Research on CarMaker and Cloud Computing Platform Co-Simulation Based on APO

Nver Ren, Rong Jiang *, Dongze Zhang
China Automotive Technology and Research Center Co., Ltd., Tianjin, China

Abstract. An cloud computing platform based on B/S architecture and docker container technology for autonomous driving simulation has been established in this paper. The map editor module of the cloud platform lets users design 3D scenes for simulating and testing automated driving systems. When the customized roadway scene for simulation created, it would be saved as OpenDrive format both for the server of cloud platform and CarMaker’s TestRun which all parameters of the virtual environment (vehicle, road, tires, etc.) are sufficiently defined. Then, based on the application online (APO) communication protocol of CarMaker, the local APO agent service was created. When the 27 parameters of vehicle dynamics received from CarMaker server, they were sent to the cloud platform in real time through UPD protocol. The process of data communication is completed by APO agent. Through the work above, a co-simulation between cloud platform and CarMaker could be successfully established for autonomous driving with seventeen-degree-of-freedom. Through the co-simulation experiment, it is found that the real-time data sampling frequency of the co-simulation is 70Hz, which completes the synchronous simulation of carmaker and cloud platform.

1. Introduction
Carmaker is a vehicle model simulation software based on C/S architecture. At present, there are some co-simulation experiments with CarMaker, such as CarMaker and Simulink were used to realize the modeling and joint simulation for automobile EPS system[1]. A co-simulation between PreScan and CarMaker has been established. The possibility of prescribing the PreScan vehicle motion (wheels excluded) was considered as a verification of the claimed S-function capabilities and, therefore, of a Prescan/CarMaker co-simulation feasibility[2]. It is not difficult to see that most of the co-simulation with CarMaker is based on C/S architecture, which is usually realized by installing multiple clients locally at the same time, and is limited to the version of the software. When we consider using C/S architecture for co-simulation experiments, there are some questions, for example, relatively high cost for development and maintenance, bad distribution, high resource cost when multi software running at the same time, low simulation efficiency and so on. Comparatively speaking, the autonomous driving simulation software based on cloud computing platform can support multi-user parallel simulation[3], besides, it has the following good properties: good distribution and sharing, not restricted by operating system, cross-platform, convenient operation and maintenance and so on. Using the advantages of Docker containers and mirror[4], It proves that the solution proposed in this thesis balance the use of resources of the nodes in cluster and improve the efficiency and stability of the co-simulation[5] experiment for autonomous driving.
With the rapid iteration of autonomous driving technology, the need to test autonomous vehicles has become more urgent. Virtual simulation testing is one of the basic testing technologies which constitute autonomous vehicle testing. In recent years, it has received extensive attention from automotive industry. So, it helps to cover all kinds of test conditions required by combining virtual simulation testing with real-world test[6].Although many companies and scientific research institutions at home and abroad have carried out the research on autonomous driving simulation platform, they are mutually independent and with their own advantages and disadvantages. Traditional simulation software such as CarSim and carmaker are mature in the field of vehicle dynamics simulation. The automatic driving algorithm in the aspect of traffic rules could be evaluated when simulating with Apoll platform. The main advantage of carcraft simulation platform is the verification of automatic driving algorithm. Panosim is mainly used in the development, testing and verification of automatic driving technology. The simulation platform launched by cognata is dedicated to providing highly realistic virtual driving scenarios[7]. In this thesis, a cloud computing platform has been built based on Docker. It is possible to support load scheduling, migration of containers, remote login and continuous delivery. In addition, we can customize roadway scenes by creating region-specific road signs and markings for co-simulation. Finally, based on the APO communication protocol of CarMaker, the local APO agent service was created. When the 27 parameters of vehicle dynamics received from CarMaker server, they were sent to the cloud platform in real time through UPD protocol.

2. Seventeen-degree-of-freedom for Vehicle[8]
To simulate the vehicle motion state under different working conditions, a comprehensive vehicle model must be established. The seventeen-degree-of-freedom vehicle model used in this paper was to consider the steering wheel angle, driving torque and braking torque input. Seventeen degrees of freedom include three axial motion degrees (x, y and z) and three axial rotational degrees of freedom (P, Q and R) of the body mass; The figure 1 shows the seventeen-degree-of-freedom vehicle model diagram, including four vertical motion degrees of mass under the front and rear springs, four spin degrees of freedom of four wheels, steering freedom of two front wheels around the main pin and the freedom of movement of tie rod.

![Figure 1. The seventeen-degree-of-freedom vehicle model diagram](image1)

![Figure 2. CarMaker coordinate systems](image2)

In the virtual world of CarMaker different axis systems for different purposes are used. They are used to simplify calculation and parametrization for CarMaker objects (including signals and variables) and to be able to represent different points of views for CarMaker objects.

In order to describe the motion state of vehicle, body and the force analysis of tire, six reference coordinate systems are established (see Fig.2). The CarMaker inertial axis system is called Fr0 (pronounced: frame zero). This is the earth fixed origin of the ‘virtual world’. O is the origin, (O, X, Y) is the horizontal driving plane (road), and Z is directed upwards (mathematically: \( \overrightarrow{X \times Y} \)).
Moving objects in the virtual world are based on their own accompanied axis system which is called Fr1. This axis system is fixed to the moving object. This means that the axis system performs all movements of the attached object like translations and rotations. X-axis positive direction points in forward driving direction, Y-axis positive direction points to the left, and Z is directed upwards (mathematically: $X \times Y$).

For every wheel there is a mountpoint (Mnt) defined within the Fr1 system. This is the center of reference of a Fr2 axis system attached to this mount-point. Mount-points are pure translations (X,Y,Z) from the Fr1 axis system. They are fixed to the Fr1 system.

There are functional dependencies (suspension kinematics and compliance) how Fr2 is orientated relatively to its mount-point. There are three generalized coordinates ($q_0, q_1, q_2$) for the movement of each Fr2 axis system. Usually $q_0$ stands for compression and $q_2$ for steer influence; $q_1$ stands for compression of the opposite wheel.

3. Map Editor
In order to meet the needs of user-defined driving scene creation, the intelligent driving simulation cloud platform software includes a map editor module. The cloud platform covers the basic elements of the driving scene, such as roads, vehicles, pedestrians, red street lights, roadside buildings and vegetation. As an essential element of driving scene, road has many attributes, such as lane direction, number of lanes, road curvature, intersection, elevation and so on. But in essence, roads can be divided into two types: straight road and curve road[9-10] the specific road modeling process is shown in Figure 3.

Users can click the curve road, spiral road or Island Road button in the function menu, when clicking the left mouse button in the drawing area to generate the road reference line, then the corresponding shape of the road was rendered. After the road was successfully added, the operations of translation, rotation, elevation setting and curvature editing are completed by dragging the control points on road reference line or setting the road property settings window. The map editor also allows users to set the direction, number and width of lanes. After the road related attributes are confirmed, according to the needs of the driving scene, traffic participants such as vehicles, pedestrians, crosswalks, traffic lights, buildings, vegetation and so on are added based on the road reference line to complete the scene generation.

![Figure 3. The road modeling process](image)

The map editor module provides basic elements editing, driving scene management and driving scene simulation. The management of driving scene is mainly to manage the roads, vehicles, pedestrians, traffic lights, buildings and vegetation added by users. The simulation of driving scene is
able to complete the corresponding simulation experiment of intelligent driving in 3D virtual engine after customizing roadway scenes.

4. APO Agent Service

The APO library is a communications library used to pass messages back and forth between server applications and a client applications. The servers and clients can be running on the same host, on two or more hosts running the same OS and being connected by a network, or on two or more hosts running a different OS and being connected by a network. The APO library is a higher level library, built over TCP (Stream) and UDP (Datagram) sockets and is intended to provide the user with a simple programming interface, while still maintaining the flexibility required by most time critical applications. APO makes passing messages easy, and eliminates the need to start from scratch each time a server/client communication interface is needed, saving time and countless debugging hours in the process.

The CarMaker application can run on a real time system being connected to one or more clients. The APO library functions are used to pass messages and quantities back and forth between the clients (IPGControl, IPGMovie etc.) residing on the non-realtime host and the CarMaker application running in realtime. This shows a single CarMaker application with one or more clients. It is also possible to have multiple CarMaker applications and multiple clients running simultaneously, since APO is not limited to a single server and multiple clients.

The steps of communication using APO are as follows:

• The first APO function in any client implementation is the call to initialize the APO. Then we should make a subscription and store the quantity pointers and other quantity information that will be used later. Our array has size 27, since we have 27 quantity that represents 17 degrees of freedom of the vehicle.
• Second, query the APO brokers for a list of servers. ApocQueryServers() is first called with a two second timeout. It is important to make sure that the list of servers is used directly after obtaining it, or shortly afterwards. Otherwise the list might be out of date.
• Third, we loop through all the servers in the list. For each server in the list, the Identity field is examined and if the identity is “CoSim”, we have found the correct server. Once the correct server is identified, ApocOpenServer() is called and the server handle(“server id”) is kept in sid. Now that the server handle is known a connection can be made(see Figure 4). ApocConnect() is first called using the server handle returned in the previous loop.
• Finally, 27 vehicle dynamics parameter that were used to describe 17 degrees of freedom of the vehicle were obtained at the frequency of 70Hz per second and saved to the parameter array. Then a UDP connection to cloud platform should be created which would enable us broadcast the received data to the cloud platform. When completing the co simulation experiment, the APO agent service must be closed.

5. Construction of simulation experiment

The map editor module of the cloud platform lets users design 3D scenes for simulating and testing automated driving systems. When the customized roadway scene for simulation created, it would be saved as OpenDrive format both for the server of cloud platform and CarMaker’s TestRun which all
parameters of the virtual environment (vehicle, road, tires, etc.) are sufficiently defined. The synchronization of cloud platform and CarMaker simulation scene is realized, as shown in Figure 5.

The specific steps of co-simulation between CarMaker client and cloud platform are as follows:

- First, log into the simulation cloud platform to open the map editing page. After drawing your custom scene, we should save the scene as OpenDrive standard format.
- Second, Start the CarMaker Application and open the scenario editor, the traffic scenario created with the cloud computing platform was directly imported for CarMaker Scenario.
- Second, According to the user's simulation requirements, the initial position and trajectory of the vehicle should be set.
- Third, we should select the experimental vehicle, and edit the driving behavior of the vehicle, including the trigger time of speed, steering, braking and so on.
- Finally, we would start APO agent service, and it helped to send vehicle dynamics parameters to the cloud platform that made the vehicle in Cloud and the car in CarMaker run in synchronization.

6. Conclusion

Virtual simulation testing is one of the basic testing technologies which constitute autonomous vehicle testing. In recent years, it has received extensive attention from automotive industry. Different software has their own advantages, so it is an inevitable trend to realize the comprehensive and efficient experiment of complex driving scenarios by joint simulation. In this paper, the B/S architecture was adopted when developing the simulation cloud computing platform. Furthermore, the advantages of CarMaker application in vehicle dynamics simulation has been made full use of. All the work made it possible to realize the co-simulation of 17 degrees of freedom of vehicles in the cloud platform, which has the advantages of low delay and high reliability.

7. References

[1] Qin Chengxian, Zhang Lei and Yan Ying 2020 CarMaker/Simulink Modeling and Joint Simulation of Automotive ESP System[J]. Mechanical Science and Technology for Aerospace Engineering. vol 39, p 1263-1270.
[2] Salvatore Battaglia. Testing the PreScan-CarMaker co-simulation. Tass International Confidential, unpublished.
[3] Su Nianle 2010 Parallelization of Simulation Model Portability Specification on Multi-core Computer[D]. National University of Defense Technology(Changsha, China).
[4] Gao Yuan 2017 Research and Implementation of A Docker-based Private Cloud Host System[D]. University of Electronic Science and Technology of China(Chengdu, China).
[5] Cornetta G, Mateos FJ and Touhafi A et al 2019 Modeling and simulation of a cloud platform for sharing distributed digital fabrication resources[J]. Computers. Vol 8, p 47-65.
[6] VISHNUKUMAR H J, BUTTING B and MULLER C et al 2017 Machine learning and deep neural network—Artificial intelligence core for lab and real-world test and validation for ADAS and autonomous vehicle: AI for efficient and quality test and validation[C]. 2017 Intelligent Systems Conference(IntelliSys)(New York:IEEE), p 714-721.
[7] LIU Jingyu, MA Hui and ZHANG Xuewen et al 2021 Research and development of autonomous driving virtual simulation technology[J]. CHINA SCIENCEPAPER. Vol 16, p 571-584.
[8] CarMaker Reference Manual Version 8.0 , unpublished.
[9] Intel open source software technology center 2009 System Virtualization: principle and Implementation Institute of parallel processing, Fudan University Tsinghua University Press.
[10] DONALD J. REI FER 2005 Practical Software Reuse. Mechanical Engineering Press(China).