ATLANTIS PRESS
*Advances in Intelligent Systems Research, volume 130*
6th International Conference on Mechatronics, Computer and Education Informationization (MCEI 2016)

# Self-driving and Highly Automated Control System for Driving Simulator

Tianbo Liu [1, a *], Nan Xu [2, b], Xiyang Liu [3, c] and Chong Xu [4, d]
School of Information & Control engineering, Shenyang Jianzhu University, Shenyang, China
[a]49357122@qq.com, [b] 365723048@qq.com, [c] 835264838@qq.com, [d] 32500103@qq.com

**Keywords:** Self-driving; Automated control system; Operator GUI; Driving simulator

**Abstract.** Self-driving car can provide road safety and build interactions between driving simulator and drivers. Except for granting freedom to the hands, the main value of the project for self-driving car is to provide road safety and build interactions between driving simulator and drivers. This paper introduces a system consists of an Operator GUI which can control SCANeR Studio for self-driving simulation, make the simulated vehicle change speed, change lanes as well as switching between automatic driving and human driving. Researchers can calculate accident risks based on the differences and justify whether the driver drive under the right condition.

## Introduction

Self-driving car has been widely used in recent years. According to a survey from BI Intelligence in 2015, in 2020, the number of self-driving cars on the road will be increased to 10 million exponentially [1]. The main value of the project for self-driving car is to provide road safety and build interactions between driving stimulator and drivers. SCANeR studio software is used to simulate the procedure. The software uses a six-degree of freedom motion platform that can move and twist in three dimensions [2]. However, at the current stage, the CARRS-Q driving simulator cannot explore the interaction between self-driving or highly automated vehicles and drivers. This interaction has not been studied much, as most research resources have been focused on developing and testing the automation technology. But challenges exist with regards to drivers and occupants of self-driving cars. For example, if the self-driving cars encounter a problem that it cannot solve, it is likely to hand control back to the driver. If the driver is not focused enough or the handover is too abrupt, it may lead to a crash [3]. Thus, it is essential to build a system that could help researchers to study the value of high automation and self-driving, in order to prevent road accidents.

This project aims at helping CARRS-Q to build a system with Operator GUI that support future study of high-automated and self-driving cars. The Operator GUI can change status of the vehicle, set a maximum speed that the vehicle can reach, display the vehicle information and make the vehicle move left or right, meanwhile, generate chart figures of different vehicle parameters like speed, accelerator and the figures keep changing when the software is running in order to produce distinctive views. The functions for operating vehicle are written in the interface [4]. By comparing driving scene with the simulated ones, researchers could calculate accident risks based on the differences and justify whether the driver drive under the right condition. This research can be applied in derivative driving alerting system, and the project structure is shown in Fig1.

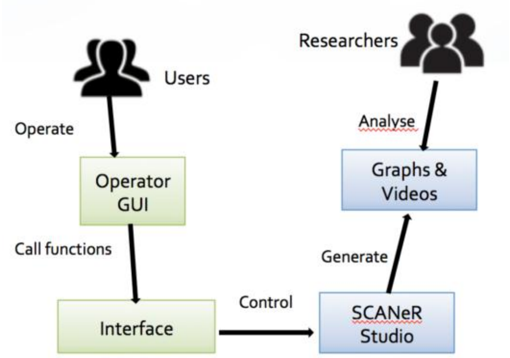


Figure 1. Project structure

### The Aim and Significance of the Search

The main purpose of driving simulator is to simulate vehicle motion and driving environment to study driver behaviors. It is designed for CARRS-Q researchers who wish to study the value of high automation and self-driving [5-7]. The most important significance for high automation and self-driving is to improve road safety. A present, the simulator cannot support this kind of automated driving, so it cannot be used to study it. There is a significant gap in research about the interactions between highly automated and self-driving cars and drivers, including on distraction, lack of awareness of the driving context, and the issue of handover. Once able to simulate self-driving car, the simulator could be used to study this problem and show that it is safe to not have your hands on the steering wheel, leading to a change of legislation. Many other studies are possible. The simulator could have interaction with drivers' cognitive responses, to detect drivers' engagement from driving speed, the distance between other cars and road surroundings such as traffic lights and sidewalk [8-9]. By comparing driving scene with the simulated ones, researchers could calculate accident risks based on the differences and justify whether the driver drive under the right condition. This research can be applied in derivative driving alerting system.

### Function Partition

The system functions can be partitioned as follows:

(1) Change Status. Switch the status between automatic driving and manual driving.

(2) Change Speed. Set a maximum speed for the car in automatic driving mode.

(3) Change Lanes. Change lanes of the vehicle to left or right.

(4) Show Graph. Display how different figures change in line charts while vehicle running.

### Core Functions Design

**Change Status.** Change status is to switch the status between automatic driving and manual driving by clicking the Change Status button. Automatic driving means the simulated car will be controlled by SCANeR Studio automatically. Manual driving means the car movements can be controlled by keyboard. At the beginning, because it is safe to start with human operation rather than automatic driving, the initial status of the car is Manual.

**Change Speed.** Change Speed function is to set a maximum speed in the textbox for the car in automatic driving mode. It indicates the maximum speed that the car can reach, rather than forces the car to the specified speed (default maximum from the scenario is 150km/h). For example, if the speed is set to a high value (e.g. 90km/h) when the car is turning, the system will calculate a proper speed that the car can have at first, then compare the speed to the given speed to ensure current speed is under that value. Thus, it is reasonable that the actual speed could be much smaller than the set speed under some circumstances. The checkbox under Change Speed button indicates whether to use the speed in textbox when switching from manual driving to automatic driving. In addition, the speed should be numeric and cannot be infinite. We set it between 1 and 200(km/h).

**Change Lanes.** Change Lanes function means to changing lanes to left or right. The car will make one movement to one direction at each pressing on Change Lanes button.

**Show Graph.** There are four Show Graph buttons at the right side of the figure list. We combine speed and acceleration, accelerator and brake, position and steering wheel angle in two figures separately. When clicking the Show Graph button, there will pop up a new window and display the parameter value in line chart, which is written in python with matlibplot library. The figure of the line chart is also keeping changing in every 0.25 seconds. X-axis is divided into 10 segments that represent 10 continuous values. Y-axis is the unit span. If there are two parameters shown in the figure, there will be a diagram form at the top right and the two line charts drawn in different colors, Line charts of speed and acceleration, steering wheel angel are shown in Fig2.

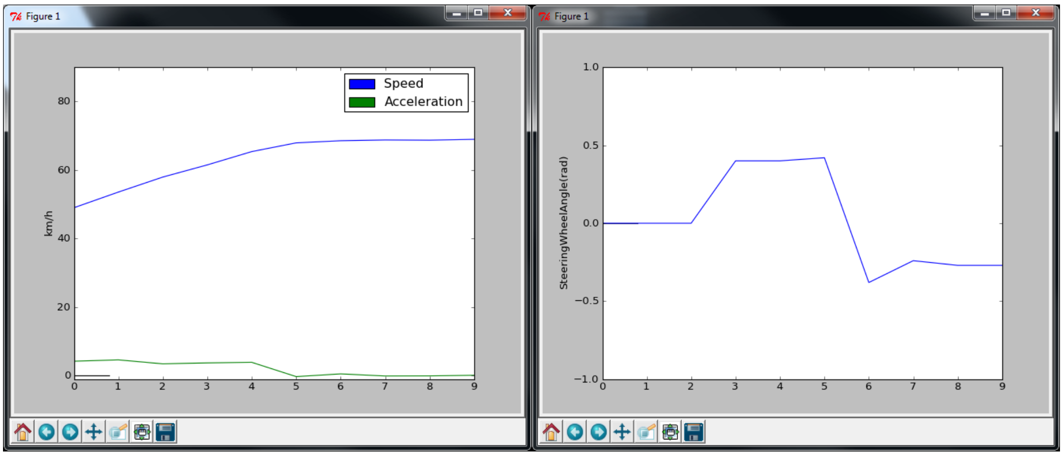


Figure 2. Line charts of speed and acceleration, steering wheel angel

## Operation Result

We use the system to test how the simulated vehicle running under the command of Operator GUI does. The initial status was Manual; we could control the vehicle by keyboard.


Figure 3. Manual driving through keyboard

Then we clicked the button Change Status from Operator GUI, the status label changed to Automatic. The figures of parameters on right side of GUI are keeping changing with the vehicle running. Manual driving through keyboard is shown in Fig. 3.

The figures below described the results of line charts. Fig. 4 showed the line chart of Steering Wheel Angle. As the vehicle was turning at the first one second, the wheel angel went below 0 at the beginning of the line chart. Then line chart value was kept in 0 rad after the turning.

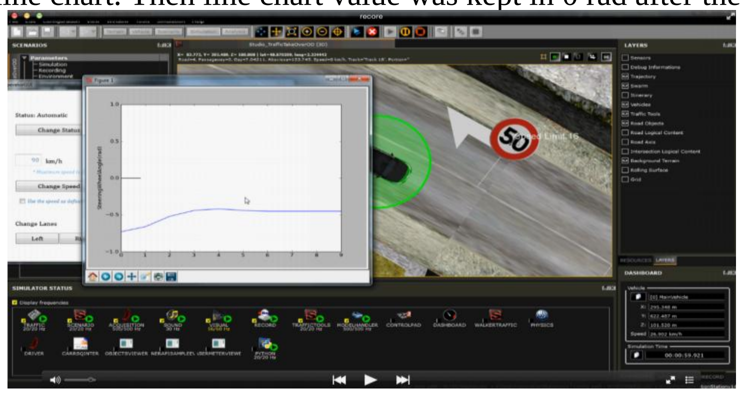


Figure 4. Steering wheel angel line chart

Fig. 5 shows the line chart of speed and acceleration. After the vehicle passing the 50km/h speed limit area, the vehicle started to accelerate. The speed increased from 22km/h to 48km/h in 5 seconds.

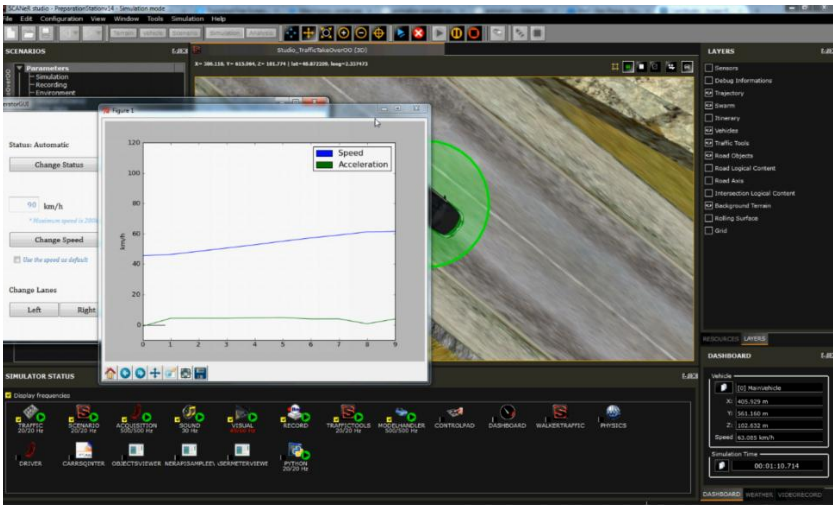


Figure 5. Speed and acceleration line chart

As the unit of y-axis is km, the change of the driving distance in 5 seconds was not quite obvious and the angle was almost 0 during a short time. However, the line chart could still show how far the vehicle had moved so far.

Fig. 6 showed the line chart of Accelerator and Brake. During the following 5 seconds, the accelerator varied greatly, it indicated the vehicle is kept increasing speed in different rates. However, at the end of Brake line chart, the brake value went up from to 100N and meanwhile, the value of accelerator went down to 0, the vehicle probably had a sudden brake.

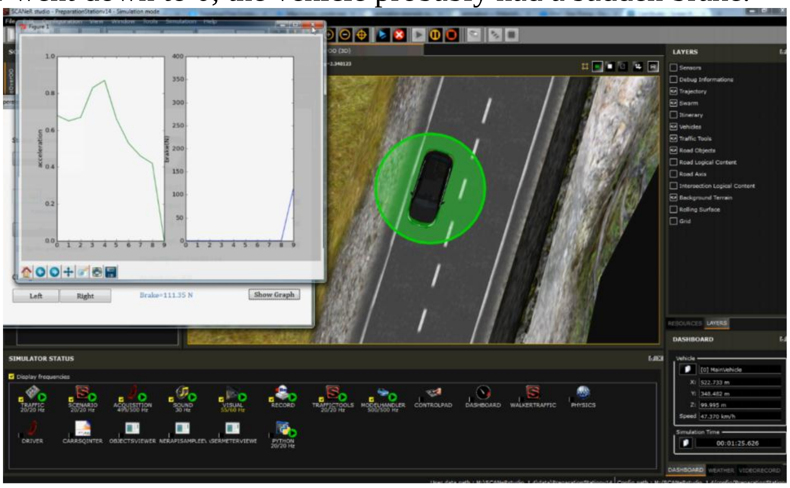

Figure 6. Accelerator and brake line chart

Then we set the maximum speed as 60km/h and click the Change Speed button after the speed was greater than 60km/h. The speed was kept increasing for 1 or 2 second and then fell to 53.07km/h with a brake.

## Conclusions

Based on the test for self-driving simulation, it can be seen that the Operator GUI provides a clear visualization of vehicle attributes in figures and line charts that keep changing with the vehicle moving. The line charts are displayed with different attributes groups that reflect different aspects of vehicle motion. For Speed and Acceleration, both of them belong to extrinsic attributes that reflect moving conditions. The self-driving vehicle could react from the commands of Operator

GUI quickly and smoothly in changing status, changing speed and changing lines. This system only covers several basic functions of vehicle movement operation. However, for a robust self-driving system, there are large amount of functions to be considered. In addition, the vehicle operation can be 100 percent manually or 100 percent automatically, but cannot be set to semi-automatic. It is more necessary to add another semi-automatic status.

## References

[1] Greenough, J. (2015). 10 million self-driving cars will be on the road by 2020. Retrieved from http://www.businessinsider.com/report-10-million-self-driving-cars-will-be-on-the-road-by-2020-2015-5-6/?r=AU&IR=T

[2] Advanced driving simulator. (n.d.). Advanced driving simulator. Retrieved from http://www.carrsq.qut.edu.au/simulator/index.jsp

[3] Abbas D,Ali P. 2013. Single Camera Vehicles Speed Measurement [C] //Iranian Conference on Machine Vision and Image Processing. Washington, DC: IEEE Computer Society, 2013:190-193.

[4] JENHUI L, JIAN L,GUANGDA H,BIN R et al. Vehicle Speed Measurement Based on Gray Constraint Optical Flow Algorithm[J]. Optik, 2014. 125(1): 289-295.

[5] MISANS P, TERAUDS M. 2012. CW Doppler Radar Based Land Vehicle Speed Measurement Algorithm Using Zero Crossing and Least Squares Method[C]//Proceedings of the Biennial Baltic Electronics Conference. Washington, DC: IEEE Computer Society, 2012: 161-164.

[6] Pan Zhao, Jiajia Chen, Tao Mei and Huawei Liang. 2011. Dynamic Motion Planning for Autonomous Vehicle in Unknown Environments[C]. IEEE Intelligent Vehicles Symposium: 284-289.

[7] Weizhong Zhang and Tao Mei,et al. Research and Development of Automatic Driving System for Intelligent Vehicles. Advances in Intelligent Systems and Computing, 2014,v215, p675-684

[8] ZELENKOV A V,LIEPKALNS D, YERSHOV A, et al. Development and Experimental Validation of Automatic Radar Recorder of the Number and Speed of Vehicles for Opposite Traffic Directions[J]. Automatic Control and Computer Sciences, 2012, 46(1): 25-33.

[9] Zhang Weizhong and Chen Gang,et al. 2014.Research of Vehicle Trajectory Tracking Based onHuman-Simulated Intelligent Control[C]//Machinery5Materials Science and Engineering Applications 2014:375-379.