Impact and Modeling of Driver Behavior Due to Cooperative Assistance Systems

Florian Laquai, Markus Duschl, and Gerhard Rigoll

Institute for Human-Machine-Communication
Technische Universität München
Theresienstraße 90, 80333 Munich, Germany
{Laquai,Rigoll}@tum.de
Duschl@in.tum.de

Abstract. Current developments in Car2X communication technology provide the basis for novel driver assistance systems. To assess the impact of such a system on a group of cars approaching a non moving traffic jam, the driver behavior resulting from a system which supports anticipatory driving is modeled and used in a sub-microscopic traffic simulation. By equipping various percentages of a simulated group of cars with the model, the effect of the system on maximum deceleration and fuel consumption can be assessed. Finally the problems resulting from switching between different driver models are discussed.

Keywords: Driver modeling, Deceleration, Traffic simulation, Anticipatory driving.

1 Introduction

Current development in driver assistance systems makes increasing use of environment information. The area scanned by state of the art radar, vision and ultrasonic sensors typically covers up to 200 meters around the car. Upcoming technologies like Car2X communication can extend this range to many kilometers by using other cars as remote sensors or just by getting their positions. As a result a period of time can be covered where no assistance was possible until now.

In the presented work we will have a look at an assistance system for anticipatory driving and its impact on a group of cars with varying percentages of equipped ones. It fosters a safer and more economic way of driving by informing the driver about future traffic and regulatory conditions in which he is forced to decelerate. This was accomplished by displaying a birdeye perspective on a simplified road visualisation. Additionally the ego-car is displayed, whereas its color and size denotes the required deceleration type (no action required / engine drag torque / braking). Also foreign vehicles and a rudimentary view of the upcoming situation are included in the perspective (see fig. 1 left).

The process flow to obtain a driver model of the deceleration phase during assisted driving and its impact on a group of cars is as follows:

- Analysis of a driving simulator experiment with the proposed assistance system
- Iterative modeling and testing with a single equipped car in a traffic simulator
- Equipping various percentages of cars in a test group with the system
- Analysis of overall fuel consumption and maximum deceleration in the group

After the final experiment analysis the occurring problems are discussed and future work regarding experiment improvements and redesign of the models will be presented.

2 Driving Simulator Experiment

A driving simulator experiment for evaluation of the system was conducted in a fixed base simulator with 3 visual channels which are projected on large screens. The test person corpus consisted of 30 drivers, 18 male and 12 female with an average age of 36 years.

For simplicity the representative situation of a non moving traffic jam on a highway behind a non visible curve was chosen. The jam stands still until the group of test vehicles reaches it, then slowly starts moving again.

The implemented assistance function takes the speeds of the ego vehicle and preceding vehicle on the same lane, the distance in between and has a model of the speed dependent engine drag force.
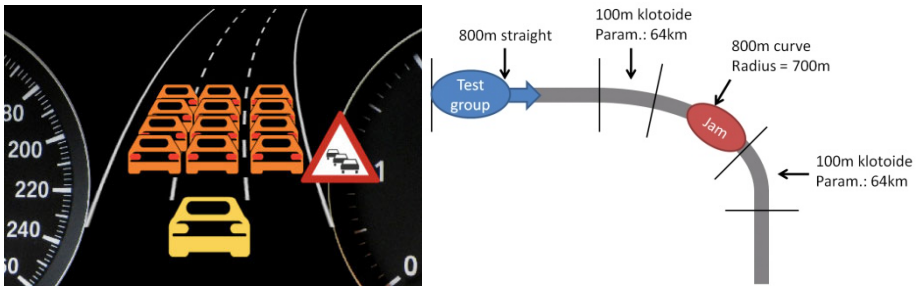

Fig. 1. Left: Birdeye Visualisation for upcoming traffic jam, Right: Test course

The assistance system logic output produced 4 possible messages:

- No action required: idle state (output "0")
- Step off gas pedal: engine drag force is sufficient (output "3")
- Comfortable braking (-0,2g): engine drag force not sufficient (output "2")
- Strong braking (-0,4g): comfortable braking not sufficient (output "1")

The logics output is then fed into the birdeye visualization, where the size and color of the ego vehicle changes accordingly. The encoding is as follows:

- Idle state: standard size, white
- Step off gas pedal: increased size, yellow
- Brake (two brake states combined): maximum size, orange

During the experiment all driver actions (pedal positions, distance keeping), dynamic properties (ego and foreign car speed and accelerations) and warning outputs are recorded for later analysis.

3 Data Analysis

In a first step the speed recordings were windowed to 35 seconds and start at the first output from the logic which was different from "0". As mentioned in [Bennet94], deceleration behavior is a function of starting speed and time. Other parameters (such as deceleration strategy advice through the ego vehicle encoding) had no significant effect on curve shapes. Consequently the data sets were allocated to speed intervals with a width of ca. 20 km/h, depending on the availability of data sets.
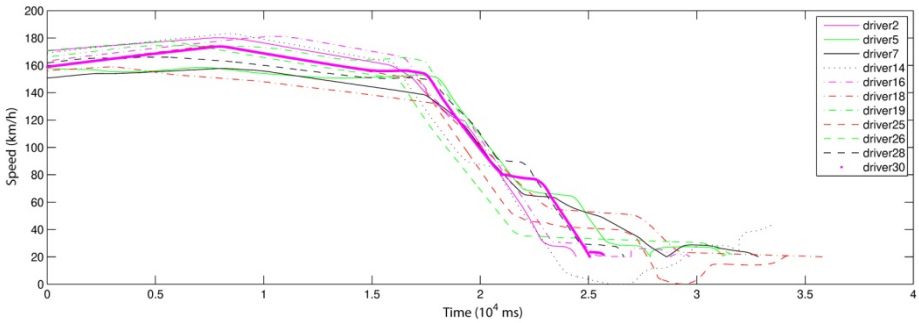

Fig. 2. Selected speed curves from driving experiment, starting at first assistance output
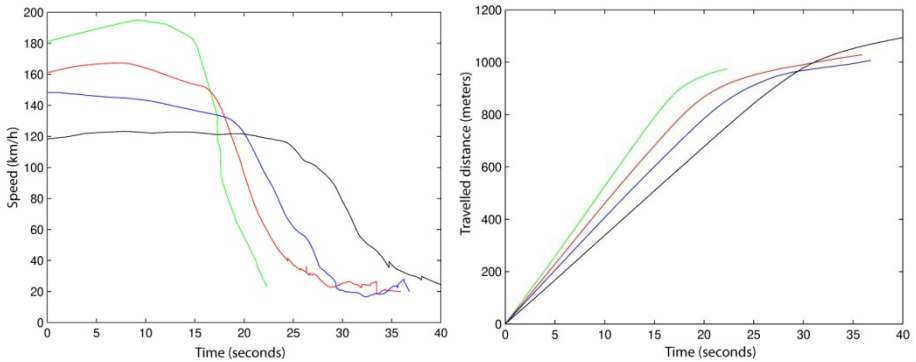

Fig. 3. Left: Averaged speed curves, Right: Integrals of speed curves (travelled distance)

Then an average curve for every speed interval is computed to eliminate individual speed oscillations. The steps during the deceleration phase which can be seen in the raw data will later be resynthesised in the model, but are eliminated in the averaging process. It can be seen that with higher starting speed, deceleration curves are steeper, the time gap between a warning output and a reaction becomes shorter and drivers

reach their target speed earlier. These effects have been already described in the literature (see [Bennet94]) and will be modeled in the behavior model described in the following sections.

It was also found that the travelled distances all approach 1000m, which was the actual distance to the jam when the system was activated. This shows that despite drivers approached the situation with different speeds, they all managed to decelerate appropriately while the steps in the curves suggest, that they reassessed the remaining distance resulting from their deceleration curve several times when it differed more than a certain threshold from the actual distance.

4   Behavior Modeling

The findings from the data analysis of the driving simulator experiment let us develop a simplified model of the deceleration phase consisting of a section with constant speed, followed by a Gaussian shaped deceleration beginning at starting speed dependent times and is delimited by the driven distance.
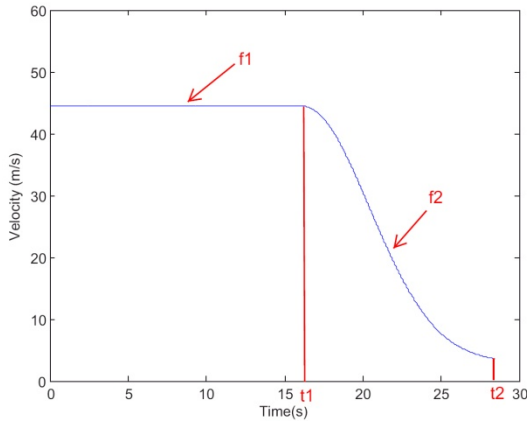

Fig. 4. Section-wise defined deceleration model

Thus the speed is defined by the following equation:

$$v(t) = \begin{cases} f_1(t), & 0 < t < t_1 \\ f_2(t), & t_1 \leq t < t_2 \end{cases}$$

$$f_1(t) = v_0 \quad with\ v_0: starting\ speed$$

The model is calibrated with a target distance of 1000m, so the proportion p is introduced for being able to apply the model for different distances d, too:

$$p = \frac{d}{1000m}$$

At first the time t_1 shall be modeled, which is fitted to the corner points of the four curves in figure 3 left:

$$t_1 = 473.13 \cdot v_0^{-0.89} \cdot p^{1.1}$$

The actual deceleration part defined by $f_2(t)$ is modeled with a Gaussian curve to create a smooth transition at the beginning and ending of the deceleration phase:

$$f_2(t) = v_0 \, exp \frac{-(t - t_1)}{2\sigma^2}$$

$$with \ \sigma^2 = \left(12649 \, v_0^{-1} - 0.016 v_0^2 + 6.467 v_0 - 523.024\right) \cdot p^{1.7}$$

As the last parameter, t_2 must be determined. This is modeled with a 4th order polynom to reach the targeted distance as closely as possible:

$$t_2 = (-0.0001192 v_0^4 + 0.01448 v_0^3 - 0.566 v_0^2 + 6.2198 v_0 + 65.92) \cdot p^{0.95}$$

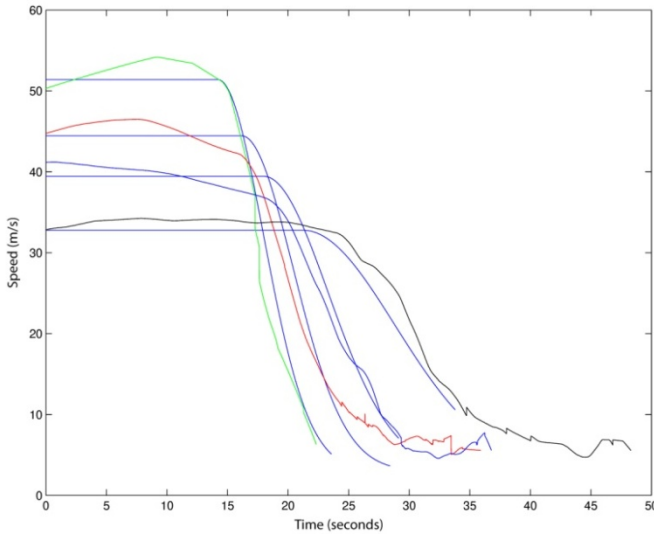The final fitting result can be evaluated in fig. 5 for various starting speeds.

Fig. 5. Real versus modeled deceleration curves

As mentioned before, the model does not resemble the original speed curves completely so far, regarding the steps in the real deceleration curves. It is concluded, that the steps have their origin in a re-estimation of the target distance by the driver due to a poor distance estimation for large distances (up to 500m).

To build an initial model of the tolerated difference between expected and real distance to the preceding car, experimental results from [Daum09] are used to create a function describing distance estimation accuracy depending on the absolute distance. In this experiment probands had to estimate absolute distances in vista space. The standard deviation between real and perceived distance in the [Daum09] experiment was initially used to create the desired function with linear interpolation between known points.
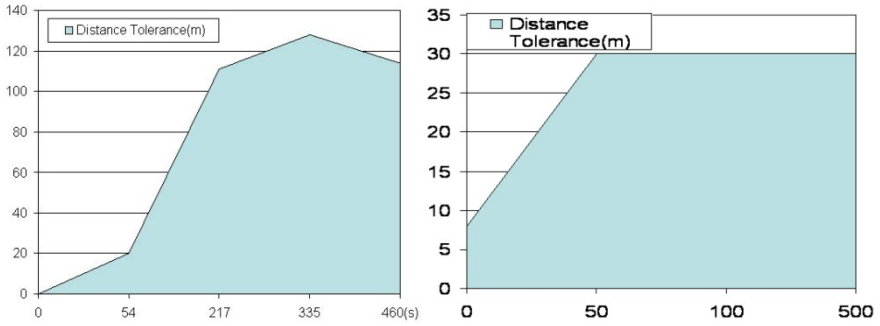
Fig. 6. Left: Tolerance curve derived from [Daum09], Right: Adapted Curve in the model

The driver model compares the expected (at the beginning of the deceleration phase) and the real distance to the preceding vehicle and rebuilds the speed curve if the tolerance is exceeded. First experiments with the left curve in fig. 6 showed that the tolerance is too large to produce the desired steps in the speed curves. An adapted version of it was then implemented to obtain the targeted results. Two adaptations were applied to the initial curve: the maximum tolerance limit is set to 30m and the minimum tolerance is 8m to prevent too frequent curve updates.
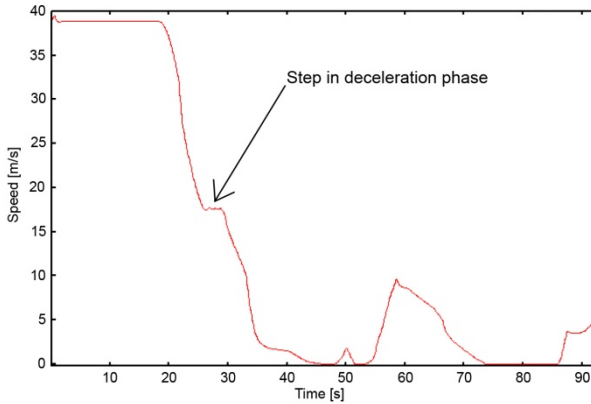

Fig. 7. Step during deceleration phase resulting from distance re-estimation

The envisaged effect of step-by-step deceleration as produced by the real drivers is shown in fig. 7.

5   Traffic Simulation

For evaluation of the model the scenario which was used for the driving simulator experiments was remodeled in the sub-microscopic traffic simulation PELOPS. As

depicted in fig. 1 on the right, the 1.8 km long test course is populated with foreign vehicles to create a stationary jam. The movement of the cars in the jam was programmed to be the same as in the driving simulation.

The tests were conducted using a Software-In-the-Loop (SIL) topology (fig. 8), meaning that the Simulink model was introduced in the processing loop of PELOPS. During design time test runs can be made quickly using the SIL topology with Socket communication, while for later testing with multiple modified driver models the Simulink model is used in a compiled binary module which is loaded by PELOPS at runtime.

The SIL interface provides detailed information about the ego-vehicle and some basic information about the surrounding foreign vehicles. Currently the only way to influence longitudinal vehicle dynamics with this interface is to forcibly set an acceleration value that can be enabled or disabled. When disabled, the normal PELOPS driver model controls the vehicle.
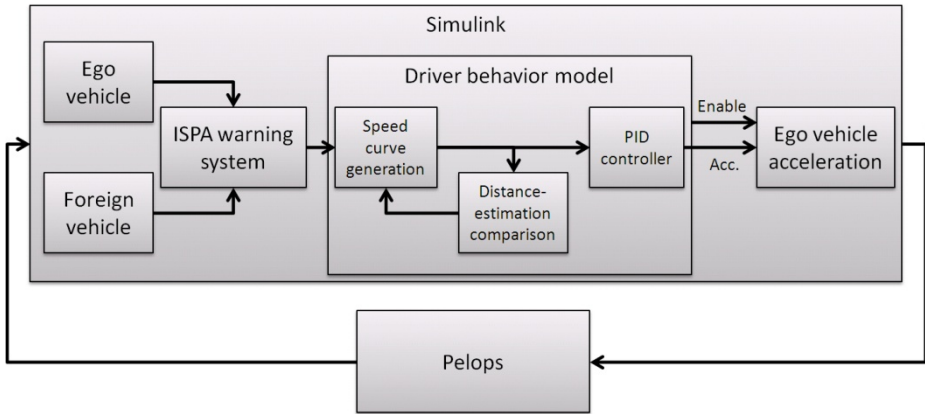

Fig. 8. Traffic simulation with Simulink model as Software-In-The-Loop model

For assessing the influences of such model on a group of 20 cars instead of just one test vehicle, the group is evenly distributed on all three lanes of the highway starting at the position where the single car was before with a speed of 140 km/h. The experiment is repeated with varying percentages of equipped cars, with the cars to be equipped selected from the head of the test group. All other cars are controlled by the standard PELOPS driver model.

6   Experiment Analysis

The goal of the assistance system is to increase safety and fuel efficiency by providing early information about future situation in which a driver is forced to decelerate. Additionally an early and smooth deceleration phase decreases the likelihood of new traffic jams and is favorable for recuperation in hybrid and electric vehicles. As there are numerous parameters to describe traffic flow, some selected ones shall be discussed here, in particular fuel consumption and maximum deceleration.

Table 1. Average maximum deceleration and average fuel consumption of test group

	Maximum deceleration (m/s²)		Fuel consumption (liters)	
Percentage	Mean	SD	Mean	SD
0%	5.6427	1.1557	0.2324	0.02162
5%	5.9134	1.1326	0.2319	0.02265
10%	6.6970	0.5212	0.2269	0.02024
20%	6.5900	1.1665	0.2292	0.02116
50%	6.5200	1.3540	0.2298	0.02445
70%	6.5200	1.3540	0.2298	0.02445
100%	6.5200	1.3540	0.2298	0.02445

For easier interpretation the data from table 1 is rendered as a bar diagram with standard deviation indicators in fig. 9.
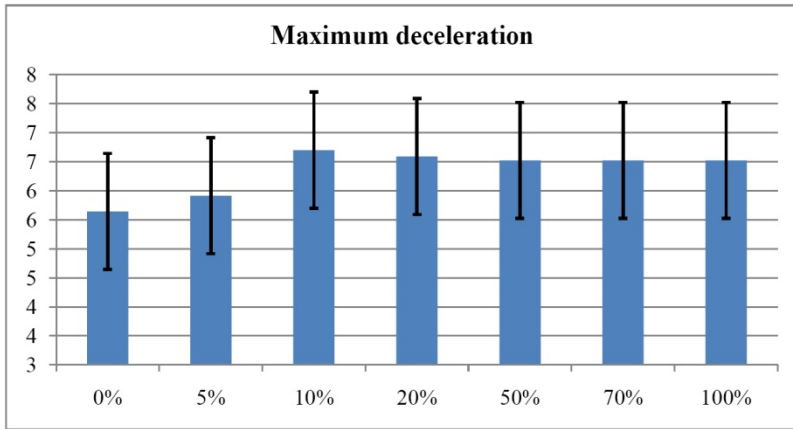

Fig. 9. Average max. deceleration [m/s²], varying percentages of equipped cars

From percentages of 10% and higher, a significantly ($\alpha = 5\%$) higher maximum deceleration can be detected. This is somewhat unexpected as the assistance system was designed to create smoother deceleration phases, while the results with our model show the opposite. Still the results from the driving simulator experiments showed the expected smoothing effect regarding maximum deceleration and also standard deviation of the acceleration.

From the view of fuel consumption, no significant ($\alpha = 5\%$) improvement could be detected. Again, the driving simulator results showed a fuel saving potential of up to 47% (for this single traffic situation and course length, see [Popiv102]). The reasons for these poor experiment results shall be discussed in the following paragraph.

7 Issues and Future Work

Both parameters discussed above (maximum deceleration, fuel consumption) do not show the expected effects as the driving simulator experiments with real drivers suggest. The main reasons for this are sudden speed changes resulting in excessive acceleration and fuel consumption. These occur when the model described above is switched off and the normal PELOPS driver model gains control again over the vehicle again. The PELOPS model keeps running parallel to the deceleration model, but its output for acceleration is overridden. As soon as the deceleration model is switched off, a large positive acceleration step is applied as the PELOPS model has a higher desired speed (see fig. 10).

   With the current implementation it is not possible to overcome this issue without substantial redesign of the Simulink controller or enhancements in the SIL interface. Regarding the SIL interface, a better parameter to control deceleration behavior might be the desired speed of the driver model instead of forcibly setting a acceleration. The other – and less elegant way – would be to implement a full car following model in Simulink also which can be parameterized at model switching events to prevent the current problems.
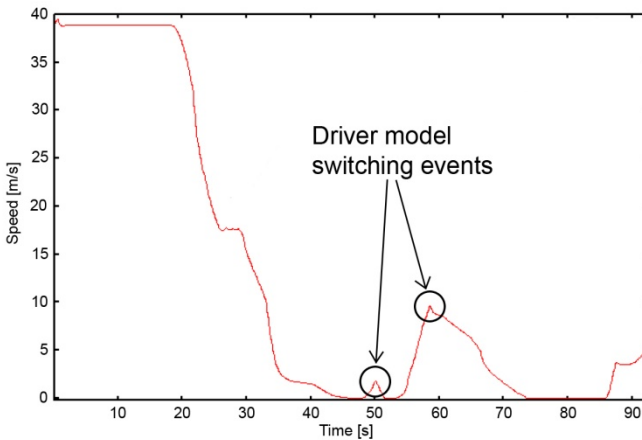

Fig. 10. Peaks in speed curve resulting from driver model switching

   A last solution would be a changeover to a different traffic simulation which exposes a more flexible and transparent interface to the user than PELOPS. This could possibly achieved with the Open Source framework SUMO (described in detail in [Kraj06]) developed by the German Aerospace Center. Investigating the proposed solutions will be part of future work on this topic as the current results are still not satisfying.

   Another point for improvement would be a switch from a distance-based re-generation of the deceleration curve to a TTC-based approach which might be a more realistic indicator (see [Kiefer06]). Also the driving simulator experiment should be conducted a second time in a different Hardware Mockup but with the same software to validate or falsify the presence of the steps in the speed curves during deceleration.

References

[Duschl10]   Markus, D., et al.: Birdeye visualisation for assisting prospective driving. In: Fisita World Automotive Congress, Budapest (2010)

[Popiv10]   Darya, P., et al.: Effects of assistance of anticipatory driving on driver's behaviour during deceleration phases. In: Humanist Conference, Berlin (2010)

[Popiv102]   Darya, P., et al.: Reduction of fuel consumption by early anticipation and assistance of deceleration phases. In: Fisita World Automotive Congress, Budapest (2010)

[Bennet94]   Bennett, C.: Modelling driver acceleration and deceleration behaviour in New Zealand. N.D. Lea International, Vancouver (1994)

[Daum09]   Daum, S.O., Hecht, H.: Distance estimation in vista space. Attention, Perception, and Psychophys. 71, 1127–1137 (2009)

[PELOPS10]   Forschungsgesellschaft Kraftfahrwesen mbH: PELOPS White Paper, Aachen (2010)

[Kiefer06]   Kiefer, et al.: Time-to-collision judgements under realistic driving conditions. Human Factors 48, 334–345 (2006)

[Kraj06]   Krajzewicz, et al.: The Open Source Traffic Simulation Package SUMO. In: RoboCup, Infrastructure Simulation Competition, RoboCup, Germany, Bremen (2006)