscope. Eventually I found that both the **turning angle and the vehicle position** can be calculated by **geometry derivation**, with only the information of the **covering distance of left and right wheels** which can be easily measured by a **wheel odometer**. The derived formula for the yaw angle is like this and the vehicle x,y position is like this, they only require two variables L1 and L2 representing the covering distance of left and right wheels.
4. Then the next step is to convert the coordinate systems. As mentioned before, the **Global coordinate system** is constructed in terms of the **initial position** of the vehicle. But the vehicle has to travel to other positions to collect more data points, so for the convenience of the trigonometry calculation, a **Local coordinate system** is created in terms of the **latest vehicle position**. The detected obstacle locations are **firstly plotted on this Local coordinates** so a conversion of the Local coordinate system to the Global coordinate system is essential to **keep the entire map consistent**. These are the derived matrix for calculating the **latest obstacle location and vehicle position**. I will not go to the details and if you are interest in the derivation please read through section 5.4 of my thesis.
5. The final step is to **enable the vehicle with autonomous driving**. This is achieved by adding a route planner. There are lots of lots of route planning designs but considering the **complexity and feasibility** on my vehicle platform that uses **cheap sensors and microcontrollers**, a relatively simple route planning algorithm was designed. Generally I have divided the nearby surrounding of the vehicle into **8 directions** This route planner is based on **comparing the total range of each of these directions**, a larger range means a larger space the vehicle can travel to, so the vehicle keep searching the largest total range to **find the** **emptiest space** of the environment. These three situations are the examples. The vehicle can **directly know the correct direction** of the next movement, after integrated on this route planner

**Implementation**

To implement the designs, **two core units** need to be integrated on the vehicle platform.

1. The first core unit is the ultrasonic radar. I mounted an ultrasonic sensor on a stepper motor to produce a **360 degree range measurement** without turning the vehicle. To **prevent messing up the wires**, the radar **turns back reversely** after spinning a full cycle. This is somewhere **alike the Lidar** unit commonly used in autonomous vehicles for mapping. But obviously the precision and speed of this ultrasonic radar is far lower than the Lidar. Since the cheapest Lidar available in the market is still unaffordable for my project.
2. The second core unit is the wheel odometer that measures the traveled distance of each wheel. Taking advantage of the **color sensitive feature** of IR sensors, I **marked the black tires with identical white labels** and use an IR reflective sensor to detect them. **Counting** the number of marks detected and **multiply** by the length of the mark, the **covering distance of the wheel**s can be obtained. Both steering wheels was equipped with these odometers, so that both wheels can be **individually measured**.

This is an overview of the components used on my vehicle. I have actuators like the DC steering motors and a stepper motor. Sensors like ultrasonic ranger and IR reflective sensors. Arduino is used to control and communicate with the actuators and sensors and Raspberry Pi is used to process the collected data and also for map display.

**Testing**

After completing the construction and coding of the vehicle, **3 tests** was performed to **examine the behavior** of my autonomous vehicle.

The first test is to test the performance of the ultrasonic radar. I **fixed the vehicle position at a static point** and let the radar **scan through a square space**. As you can observe from the resulting map that there are **two major problems** of the ultrasonic radar, one is the **imperfect mapping at corners** and another is the **curved straight walls**, these are all caused by the **wide beam and low accuracy** of the ultrasonic sensor.

The second test is under the environment of an **enclosed rectangular space.** Because of the **angle limit and range limit** of the ultrasonic sensor, the vehicle have to dynamically move to other positions to collect more data points. In this test, three mapping positions were selected. After finish collecting the data points at each of the 3 mapping positions and convert them all in one Global maps, a rough **rectangular shape emerged**. Again, the major problem is **still at the corners** and many **ridiculous points** exists.

The final test is to examine the **behavior of the route planner**. A **non-enclosed “L” shape space** was used for this test to see if the vehicle with route planner **can turn at the corner** and **find the “exit”** located on the top. As a result, the vehicle successfully avoid crashing at the corners and found the ‘exit’ of this space together with an 80% accurate map. Take a look at the travelled path of the vehicle which is represented by the red dots, the vehicle actually planned an efficient route towards the exit. However, from the displayed map, the problems at the corners still exist.

This video shows the mapping process in this test.

Conclusions

In summary, this project **generally completes the aim** which is to develop an autonomous mapping vehicle. And the strength of this vehicle is low cost, simple control and real-time map display is achieved, while there are also limitations like errors in corners and straight walls, and it’s also a relatively slow method as the vehicle should **wait the ultrasonic radar complete a full cycle** to move to the next position. Improvements like using a Lidar together with an occupancy grid map can be made to solve the problems.

So this is the overview of my presentation,