2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Autonomous RC-Car for Education Purpose in iSTEM Projects

WeiHang Feng*1, Yun Pei*2, Bin Hai*3, WuLing Huang*4, XiaoYan Gong*4, Fenghua Zhu*4

Abstract— **Software simulation and real environment running parallel execution is a lately proposed method, which also provides full coverage and convenience to accomplish autonomous driving education purpose. This paper introduces a new high-school iSTEM program of autonomous vehicles education. This program uses a Scaled RC-Car platform with several sensors and Raspberry Pi embedded platform, to build an autonomous driving car in scaled indoor simulation environment. The RC-Car is capable of safely autonomous driving. Many existing algorithms are put together to provide the necessary functions of autonomous driving, such lane detection, obstacle detection, lane following, vehicle control etc. In this paper, we provide the details of this program, hardware and software components of the RC-Car, Deep learning end-to-end approaching of algorithm deployment, and future works.**

I. Introduction

Autonomous Vehicle (AV) is a thriving emerging thing that already has had great impact, which may revolutionize transportation, while substantially enhancing traffic safety and efficiency. The autonomous driving applications are expected to booming in the future. It is necessary to prepare for it now so that the high school students can build their perfect application in the future. It is accepted that robotics competitions, camps, and clubs all increase interest in Science, Technology Engineering and Mathematics (STEM), which is an exciting and good direction. For some exist programs, students build expensive robots for their competitions. The cost of the platforms is very high and the competitions require the students to build both the hardware and complex software. Most existing programs are unable to address the challenges of the students for their knowledge and learning efficiency, which do not have room for students to design and implement relatively complex existing computer programs.

Currently, an iSTEM program is proposed, which uses several sensors and Raspberry Pi embedded systems for an 1/10-scale RC-Car platform, which aims to build an autonomous car prototype in scaled indoor simulation traffic environment, special designed for AVs education purpose, with several RC-Cars, roads, signs and traffic signals. Several sensors are used to provide necessary data to the car, which is capable of safely and intelligently driving. Many existing algorithms are put together to provide the necessary control to the car, such of lane detection, obstacle detection, car control etc.

This work was partly supported by National Natural Science Foundation of China under Grant 91520301.

*1WeiHang Feng, is with Beijing NO.161 High School. *2 Yun Pei, is with Jilin University. *3 Bin Hai, is with Beijing Institute Of Fashion Technology.

*4WuLing Huang, XiaoYan Gong, Fenghua Zhu are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China.

This program provides built RC-Car platforms and introduces how to carry out the AVs education in simulated traffic environment with or without V2X connection and accession of the traffic information. With the evaluation methods and procedures, we could demonstrate the AVs development and the key algorithms, which is useful reference both for AVs and for the next Intelligent Transportation System. The students only need to focus on software layers applications, learn through several challenging driving tasks assignments. And finally, students should finish a project in a collaborative fashion.

In this paper, we provide the details of the autonomous RC-Car for the iSTEM Projects, especially hardware and software components of the RC-Car, typical algorithm development and future works. The second part is about AVs education for iSTEM Purpose, including using simulation methods and scaled indoor environment to accelerate the education process. The third part is about AVs functions for iSTEM education purpose, including lane detection algorithm, driving with neural network leaning. The final part is our future works.

II. Education of Autonomous vehicles for iSTEM Project

A. *Autonomous Vehicles Education for iSTEM Purpose*

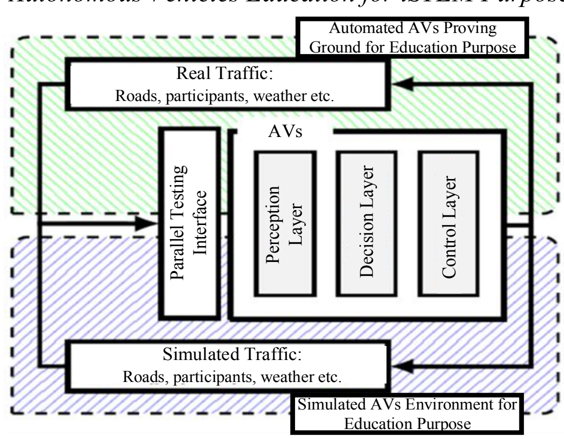


Figure 1. AVs Simulation and Filed Running Parallel Education

The Parallel Education of AVs is an integration of simulation and real systems, including the driving simulation platform [5]. The real RC-Car with different sensors, vehicle actuator and controller, traffic scene definition, driving functions, which provides a method for autonomous driving perception, planning and control functional education, as shown in Figure 1. Besides the real RC-Car, the simulation environment also provides a comprehensive autonomous driving education solution, with vehicle sensors, communication system and function modules, tested with simulation, HIL, and VEHIL methods [7].

For education purpose, the run and test driving of AVs method has its limited, because of the running and testing results are hard to manipulated and sometime costly, especial getting the results of the half-state AVs functions in develop state involved in the education cycle. To overcome the shortcoming of this, a solution (VIVUS) to combine the advantages of simulations with the representativeness of test drives, by extending the HIL environment from vehicle level to the traffic level, is needed [6].

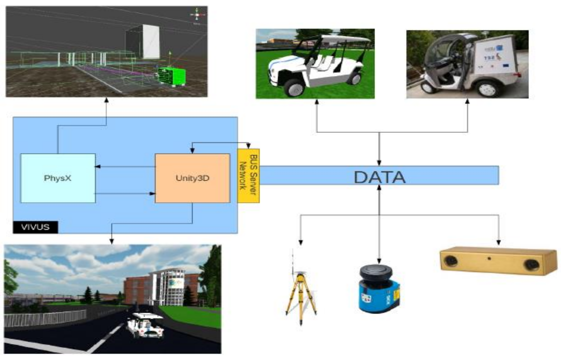


Figure 2. Simulation Methods for AVs Education in iSTEM Projects (VIVUS Referent Architecture)

At the first stage of AVs in iSTEM Projects, the students can run the software in the simulation environments, with necessary sensors components such as radar, lidar, cameras, infrared etc, as Figure 2. At the second stage, students can use scaled or not called autonomous vehicle with VEHIL methods supported, which adds value to the education process of autonomous vehicles, with a lot of advantages. Education with VEHIL methods are performed in a reproducible and flexible way, allowing the detailed explanation with safety-critical, precise and repeatable variation.

B. *Using Simulation Methods to Accelerate the Education Process*

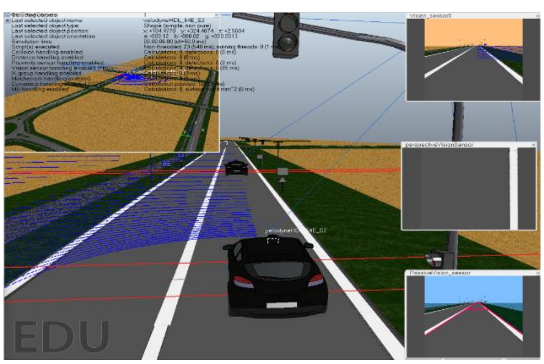


Figure 3. AVs Simulation Environment for iSTEM Projects

There are several simulation and test drive mixed methods for reference, such as ROS Rviz, Gazebo and V-REP etc. We have set up an autonomous driving simulation testing environment, which includes sensors, perception modules, navigation stacks, motion planning, and vehicle dynamic modules etc. We use V-REP, which is a simulator with numerous functions, features or APIs. As Figure. 3 shown, autonomous driving can be simulated, with dynamically load 3D road model, obstacles, vehicles and pedestrians. Autonomous driving software stacks are ROS nodes, including simulated vison, Lidar, IMU+GPS sensors, and perception, planning and control ROS nodes. Therefore, AVs functions can be tested with in various scenarios simulation data collected [4].

C. *Using Scaled Indoor Traffic Environment for AVs iSTEM Projects*

Autonomous vehicles can be tested by driving a certain mileage in the typical real traffic environments with assessment of driving quality, which can also be used in educational purpose, as Figure 4 shown [1], the educational driving tasks are assigned along the route, with different task scenarios. Lists of simple function cases are selected and assembled into different education processes, which are used to further educate the autonomous vehicle Perception Layer Functions, Decision Layer Functions, Navigation Layer Functions and Action Layer Functions [2].

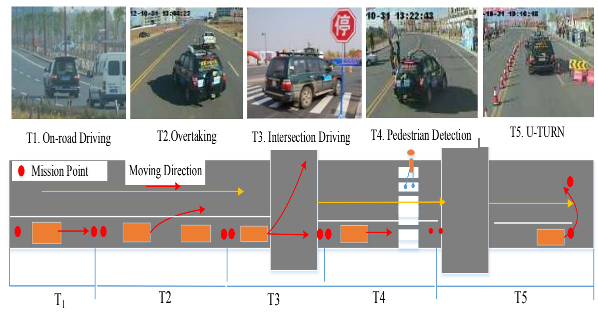


Figure 4. AVs testing cases used for education purposes

The AVs driving education can be also deployed in a certain closed area with scaled and simulated roads, with V2X network, and other traffic facilities support, such as shown in Figure 5. The typical scaled road section is with intersections, traffic signs and traffic lights, which is consistent with the common roads. The testing infrastructure and facilities includes simulated RF base station, Wi-Fi communication station, on-board and roadside units, V2I supported traffic lights controllers, and others. Wi-Fi network simulates the V2X support the vehicle and road communication, makes the automated testing possible. The V2X network provides access to network services or cloud services. The testing data are therefore logged and viewed for algorithms explanation and education.

Figure 5. Scaled Indoor Environemt for AVs Education Purpose

III. Scaled Autonomous Vehicle for Education Purpose

A. *Scaled AV for iSTEM Education Purpose*

1) Autonomous RC-Car Hardware Design

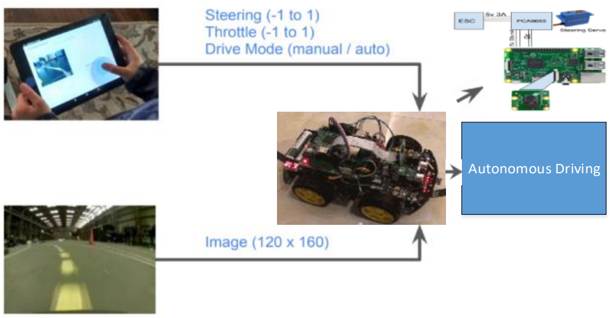


Figure 6. Autonomous RC-Car for Education Purpose

A pre-built four wheel drive (4WD) chassis is used as a base on which following hardware components are fit: Raspberry Pi (rev C) for GPU and CPU computations, Wi-Fi 802.11n dongle to connect to Pi remotely, Motor driver IC L293D which can control two motors, Ultrasonic sensor to detect obstacles, Pi camera, batteries to provide power. The Raspberry Pi is a credit card-sized single-board computer. The BCM2835 Compute Module is an entirely different form factor and cannot be used standalone. In this project, we have used the model C Rev 3. It comprises of a 512 MB RAM model with two USB ports and a 10/100 Ethernet controller. It is the camera shipped along with Raspberry Pi which can be used to take high-definition videos as well as still photographs. Ultrasonic sensors evaluate attributes of a target by interpreting the echoes from radio or sound waves respectively. In this project, they are used to detect the distance of obstacles from the car [8].

2) Software of Raspberry Pi

Raspbian is a free operating system based on Debian Linux, which is available for free from the Raspberry Pi website. Python is a widely used general-purpose, high-level programming language. Its syntax allows the programmers to express concepts in fewer lines of code when compared with other languages like C, C++ or java. The RPi.GPIO Python library allows you to easily configure and read-write the input/output pins on the Pi's GPIO header within a Python script. OPENCV is a library of programming functions mainly aimed at real-time computer vision. It has over 2500 optimized algorithms, including both a set of classical algorithms and the state of the art algorithms in Computer Vision, which can be used for image processing, detection and face recognition, object identification, classification actions, traces, and other functions. In our project is used to detect the roads and guide the car on unknown roads. It is based on C++ but wrappers are available in python as well.

B. *Autonomous Vehicle Architecture for iSTEM Projects*

The architecture of an autonomous vehicle for education purpose should be very formal and should base on the general driver behavior [3], follow a sensor-based and actuator-based autonomous system architecture, consisting of Perception, Decision and Action Layer, as Figure 7 shown.

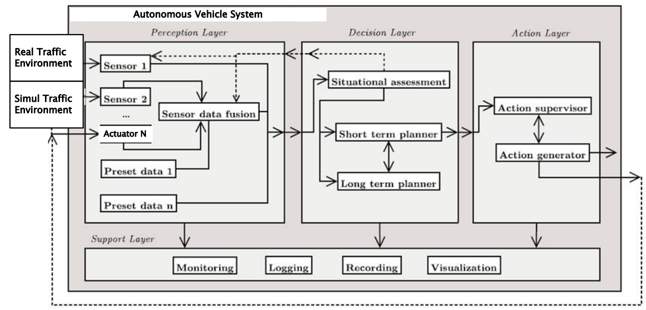


Figure 7. Autonomous Vehicle Architecture for Education

The education of Perception Layer should cover the functions of data acquisition and the data fusion, such as data from vision or radar sensor, which is merged into a unique fusion map, including the detection of pedestrians, lane, traffic signs, cars and other things around the driving area elements.

The education of Decision Layer should cover the interpretation of all incoming data from Perception Layer, which generate a reasonable output to the Action Layer, the short and long-term planners and accomplish the overall goal. It includes Artificial Intelligence algorithms, such as Neural Networks, Machine Learning, commonly used mainly due to the highly non-linear behavior of real environment.

The education of Navigation Layer should cover higher level driving tasks such as controlling the global objectives, trajectory planning, efficiency and commodity, which considers of the driving conditions.

The education of Action Layer should descript the commands from Decision Layer into the action supervisor, which sets up the abstract decision into set points to be fed by the actuators' controllers. The action generator denotes the system controllers and performs the low-level actions in the actuators, also monitoring the feedback variables to further process the new actuating variables.

IV. AV Education Examples in ISTEM Projects

For education purpose, we simplify the functional levels, with simple Perception Layer Functions, Action Layer Functions are used. We implement the functions in scaled indoor environment, with simple traffic scenario. First, we test simple lane detection functions in the scene.

A. *Lane Detection with CV Algorithm*

The road lanes could be detected by model-based or feature-based algorithms. The model-based technique uses a few parameters to represent the lanes, assuming the lane can be presented by straight or curve lines. To estimate the parameters of lane model, the likelihood function, Hough transform, and the chi-square fitting, etc. are applied to the lane detection. The feature-based technique localizes the lanes in the images by combining the low-level features, such as painted lines or lane edges etc. In general, the well-used algorithm of lanes detection is a combination of feature and model base is used, which is valid for all kind of roads (whether they are marked with white lanes or not).

The overall method consists of several major steps. Firstly, a binary image of the current view is created with extraction of the appropriate upper and lower range to determine the color range for the road. Then, it needs to define the region of interest of the camera image. This removes the sky area from the visual field and saves unnecessary computations. Then convert complex region of interest into simple shape, which is an important step in determining the boundary of the road. In the region of interest, the contours are determined, which contains the central part of the bottom region is the road and approximate the contour into a simpler polygonal curve. This curve (basically a contour) represents the potential road.

Using the Hough transform method to get the candidate straight line of the lanes. Several Hough lines are obtained along the left and right edges of the road. According to the angle of the left and right lanes and the lane line width, select the properly lanes.

Line which is tilted towards left and lies entirely in the left half the image is discarded. Similar lines are also discarded from the right half. Similarly, the left edge and the right edge of the road are processed. The edges of the road cannot change the angle in discrete fashion. Any line which is far away from the line in previous frame is simply discarded. Finding the left and right edges in these different sections can be used to find the possible edges of the road as is depicted by two dotted lines connecting the bottom of the ROI to the top.

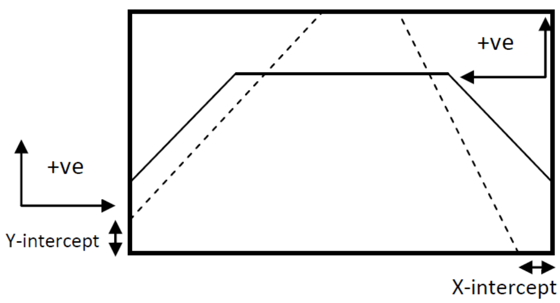


Figure 8. Lane detection from a camera image

B. Using Neural Network to Drive RC-Car

1) RC-Car Overview

The RC-Car is a very simple car, based on a Raspberry Pi, a camera, and a servo shield board to interface with the RC-Car, shown as Figure 6. Driving the vehicle around a lined track to capture images and vehicle headings (heading left, forward and right), which trains a neural-network autopilot to drive itself around the track.

While collecting training data, the car itself doesn't do all that much. It basically takes pictures and sends them to a GPU supported PC server and gets servo commands in return. The server is important to implement the autonomous driving functions. Firstly, it collects the images and driving information from the user manually driving the car around the track, by the default way delivered by the server. The server records data from a person driving the car, then uses those images and heading variables to train a Keras/TensorFlow neural network model in software. This happens quickly if GPUs are used. Once trained, the model can be loaded on the car and the car should be able to drive like the manual driving style.

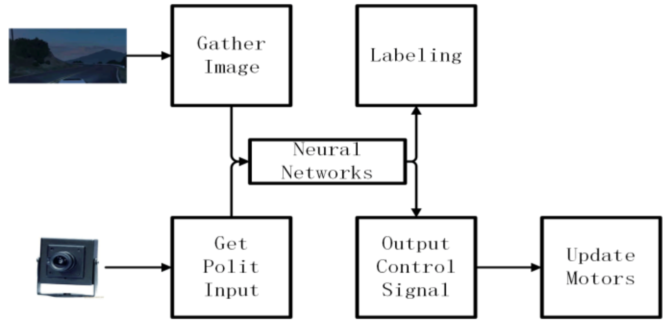


Figure 9. Run the "vehicle loop" 30 times per second.

(1) Get the driving data

The dataset is composed of about 2600 raw images, which are manually categorized under 3 folders (3 classes). After the data generator, each raw image can generate 9 deformed images, which will increase the amount of dataset to about 23000. The dataset consists of 2 numpy arrays. X are the image arrays and Y is an array of the corresponding classes.

(2) Split Data

Here we'll shuffle our data and separate the data into two parts. Training data will be used to train our driving model, Validation data is used avoid overfitting the model.

(3) Augment Training Data

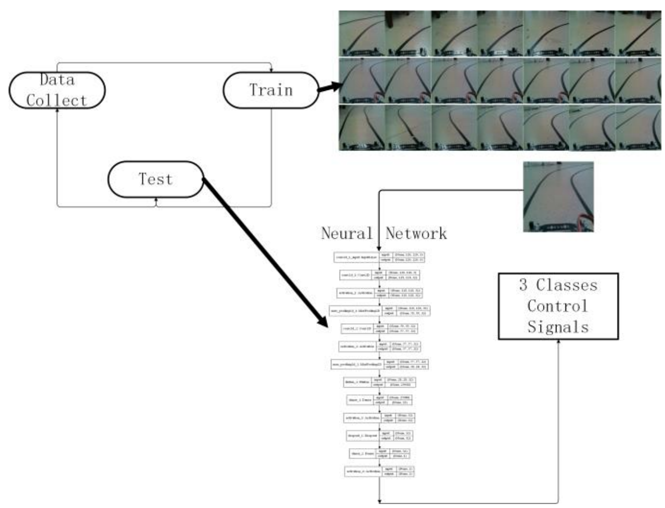


Figure 10. Train and Test the Autopilot Functions

To further double our training data and prevent steering bias, we flip each image and classes and add it to the dataset. There are additional ways to augment driving data using translations and fake shadows but not use those for this autopilot.

(4) Build a driving model

This driving model will be an end-to-end neural network that accepts image arrays as input and outputs a steering angle between -90 (left) and 90 (right). To do this we'll use a multi-layer convolution network with one fully connected layer. This model is based of Otavio's Carputer but does not produce a throttle value output, does not use past steering values as input into the model, and uses one less convolution layer.

TABLE I
NEURAL NETWORK DESCRIPTION

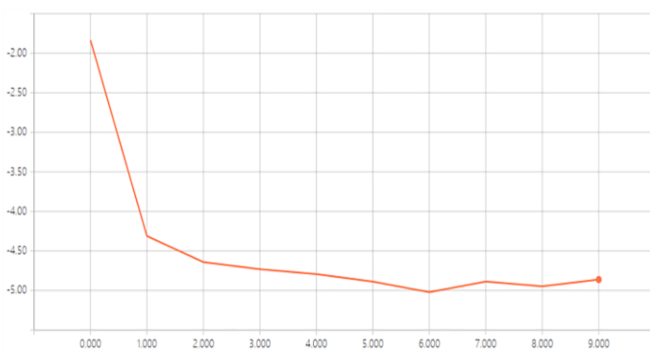
Layer (type)	Output Shape	Param #	Connected to
conv2d_1 (Conv2D)	(None, 118, 118, 32)	896	Input (shape=(120, 120, 3))
activation_1 (Activation)	(None, 118, 118, 32)	0	convolution2d_1[0][0]
max_pooling2d_1 (MaxPooling2D)	(None, 59, 59, 32)	0	activation_1[0][0]
conv2d_2 (Conv2D)	(None, 57, 57, 32)	9248	maxpooling2d_1[0][0]
activation_2 (Activation)	(None, 57, 57, 32)	0	convolution2d_2[0][0]
max_pooling2d_2 (MaxPooling2D)	(None, 28, 28, 32)	0	activation_2[0][0]
flatten_1 (Flatten)	(None, 25088)	0	maxpooling2d_2[0][0]
dense_1 (Dense)	(None, 32)	802848	flatten_1[0][0]
activation_3 (Activation)	(None, 32)	0	dense_1[0][0]
dropout_1 (Dropout)	(None, 32)	0	activation_4[0][0]
dense_2 (Dense)	(None, 1)	33	dropout_1[0][0]
activation_4(Activation)	(None, 1)	0	output[0][0]
Total params: 813,025 Trainable params: 813,025 Non-trainable params: 0			

(5) Train the model.

The autopilot won't work if you don't train it correctly. The biggest problem is over fitting the model so that it would not work in evenly slightly different scenarios.

(6) Evaluate Performance

We can check if the models predictions are reasonable by plotting the predictions vs the actual classes. The loss function and accuracy curve are shown in Figure 11.



(a) Loss function

(b) Accuracy curve

Figure 11. Evaluate Performance

V. CONCLUSION

This paper introduces a program, a new high-school iSTEM program in autonomous driving vehicles education. This program uses several sensors and Raspberry Pi embedded systems for an scaled RC-Car platform, to build an autonomous car prototype in scaled indoor simulation traffic environment. We implement some basic function of AVs, such lane detection, obstacle detection etc.

It provides several challenging driving tasks, from traditional CV algorithm to deep learning methods. The autonomous RC-Car is a good platform for ISTEM project, with hardware and software components introduction. But how to get the RC-Car autonomous driving by using neural network and implement more functions are our future works. The model will be improved when adding more driving data. A more sophisticated model would speed up on straightaways and slow down before curbs, with predict outputs both throttle and steers values.

REFERENCES

[1] W. Huang, D. Wen, J. Geng, N.-N. Zheng, "Task-Specific performance evaluation of UGVs: Case studies at the IVFC," IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 5, pp. 1969-1979, 2014.

[2] Junior: The Stanford Entry in the Urban Challenge, Michael Montemerlo, Jan Becker, .., and Sebastian Thrun. 2008. Junior: The Stanford entry in the Urban Challenge. J. Field Robot. 25, 9 (September 2008), 569-597.

[3] Li L, Huang W, Liu Y, Zheng N-N, Wang F, Intelligence testing for autonomous vehicles: A new approach [J], IEEE Transactions on Intelligent Vehicles, 2016, vol. 1, no. 2, pp. 158-166.

[4] Huang W, Wang K, Lv Y and Zhu F, Autonomous vehicles testing methods review [C], 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, 2016, pp. 163-168.

[5] F.-Y. Wang, N.-N. Zheng, D. P. Cao, C. M. Martinez, L. Li, and T. Liu, “Parallel driving in CPSS: a unified approach for transport automation and vehicle intelligence,” IEEE/CAA J. of Autom. Sinica, vol. 4, no. 4, pp. 577-587, Oct. 2017.

[6] Miquet C. New test method for reproducible real-time tests of ADAS ECUs:“Vehicle-in-the-Loop” connects real-world vehicles with the virtual world[C]//5th International Munich Chassis Symposium 2014. Springer Fachmedien Wiesbaden, 2014: 575-589.

[7] Gechter F, Dafflon B, Gruer P, et al. Towards a hybrid real/virtual simulation of autonomous vehicles for critical scenarios[C]//The Sixth International Conference on Advances in System Simulation (SIMUL 2014). 2014.

[8] https://www.raspberrypi.org/blog/self-driving-car