Research Statement: Coupling Vehicle Simulators

Dominik S. Buse
Distributed Embedded Systems Group, Heinz Nixdorf Institute, Paderborn University, Germany
buse@ccs-labs.org

***Abstract*—Obtaining the right data is the key to improving vehicle efficiency. Using Hardware-in-the-Loop (HiL) systems, detailed information about certain components can be gathered. Traffic simulators can run large-scale scenarios to give a view on traffic as a whole. Upcoming Inter-Vehicle Communication (IVC) technology will allow future vehicles to actively interact and cooperate—and in simulation, that technology is already available. As part of the Hy-Nets project, I want to combine all these technologies to enable a holistic view on vehicle behavior in the context of intelligent traffic scenarios. With such systems, stemming from different domains, a broad range of research question arise on how a link between them could be established.**

I. Introduction

The ever increasing number of cars on the world's roads makes improving the efficiency of vehicles one of today's great challenges. Car manufacturers go to great lengths to optimize their products. But to do so, they first have to measure the component's behavior.

To conduct such tests—especially before most of the components are ready to be integrated as a whole system—Hardware-in-the-Loop (HiL) systems are applied. These can emulate the behavior of individual parts (or even complete cars) in order to provide a mock-up to the component under test. Thus, the system under test is embedded in an controlled environment that makes it experience seemingly real test situations. Using e.g. a HiL-enabled engine test bed, the engine of a car can be analyzed in conjunction with a virtual car, experiencing real friction provided by the test bed, enabling engineers to analyze performance, power consumption, and emissions in high detail. In such a setup, the whole rest of the car can be simulated by the HiL system. Thus, a vehicle composed of real hardware and simulation models forms one logical entity. We call that vehicle the *vehicle in the loop*.

However, while current HiL systems are able to simulate the vehicle in the loop itself and physical properties of the environment, e.g., the shape and slope of the road, intelligent traffic is outside of their scope. The vehicle in the loop may detect obstacles such as other vehicles via (simulated) sensors, but not interact with them. Designed to be hardware test platforms, HiL systems do not include complete driver models to reproduce traffic behavior—especially for vehicles other than the vehicle in the loop. Thus, the actions of the vehicle in the loop can not influence the other vehicles and vice versa. A traffic simulator is needed to simulate this.

The simulation of road traffic is a another domain of its own, which relies on different tools. Traffic simulators employ a coarser level of detail to be able to simulate a large number of vehicles. Numbers range up into several thousands for city-scale traffic models.

A traditional vehicle's reactions to the behavior of neighboring vehicles can be described as passive interaction. Vehicles register their surroundings and adjust their own actions but do hardly cooperate or coordinate. Upcoming IVC technology will complement this with active interaction. Using wireless transmission, traffic participants can be informed of upcoming events and send out information themselves. This allows integration of knowledge about the near future, e.g., traffic light phases, or even negotiation of coordinated driving maneuvers such as platooning. Simulation of IVC is again carried out by specialized tools modeling transmission, protocols, and vehicle behavior.

Combining all these systems promises an unprecedented degree of flexibility and detail in vehicular simulation.

- Real hardware can be analyzed without any modeling or simulation error.
- HiL systems provide the remaining components of the car not yet available as for physical integration.
- Traffic simulators allow to embed the vehicle into full traffic scenarios and simulate the reactions of close-by cars.
- IVC enables researching future interaction of interconnected traffic participants to cooperatively improve traffic.

In summary, this would allow to analyze the vehicle both as a whole unit and as an entity within an interactive environment.

II. State of the Art

A. HiL Systems

HiL systems are designed to embed a component under test into a simulated environment. They can replace single components or whole systems (such as vehicles) by mathematical models [1]. A HiL system excels a testing closely specified test scenarios, e.g., a crash test [2]. After being considered as complete systems, they are now opening up to be allow interfaces to other simulation systems [3].

B. Traffic Simulators

Traffic simulators reproduce the movement of road users, such as cars or motorcycles. They are typically implemented as event-based systems that steps from one state into the next.

In each transition, all vehicles are advanced according to the time between the current and previous state [4]. In contrast to HiL systems, which are embedded systems that have to fulfill real-time requirements to interact with physical test beds and components under test, traffic simulators are more like traditional computer applications. They work on a best-effort basis on general purpose hardware and follow discrete, deterministic simulations models.

C. IVC Simulators

Simulation software covering both road traffic and wireless communication is already available for research. Sommer, German, and Dressler [5] propose Veins, a simulation framework coupling the SUMO traffic simulator [4] and IVC implementations built on top of the OMNeT++ network simulator.

III. Research Questions

Coupling HiL systems (including real hardware) and traffic simulation opens up a number of research questions I'd like to pursue:

- *Synchronization of time-bases:* While HiL systems operate on the order of milliseconds, traffic simulators typically have a step size in the order of tenths to a few seconds. To traffic simulators, it is typically irrelevant how much (wall clock) time passes during the computation of one simulation step. In contrast, a HiL system has to keep up with connected hardware components in real-time and missing a timing deadline or pausing the simulation is not possible.
  How can these discrepancies and differences in granularity be handled? How do network protocols have to be designed in order to do so?
- *Translation of entity representations:* Both simulation systems represent important entities such as road paths, longitudinal and lateral movements, or object coordinates in inherently different formats.
  How can they be share both on- and off-line? How to translate between them reliably?
- *Discontinuities:* Even with a way to translate entity representations, how to handle discontinuities, e.g., a vehicle entering an intersection or leaving the scenery, that are treated differently by the individual simulators?
- *Driver Behavior:* Both simulators have to implement some kind of driver model that reacts to road and traffic conditions such as speed limits, traffic lights and neighboring vehicles. These driver models may follow different strategies that lead to diverging vehicle behavior.
  How can vehicle states be kept synchronized under such circumstances?
- *Ownership of controlled entities:* Some shared entities may need a single controlling simulator that owns it at a given point in time. It may also be necessary to transfer such ownership, e.g., if a vehicle enters of leaves the horizon of the HiL. Such a transfer requires a decision on which entities to consider for a transfer in the first place. That decision will have to take multiple variables and conditions into account, such as performance reserves and real-time feasibility.
  How do such decision functions have to be designed and parameterized? How to reliably perform the transfer of control?
- *Real-Time performance of traffic simulation:* As stated before, traffic simulators are not designed to run under real-time constraints.
  How can such a simulator be managed to keep completing computation steps in real time? What to do if it falls behind?
- *Startup and Shutdown:* Systems involving real hardware, such as a car engines, can not be simply started, paused or stopped in arbitrary states.
  How can startup and shutdown procedures be coordinated among simulators and real hardware?

I plan to work on these topics during the Hy-Nets project. The project aims to improve the efficiency of electric hybrid vehicles by using vehicular communication. Its focal point is the examination of the interaction between the real hardware and software of the power train and complex traffic scenarios. To do so, we will install a hybrid engine prototype in a HiL-enabled engine test bed, which in turn will be coupled with Veins running a traffic scenario of the city of Paderborn. Our contributions are the link between the HiL and Veins (including the coordination of the simulators) and the traffic scenario.

References

[1] H. Hanselmann, "Hardware-in-the loop simulation as a standard approach for development, customization, and production test of ecu's," SAE Technical Paper, Tech. Rep., 1993.

[2] G. Hordys and A. Rolfsmeier, "Drive safely – virtual euro NCAP tests," *Automobil Elektronik*, 2014.

[3] A. Himmler, "Hardware-in-the-loop technology enabling flexible testing processes," in *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, vol. 10, 2013, pp. 6–2013.

[4] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, Dec. 2012.

[5] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, Jan. 2011.