Research on the Controlling Process in Autonomous Driving

Chuyu Xiao
Beijing No.80 High School, Beijing, China
gaoming@cas-harbour.org

Abstract. With the development of science and technology, self-driving automobile will become increasingly available in the future, and our lives will become more convenient with them. Autocar is in an area of robotics, belonging to the service field. It can be used to make our lives better and reduce the cost of human driving the car, thus will have a great effect on human’s life. However, people do not understand the mechanism of self-driving quite clearly, it is necessary to present some basic operating process and related algorithms of the automatic cars. Meanwhile, in order to show how a self-driving car could control itself using various sensors and algorithms, which is also the basic idea of the automobile company to make a new autocar. Different sensors that are able to detect the behavior of the self-driving cars are used in this paper to fit in different algorithm that follows the principles of motion and robotics. It is concluded that the car could theoretically drive safely, and could avoid crashing to other cars, people or barriers with the safety algorithm. Also, the car could follow the traffic rules properly and plan the most efficient route to get to the final location.

1. Introduction
Nowadays, with the development of new energy sources and artificial intelligence, some technical companies designed the autonomous cars, but currently there are no companies that are able to offer a fully autonomous ride in any conditions, on any road, with no human overseer. [1] Controversy emerges when it comes to the driving danger issues, as the AVs behavior will not be based on individual moral or impulsive reactions, but on ethical guidelines previously coded in the vehicles’ software.[2] Thus, this article focus on the analysis of how an autocar processes the data from its sensor to control its motion and the research will be conducted with some related references that show current knowledge of dynamics and robotics. Meanwhile, a plenty of calculation will be implemented and the equation of the process will be obtained. Through this paper, researchers and people who are interested in this issue will better understand how the “brain” of robot controls the car and engineers can get a prompt to produce smarter cars.

2. Method

2.1. The rules that robot drivers need to follow
A robotic driver needs to follow some significant traffic rules, so does the model used in this paper:
   (1) Driving on the lane between the dashed lines on the street. DO NOT drive cross the yellow lines.
   (2) Before staying at the crossing, the car should change to the lane associated with the direction of turning.
   (3) When the traffic light turns red, the car should stop, and when the traffic light turn green, the car should go. When the traffic light turn yellow, the car should reduce its speed and stop.
(4) All cars should drive in a limited speed, but the speed should not be too low. There mainly are three steps: sensing, processing and reacting, just like the reflex arc of a person to perform an action. Therefore, sensors, algorithms and machinery are involved in the operation process.

2.2. Sensors
The sensors include two photoelectric sensors and a radar. The photoelectric sensor at the bottom is mainly used to track different lines by spotting white and yellow colors, while the photoelectric sensor on the top is used to check the traffic light and signs. The radar could track the traffic around. These sensors make it possible for the car to make changes.

To detect the velocity the car and the front car, an angular velocity sensor is placed at the axis of the wheel and a laser velocity meter is used in front of the car. Also, a gradient sensor in the car could detect the slope of the road, and an angle sensor is placed at the front wheel to detect the steering angle. Of course, a scale should be used to weight the total mass and weight of the passengers and freight.

In addition, a GPS chip and a compass should be used to identify its position and direction. An atmospheric pressure elevation sensor is also used so that it is able to detect the height of the car. The data gathered is used in combine with a digital map.

3. Algorithm and Mechanics

3.1. Lines on the Ground
In this system, the photoelectric sensor is used so that it is able to detect different colors. It is known that there are a plenty of white lines on the street. The system could get those white lines by catching RGB at different point and then use them to draw a best-fit line to identify the lines. There are six methods of finding the best-fit line: Split-and-merge, Line regression, Incremental, RANSAC, Hough Transform (HT) and Expectation Maximization (EM). [4]

A Split-and-Merge method is selected to find the line on the ground because it is one of the fastest and the most accurate ways. There is a table that present the method of calculation of the complexity.

|                          | Complexity            | Speed[Hz] | False Positives | Precision |
|--------------------------|-----------------------|-----------|-----------------|-----------|
| Split-and-Merge          | \( N \cdot \log N \) | 1500      | 10%             | +++       |
| Incremental              | \( S \cdot N_2 \)    | 600       | 6%              | +++       |
| Line-Regression          | \( N \cdot N_f \)    | 400       | 10%             | +++       |
| RANSAC                   | \( S \cdot N \cdot N_{trials} \) | 30       | 30%             | +++       |
| Hough-Transform          | \( N \cdot C + S \cdot N_R \cdot N_c \) | 10       | 30%             | +++       |
| Expectation maximization | \( S \cdot N_1 \cdot N_2 \cdot N \) | 1        | 50%             | +++       |

In the table, \( N \) represents the number of points in the input scan; \( S \) means the number of line segments extracted; \( N_f \) means sliding window size in Line-Regression; \( N_{trials} \) is the number of trials for RANSAC; \( N_c \) stands for the number of columns and \( N_R \) stands for the number of rows for Hough accumulator array; and \( N_1 \) and \( N_2 \) represent the number of trials and convergence iterations.[3] It is obvious that Split-and-Merge method has the highest speed and relatively low false positives, so it would be suitable for a fast-running car.

About how to guarantee driving between the lines, there are two paralleled straight lines which could connect the front and back wheel. If the left line intersects the identified white line, then the car will need to turn a little to the right, and vice versa. It can be shown in Figure 1.

Of course, this method could be helpful as the car is driving on the same road, but it is not suitable to be used when changing to another road or parking aside. And this will be discussed later.
Figure 1. The red line is envisioned by the car. In graph(a), there are no intersections between the red line and the white dashed line, so the car goes straight forward. In graph(b), the left red line intersects the white dashed line at the blue dot, so it turns right a bit. In graph(c), the right red line intersects the white dashed line, so it turns a little to the left.

3.2. Speed and Steering

3.2.1. Speed measurement. Stationary barriers and moving cars could be regarded as the same because motion is relative. The detection of related speed with the front object is done by a laser speed meter, and this measurement is essential for safety on the streets. The linear speed of the car could be detected by the angular speed sensor when multiply the angular speed by the radius of the wheel. The distance between the front object and the car could be known by the detection of radar. Suppose the previous object has a distance x with the car, which is detected by the radar, and the relative speed is $v_{\text{initial}}$, then the relative speed should be reduced to 0 without hitting the object. There are rules about the distance between cars, for example, at 50 mph, that distance is 229 feet. However, at 70 mph, it increases to 387 feet. [4] The speed of the car related to the ground is already calculated, so the final distance could be known. Then the acceleration a could be derived.

In equation (1), a positive value means that the car should be pushed forward while a negative value means that the car should be braked. In a real life, the algorithm should be done continuously so that it can adapt to various velocities.

$$a = \frac{v_{\text{initial}}^2}{2(x-d)}$$

(1)

3.2.2. Force and acceleration. Let’s move on to acceleration. Newton’s second law will be used in this section. The mass of the car $m$ is calculated by adding the mass of the car itself and the mass of passengers and freight.

The difficulty is to find the net force. In this case, the gradient sensor comes in handy. The car should be pulled or braked by different forces.

Figure 2 shows the forces on a wheel. Actually, these three forces are acted on the whole car instead of a single wheel, but the same principle applies. The angle detected by the gradient sensor is the angle between the incline and the horizontal. Let’s call it θ. The net force of the wheel if no pulling

Figure 2. Diagram of forces on a wheel, notice that the frictional force is stationary up and down a hill.
and braking force applies could be derived using the graph. Suppose a force $F$ is applied and the car is going downward, positive force value means accelerating and negative force value means decelerating. The following equation:

$$F + F_{\text{Gravity}} \sin \theta = F_{\text{Friction}} \tag{2}$$

The frictional force could be represented as $\mu_s F_{\text{Normal}}$, where $\mu_s$ is the coefficient of static friction of the rubber tire, then the equation could be written as:

$$F + F_{\text{Gravity}} \sin \theta = \mu_s F_{\text{normal}} \tag{3}$$

Then $F$ could be represented:

$$F = \mu_s F_{\text{Normal}} - F_{\text{Gravity}} \sin \theta \tag{4}$$

Then a if-method is used to consider whether the brake or the engine will be used and the strength of the force it should exert.

3.2.3. Steering system. Finally, steering, commonly known as changing direction, is considered in the process. A new concept must be introduced: centripetal force. Centripetal force is the force required to keep a moving mass in a circular path. [6] It is calculated by the following equation:

$$F_{\text{Centripetal}} = \frac{mv^2}{R} \tag{5}$$

$R$ is the radius of the arc determined by the road itself. Also, for a car on level ground, the centripetal force is just the static frictional force between the wheel and the ground. So using the equation $F_{\text{friction}} = \mu_s, \text{max} F_{\text{Normal}}$, the maximum frictional force and therefore the velocity suitable for making this turn is determined. Then equation (6) is employed to achieve such a speed.

$$a = \frac{v_{\text{initial}}^2 - v_{\text{final}}^2}{2d} \tag{6}$$

Now the steering-front wheel system used to control the car will be discussed. The Ackermann angle should be known in this section. Ackermann Steering refers to the geometric configuration that allows both front wheels to be steered at the appropriate angle to avoid tire sliding. [6] The control of the Ackermann’s angle will be discussed later in Section 3.3.

3.3. Navigation

Navigation is essential for a car to drive automatically. It helps the car to find the best route from one point to another. Locating the car is the prerequisite of a good navigation. The GPS sensor in the car accurately find the location of the car as a point on the Earth and the direction of it. Any point on the earth could be represented by a longitude value and a latitude value. However, our city is cubic, so an atmospheric pressure altitude sensor measuring the height could more accurately find the car’s location in a 3D setting. Nevertheless, to simplify the problem, altitude is not included in the following algorithm. Thus, the three values (longitude, latitude, and direction) are put into a $1 \times 4$ matrix:

$$p = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} \tag{7}$$

In this matrix, $x$ represents longitude, $y$ represents latitude, and $\theta$ represents the direction of the car. However, GPS location is not always available during the whole journey. In a cave or a tunnel where the signal is weak, GPS would not work very well. Nevertheless, the car could locate by its own perception, just like a taxi driver know where he is while driving somewhere in the city. To find its location without GPS, the car needs to use geometry and calculus to find the final position if its starting point is known.

The final angle changed by the car could be regarded as the Ackerman angle divided by 2, as shown in Figure 3. Notice that there are relations between different parameters of the location, let’s say that $\Delta s$ represents the displacement of the car in its direction. Thus, the car’s change the position and direction in terms of the location is shown in Figure 4.
Thus the following equations is derived from the graph:

\[ \Delta x = \Delta s \cos(\theta + \Delta \theta/2) \]  \hspace{1cm} (8)

\[ \Delta y = \Delta s \sin(\theta + \Delta \theta/2) \]  \hspace{1cm} (9)

Then the final position of the car could be represented as the following:

\[ p_{\text{final}} = \begin{pmatrix} x_{\text{initial}} + \Delta s \cos(\theta + \Delta \theta/2) \\ y_{\text{initial}} + \Delta s \sin(\theta + \Delta \theta/2) \\ \theta_{\text{initial}} + \Delta \theta/2 \end{pmatrix} \]  \hspace{1cm} (10)

Nevertheless, the trace of the car is usually curved, so calculus should be employed. Let’s see how it work. The curve could be seen as plenty of infinitesimal segments adding together, so each of the delta values could be limited to zero and integrated:

\[ p_{\text{final}} = \begin{pmatrix} x_{\text{initial}} + \lim_{\Delta s \to 0} \sum \Delta s \cos(\theta + \Delta \theta/2) \\ y_{\text{initial}} + \lim_{\Delta s \to 0} \sum \Delta s \sin(\theta + \Delta \theta/2) \\ \theta_{\text{initial}} + \lim_{\Delta \theta \to 0} \sum \Delta \theta/2 \end{pmatrix} \]  \hspace{1cm} (11)

Rewrite the equation to integral form in terms of time \( t \), assuming the steering angle \( \Delta \theta \) is constant:

\[ p_{\text{final}} = \begin{pmatrix} x_{\text{initial}} + \int_0^t v \cos(\theta + \Delta \theta/2) \, dt \\ y_{\text{initial}} + \int_0^t v \sin(\theta + \Delta \theta/2) \, dt \\ \theta_{\text{initial}} + \int_0^t (\Delta \theta/2) \, dt \end{pmatrix} \]  \hspace{1cm} (12)

An algorithm in real life usually is more complicated than what has been described above, and this model is just a basic representation. However, it could be a standard for more complicated algorithms within a car for its navigation.

Finding the route of the car is just like finding the best way to get out of a maze. However, instead of the length of the whole route, the car should also find the total time to compare them. It would add different segments of the roads with different speeds and consider stopping at the traffic lights. Of course, the accident and the traffic flow should be noted. Assume the route is divided into \( n \) parts, then the following equation is used to find the total time:

\[ t_{\text{total}} = \sum_{i=1}^n \frac{s_i}{v_i} \]  \hspace{1cm} (13)
Where \( s_i \) and \( v_i \) represent the length and average speed of the ith road segment. After finding different routes to the destination, the only concern is to apply this equation to compare different routes in terms of time. Finally, the car could use reinforced learning and genetic algorithm to be more familiar with frequently driven routes.

3.4. Traffic lights and road signs

As electronic navigation becomes available, every road sign could be represented on the electronic map and a warning would be given out by the system when the car is approaching the sign. When the navigation system in the self-driving car receives a warning, it responds. For example, when it encounters a speed limit sign, it will change its speed to fit in with this sign without interfering with other cars. The mechanism discussed in 3.2.2 will be an auxiliary.

The identification of traffic light is similar to what is discussed in 3.2.1, but at this time, the photoelectric sensor at the top only needs to identify the color of the light by receiving an RGB from a visual angle in front of it. Figure 5 shows the procedure to deal with a traffic light.

![Figure 5. The controlling process of a car at the traffic lights](image)

4. Conclusion

Using the algorithms that have been discussed in this paper, it could be concluded that the car is theoretically able to control itself on the road and park in a lot of places without the help of anyone.

The model in this paper is proposed and simplified to solve the problem of the self-driving cars. However, the proposal is just based on known reasoning and has not been executed in a real-life. To test the car, a modeled or an experimental car should be first constructed and tested in a virtual environment, but the conditions for doing so have not been acquired yet.

In the future, the car’s motion in a 3D range is likely to be analyzed and equations should be used to determine its Ackermann angle more accurately. To test the car, a computer model should first be built to simulate the car and the different roads and terrain. Then a real model of the car will be built and run in a sand table of a terrain. If the trials are successful, the car could be constructed and run in real roads to test its function. There should also be plenty of cars to accompany it so that it would be tested to adapt to the environment.

Acknowledgement

Thanks Professor Fumiya Iida from Cambridge University and my teaching assistant Jiahui Zhu from Shanghai Tech who offered a lot of useful information about mechanics and robotics to me. And I will thank Cetus Talk for instructing me to understand the information, doing the research, and writing the paper. Then, thanks Joy Wang who gave me feedback efficiently and Mandy Han who taught me the...
way to do research and supervise my paper.

References
[1] Adam Kaslikowski. (2019). Everything you need to know about autonomous vehicles.
[2] Margarita Martínez-Díaz, Francesc Soriguera. (2018). Autonomous vehicles: theoretical and practical challenges. Transportation Research Procedia, 33, 275-282.
[3] Bonnie Gringer. (2018) History of the Autonomous Car.
[4] Roland Siegwart, Illah R. Nourbakhsh, Davide Scaramuzza. (2011). Introduction to Autonomous Mobile Robots (second edition). Intelligent Robotics and Autonomous Agents, 9, 242-262.
[5] Greg. (2021). What is a Safe Distance Between Cars at Highway Speeds in Texas?. Baumgartner Lawyers.
[6] Gooch J.W. Gooch j.w. (2011). encyclopedic dictionary of polymers. Springer, New York, NY.
[7] Jonathan Vogel. Tech Explained: Ackermann Steering Geometry. Racecar-Engineering.