Towards a Hybrid Real/Virtual Simulation of Autonomous Vehicles for Critical Scenarios

Franck Gechter, Baudouin Dafflon, Pablo Gruer and Abderrafiaa Koukam

IRTES-SeT, UTBM F-90010 Belfort Cedex
Email: firstname.lastname@utbm.fr

***Abstract*—Developing control and perception algorithms for autonomous vehicle is a time-consuming activity if one performs it directly on vehicles hardware level. Moreover, some test cases are hard to reproduce. For this reason, many laboratories and companies are generally using simulation tools. The goal of these tools is to benefit from a testing environment as close as possible to reality and able to reproduce specific testing cases. The main problem with standard simulation tools is that they may not accurately represent real conditions. In order to increase the quality of the simulations, hardware is generally introduced in the loop.**
The goal of this paper is to present a ”work-in-progress” adaptation of IRTES/SeT-Lab simulation tool named VIVUS so as to be able to introduce hybrid simulation which consists in both introducing hardware in the simulation loop and/or software simulation in the hardware experimental loop.

Keywords–hybrid simulation; autonomous vehicle algorithms; sensors simulation; augmented reality

I. Introduction

Developing control and perception algorithms for autonomous vehicle time-consuming activity if one performs it directly on vehicles hardware level. This time cost is linked to hardware issues, vehicle availability, etc. Moreover, some test cases are hard to reproduce (dealing with moving obstacles for instance) or forbidden so as to preserve vehicle integrity (testing collision avoidance algorithm for instance). For this reason, many laboratories and companies are generally using simulation tools. The goal of these tools is to benefit from a testing environment as close as possible to reality in terms of perception and control and able to reproduce specific testing cases. The main problem of standard simulation tools is their distance with real conditions since they generally simplify vehicle/sensors physical models and road topology. Virtual cameras are generally reduced to a simple pinhole model without distortion simulation and vehicle models do not take into account any dynamical characteristics and/or tyre-road contact excepted in specific automotive industry tools such as Calas [1] for instance. Some other tools such as Pro-Sivic [2] are focused on the quality of the virtual sensors having introduced hydro-meteor (rain drops) and other kind of perturbations (fog, etc.) in the sensors simulations. Many laboratories simulators are also exiting focusing each on one specific aspect such as mechanical link between elements, electrical conception [3], platoon control [4] or ergonomic considerations [5]. Despite the quality of these simulations on both physical and perceptual points of view, one needs to increase the quality of the simulations by introducing hardware in the loop. This introduction allows to improve simulation quality while keeping some interesting parts of simulation tools. This hardware in the loop simulation is not really new and is widely used in car manufacturer bench but also for Unmanned Aerial Vehicles (UAV) [6] or Unmanned Underwater Vehicles (UUV) [7]. Generally, the car bench is used to test vehicle dynamical behaviour under stressed situations or for training purposes. Nevertheless, car manufacturers still widely use experiment campaigns for the tuning of vehicle components such as brake systems, driving assistance, etc.

The goal of this paper is to present a ”work-in-progress” adaptation of IRTES/SeT-Lab simulation tool named VIVUS (Virtual Intelligent Vehicle Urban Simulator) [8], [9] so as to be able to introduce hybrid simulation. Hybrid simulation consists in both introducing hardware in the simulation loop and/or software simulation in the hardware experimental loop. If the first kind of simulation is well known in literature, the second aspect is more scarcely represented. This approach allows for instance, to simulate a virtual sensors detecting a moving obstacle and sending data to a real vehicle that will behave following its embedded algorithms (cf. Figure 1).

The paper is structured as follow: Section II presents the global architecture of the VIVUS simulation tool and sketches up the available vehicle modes. Section III gives a short overview of the simulated elements. Section IV introduces the hybrid simulation principle and gives some use case for each situation. Finally, section V concludes this paper while giving some future applications of this work.

II. VIVUS Architecture

This part presents the simulator architecture. After an overview of global structure, this section will focus on specific interaction between external interfaces and VIVUS.

A. Global structure

VIVUS is a 3D based simulator, which supports a 3D render and physical simulation. These two components are ensured by a third party application called Unity3D. The choice of Unity3D was made to limit development time and to enable cross platform applications.

The architecture is based on client-server communication processes. In/out communication can be seen as an external Application Programming Interface (API) where the simulator

 ISBN: 978-1-61208-371-1

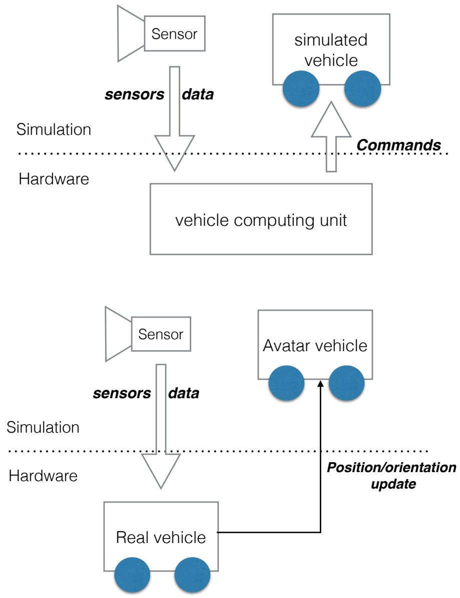

Figure 1: Hardware in the simulation loop (top) / Simulation in the hardware loop (bottom)

offers a list of services aimed at using simulated items or vehicles (cf. Figure 2).

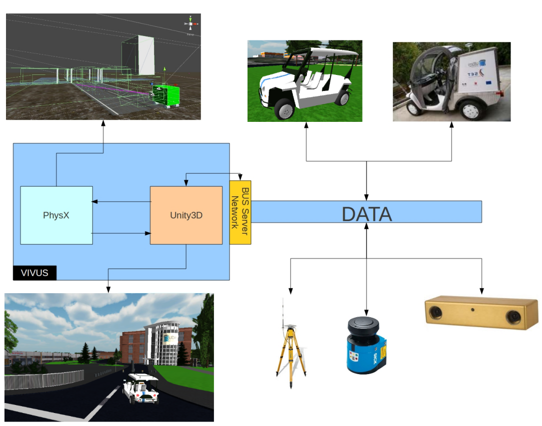

Figure 2: Vivus architecture

The simulator structure is made to allow as much flexibility and adaptability as possible. Each vehicle is considered to be autonomous in terms of perception and behaviour. The structure allows then to have heterogeneous team of vehicles each having a specific equipment and behaviour. At each simulation step, the kernel executes a list of instructions that can be summarized by: (1) *update item perception* ; (2) *update item setpoint* ; (3) *update physics*. The steps *update item perception* and *update item setpoint* are in/out accessors. These can be used to send data from the simulator to an external program or from an external program to the simulator. *Update physics* has got two behaviours depending on the vehicle mode chosen (see section II-B).

B. *Vehicle modes*

As for motion, two modes are allowed by the *update physics* function. The first is a classic motion based on Newtonian laws. The second mode, called Avatar mode, is used for only sensors simulation.

1) Classic motion: Classic motion follows Newton law of motion, based on $\sum F = m.\gamma$ where m is mass, γ is acceleration and solved by physics engine. In this mode, vehicle (or pedestrian) is controlled by an acceleration vector $(acceleration, steeringangle)$ and is limited by its physical capacity.

2) Avatar mode: Avatar mode corresponds to a scan application. Car position and orientation are set, vehicle is then teleported without taking into account physical laws, dynamics and internal parameters. However, sensors are normally simulated linked to their position on the vehicle.

III. Simulated Elements

The built-in simulation kernel of VIVUS proposes a set of customisable vehicles and sensors.

A. *Vehicles*

The physical behaviour of the vehicles can be customised following classical parameters such as maximum acceleration, maximal speed, mass, maximum steering angle, etc. In addition, some vehicles are available directly without setting them. VIVUS offers two simulated vehicles inspired by the real experimental platforms developed by SeT laboratory and which are named SeT-Car. This car model has been designed using PhysX engine requirements. The Model used is based on a composition of PhysX elementary objects (cf. Figure **??**). The SeT-Car is then considered as a rectangular chassis with four engine/wheel components. This choice can be considered to be realistic, the chassis being made as a rectangular and undeformable shape.

Figure 3: PhysX elementary objects for a SeT-Car

B. *Sensors*

In addition to vehicle models, VIVUS offers a collection of sensors usually found in experimental autonomous vehicles. These sensors are placed relatively to the vehicle without material link since the system does not take into account attachment points. Each sensor can be tuned using a 3D position, a 3D orientation, a frequency rate and its intrinsic parameters. The available sensors are :

- LMS: A Laser Measurement Sensor (Laser Range Finder) able to provide information as a set of distances retrieved from a laser scanning. Intrinsic parameters are aperture and resolution. The default provided laser range finder sends a data set of 181 distances with a resolution of 1 degree and a scanning angle of 180 degrees. (This corresponds to a Sick LMS 200)
- IBEO: Ibeo is a special laser range finder able to convert brightness values from a laser scan. A software driver has been developed to find the position of the three rear beacons laboratory vehicles.
- GPS: Our Global Positionning System receivers are providing NMEA frames. Each frame contains specific dats according to their definition. Intrinsic parameters are: data format (WGS84, Lambert2e, etc.) and origin point.
- Camera: Camera is a sensor able to send over network several video frames. Resolution, framerate and focal distance can be set.
- Odometry: Odometry is measuring the wheel movement. It computes the wheel rotation velocity in Round Per Minute.
- Proprioceptive sensor: This sensor is a customized one, able to provide all intern states of a vehicle (velocity, commands, position, etc). It also simulates an inertial measurement unit.

Each sensor sends a frame over the network.

C. Perturbations and metrics

1) Perturbation: Some perturbations are available in VIVUS. Perturbations represent events that may occur during the movement of vehicles in the environment. They deteriorate sensors and car behaviours. Perturbation can be summarized by :

- Command Perturbations: Command perturbations represents disturbances occuring on command sent to vehicles. They simulates the modifications such as communication loss, magnetic disturbance or interpretation errors, which can happen commonly during experiments. In VIVUS, a command is represented by a 2D vector holding the desired speed (in km/h) and the desired steering angle (in degrees).
- Motor Perturbation: Motor perturbation simulates the possible engine failures. Because the vehicles are electrically powered, a battery discharge is even possible. This discharging produced a disturbance into electrical engine (loss of power), which disturbs vehicle regular behaviour.
- Wheel Perturbation: Wheel perturbation affect a selected wheel of a vehicle. To affect more than one, multiple perturbation must be defined. This perturbation simulates the grip affected by the weather (rain, snow, ice), or by a tire problem.
- Sensor Perturbation: Sensor perturbation is designed to emulate alteration or loss of the vehicle perception. This perturbation is specific to each sensor.

2) Metrics: In many cases, when a simulation is done, you have to make a conclusion from what have happened. Metrics have been also defined to record some parameters during simulation time. They are useful to exploit post-simulation results.

- Inter-vehicles Metrics: Especially designed for platoon algorithm development, these metrics records different distances representing gap between two vehicles.
- Command Integrity Metric :The command metric allow to record all differences between commands sent to a given vehicle and the effective order. These metrics respects the VIVUS command representation (Velocity/Steer Angle).
- Localisation Metric: This metric record the successive positions of a selected vehicle.
- Physical Integrity Metric: Physical integrity is the closest distance before a collision. Time to collision is also evaluated using the current velocity vector.

IV. INTRODUCING HYBRID SIMULATION

As explained in the introduction, hybrid simulation consists in mixing together real (hardware) elements and simulated ones. The VIVUS architecture allows to make real and virtual entities communicate together thanks to a network communication based middle-ware. Then, one can define two categories of applications: Hardware in the simulation loop and simulation in the hardware loop. Of course, these two strategies are not mutually exclusive and can be mixed up for specific purposes.

A. Hardware in the simulation loop

Hardware in the simulation is now a wide-spread activity when one want to simulate precisely the behaviour of one specific hardware part of a system. This is used generally for testing components (effectors/actuators, central processing unit, embedded software, etc), testing sensors integration, measuring the time response of a hardware process from sensors data to actuators, ergonomy and design validation, etc. This kind of experiments is used by almost all car manufacturers including the intensive use of car benches.

As for the middle-ware part that allows virtual/real element to communicate together, RTmaps [10] has now become a standard even if some other solutions such as Effibox [11] start to be used in academic applications.

Concerning VIVUS, this has been applied to validate the model of the simulated sensors as compared to real ones. This comparison has been done dealing with sensors data output structure and sensor operation. For instance, the way the laser range finder works has been reproduced in simulation using ray tracing and introducing specular reflexion on objects depending on their texture. Then we compared the obtained result using same kind of detectable objects on both sides.

B. Simulations in the hardware loop

References on this kind of experiments are scarcely represented in literature. The goal is to be able to test real vehicle behaviour in critical cases, such as testing an obstacle

 ISBN: 978-1-61208-371-1

Figure 4: Augmented reality viewer for VIVUS

avoidance algorithm, for instance. The main interest of these kind of configurations is to be able to reproduce precisely the scenarios in terms of moving obstacles trajectories and of perturbations on sensors level (data transmission loss, false data transmissions, optical perturbation due to light exposure, etc.).

Now, let us take an example of application of the use of simulation in the hardware loop. Since a couple of years, IRTES-SeT laboratory is developing driving assistance and automatic control algorithms for autonomous vehicles. Among these, we use, for instance, simulated sensors and real vehicles to test the performances of multi-agent based obstacle avoidance algorithms [12]. To that way, we simulate sensors and moving obstacles in the virtual world. Sensors data is transmitted to real vehicle on which the obstacle avoidance algorithm is running. The real vehicle behaviour is tracked with a cm-precise Real Time Kinematic differential GPS. The real position of the vehicle is then transmitted to VIVUS so as to update correctly virtual perception. In this situation, the avatar mode of VIVUS is used in order to have a representative of the real vehicle into the virtual world. This avatar is required to be able to tackle with geometrical issues linked to real vehicle dynamics, to sensors' positions and to obstacle dynamics. In this example, the use of simulated perception is an important progress by contrast to the classical experimental protocol that are using boxes or to represent static obstacles. Moreover, it allows to perform detailed experiments in terms of combination of sensors and algorithms comparisons.

C. Virtual and real vehicles working together: an interesting side effect

One interesting side effect of the possibility to merge real/virtual vehicles and real/virtual sensors in a same simulation is to be able test multi-vehicle navigation algorithms. For instance, hybrid real/virtual vehicle platoon have been successfully tested. In the tests performed, the leader vehicle of the train is a real vehicle with real sensors. The first follower is a virtual vehicle with virtual sensors that are able to detect the avatar of the train leader. The second follower is a real vehicle with virtual sensors. Moreover, we also develop a augmented reality viewer, which can observe both real and virtual vehicle on a real playground. Figure 4 shows a view of a virtual vehicle over the real playground.

V. CONCULSION AND FUTUR WORK

The Simulation/Hardware features introduced into VIVUS enable to perform more precise autonomous vehicle algorithms evaluation due to the introduction of hardware element into the simulation loop. Even if "hardware in the loop" simulations are widely spread, the fact of using simulated, near real, elements such as sensors into the hardware loop is particularly interesting. Indeed, it allows to test real vehicle behaviours under critical condition following dangerous/forbidden scenarios. For the moment, the coupling between real vehicle and simulated elements is linked to the possibility to measure precisely the real absolute position of the vehicle in its real environment. This measurement requires, for the moment, expensive materials such as differential GPS. From now, we are focusing on this part, trying to figure out if cheaper sensors can be used to solve this issue while maintaining an acceptable precision. To this end, some experiments with an inertial sensor and accelerometers have been done. As soon as the correct precision is reached, we will perform experiments with real vehicles and software sensors such as obstacle avoidance in emergency situations, mixed real/virtual vehicles platoon control, etc We also plan to integrate this tool for tuning our autonomous vehicle algorithms. Eventually, one can say that this feature allows to integrate an intermediate step between classical simulations and experiments.

ACKNOWLEDGEMENT

This works was done with the support of the French ANR (National Research Agency) through the ANR-VTT SafePlatoon 4 project (ANR-10-VPTT-011). The authors would like to thank Jean-Michel Contet, Maxime Gueriau and Etienne François for their development.

REFERENCES

[1] http://www.thales safare.com, [last accessed October 2014].

[2] http://www.civitec.com, [last accessed October 2014].

[3] G. Tsampardoukas and A. Mouzakitis, "Deployment of full vehicle simulator for electrical control system validation," in Control (CONTROL), 2012 UKACC International Conference on, Sept 2012, pp. 551–556.

[4] K. Kim, D. il Cho, K. S. Huh, and K. Yi, "Real-time heterogeneous multi-vehicle simulator for platoon control," in SICE 2003 Annual Conference, vol. 1, Aug 2003, pp. 617–621.

[5] M. Kallmann, P. Lemoine, D. Thalmann, F. Cordier, N. Magnenat-Thalmann, C. Ruspa, and S. Quattrocolo, "Immersive vehicle simulators for prototyping, training and ergonomics," in Computer Graphics International, 2003. Proceedings, July 2003, pp. 90–95.

[6] V. Chandhrasekaran and E. Choi, "Fault tolerance system for uav using hardware in the loop simulation," in New Trends in Information Science and Service Science (NISS), 2010 4th International Conference on, May 2010, pp. 293–300.

[7] B. Davis, P. Patron, and D. Lane, "An augmented reality architecture for the creation of hardware-in-the-loop hybrid simulation test scenarios for unmanned underwater vehicles," in OCEANS 2007, Sept 2007, pp. 1–6.

[8] F. Gechter, J.-m. Contet, O. Lamotte, S. Galland, and A. Koukam, "Virtual intelligent vehicle urban simulator: Application to vehicle platoon evaluation," Simulation Modelling Practice and Theory (SIMPAT), vol. 24, 2012, pp. 103–114.

[9] O. Lamotte, S. Galland, J.-m. Contet, and F. Gechter, "Submicroscopic and physics simulation of autonomous and intelligent vehicles in virtual reality," in 2nd International Conference on Advances in System Simulation (SIMUL10). Nice, France: IEEE CPS, 2010.

[10] http://www.intempora.com, [last accessed October 2014].

[11] http://effistore.effidence.com, [last accessed October 2014].

[12] B. Dafflon, F. Gechter, P. Gruer, and A. Koukam, "Vehicle platoon and obstacle avoidance: a reactive agent approach," IET Intelligent Transport Systems, no. 3, Sep. 2013, pp. 257–264(7).