Chapter 8

Conclusions

Driving behavior research is an exciting and very diverse field. Simulation is one available instrument for obtaining data necessary for hypotheses testing and model parameter tuning.

Reviewing simulator technology shows that a high level of sophistication has been reached in many aspects, for instance the simulation of motion, the realism of vehicle cabins and control elements or the detail level of the three-dimensional scene. Only a few universities and big automotive companies can afford simulators that comply with the state of the art, though.

It strongly depends on the specific subject of investigation whether or not a state-of-the-art driving simulator is required: When strong accelerations are involved (e.g. due to fast driving through sharp bends), realistic motion simulation is a mandatory component of the used simulator [29]. When continuous highway driving with very small acceleration forces is investigated, motion simulation is by far less relevant than in the previous case.

Scenario scripting systems exist [63] in order to exactly specify both road layout and surrounding vehicle behavior. Precise situations can be defined and rendered in a simulator run. This way, a programmed situation can be repeated as many times as needed.

When it comes to studying lane change behavior, pre-programmed scenarios can not account for dynamically evolving scenes, e.g. when regarding a lane change situation on a two lane highway with three involved vehicles: The subject vehicle on the right lane driving at intermediate speed with a slow lead vehicle, and a fast vehicle approaching from behind on the left lane. The driver of the subject vehicle needs to decide whether or not to wait until the fast vehicle has
passed before lane changing. When he decides to no wait and perform the lane change, the fast vehicle might need to react in order to avoid a collision or to reduce the risk of it. In turn, the driver of the subject vehicle will observe the behavior of that other vehicle and possibly alter his own behavior in the next lane change situation as a reaction to that.

It is possible to use autonomous vehicles in a driving simulator, i.e. to control surrounding vehicles by an algorithm that analyzes the situation and reacts accordingly. In the considered scenario, the algorithm would need to decide for the fast vehicle whether and how to brake when the subject vehicle pulls in. Thus, a model of this kind of braking behavior would be required to run an experiment on lane change behavior. One could think about creating a model for the mentioned kind of braking behavior and running experiments for calibrating it prior to the real experiment. As illustrated above, the braking behavior affects the lane change behavior, and hence a bad model or bad calibration would compromise the obtained behavioral data.

A solution to this is to have a human driver control this fast vehicle, too. With the multi-user approach it is possible to bootstrap the simulation an entire traffic scenario by giving the right instructions to human drivers. No algorithm will control a vehicle more human-like than a human driver.

For creating our multi-user simulator, some technical problems had to be solved first.

1. **Synchronicity is always relevant in parallel systems.** A master clock is generating a reference time signal. All participating instances (nodes) need to synchronize with the reference time at certain moments in order to compensate deviations. The normal way of synchronization is to overwrite the current time signal of each node with the master time signal. If the master clock is ahead of the node clock, the difference time interval will be skipped. In the contrary case, it will be passed twice. Both situations would cause problems to the physical simulation and also the data recording. Thus, a more elaborate method has been developed: The used synchronization is leap free. Rather than overwriting the node’s internal time, a drift between internal and master clock (hosted by the server) is observed over a 4 second time interval, and the internal clock speed is then tuned such that the drift is compensated and the time offset is reduced within the next observation interval.

2. **Robustness against network failure is similarly relevant.** All simulator
nodes exchange information with the central server via UDP. Each simulator node sends its state data (time stamp, vehicle position, moving direction, velocity etc.) in 40 ms intervals to the server. The server gathers data from all nodes, compiles them into one large data chunk and broadcasts that back to all nodes. The points in time when the server sends the data packages are used for synchronization, described in the previous paragraph. If the server waited for all data packages before sending, the delayed arrival of a data package of one node would delay the delivery of the compiled data packages, too, and synchronization of all clocks would be disturbed. Hence, the server uses a time-out for the arrival of data from the nodes and always delivers data in time. If the data package of one node has not arrived by then, it is replaced by an estimate based on the previous data package. The same happens on the simulator node side: When data from the server are needed, the latest available are used. Also in this case, dead reckoning is applied to extrapolate the current vehicle position from the last known one, the associated velocity and the difference in time. Actually, dead reckoning is always applied for compensating network travel time.

3. In order to capture the lane-change behavior of a subject statistically, a sufficient number of lane-change events needs to occur within reasonable time. Since the number of subjects participating in an experiment is limited (in our case to ten), fast vehicles have to encounter each slow vehicle more than once. This is achieved using a section of road where beginning and end are connected (periodic boundaries) without having to use a circuit. A circuit is not desirable because the required strong curvature would not fit with a highway situation. Instead a straight or slightly curved shape of the road is used and whenever a vehicle passes one boundary of the track, it is inserted at the other boundary. Looking straight ahead from a position close to the end, objects close to the beginning of the track can be seen as if they were copied. Actually, beginning and end are not recognized by the subjects, and neither is the leap from one to the other.

In our study on lane change behavior (described in chapter 7) we were able to show that the subject’s driving behavior in a multi-user scenario differs from the behavior observed in an otherwise comparable single-user scenario. We decided to use the above mentioned lane change task with two adjacent vehicles: a slow
lead vehicle that has to be overtaken and an approaching fast vehicle in the
target lane.

We conjectured that subjects would accept higher risk in the single user case.
Firstly, because in that case they would not endanger other human drivers (even
though the danger is only virtual). And secondly, because the behavior of the
computer controlled vehicles would be easier to predict. So, we need to quantify
risk in order to be able to compare.

More easily than risk, its inverse can be quantified: Safety. We feel safe if
there is no dangerous object within a certain range. When following another
vehicle, for instance, we maintain a certain safe distance. The actual safe dis-
tance depends on the velocity: At higher velocities, the safe distance is higher,
too. If we divide safe distance by velocity, we obtain the quantity time headway
(THW), which is velocity independent. Time headway is a relevant quantity
when following another vehicle at similar speed. Different rules apply when we
approach another vehicle that drives at a much lower speed than we do, which
is an imminent collision situation. Here, a quantity for safety is time-to-collision
(TTC), which is computed dividing distance by relative velocity. This quantity
indicates the time until one vehicle will collide with another, given both keep
moving at their current velocities. It could be shown that every human driver
has an individual TTC threshold — if the current TTC falls below that in a
driving situation, the driver brakes, see [84].

Applying this information to the hypothesis in our overtaking scenario sug-
ests that subjects accept lower TTC values in the single-user case than in the
multi-user case. Since the study was designed such that the velocities were sim-
ilar for the four types of lane changes (tab. 7.1) it was sufficient to compare
distances.

As expected, large differences between average distances in the multi-user
and the single-user set-up were found (see tab. 7.4), many of which were sig-
nificant in spite of a rather small number of samples and very large standard
deviations. Indeed we can claim that there is a quantitative difference in be-

der behavior between the two set ups. This information should be considered when
designing experiments on driver-driver interaction using a single-user simulator.

Independent of the used simulator, the quality of data obtained in any kind of
experiment depends on a proper application of the used tools. The experience so
far has shown that it is especially crucial to give subjects the right instructions
— a compromise has to be found between very strict rules on the one hand
which make experimental conditions sharp, but may cause subjects to behave
completely unnatural, and a large degree of freedom, which lets subjects drive naturally, but may not generate sufficiently many relevant situations.

The multi-user approach is a very valuable enhancement to existing simulator technology. In the future, multi-driver simulators will enable experiments that would not be possible otherwise, e.g. when predicting the effect of ADASes that use car-to-car communication. At present, the first results of studies using multi-user driving simulators are available, see [55] and also section 2.5.3. Multi-user driving simulators may also help to develop comprehensive models of driving behavior, for which interaction is a crucial component.