BEHAVIOR ON THE STABILIZATION LEVEL FOR THE COGNITIVE DRIVER MODEL

Dipl. Ing. Plavšić Marina, Guzmán Emitzá, Prof. Dr. Bengler Klaus, Prof. Dr. Bubb Heiner
Lehrstuhl für Ergonomie, Technische Universität München
Boltzmanstrasse 15, 85 747 Garching
plavsic@lfe.mw.tum.de, emitza.guzman@mytum.de, bengler@lfe.mw.tum.de, bubb@lfe.mw.tum.de

ABSTRACT: This paper deals with the modelling of driver's behaviour on the stabilization level in a dynamic, simulative environment in real-time, while taking into consideration the driver's cognitive processing for steering performance. The driver's cognitive processing on the stabilization level is analyzed and different mathematical approaches for its presentation are discussed. The driver's behavior is implemented with Time to Line Crossing (TLC) function and tested with the driving simulator software SILAB. It is shown that this approach takes into consideration the most important cognitive cues for steering and simulates the stabilization maneuvers efficiently in real-time.

1 INTRODUCTION

Up to recently, the focus of driver modelling was on the modelling of the driver control behavior while keeping the vehicle in the lane. Such models have been greatly used for the improvement of vehicle dynamics and for the development of Advanced Driver Assistance Systems (ADAS) for the stabilization level such as Anti-lock Braking System (ABS) or Electronic Stability Program (ESP). With the growing interest and technical feasibility of the driver assistances for the guidance level, the interest in modelling driver cognition has increased.

Several cognitive driver models are being developed. Some of these models are implemented within existing cognitive architectures such as ACT-R [1], Soar [2] or QN-MHP [3]. The advantage of these models is that they can use the already validated psychological knowledge of human cognitive limits and performance. However, even though highly valuable for the simulation of traditional user interfaces and cognitive tasks, these models have features which make them inappropriate for the usage in dynamic environments and for simulating multi-tasking [4].

Therefore, a lot of models have been developed taking only into account human cognition relevant for the driving task like COSMODRIVE [5], SSDRIVE [6] or ACME [7]. These models are mainly not available for public usage. Because of this, a new cognitive driver model (DriMoS) started to be developed at the Chair of Ergonomics, Technische Universität München [8]. This paper reports on the implementation of the stabilization level of driving in this model, while taking into account the relevant cognitive processing. In contrast to some other driver cognitive models, which are solely focusing on the guidance level, DriMoS tends to be an integrative model of the driver behaviour taking into consideration both guidance and the stabilization level. In the work of [4], it is, for example, shown that while performing a turn at an intersection, the significant amount of gazes and attention serve for stabilizing the vehicle.

These gazes are performed at the costs of scanning for the potential conflict vehicles and crossing pedestrians. From this reason, the cognitive behaviour on the stabilization level should not be neglected while modelling the driver’s cognition.

The objective of the work is the analysis of the driver’s cognitive processes while performing stabilization task and their implementation into the computer program. Hereby, different approaches for modelling driver's stabilization behavior regarding their advantages and 2

drawbacks are discussed and the implementation of the driver's control behavior in DriMoS is described.

2 RELATED WORK

Within this section, a brief sketch of DriMoS architecture is first presented. Afterwards, the related work with respect to models describing the stabilization task in driving is discussed.

2.1 Driver Model of the Chair of Ergonomics

The cognitive, simulative driver model DriMoS started to be developed in the thesis of [8]. The model is defined according to software-ergonomic criteria and it is applicable for real-time simulation and for coupling with the driving simulator software SILAB [9]. Furthermore, the model is defined to be extensible and modular and as such it can be developed within short-term student works.

The model consists of several stand-alone modules with clear interfaces to each other. In this way, each module of the model can be easily exchanged or adapted. Also, the communication between modules is clearly defined, simple to use and adapt. The interface of the driver model to the driving simulator is implemented in C++ and C#.

The heart of the program is the Driver, the module which considers the driver behavior on the stabilization, guidance and navigation levels. Other modules shown in Fig. 1 are responsible for the communication with SILAB. The model is configured with an XML file, which is loaded on the start. It consists of a description of all the modules and their states. Each module has its function and communicates with other modules over a configuration file. The Loader module is responsible for the correct initialization of the program. The Communication module enables the exchange of data between SILAB and DriMoS over UDP sockets and the Street Reader module possesses information about the street course. In this way the model can be implemented within an arbitrary environment.

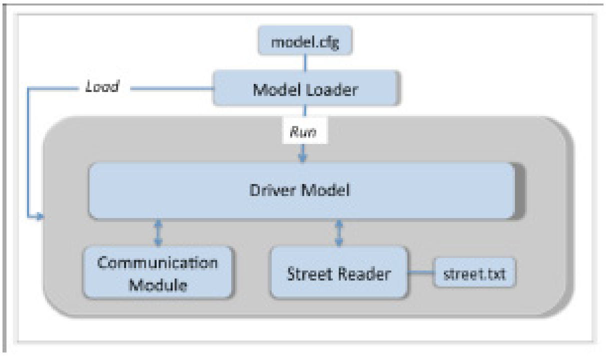

Fig.1. Sketch of the structure of the DriMoS, adopted from [8]

The development of DriMoS is planned to be a long-term, iterative process, which occupy both top-down and bottom-up approaches. Until now, the driver's behavior on the stabilization level has been implemented and the implementation of the guidance and navigation level is planned for the future. The details of how the guidance level of Driver is modeled can be found in [9] The next section discusses different models used to simulate driver steering behavior. 3

2.2 Models of control behavior

The majority of models that account for the driving task on the stabilization level describe the driver-vehicle interaction with the control theory paradigm. One of the first descriptions of a human as a controller has been given by [10]. He described human behavior as a linear system with a non-linear, remnant element. Such systems are called quasi-linear systems. Accordingly, the driving task presents the reference input (R) of the control circuit, which is constantly compared to the output of the system (Y) (compare with Fig. 1).

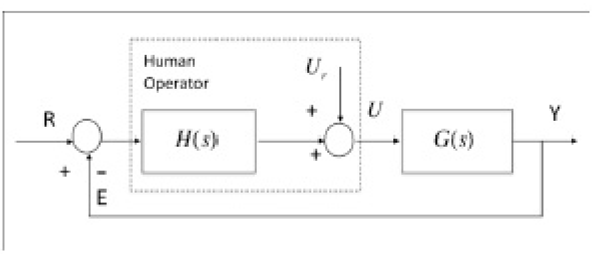

Fig.2. Control theoretic model of human operator from [10]: R=Reference input, E=Error, Y=Output variable, U=Control variable, Ur=remnant

This model has served as a base for many other models that appeared over the years. One of such groups of models is so called crossover-model group. The cross-over theory describes the driver and the vehicle together and just the total system behavior is considered. The determination of parameters for such models was a topic of plenty of scientific works for different system environments and signal types. These models have the advantage that they are efficient and low-cost, but the problem is that they do not exactly reflect what is happening in the driver's brain. As they were mostly developed to test an influence of the vehicle design on the handling quality, they ignored the driver's cognition and resulting anticipation.

While steering, drivers do not have an internal representation of complex mathematical forms but they have very simple models from desired movement to the corresponding command. The driver previews a further path and sets an ideal course and tries to drive that ideal course. In this way, the driver follows his mental model of the course by getting optical as well as kinaesthetical feedback of lateral acceleration. In order to model the way the driver steers with respect to driver’s cognition, it should be first understood which cues from the environment are used to set the ideal course and to determine its deviation and how they are processed into action. Additionally, the solution that allows the implementation of these models in computational software so that the simulation of driver behavior is efficient and possible in real-time should be found. However, the way in which the driver perceives information necessary for the steering is a disputed topic.

The models that started to take into account the driver’s anticipation are open-loop control models, which started with Fiala’s work [11]. The open-loop element deals with previewing the future desired path and it generates the major part of the driver commands. It is named like that because it does not depend on the vehicle motion. A closed-loop part is a compensatory element, which produces correction movements. The open loop from [11] depends on three elements: head-light orientation or preview from the street, the direction of the vehicle and lateral distance.

Based on these models, plenty of further models have appeared that aimed to model the driver’s behavior as exact as possible, introducing second or even third order differential equation. They differ by parameter values and input elements or by additional non-linear elements. In [12], a detailed overview and classification of these models is given. The 4 majority of them have the steering wheel angle as an output, but they differ in input parameters. In [12], even 21 different input parameters have been identified.

The existing cognitive models of driver’s behavior either do not take into account the stabilization behavior or apply an unsuitable approach. For example, the strategy implemented in one of the most popular human cognitive models, ACT-R, is the Two visual points approach [13]. This approach is based on the results from [14] and is widely applied in other models. Opposite to this approach, investigations show that the driver in conjunction uses multiple sources of information while steering rather than just a small part of the driver's visual field. In [15] it is shown that surface texture, the density of optical flow, peripheral vision and vestibular and kinaesthetic perception of velocity present the most important cues for steering. Already in [16], an explanation can be found of how novices and experienced drivers apply different strategies for lateral control. Novices control lateral distance, whereby experienced drivers control mainly the heading angle. A detailed analysis of controlling strategies relevant for the steering orientation is given in [17].

Apart from being criticized for its psychological validity, when implemented in

DriMoS, the Two visual points approach showed to be prone to errors and not precise enough. A special problem in DriMoS, while using this approach was to get vehicle back to the pavement road, once it departed from the course [8]. The similar problems are reported by [18].

3 APPROACH AND IMPLEMENTATION

From the related work, it can be concluded that the whole human body acts as an organ of perception and that the visual cues and applied strategies cannot be precisely predicted even with an extensive knowledge of the current traffic situation and an appropriate mathematical model of non-linear driver's elements. Therefore, appropriate elements should be chosen which have an empirical grounding with regard to the driver's anticipation, but the model should also be simple enough, so that the coupling with the driving simulator is possible in real time without demanding too much computational resources.

In contrast to haptic and acoustic feedbacks, which improve the control of the vehicle, the visual feedback is actually necessary for steering. In the absence of visual feedback, subjects failed to produce a sufficient steering movement resulting in a systematic deviation of the final heading [19]. This hypothesis is also confirmed by [20] who found the correlation of r=0.8 between lane keeping errors and maximal distraction time. This means that an input to the driver model relevant for steering can be sufficiently modelled with visual cues.

For an experienced driver in the rural or highway roads, the most important cues are the velocity, yaw angle, the radius of the curvature the vehicle follows and the radius of the curvature of the upcoming road segment, whereby the error of yaw angle presents the most important optical information [21]. This error is perceivable as a displacement of Focus of Expansion, detectable in a far area of the visual field. Lateral distance is observable as a disturbance of symmetry and is detectable just in close point.

These values allow the definition of the mathematical model and are determining the Time to Line Crossing (TLC) parameter. TLC is the point in time at which the vehicle will cross the edge line of the road. In the work of [22] it is also argued that experienced drivers can directly deduce TLC from the optical flow. Therefore, TLC can be seen as a parameter that is connecting the stabilization and guidance level of the driving task.

The precise equation for calculating TLC is:

$$TLC_3 = \frac{-\upsilon \tan\theta + \sqrt{(\upsilon \tan\theta)^2 + 2\upsilon^2 d_r (\frac{1}{r_k} - \frac{1}{r_s})}}{\upsilon^2 d (\frac{1}{r_k} - \frac{1}{r_s})} \qquad (1)$$

This equation incorporates information about velocity (v), the yaw angle relative to the lane centreline (θ), the radius of curvature the vehicle is following (r_k), the radius of curvature of the upcoming road segment (r_s), and the distance between the outside edge of the tire and the lane boundary (d).

An implementation of the precise TLC trigonometric formula is complex and can require a lot of computer power. But a TLC approximation that takes into account the lateral distance and its first two derivatives describes very well the run of TLC curve for the area within the lane [23]. This TLC, named TLC2 is presented by the equation:

$$TLC_2 = \frac{-\vartheta_q + \sqrt{\vartheta^2 + 2a_q d}}{a_q} \quad (2)$$

Whereby ϑ_q designates the lateral speed towards the lane boundary, the lateral acceleration and d the distance to the lane boundary. (3)

Therefore, TLC2 was implemented in DriMoS by taking into account error-neglecting times and the fact that for a larger TLC, smaller steering corrections are necessary. In [22] it was shown that drivers do not immediately react to error but takes some time, called error-neglecting time, before performing correction

Implementation in DriMoS.

Even though SILAB offers predefined values for TLC, they often give unexpected results. Therefore, a new programming unit is written for calculating TLC values. This unit accepts the following values from SILAB: lateral velocity, distance to the left lane boundary, distance to the right lane boundary and orientation of the car (see Figure 2). The returned value from this unit is used as an input to the steering wheel angle calculation formula. The biggest problem concerning the implementation of steering behavior is to find the correct steering wheel angle on the base of TLC.

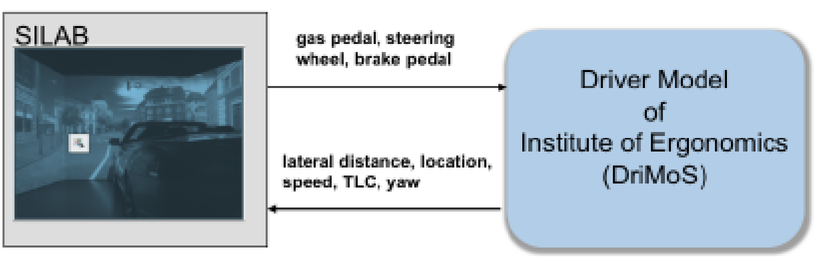

Fig. 3. The connection between the driving simulator software SILAB and DriMoS

This is accomplished by modelling the driver's reactions with a Proportional-Integral- Derivative (PID) controller, which is found to be appropriate for describing human operating behavior [24]. A PID Controller corrects the error measured between a variable and a desired setpoint by calculating the corrective action, which can adjust the process rapidly. The mathematical formula for a PID controller is given in the equation:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de}{dt}(t) \quad (4)$$

where u(t) represents the control output, t the time, e(t) the error given by the controlled variable, and a subjective integration variable. The first member in equation (4) represents a proportional part, the second an integral and the third a derivative part. , K_p, K_i,and K_d are the respective tuning parameters.

The proportional part determines the reaction to the current error, the integral part determines the reaction based on the sum of recent errors and the derivative part determines the reaction based on the rate at which the error has been changing. For determining the values of the tuning parameters the heuristic Ziegler-Nichols Method was used and tested on computers with different performance capabilities. The point of entry for our PID Controller was the inverse proportional of the previously calculated TLC. In our model we took the driver's neglection of errors with TLC values larger than 1.3s into account. PID Controllers were also used in order to regulate the initiation of pressure on the speed and brake pedals while driving.

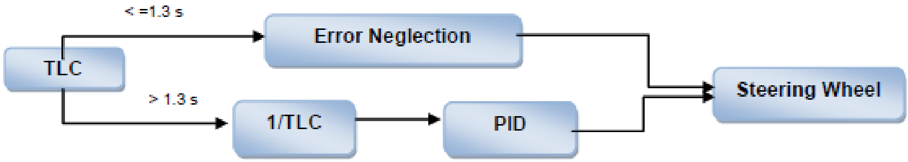

Fig. 4. Incorporation of PID Controller for Steering Wheel Control.

Turning at intersections is also modelled. In the process of turning in an intersection the steering wheel is continuously turned until the car yaw reaches a desired predetermined value. This predetermined value is associated with the position in the lane the car should have after turning. Furthermore, the longitudinal control is also modelled and coupled to the lateral control while driving through an intersection.

At the end, DriMoS is coupled with the driving simulator software SILAB and the behavior of the model is tested on different road curvatures. In order to execute the tests a road segment is built in the simulator, this segment consists of 3.680 kilometers and is a combination of different curvatures, a description of the road segment is shown in table 1. The initial conditions of our experiment are the following:

The road segment has a width of 3.66m

The car is 1.8m width

The car is initially positioned in the middle of the lane

The velocity of the vehicle remains at an approximate speed of 35 km/h.

The ranges of the curvatures in our road segment are from -500 radius to 750 radius

The vehicle ran the road segment several times and it prevailed on the lane. 7 Type	Length (m)	Radius (m)
Straight	50	
Curve	200	600
Curve	200	700
Straight	150	
Curve	400	750
Curve	400	-450
Curve	300	450
Curve	300	500
Straight	200	
Curve	180	800
Curve	100	480
Curve	300	-300
Curve	300	500
Curve	300	-500
Straight	50	
Curve	100	350
Curve	100	-350
Curve	50	300

Table 1. Description of road segment in which driving tests were executed.

4 CONCLUSION

The presented work emphasizes the importance of modelling the stabilization level within the cognitive driver model and suggests TLC2 as the compromise for both an efficient and accurate approach. The implemented model, together with the PID controller proved to be efficient for lane keeping in both straight and curved roads. Future work is composed of refining the model by implementing different heuristics for maneuver planning and stronger integration with the guidance level. In the first step, driver’s eye movements for the perception of the needed input will be modelled.

5 REFERENCES

[1] ACT-R, Theory and Architecture of Cognition, http://act-r.psy.cmu.edu/

[2] Aasman, J.: ‘Modeling driver behavior in Soar’, Dissertation, Rijksuniversiteit te Groningen, 1995

[3] Liu, Y. and Feyen, R., and Tsimhoni, O.: ‘Queueing Network-Model Human Processor (QN-MHP): A computational architecture for multitask performance in human-machine systems’, ACM Trans. Computer-Human Interaction, 13(1):37–70, 2006

[4] Plavsic, M.: ‘Analysis and Modelling of Driver Behavior for Assistance at Road Intersection’, Dissertation in process, Technische Universität München, 2010

[5] Bellet, T. and Bailly, B. and Mayenobe, P. and Georgeon, O.: ‘Cognitive Modelling and Computational Simulation of Drivers Mental Activities’ in Cacciabue P.C. (Ed.) ‘Modelling 8 Driver Behaviour in Automotive

Environments' (Springer Verlag, 2007),18, 315-344

[6] Cacciabue, P. and Re, C., and Macchi, L.:'Simple Simulation of Driver Performance for Prediction and Design Analysis' in in Cacciabue P.C. (Ed.) 'Modelling Driver Behaviour in Automotive Environments, (Springer Verlag, 2007) 19, 344–377

[7] Krajzewicz, D.: 'ACME Driver–ein kognitives Fahrermodell', In Proceedings der 5. fachüebergreifenden Konferenz Mensch und Computer, p. 105–110. Wien:Oesterreichische Computer Geselschaft, Auginer, 2005

[8] Popova, S: 'Analyse und Entwicklung eines Modells zur computergestüzten Verhaltenssimulation von Autofahrern'. Master's thesis, Fakultät für Informatik, Technische Universität München,2008.

[9] SILAB. http://www.wivw.de/ProdukteDienstleistungen/SILAB/index.php.de.

[10] Tustin A.: 'The nature of the operator's response in manual control, and its implications for controller design' J. Inst. Elec. Engrs., pages 190–207, 1947

[11] Fiala, E. :'Die Wechselwirkungen zwischen Fahrzeug und Fahrer', AutomobiltechnischeZeitschrift, 69(10):345–348, 1967.

[12] Jürgensohn, T.: 'Hybride Fahrermodelle'. PhD thesis, TU Berlin, ZMMS Spektrum, Band 4. Würzburg: Pro Universitate Verlag, 1997.

[13] Salvucci, D.: 'Modeling driver behavior in a cognitive architecture', Human Factors, 48(2):362–380, 2006.

[14] Land, M. and Horwood, J.: 'Which parts of the road guide steering?. Nature, 377(6547):339–340,1995

[15] Chatziastros, A. and Wallis, G.M. and Bülthoff, H.H.: 'The use of optical flow and splay angle in steering a central path', Vision in vehicles VIII. Amsterdam: North-Holland, 1999.

[16] Bubb, H. Systemergonomie, chapter 5.2-5.4, pages 333–420. Carl Hanser Verlag, München, 1993.

[17] Eckstein, L.: 'Entwicklung und Uberprufung eines Bedienkonzepts und von Algorithmen zum Fahren eines Kraftfahrzeugs mit aktiven Sidesticks', Fortschritt Berichte - VDI Reihe 12 Verkehrstechnik Fahrzeugtechnik, 2001.

[18] Möbus, C., Hübner, S., Garbe, H.' Driver Modelling: Two-Point- or Inverted Gaze-Beam-Steering', Prospektive Gestaltung von Mensch-Technik-Interaktion, 7. Berliner Werkstatt Mensch-Maschine-Systeme, Fortschritt-Berichte VDI-Reihe 22, Nr. 25, 483 – 488, 2007

[19] Wallis, G. and Chatziastros, A. and Bülthoff, H.: 'An unexpected role for visual feedback in vehicle steering control', Current Biology, 12(4): 295 – 299, 2002

[20] Schweigert, M.: 'Fahrerblickverhalten und Nebenaufgaben', Dissertation,

Lehrstuhl für Ergnomie, Technische Universität München, 2003

[21] Donges, E.: ‘A two-level model of driver steering behavior’, Human Factors, 20(6):691–707, 1978

[22] van der Horst A.R.A.: ‘A time-based analysis of road user behavior in normal and critical encounters’, Dissertation, TNO Institute for Perception, 1990.

[23] van Winsum, W. and Brookhuis, K.A. and DeWaard, D.: ‘A comparison of different ways to approximate time-to-line crossing (TLC) during car driving’, Accident; analysis and prevention, 32(1):47, 2000.

[24] Wolf, H.: ‘Ergonomische Untersuchung des Lenkgefühls an Personenkraftwagen’, Dissertation,Technische Universität Munchen, 2009