Investigating Human Lane Keeping through a Simulated Driver

Sebastian Noth[1], Iñaki Rañó[1], Gregor Schöner[1]

(1) Institut für Neuroinformatik, Ruhr-Universität Bochum, 44780 Bochum, Germany, E-mail : {inaki.rano, sebastian.noth, gregor.schoener}@ini.rub.de

Abstract *– Human driving behaviour modeling is an active and exciting research field with many applications for the automotive industry, ranging from safety to driving automation. Lane keeping is a well-studied problem, but formal models of human driving for this task are scarce. This paper implements a methodology to obtain naturalistic driver models through Systems Identification techniques. We obtain two different representations for individual human driver controllers based on their driving in a simulated environment. The models capture the way humans steer cars and can generate stable trajectories when controlling a simulated vehicle.*

Key words: *lane-keeping model, systems identification, driver model.*

Introduction

Lane keeping for autonomous driving is a well-studied problem, and many good solutions can be found in the literature. Moreover, lane departure warning systems (LDW) and lane keeping systems (LKS) can be already found in several commercial cars nowadays. While LKS function by using some kind of bang-bang controller, many others actually work continuously in time implementing lane keeping mechanisms. However, most of the existing lane- keeping controllers try to keep the car in the centre of the lane through standard control techniques. Therefore these mechanisms might not match the criteria used by human drivers when keeping the car in the lane and result in unnaturalistic driving from the user point of view. Having a mechanism that mimics human driving has many other advantages like: detection of abnormal driving, teaching and training of drivers, substitution of human driving in tests or experiments, among others. This paper presents a novel approach to obtain naturalistic driver controller models based on data from real drivers.

Like humans do, many of the lane-keeping systems found in the literature rely on visual information from the road, since it is relatively easy to obtain and provides adequate information to feed into a controller. The pioneering works on visually guided vehicles go back to the early 90s (see [Dic1] and references therein), and therefore, an exhaustive review of the relevant works in this area would need to be quite extense. An elegant analysis of the effect of the look-ahead distance is presented in [Kos1], where a vision-based lane keeping controller is implemented. Moreover, by analysing the root locus of the system, they prove that a larger look-ahead distance helps stabilising the motion of the vehicle. In [Cer1] another vision-based controller is presented and designed using loop-shaping, a frequency domain technique that, among others, keeps the vehicle within some specified lateral position and velocity constraints. Since their vehicle works at high velocities, they need to account for the steering actuator and parameter uncertainty of the car. In some cases visual information is used jointly with other sensors to perform lane keeping, like the work presented in [Mar1]. The authors implement a double loop controller based on vision and a gyroscope measuring the yaw rate of the vehicle. The internal loop is a PI controller of the yaw rate while the external loop provides the reference yaw rate based on the visual system and a PID controller. Their controller is robust to changes in relevant parameters of the model like, among others, the vehicle speed and mass.

A common feature among the vision-based lane keeping mechanisms is that they rely on the deviation from the centre of the lane at a single point. If we intend to mimic or model the way humans perform that task, it seems appropriate to use the same sort of information humans use. In fact, it is well known that humans use two reference points to steer a car [Lan1], and, therefore, we will focus on two point based lane keeping mechanisms. While the car is kept in the centre of the lane mainly using an angular reference to a point right in front of the car (near point), the alignment with the road is performed using a point of the road close to the line of the horizon (the far point). The first two-point model for lane keeping that can be found in the literature is presented in [Sal1]. This work compares the behaviour of the proposed model with real drivers and demonstrates that their lane keeping model captures accurately the way humans drive. However, the parameters of the model are empirically adjusted to qualitatively match the drivers' behaviour and, therefore, their estimate is just an approximation. Another work relying in a two-point controller to imitate human driving is presented in [Sen1]. There the authors use the standard car model presented in [Ack1] and propose an improved driver model derived from one of the earliest works in driver modelling [Hes1]. Even though the parameters of their model are adjusted through system identification techniques, they only use data obtained from simulated drivers implemented through optimal control techniques, and therefore their work does not represent human drivers' behaviour.

The standard way of modelling the car for a constant forward velocity is through a linear approximation of the car dynamics, and therefore several linear control models have been proposed in the literature to model human driving. A PD controller is used in [Hes1] with a single reference point, while a phase-lead and a proportional controller are used for the near and far points respectively in [Sen1]. A PI and a proportional controller are used for the far and near point respectively in [Sal1]. However, none of these works really compares the theoretical results of the control systems with known empirical findings of human driving [Lan1]. This paper contributes to the modelling of human drivers in several ways. First, we present a new perception model for the driver that allows to use a two point steering control mechanism in the same way humans do. Second, we fit data obtained from human drivers with system identification mechanisms to produce human equivalent linear controllers that can generate naturalistic driving. Our long term goal is to integrate this lane keeping system in a general driver model that includes modelling of human decision taking [Pel1].

The rest of the paper is organised as follows: The next section presents the car and driver models jointly with the setting used to gather the data needed to train the models. The experimental procedure and results are shown in the following section, while the last section accounts for the conclusions and future work.

The Car and Driver Models Integrated Within the Simulator

This section first presents the car model we used, a model that includes a new human-like output equation for the far angle. We then present the chosen potential models of human lane keeping behaviour, and finally, we describe the driving simulator used to gather human driving data.

The Car-Perception Model

Our model for the vehicle dynamics is adapted from [Sen1], though we discard the wind force as a disturbance since our focus is modelling human driving under normal conditions. The car model consists of a linear state space representation where the state of the car is described by the vector $x^T = [\beta, r, \psi_L, y_L, \delta_d, \dot{\delta}_d]$, with β representing the slip angle, r the yaw rate, ψ_L the relative yaw angle, y_L the lateral offset from the centre of the lane and δ_d and $\dot{\delta}_d$ the steering wheel angle and velocity respectively. The driver input u to the model is the torque Γ applied to the steering wheel, while, as already stated, we will consider the road curvature ρ as the only disturbance input w. Therefore, the dynamics of the vehicle is described by the equation:

$$\dot{x} = Ax + B_u u + B_w w, \qquad (1)$$

where matrices A and B_u are the same as in [Sen1], whilst the perturbance matrix B_w is in our case: $B_w^T = [0\ 0 - v\ 0\ 0\ 0]$, v being the linear velocity of the car. The numerical parameters of the simulated car are presented in table 1.

Table 1. Parameters of the car model.

C_r	C_f	l_f	l_r	m	I_z
60000	47000	0.88	1.5	867	1146

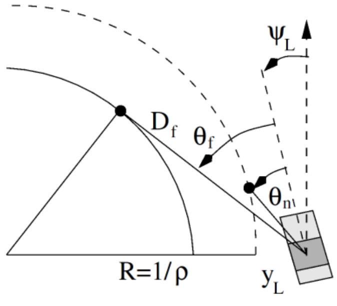

Fig. 1. Reference system of the model variables.

Since humans use near and far angles to steer a car (see Figure 1), the output equation should provide values for such variables. However, the existing works [Sen1] [Kos1] model the perception of the near (or far) point without accounting for the road curvature. While the near point angle (θ_n) can be computed from the near distance (D_n), yaw angle (ψ_L) and lateral position of the vehicle (y_L) as, $\theta_n = \frac{1}{D_n} y_L - \psi_L$, the far angle has to be computed using the road radius. The far point angle can be computed from the road radius (R), the car lateral position relative to the middle of the road and the yaw angle as:

$$\theta_f = asin\left(\frac{D_f^2 + y_L^2 + 2Ry_L}{2D_f(y_L+R)}\right) - \psi_L, \qquad (2)$$

where we neglected the width of the lane compared to the radius of the curve. Linearizing this equation around the middle position of the lane ($y_L = 0$) and a straight road, we obtain the output equation for the approximated far point angle:

$$\theta_f = \frac{1}{D_f} y_L + \frac{D_f}{2}\rho - \psi_L, \qquad (3)$$

where $\rho = 1/R$ is the curvature of the road and D_f is the distance to the far point. Therefore, matrices C and D for our output equation $y = Cx + D_w w$ can be stated as:

$$C = \begin{bmatrix} 0 & 0 & -1 & 1/D_n & 0 & 0 \\ 0 & 0 & -1 & 1/D_f & 0 & 0 \end{bmatrix}$$

and

$$D_w^T = [0 \quad D_f/2 \quad 0],$$

where the output vector is $y^T = [\theta_n \quad \theta_f]$. Even though the perception is carried out by the driver, it is convenient to consider it as an output from the car/road system, such that it can be directly fed into the controller model.

The Driver Model as a Black-Box Model

Systems identification techniques have been used as a tool to model controllers, for instance in robotics [Aka1], for complex tasks. This methodology presents a set of desirable features. On the one hand, it provides a way of obtaining controllers that perform appropriately under real world circumstances, as their parameters are estimated from real data and therefore robustness to noise is already accounted for. On the other hand, they provide a way of obtaining controllers through a process of learning by demonstration if the system is guided by a teacher while the training data is gathered.

Following a similar approach, we consider human drivers' lane keeping behaviour can be properly modelled by ARX models [Lju1] with the perceived near (θ_n) and far angles (θ_f) as an input and the torque (Γ) acting on the steering wheel as an output. This means we need to identify two ARX models simultaneously from the data, one for each input variable of the driver. Since the car model is a linearization and the driver is modelled also as a linear system, standard techniques can be applied to analyse the whole car-driver dynamical system once the controller is learned. Moreover, this approximation assumes no interaction between the near and far controllers, i.e. the global output comes from two independent contributions.

As we already explained in the introduction, some works model the driver as a state-space model of dimension five [Sen1]. This represents a grey-box model that imposes constraints to the internal control mechanism that generates human driving, constraints which may not be appropriate. In order to grasp a deeper insight of the internals of the lane keeping control mechanism, we decided to compare the ARX models with state-space black-box models that impose no restrictions to the controller form. This would allow to investigate whether there is an internal interaction between the near and the far point controller through some state-space variable.

The Driving Simulation Lab

To gather data from real drivers, we used our Driving Simulator Lab, which has seats for up to ten drivers in individual driving cabins as shown in Figure 2. The cabins are equipped with a force feedback steering wheel and a pedal console of a throttle and brake pedal. A realistic virtual environment is rendered to a 24" computer screen mounted in front of the subjects at an approximate distance of 60 cm, generating a field of view of 43°. For every cabin, the simulation software, TrafficSimulation [Not1], a commercial, fully customisable software with different modules and plug-ins, is run on a standard PC. For the present experiment, two adjacent cabins were used with simulators running in standalone-mode, such that drivers could drive in couples while talking to each other, therefore, instead of only focusing on the simulated environment, a rather automatic driving behaviour is induced.

Fig. 2. Two cabins of our driver lab.

Since the steering wheel used in our lab does not provide an accurate force feedback, the steering column generates a torque proportional to the angular displacement. Therefore, the torque the user needs to apply on the steering wheel is proportional to the angle, and the parameters on the car model, equation (1), are set accordingly to account for this drawback of our system. Apart from this issue, the simulator software runs the car model presented earlier in this section in a configurable simulation environment, which includes, among others, traffic signs, buildings or trees.

Experiments and Identification Results

This section presents the experimental identification setting and process, and it also includes the results and analysis of the obtained models.

Data Acquisition Procedure

In order to sample driving behaviour under different conditions, we created a specific road layout for the simulated environment. The test track is, after a 9 km long accommodation stretch, subdivided into 3 parts, each of which has a different speed limit (80, 100 and 120 km/h), which was indicated to the driver through traffic signs alongside the road. Every part contains sections of different curvatures, alternatingly 5 right and 5 left turns. Curves are separated by 300 m long straight stretches to let the drivers stabilise the car to the centre of the lane. Additional 500 m long straight stretches are inserted between the parts for the required speed adjustment. The five different curvatures are logarithmically equidistant steps, ranging from light curves of 8 km radius to rather sharp curves of 800 m radius. The track consists of two 3.8 m wide lanes. The ride takes altogether about 20 minutes. The DataWrite plug-in for TrafficSimulation allowed us to record driving data at a rate of 25 Hz, specifically the system state data recorded are steering wheel torque Γ, lateral position y_L, relative yaw angle ψ_L, yaw rate $\dot{\psi}_L$, track position s, and velocity v.

Prior to driving, subjects were asked to fill out a small questionnaire about their driving habits. We asked them when they received their driving license and some general personal questions. For some of the subjects the screen distance, seat height and distance to the pedals had to be adjusted. Subjects were recruited among the staff of the Institut für Neuroinformatik. Eleven subjects were accomplishing the driving task, 5 females and 6 males. Average age was 35.9 years with a minimum of 27 and a maximum of 56 years. Subjects held their licenses for 17 years on average, and reported to be driving 12120 km per year on average, thereof 5760 km on highways. The accommodation part was excluded from the recorded data in order to properly train the models.

Model Fitting

Since the TrafficSimulation software provides us with the state in time for all the trajectories, we used the already presented output equations to compute the inputs to identify the driver model. The model output was actually obtained from the steering wheel. For every subject, the data was resampled at a frequency of 5 Hz after being filtered in order to avoid aliasing problems. The two point lane keeping model requires distances of the far and the near point, D_f and D_n. We assume $D_f = 75$ m and $D_n = 5$ m. From the lateral position, the relative yaw angle and the track position, we can compute the far and the near angle, θ_f and θ_n, which are used as inputs to the driver model.

A key problem when trying to identify a system using a black-box methodology is to identify the system structure and its order. As we already stated, we decided to use a black-box state-space representation and an ARX model, but the corresponding orders needed to be selected. We tested a large range of possible dimension for the state-space model and a set of possible parameters for the ARX model. Specifically the range of state-space dimensions was from 2 to 15, while for the ARX model we tested values for the number of poles (n_a) from 2 to 10 and several values for the number of zeros (n_b) for both inputs, always taking into account that the controller model has to be causal, i.e. the number of poles has to be higher or equal to the number of zeros. Figure 3 shows the average result of the computed mean square errors for different models trained with the data set of 11 drivers, jointly with bars indicating the standard deviation of the model error from the mean. In the case of the state-space models (right figure) we can see a global decreasing trend in the error mean as we increase the number of dimensions. However, as there is a trade-off between the descriptive capability of the model and its complexity, we decided to use a state-space representation of dimension 5, which coincides with the dimension of existing models [Sen1]. Since the ARX models include three parametric dimensions (the number of poles and the number of zeros for both inputs), we only represent the case in which both transfer functions had only one zero in figure 3 (right). In the case of the average error for the ARX models, the decreasing trend is not so clear, since, in fact, it seems to stabilize for a number of poles higher than 5. Theoretically the two representations, the state-space and ARX model, should be equivalent, and therefore it is normal that the optimal dimension and the best number of poles are the same, moreover, the mean square average errors for the eleven drivers is similar in both cases.

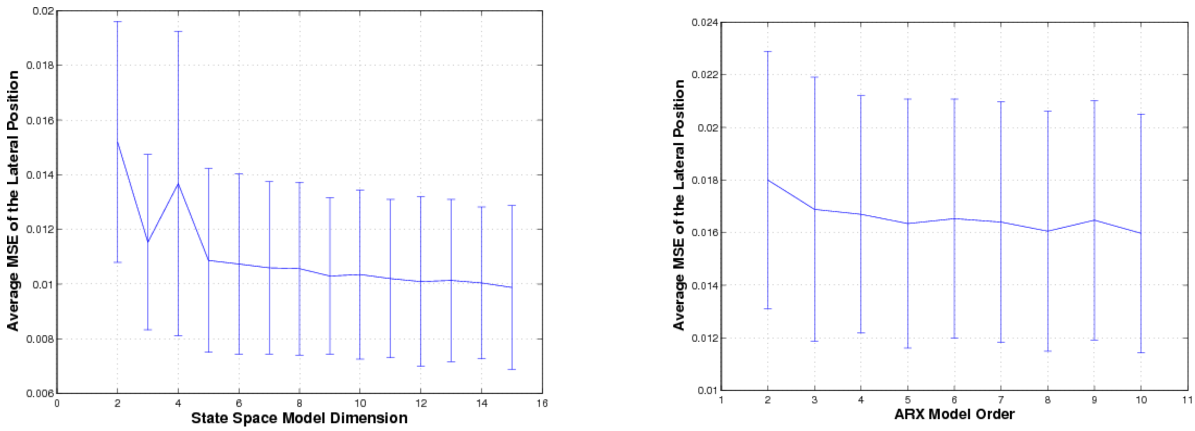

Fig. 3. Mean square errors for different state-space and ARX models.

The least-square errors were computed by simulating the obtained models fed with the corresponding near and far angles from the data recorded from the simulator and comparing the output of the model with the actual response of the driver under the same conditions. This corresponds to an open-loop simulation of the models, which is shown in Figure 4. The left part of the figure represents the inputs, near and far angle, while the right part plots the outputs of the state-space and the ARX model superposed to the actual command of the driver (the steering torque). The figure corresponds to a 160 seconds chunk of the experiments carried out by driver Dr04. As it can be seen, the range of variation for the near angle input is bigger that for the far angle. This is a joint effect of the near and far distance combined with the relatively small curvature of the road (even though the plot shows a section including a curve with small radius). On the other hand, the figures on the right show how both models capture the general shape of the steering torque generated by the user, but with a much less oscillatory behaviour. This will become more evident in the next section were closed-loop simulations are presented. In general it can be claimed that both models capture the global tendency of the driver, though they probably do not have enough descriptive capacity to account for the small details of the driving, i.e. high frequency signals.

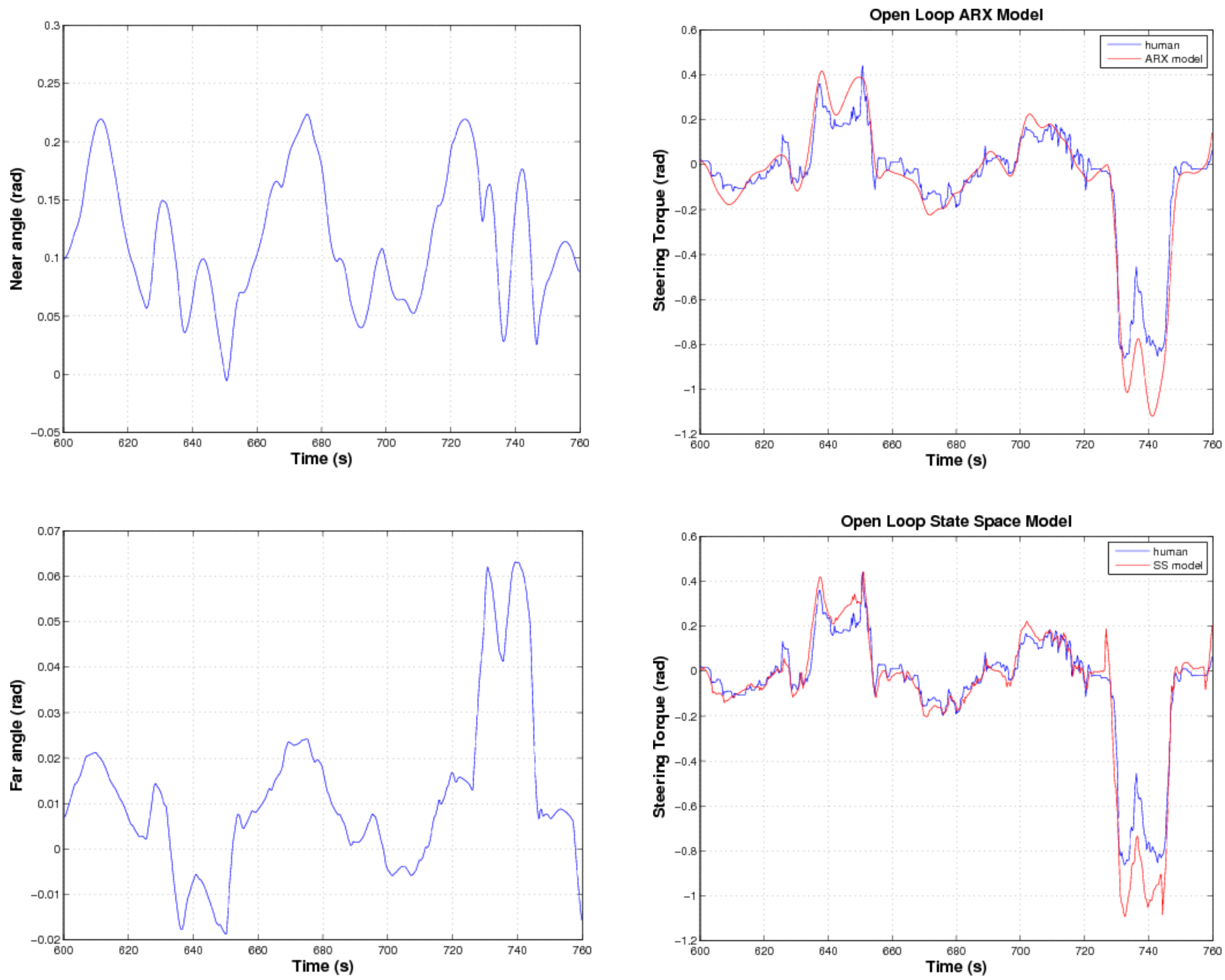

Fig. 4. Near and far angle based on recorded data, recorded steering torque output data and open-loop simulation based on the near and far angle

Closed-Loop Simulations of the Model

In order to test the real performance of the lane keeping controller models obtained, we need to test them in a simulated environment where they evolve to keep a car centered on the lane. First, we tested the stability of the whole driver model/car system beforehand, and we saw that all the learned controllers trained in the previous section were stable in closed-loop. Figure 5 shows the result of two models obtained from driver Dr03 superposed to the actual driver-car trajectories. The learned controllers were tested in an identical setting as the real drivers, actually obtained from the road specifications used on the session recorded using the driving simulator. Even though the models were simulated in Matlab, the configuration parameters of the car were identical to those in the simulated car. As it can be seen in the left figure, both controllers, the state-space and ARX models, capture the general trends of the driver Dr03, but they cannot cope with the fast variations present in the steering torque generated by the human driver. This can be interpreted as the model not having enough degrees of freedom to cope with these variations, but the most plausible reason is that the actual control mechanism used by the human drivers is non-linear. In fact, it is quite plausible that humans do not actually try to keep the car perfectly centered in the lane, but use instead some range where they do not control the steering wheel, a bang-bang controller for the near point. This could be modelled as a dead-zone in the controller models, but obviously it cannot be captured by a purely linear model. The right figure shows the actual lateral trajectory followed by the driver and the models. As it can be seen, there is a bigger difference between the real trajectory and those generated by the models. This is due to the dynamics of the car integrating and amplifying the small variations of the controllers, since what it was minimized during the training phase is actually the error on the steering toque and not on the lateral position of the car.

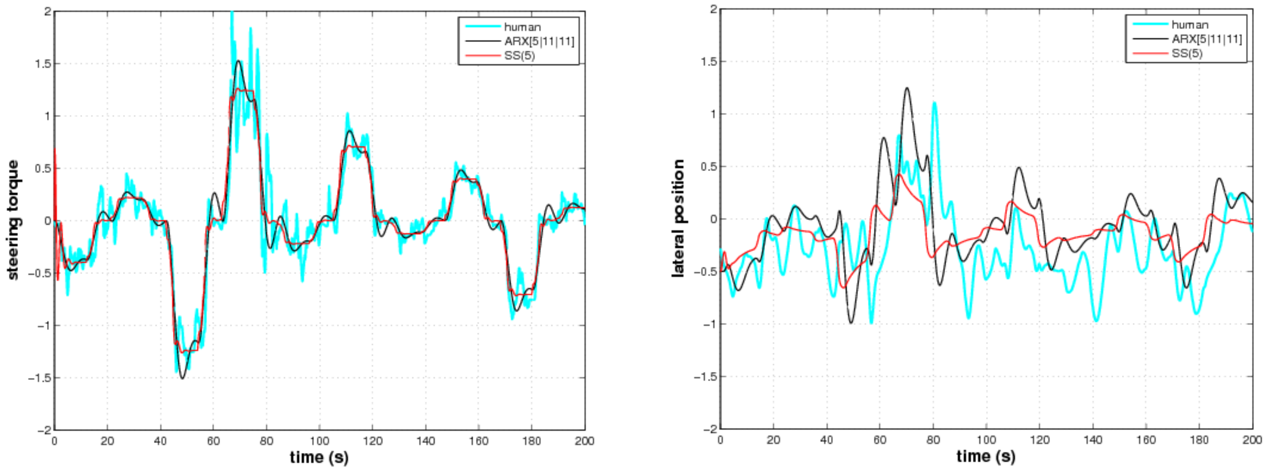

Fig. 5. Steering torque generated by the ARX and state-space model fitted with data from driver Dr03 and the resulting lateral position, compared to data from the driver.

Conclusions and Future Work

This paper presents a novel approach to obtain naturalistic driving controllers through the use of Systems Identification techniques with data recorded from human drivers. Existing works either did not fit real driving data or used heuristic methods to obtain the controllers, therefore, to the best knowledge of the authors, this represents the first time a formal identification technique is used to model human lane keeping. As already stated, all the works proposing linear models present limitations since they cannot cope with non-linear effects like dead-zones in the control variables. However, some extra simulations we performed switching off one of the models obtained, matched the known results from humans driving only using the near or the far point [Lan1]. This is an interesting topic which deserves a deeper analysis outside the scope of the present work.

Further analysis needs to be carried out using the state-space representation to test the hypothesis of total independence on the near and far controllers, since the output torque is computed from the five dimensional state-space vector which dynamically links the near and far input variables. Another important issue is that humans can adapt their driving to different cars, and, therefore, the robustness of the driver model to change on some parameters (like velocity or car model parameters) has to be assessed. Finally, we plan to integrate this mechanism as a part of a driver model that reaches the tactical level of driving.

References

[Aka1] Akanyeti O., Rañó I. "An application of Lyapunov stability analysis to improve the performance of NARMAX models". *Robotics and Autonomous Systems*, 2010, 58(3), pp. 229-238.

[Ack1] Ackermann J., Bartlett A., Kaesbauer D., Sienel W., Steinhauser R. "Robust control: Systems with uncertain physical parameters". *Springer*, 1993.

[Cer1] Cerone V., Chinu A., Regruto D. "Experimental results in vision-based lane keeping for highway vehicles". *Proceedings of the 2002 American Control Conference*, 2002, pp. 869-874.

[Dic1] Dickmans E. "Vehicles capable of dynamic vision". *Proceedings of the International Joint Conference on Artificial Intelligence*, 1997, pp. 1577-1592.

[Hes1] Hess R., Modjtahedzadeh A. "A Control Theoretic Model of Driver Steering Behavior". *IEEE Control Systems Magazine*, 1990, 10(5), pp. 3-8.

[Kos1] Kosecká J., Blasi R., Taylor C., Malik J. “Vision-based Lateral Control of Vehicles”. *Proceedings of the IEEE Intelligent Transportation Systems Conference*, 1997.

[Lan1] Land M., Horwood J. “Which parts of the road guide steering?”. *Nature*, 1995, 377, pp. 339-340.

[Mar1] Marino R., Scalzi S., Orlando G., Netto M. “A Nested PID Steering Control for Lane Keeping in Vision Based Autonomous Vehicles”. *Proceedings of the 2009 American Control Conference*, 2009, 9(2), pp. 2885-2890.

[Not1] Noth S., Edelbrunner J., Iossifidis I. “A Versatile Simulated Reality Framework: From Embedded Components to ADAS”. *International Conference on Pervasive and Embedded and Communication Systems*, 2012.

[Pel1] Pellecchia A., Igel C., Edelbrunner J., Schöner G. “Making Driver Modeling Attractive ”. *IEEE Intelligent Systems*, 2005, 20(2), pp. 8-21.

[Sal1] Salvucci D., Gray R. “A two-point visual control model of steering”. *Perception*, 2004, 33, pp. 1233-1248.

[Sen1] Sentouh C., Chevrel P., Mars F., Claveau F. “A Sensoriomotor Driver Model for Steering Control”. Proceedings of the 2009 IEEE International Conference on Systems, Man and Cybernetics, 2009, pp. 2462-2467.

[Lju1] Ljung, L. “System Identification: Theory for the User”. *Prentice Hall*, 1999.