# Mini Auto Race Car (MARC)

Tianjian Huang
*School of Science and Engineering*
*The Chinese University of Hongkong, Shenzhen*,
Shenzhen, China
E-mail: 117010099@link.cuhk.edu.cn

***Abstract*—Combining the concepts of smart car and light-weighted hardware platform, Mini Auto Race Car (MARC) project aims at developing a 1/24 scale car model as a powerful tool of academic research and environment. Compared with other similar smart car projects such as the MIT RACECAR, MARC is lighter, cheaper, and easier to use. To achieve MARC's targets, I adopted Ackermann steering structure to the chassis and used lightweight control boards. In addition, I also designed a passive power marker so that the smart car can navigate automatically. Experiments had illustrated that the alpha version of MARC is available to complete its basic functions.**

## I. Introduction

Smart car is a cutting-edge field of study which has appealed to many top scientists and enterprises. Meanwhile, there is a trend of lightweight platforms throughout the engineering design, such as thinner laptops and smaller robots. A combination of these two ideas is a Mini Auto Race Car (MARC), which is a lightweight race car with AI directing its path [1].

A MARC is designed as a 1/24 scale model car. The low-level control of motors and steering servos is taken by a STM32 control board, and high-level algorithms such as computer vision and path planning will be done by a mini computer board like Raspberry Pi. It also supports other peripheral devices like a monocular camera, IMU and odometer. Such a lightweight hardware design allows it to be neat, inexpensive and relatively easy to develop. Currently CUHK(SZ) has one course teaching basic of AI (CSC3180), and it lacks a cost-efficient and intuitive platform which helps students learn how AI works in real-world and perform hands-on experiments about robots and AI. MARC can fill this gap and make a significant contribution to our undergraduate education. Moreover, the MARC is also an open-source platform which allows further developments. With our lightweight, compatible race car platform, students and researchers can implement advanced algorithms to make the car even faster and organize some competitions in order to encourage innovative spirit and give publicity to our university.

The MARC project is not the only racecar project. In fact, MIT has already developed a similar platform called MIT RACECAR [2]. It has been used in multiple courses and workshops since 2015. Hypha ROS is another racecar project which is significantly cheaper and lighter than MIT RACECAR [3]. Hypha ROS racecar needs less gadgets and peripheral devices, and it only costs less than USD$600, while MIT RACECAR costs more than USD$4000. However, they still have some thresholds to a beginner level user. For example, 600 dollars is still a relatively high price for students, and either car's chassis is 1/10 scale, which means the car should be ran in a wide environment. In addition to price and size, many existing smart cars choose single board computer to do all tasks, such as [4] and [5]. Students without advanced hardware knowledge can find it difficult to program on such kind of platforms.

## II. Problem Statement

Constructing a mature product of smart car is an elaborate project, and it needs several groups of engineers and researchers to finish. Since I was the first generation of researcher who start this project, my mission was to build up the alpha version car model which can accomplish basic functions.

A basic set of smart car includes three parts: the mechanical structure, the computer and control structure, and the special marker to direct the car.

### *A. Mechanical Structure*

The first problem is how to choose the mechanical structure. There are several criteria, including size, weight, material of key components, way of driving and steering, stability and robustness, and the total cost. The size and weight should be as small as possible, but not too small to hold computer hardware or battery. The car should be robust enough to support itself and bear accidental crash on

obstacles. As for driving and steering method, we are considering two mainstream designs, which are differential drive design and Ackermann steering design.

### *B. Computer and Control Structure*

Another crucial problem is how to construct the intelligent part, which is the computer and control structure. This problem can be divided into many sub-problems. For example, what kind of hardware architecture should we choose? What software application should we develop? How to connect the control structure with mechanical structure? Also, the size and weight of computer hardware should fit the chassis, and computational performance is also an important factor.

### *C. Marker Design*

Third, we also need to consider the design of marker. A marker is like a landmark which will guide the direction of the smart car, so that the smart car can run in a regular routine. According to markers' signal transmission method, they can be classified into active power marker and passive power marker. Examples of active power marker include traffic light, magnetism transmitter and radio transmitter, which has an internal power source and actively transmit signal to their surroundings. Passive power marker, such as QR code or zebra crossing lines, has no internal power and its signal transmission relies on energy in the environment. Criteria to evaluate a marker include accuracy, minimal and maximal distance can be detected, stability and the difficulty to set up.

Although the number of factors is large, all selection criteria are pointing to the same purposes. The alpha version car model should be cheaper and more lightweight than the average of existing smart car, and it should be relatively easy to use.

## III. INTERMEDIATE RESULTS

Base on the previous analysis of problems, the alpha version car model used the designs as follows:

I chose a 1/24 scale Ackermann steering car model as the chassis. It is driven by two independent DC motors and steered by a servo. Using Ackermann steering structure has several advantages. First, it is relatively easy to complete straight line movement. Compared with differential-steer drive, Ackermann-steer drive has only one pair of rear wheels and the rotation speeds of rear wheels are easier to maintain the same. Second, Ackermann steering structure runs faster and has more motility. Turning process of Ackermann steering vehicle depends on the servo and a set of rods connecting front wheels and the servo. Assuming all rods are rigid bodies, turning speed is directly related to servo's torque and responsive time. The larger the torque and the shortest the responsive time, the faster the vehicle can turn. On the contrary, turning speed of differential-steer drive is restricted because the differential wheels rotating too fast will results in slipping. Third, Ackermann steering structure is more cost-efficient. While an Ackermann vehicle needs at most three motors/servos, a differential drive vehicle needs one motor for each wheel. Differential drive structure has a better performance in outdoor, cross-county environments. However, since MARC is for indoor, car-race scenario, Ackermann steering structure is the reasonable choice.

For computer hardware, I decided to use a double control structure. This design has two control boards, one is for lower level control (LLC), another one is for higher level control (HLC). HLC is the major control board that process most of the tasks, including sensor data processing, path planning and sending velocity commands to LLC. HLC also provides interfaces to allow users to write applicational programs. LLC is responsible for receiving commands from HLC and control mechanical hardware, and sending hardware status to HLC. This design separates low-level implementation and high-level application to the most extent, leaving all controlling details to the LLC, so that users can focus on other tasks in HLC. Considering both financial and performance issues, I chose Raspberry Pi 3 Model B as the HLC, and STM32F301 micro control board as the LLC.

For the software design, I used Robot Operating System (ROS) to construct all programs. As a powerful tool of distributed computer system, ROS provides convenient APIs for communication, configuration, launching and testing robots. It is also a worldwide popular software architecture which has abundant learning resource and mature community. Using ROS not only simplifies software design but also fulfills teaching purposes.

Choice of marker is related to what types of peripheral sensors are allowed. The alpha version uses QR code as its marker and only monocular camera as the sensor. QR code is a kind of passive power marker. It is simple, easy to generate, and can convey abundant information. Moreover, ROS has

software package which supports QR code pose translation using monocular camera. The simple combination of QR code and monocular camera can satisfy basic functions such as navigating and following, and minimize the costs.

## IV. SIMULATION AND EXPERIMENT

The alpha version of smart car has the size of 21.1 cm × 15.2 cm, and it weighs less than 0.5kg. The simplified hardware largely reduces the number of gadgets and sensors, so the smart car only costs less than 2000RMB, about US$285. The lightweight and cost-efficient purposes have accomplished, and the final trial is that whether it can run.

To verify the designs, I programmed and tested a simple path planning algorithm. This algorithm can translate pose information from QR code and generate a sub-optimal global path using Bezier curve. Bezier curve is a traditional way to accurately describe a curve in computer graphics. The formula for 3rd order Bezier curve is as (1) shows.

$$B(t) = P_0(1-t)^3 + 3P_1t(1-t)^2 + 3P_2t^2(1-t) + P_3t^3,\ t \in [0,1] \quad (1)$$

Bezier curve may not be the most optimal path, but it is simple and fulfills the needs of navigating the smart car. It has an additional advantage. Figure 1 shows an example of Bezier curve path [6]. $P_i$ and $P_f$ are the end points, and $C_1$ and $C_2$ are the auxiliary points. According to the definition of Bezier curve, line $P_iC_1$ is tangent to the curve at $P_i$, and line $P_fC_2$ is tangent to the curve at $P_f$. Assume $P_f$ is the position of QR code, and QR code is facing at the direction of $\overrightarrow{P_f\ C_2}$. Line $P_fC_2$ is perpendicular to the QR code, so the smart car will pass the QR code vertically if following the path. This feature provides a good criterion to determine whether the smart car is really following the marker, and it also provides a way to set up the marker. As Figure 2 shows, the QR code can be attached to the top center of a door frame. Theoretically, if the car follows Bezier curve and the door is wider than the car, the car will pass through the perpendicular line of the door without scraping the door frame.

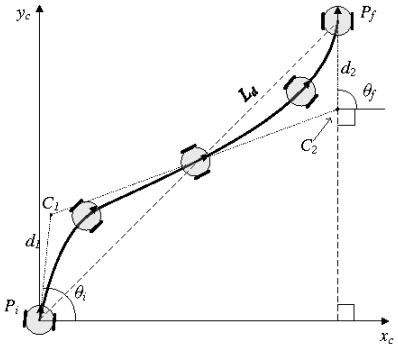


Fig. 1. An example of Bezier curve path [6].

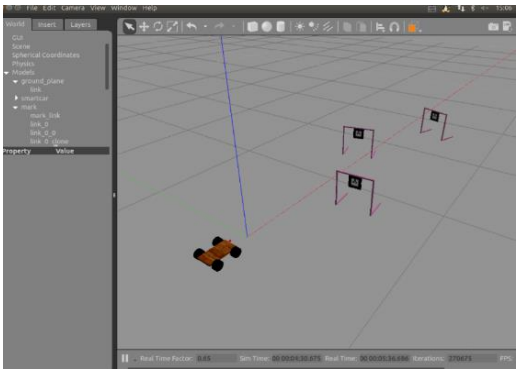

Fig. 2. Gazebo simulation scene. There are 3 doors in front of the smart car, and each door has a marker attaching to its top-center position.

In real experiment, I set the door as 30cm width and 35 cm height, attaching a QR code in the top center of the door. Assume the car was at (0,0) of the ground reference frame, and the car was facing to positive direction of the x axis before the algorithm ran. If the door with marker was detected by the camera, the smart car could walk through the perpendicular line of the door. That justified the effectiveness of my architecture stated above.

## V. CONCLUSION

Currently, the basic structure of MARC has been established. I would like to divide my major contributions into 3 aspects. At the lower level of software design, I constructed a special-purposed Linux environment base on Raspberry Pi 3 Model B, and I wrote basic ROS packages to provide a skeleton for MARC. At the level of algorithm, I programmed and tested a fundamental algorithm demo of path planning. When detecting a QR code, the algorithm will quickly obtain the pose of that code and generate a sub-optimized global path using Bezier curve. Though it is a simple application, it is comprehensive for students who just start to learn and thus fit for teaching purposes. Finally, at the

level of simulation, I also make a Gazebo simulation platform with complete model of the car. Students can test their SLAM algorithm in this virtual platform before putting to a real car. More future works can be done. For example, the Bezier curve path can be more optimal so that the car can move along a shorter path. Algorithms such as sensor data fusion can be added to the navigation module in order to make up disadvantages of monocular image data.

Although MARC still has a long way to be completed, I conquered most of the early stage difficulties and led a promising start for my successors. In conclusion, my missions are completed with all my efforts, and I have gained lots of experience from MARC project.

## REFERENCES


[1] "MARC-Project," GitHub. [Online]. Available: https://github.com/MARC-Project. [Accessed: 14-Jan-2020].

[2] "RACECAR," *RACECAR*. [Online]. Available: https://mit-racecar.github.io/. [Accessed: 01-Jan-2020].

[3] "Hokuyo Racecar BOM (Classic)," *Google Sheets*. [Online]. Available: https://docs.google.com/spreadsheets/d/1uBImPVY82Y1d5BRhywwjz3CI9nG4AKZsTNbz1H29JDM/edit#gid=0. [Accessed: 01-Jan-2020].

[4] X. Zeng, "Design of Smart Car Control System of Path Information Identified Based on CCD Camera", *Journal of Hubei Automotive Industries Institute*, vol. 22, no. 2, pp. 72-76, 2008. Available: http://en.cnki.com.cn/Article_en/CJFDTotal-HQCG200802017.htm.

[5] Y. Deng, H. Zhou and Y. Tan, "Design and Realization of Smart Mini-Car Based on MC9S12DG128", *Research and Exploration in Laboratory*, vol. 26, no. 1, pp. 67-69, 130, 2008. Available: http://en.cnki.com.cn/Article_en/CJFDTotal-SYSY200801024.htm.

[6] J. Jiao, Z. Cao, P. Zhao, X. Liu and M. Tan, "Bezier curve based path planning for a mobile manipulator in unknown environments", *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2013. Available: 10.1109/robio.2013.6739739 [Accessed 1 January 2020].