Testing Automated Driving Systems: Framework for Graph-Based Description and Generation of Test Scenarios

Master’s Thesis

Supervisors: Prof. Dr. Alexander Pretschner, Florian Hauer
Email: {alexander.pretschner, florian.hauer}@tum.de
Phone: +49 (89) 289 - 17885
Starting date: immediately

Context

Driver assistance systems exist for over three decades now with increasing functionality and the overall goal of highly autonomous driving seems to be not out of reach anymore [1]. However, the systems are getting increasingly complex as they are not only passive, but active systems interfering with the driver. Thus, for advanced driver assistance systems (ADAS) extensive testing needs to be performed, before they can be deployed for series production [2][3].

For an autonomous highway pilot, it is estimated that approximately 6.62 billion kilometers of test driving on highways are necessary [4]. Considering this and other complexity and feasibility issues, simulation is arguably the most practical and effective way of testing software systems used for autonomous driving [5].

A lot of such simulation tools exist, e.g. CarMaker by IPG Automotive [6]. However, within these tools, test scenarios are created in a manual and very ad hoc manner. Every single test parameter (street position, pedestrian position and walking direction, timers for lights, ...) are set for every test case.

To improve this process, graph-based structures could be used. In [7], graph structures are used, which are similar to control flow graphs, to describe test cases. With that, different variants of a scenario can easily be set and maintained. From this structure the runnable test cases are generated. However, there are further dependencies between test cases, which can be used for improvement.

Goal

The main goal of this thesis is the development of a framework for test scenario generation. From a user-provided graph structure, describing variants of a scenario, runnable test scenarios are generated. This includes several partial goals:

1. Variation of scenarios happens on different abstraction levels. A whole street section could be replaced by another or the speed of a pedestrian could be slightly increased. For this, a suitable graph representation and description elements have to be developed.
2. Dependencies between variants of a scenario can be used for efficient test case generation, since not every detailed has to be generated again. A suitable backend data structure has to be developed.
3. Test cases can not directly be executed from such a graph structure. A generation mechanism has to be developed, which takes such a graph as input and returns the executable test cases.
Working Plan

1. Get familiar with CarMaker and understand how simulation based testing is done
2. Develop a suitable graph structure for description of scenario variants
3. Develop a suitable backend data structure to use dependencies among variants
4. Implement a user editor for the graph structure
5. Implement the test case generator by connecting your editor to CarMaker
6. Evaluate your framework by applying it to a lane-keeping system

Deliverables

- The framework’s source code and modules
- A demo of the framework, including instructions on how to run the demo
- Technical report with comprehensive documentation of the implementation, i.e. design
decision, architecture description, API description and usage instructions
- Final thesis report written in conformance with TUM guidelines

References

[3] Stellet, Testing of advanced driver assistance towards automated driving: A survey and
taxonomy on existing approaches and open questions, 2015
and Neural Networks, 2016
simulation-software/carmaker/, retrieved 23th February 2017
Media/Presentation/Apply_Innovate_2016_ITK_Surmund.pdf, retrieved 16th June
2017