Naturalistic Lane-Keeping based on human driver data.

I. Rañó H. Edelbrunner G. Schöner

Abstract— Autonomous lane keeping is a well studied problem with several good solutions that can be found in the literature. However, naturalistic lane keeping mechanisms, in the sense of imitating human car steering, are not so common. Based on existing knowledge of human driving, this paper analyses several controllers prone to generate human-like lane keeping behavior. Using systems identification, we compare how well the control models fit with real human steering data gathered in a simulator. Experimental results points towards a parsimonious control mechanism where angles relative to the direction of the road are directly used by the driver to steer the car and keep it on the lane. This result can be used to design human like autonomous lane keeping mechanisms or to improve the design of ADAS systems adapting them to individual drivers.

I. 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 (see for instance [2] and references therein). While LKS implement some kind of bang-bang controller, many other approaches actually work continuously in time implementing lane keeping mechanisms. However, most of the existing lane keeping techniques just try to keep the car in the center of the lane through standard controllers, producing unnaturalistic driving from the user point of view. This might make the user feel uncomfortable and untrusting of an automatic driving vehicle. Besides generating comfortable motion for humans, having a mechanism that mimics human driving has many other advantages like: allowing the detection of abnormal driving, teaching and training unskilled drivers, or substituting of human drivers in tests or experiments.

Like humans do, many of the lane keeping systems found in the literature rely mainly on visual information from the road [9], since it is relatively easy to obtain and adequate to feed into a controller. The pioneering works on visually guided vehicles go back to the early 90s (see [4] and references therein) and therefore many relevant works in this area can be found. An elegant analysis of the effect of the look-ahead distance is presented in [6], where a vision based lane keeping controller is implemented. Moreover, the authors prove, by analyzing the root locus of the system, that a longer look-ahead distance helps stabilizing the motion of the vehicle. The work in [3] presents another vision based controller designed using loop-shaping, a frequency domain technique, that keeps the vehicle within some specified lateral position and velocity constrains. Since their vehicle works at high speeds they need to account for the steering actuator and parameter uncertainty of the car. In some cases, like [8], visual information is used jointly with other sensors to perform lane keeping. 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 control system is robust to changes in relevant parameters of the plant like, the vehicle speed or mass, among others.

A common feature shared by many of the visual based lane keeping mechanisms is that they rely on the deviation from the center of the lane at a single point. However, it is well known that humans use two reference points to steer a car [7], one point close to the front of the vehicle, the near point, and one point close to the horizon, the far point. While the near point is used to keep the car centered on the lane the far point helps keeping it aligned with the road. If we intend to mimic or model the way humans perform the lane keeping task it seems appropriate to use the same sort of information humans use. Therefore, in this work we will focus on two point based lane keeping.

The first two point model for lane keeping that can be found in the literature is presented in [13], where a PI and a proportional controllers are respectively used for the far and near angles as viewed by the driver. This work compares the behavior of their proposed model with real drivers and demonstrates their lane keeping model can account for the way a human drives. However, the assumption that the driver keeps the car in the center of the lane is not realistic, therefore, their estimate is a rough approximation of the real driver behavior. Another work relying on a two point controller to imitate human driving is presented in [14], where the authors use the standard car model of [1] and improve a driver model derived from one of the earliest works in driver modelling [5]. One phase-lead and one proportional controller are used for the near and far points respectively. Even though the parameters on their model are adjusted using system identification techniques, they rely on data obtained from simulated drivers implemented with optimal control techniques, and, therefore, their results do not actually match human drivers. The work in [11] provides a comparison of ARX and State Space black box models using systems identification techniques. This work also relies on

Iñaki Rañó is with the Intelligent Systems Research Centre, University of Ulster, UK i.rano@ulster.ac.uk.
Hannes Edelbrunner is with NISYS GmbH, Technology Center Ruhr edelbrunner@nisys.de
Gregor Schöner is with the Institut für Neuroinformatik, Ruhr-Universität-Bochum gregor.schoener@ini.rub.de.
a double point steering model but presents the drawbacks of black-box models, i.e., they cannot not provide a broad understanding of how the control process is performed.

None of these works capture through the theoretically proposed controllers the known qualitative empirical findings of human driving [7]. This paper contributes to the modelling and implementation of naturalistic car driving by presenting an experimental comparison of possible human-like driver controllers based on known perceptual clues guiding driving behavior. This naturalistic motion generator can be actually used for path following tasks, a research area in robotics and automated guided vehicles. A proper way to model human driving would allow to characterize properties like individual ways to take curves, or to account for individual differences among drivers. Investigating further the parameters of the learned models can help identify and distinguish different types of driving style, cultural differences, and different levels of experience or performance. Such a system running continuously in a car would also enable to compare the current way of driving with a long term average in order to infer information about the state of the driver in terms of attention and it can be used to improve LDW systems. We also present a new linear perception model that allows to compute the two points used by humans, the near and far points.

The rest of the paper is organized as follows. Section II presents the methods and materials used in this work. It introduces the car model used and the proposed driver models, which includes a modified perception equation to match human control variables. It also gives a short introduction of our driver simulation lab. The experiments carried out and results from the model fitting are presented in section III, followed by a discussion of the main findings. Finally section IV draws some conclusions and presents future working directions.

II. THE CAR AND THE DRIVER MODEL

This section presents the car model including an adapted measurement equation for computing the far point from the car state vector and the road curvature. The linearized approximate commonly used to obtain the near point angle relative to the angle of the car assumes a straight road. This assumption does not hold for the far point in a curve, and the curvature of the road must be accounted for. However, in a real road situation, the relative angle of the far point can be obtained using, for instance, computer vision techniques. The selected models for the human controller are also presented in this section. Finally, we describe the specific driver simulator we used to gather human driving data in this work.

A. The Car-Perception Model

Our model of the vehicle dynamics is adapted from [1], though we will not consider the wind force disturbance since our focus is modelling human driving under normal conditions. The car model consist of a state space representation where the car is described by the state vector $\mathbf{x}^T = [\beta, r, \psi_L, y_L]$, with $\beta$ representing the slip angle, $r$ the yaw rate, $\psi_L$ the relative yaw angle and $y_L$ the lateral offset from the center of the lane. The driver input $u$ to the model is the steering wheel angle, while, as already stated, we will consider the road curvature $\rho$ as the only disturbance input $w$. Therefore, the linearized car dynamics is described by the equation:

$$\dot{\mathbf{x}} = A\mathbf{x} + B_u u + B_w w$$  \hfill (1)

where matrices $A$ and $B_u$ are a simplified version of those in [14], whilst the perturbation matrix $B_w$ is in our case:

$$B_w^T = \begin{bmatrix} 0 & 0 & -v & 0 \end{bmatrix}$$

$v$ being the linear velocity of the car. The numerical parameters of the simulated car are presented in Table I (cornering stiffness of the tires, length and mass of the car...)

Since humans use near and far angles to steer the car (see figure 1), the output equation should provide values for such variables. However, the existing perception models [14] [6] model the near (or far) point without accounting for the road curvature. While the near point angle ($\theta_n$) can be computed from the near distance ($d_n$), yaw angle ($\psi_L$) and lateral position of the vehicle ($y_L$) as, $\theta_n = \psi_L - \psi_L$, the far angle needs to account for 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:

$$\theta_f = \arcsin \left[ \frac{D_f^2 + y_L^2 + 2Ry_L}{2D_f(y_L + R)} \right] - \psi_L$$  \hfill (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 approximated output equation for the far point angle:

<table>
<thead>
<tr>
<th>$C_p(N/\circ)$</th>
<th>$C_I(N/\circ)$</th>
<th>$I_f(m)$</th>
<th>$I_v(m)$</th>
<th>$m(Kg)$</th>
<th>$I_z(m^2Kg)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>60000</td>
<td>47000</td>
<td>0.88</td>
<td>1.5</td>
<td>867</td>
<td>1146</td>
</tr>
</tbody>
</table>

TABLE I
PARAMETERS OF THE CAR MODEL.

Fig. 1. Reference system of the model variables.
\[ \theta_f = \frac{1}{D_f} y_L + \frac{D_f}{2} \rho - \psi_L \]  

(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 \( y = Cx + Dw \) can be stated as:

\[
C = \begin{bmatrix} 0 & 0 & -1 & \frac{1}{D_f} \\ 0 & 0 & -1 & \frac{D_f}{2} \end{bmatrix}
\]

and

\[ \begin{bmatrix} D_w^T \\
\end{bmatrix} \]

where the output vector is \( y^T = [\theta_n, \theta_f] \). The perception is carried out by the driver, but it is convenient to consider it as an output from the car/road system to be able to model separately the driving control mechanism.

**B. The Driver Model as a Grey-box**

When a controller has to be designed to steer a car, one reference point, the lateral distance from the car to the center of the lane, is used. This simplifies the control problem as under these conditions it is just a SISO system. However, this implies that the controller has to be adequate to keep the car centered on the lane and aligned with the road based on one reference point. The point where the angle is measured has to be properly selected since for short distances of the point to the car, oscillatory behavior might appear, even leading to unstablility. Moreover, a derivative component in the controller is necessary to quickly stabilize the car on the lane. Considering more than one point makes more complicated obtaining formal stability results.

On the other hand, humans, by relying on a two reference point control mechanism, implicitly use the change on the road shape, which is related to the temporal rate of change of the center point through the linear velocity. Therefore, using two reference points for a controller provides valuable information, as the far point gives an estimate of the upcoming road shape, which is related to the future rate of change of the near point. An important issue when modelling human drivers is finding out whether the controllers for the near and far points interact with each other or they produce independent outputs that are later combined. An empirical evidence pointing towards independence of the controllers is the fact that humans can steer a car relying only in one reference point when the second one is occluded [7]. However, it is also plausible that the strategies to steer a car using two or only one reference points will be completely different. In our previous work we modelled humans using black-box models through the steering torque and near and far points [11]. Specifically we trained ARX and state space models to fit data from human drivers. Even though the modelling capacity of the state space models for a given dimension is higher, and they produce a smaller prediction error, we found that the ARX model captures better the way humans really drive. Consequently, we concluded that the dynamics of the near and far points are not mixed and it is more plausible that two independent controllers are used by humans.

In this paper we compare four possible driver models or potential control techniques a human could use to steer a car. Since we used a linearized car model, and the deviations of the car direction with respect to the direction of the lane are small, we postulate that linear models would be a good approximation to capture human driving. Since the response of the human to the perceptual inputs is delayed by the sensori-motor and cognitive systems all the proposed models include a delay in their response. The near and far angles relative to the car direction, \( \theta_n \) and \( \theta_f \), correspond to the inputs of the controller modelling human drivers, while the output of the controller will be the angle \( \delta_d \) of the steering wheel.

1) **Single point phase-lead model**: The standard single point phase-lead controller is also used as a potential model for the drivers. This model assumes that humans use a single reference to steer the car jointly with its time evolution as estimated by the driver. It has been shown that humans can drive under such circumstances (when only one reference point is used), and therefore we include this model for completeness of our analysis. However, it is likely that such a mechanism is used for single reference point steering. The prediction of the road direction this controller provides is much shorter in time than using a far reference point. The transfer function of the model is:

\[
\delta_d(s) = K_n \frac{(1 + T_d s) e^{\tau s} \theta_n(s)}{1 + \alpha T_d s} 
\]

(4)

where \( K_n, T_d, \alpha \) and \( \tau \) are the parameters of the model, and it is the controller used to steer the car in [6].

2) **Two proportional controllers model**: This model represents the simplest two point control mechanism, where the steering of the car is just a combination of the effect of the near and far angles as viewed by the driver. The parameters of the model are the proportionality constants \( K_n, K_f \) and the reaction time \( \tau \).

\[
\delta_d(s) = K_n e^{\tau s} \theta_n(s) + K_f e^{\tau s} \theta_f(s) 
\]

(5)

3) **Phase-lead and proportional controller model**: This model is a combination of the two above, and it assumes a direct use of the far point information with a prediction estimate of the near point angle. It has been used to model human driving in [14]. Therefore, this controller assumes humans try to correct the deviation of the car’s heading direction using a prediction of how the car moves. The corresponding model can be stated as:

\[
\delta_d(s) = K_n \frac{(1 + T_D s)}{1 + \alpha T_D s} e^{\tau s} \theta_n(s) + K_f e^{\tau s} \theta_f(s) 
\]

(6)

where \( K_n, K_f, T_D, \alpha \) and \( \tau \) are the parameters of the model.
4) Integral (PI) and proportional controller model: This model assumes a proportional control of the far point and a proportional integral controller for the near point as presented in [13].

\[ \delta_i(s) = \frac{K_n (1 + T_I s)}{T_I s} e^{\tau \theta_n(s)} + K_f e^{\tau \theta_f(s)} \]  

where the parameters of the model are \( K_n, K_f, T_I \) and \( \tau \). The underlying assumption of this model is that humans try to keep the car centered on the lane, such that, if the road is straight, in the long term the car will drive exactly on the center of the lane.

C. The Driving Simulation Lab

The data needed to fit the models was acquired in our driving simulation lab, which has seats for up to ten drivers in individual driving cabins. The cabins are separated by panels and equipped with a force feedback steering wheel and a pedal console. The steering wheel and the pedals are attached to a computer with a screen displaying the simulated environment.

A realistic virtual environment, as shown in figure 2, 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 around 43°. For every cabin, the simulation software, TrafficSimulation [10], a commercial, fully customizable software with different modules and plug-ins, is run on the corresponding standard PC. Even though a key feature of this simulator is its support for multiple subjects driving simultaneously in a shared environment, in the present experiment, only two adjacent cabins were used with simulators running in standalone-mode.

![Fig. 2. Screenshot of the TrafficSimulator.](image)

The simulator software runs the car model presented earlier with the parameters shown in table I in a configurable simulation environment, which includes, among others, traffic signs, buildings and trees. The distribution of the elements in the environment can be configured in the software at start up time. The simulator also allows to have simulated cars driving in the same environment driven by an control architecture that allows them performing different maneuvers that include overtaking if necessary, depending on their goals.

III. EXPERIMENTS AND DRIVER IDENTIFICATION

This section presents the procedure followed to acquire the data and the results of fitting the proposed models.

A. Data Acquisition and Model Fitting

Previously to the the data acquisition process the location and height of the pedals and chair in the cabins were adapted to the drivers. Two cabins were used simultaneously with two independent simulated environments, generated by the same configuration file. Under this configuration drivers could drive in couples while talking to each other, inducing a rather automatic driving behavior instead of only focusing on the simulated environment. This makes the results more realistic and helps avoiding the boredom of the subjects that perform a repetitive task for a long time.

In order to sample driving behavior under different conditions, we created a specific road layout for the simulated environment. The initial part of the test track includes a 9 km long accommodation stretch with curves of different radii such that the users adapt to the driving environment and conditions. Then the road is subdivided into three parts, each of which with a different speed limit (80, 100 and 120 km/h), indicated to the driver through traffic signs alongside the road. Every part contains sections of different curvatures, alternating 5 right and 5 left turns. Curves are separated by 300 m long straight stretches to let the drivers stabilize the car in 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 standard 3.8 m wide lanes.

The ride takes altogether about 20 minutes and the accommodation part was excluded from the recorded data in order to properly train the models. Several visualization plug-ins allow displaying different elements on the environment like; other vehicles, trees, buildings, billboards and guard rails. For the experiment at hand the environment contained only trees and buildings randomly distributed along the road. One of the plug-ins of TrafficSimulator, the DataWrite plug-in, allowed us to record driving data at a rate of 25 Hz. Specifically, the system state data recorded are: steering wheel angle \( \delta \), lateral position \( y_L \), relative yaw angle \( \psi_L \), yaw rate \( r \), track position \( s \), and velocity \( v \). This information is needed to compute the inputs required by the proposed driver models.

Thirteen subjects accomplished the driving task, five females and eight males. The subjects were instructed to keep the car in the lane, but trying to keep the car centered was not required, so each driver was allowed to steer the car freely in a way he/she would find comfortable. For each subject a data-set was obtained and the parameters of the four models were identified. The average age among the subjects was 35.9 years with a minimum of 27 and a maximum of 56 years, and they held their licenses or 17 years on average, and reported to be driving 12120 km per year on average, thereof 5760 km on highways. Therefore the sample includes rather experienced drivers, but none of them professional.
B. Results of the Lane-Keeping Identification

Since the TrafficSimulator software provides us with the state in time for all the trajectories, we used the output equation presented in section II 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 10 Hz after a filtering process 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 assumed to be \( D_f = 75 \text{ m} \) and \( d_n = 5 \text{ m} \). For the single point controller model we used a look ahead distance of 22 m, since smaller distances will lead to unstabilities in the closed loop control system. Moreover, it is plausible that a human steering a car with one reference point will adapt the distance to improve the stability on the lane. Obviously, a too close distance in the single point model would turn out the fitting procedure very difficult as the controller might not be able to stabilize the car in the middle of the lane. Using these parameters and the lateral position, the relative yaw angle and the track position, we can compute the far and the near angle, \( \theta_f \) and \( \theta_n \), which are the inputs required to train the driver model.

The gathered data was used to identify all the proposed driver models through a Prediction Error Method (PEM) algorithm, where the delay was converted to a first order Padé approximation. To asses the relative performance of the models in capturing the drivers’ behavior, we computed the average errors and standard deviation between the data and the obtained driver model for all the drivers. The results are presented in the last two columns of table II. As it can be seen the difference between the one point model and two points model is of one order of magnitude, while the including a more complex model does not bring much once we use the two points to steer the car. The table includes the stability analysis in closed loop of the control models as the percentage of models for different drivers that are stable in closed loop. The stability test was performed by building the complete car-driver linear model and analyzing the eigenvalues of the corresponding matrix, such that if at least one eigenvalue has positive real part the whole system is unstable. As it can be seen, the model which generates the most close loop stable systems is the one with two proportional controllers for the near and the far point. The data set for one of the drivers could not be fit to generate stable closed loop behavior but the only positive eigenvalue of the closed loop system had a time constant of 500s, and therefore it seems to be stable for short simulations. In the opposite side is the controller with the integral part. Since usually humans do not keep the car centered on the lane, the model cannot account for the data, and the unstable eigenvalues produce in general much faster divergence.

Figure 3 presents the real trajectory of the steering wheel angle of one of the drivers and the trajectories obtained through the different models for that driver during a period of 100s. The models were used to predict the steering angle based on the near and far angles from the data sets. It can be seen that the steering wheel angle predicted by the two points models (dashed lines) match quite well the real angular trajectory (continuous black line). That means these are suitable models of the control mechanism used by real driver. Moreover, the model using an integral component for the near point can also predict quite accurately the steering angle even though as shown in table II it cannot generate proper commands to steer a car in closed loop. This is because the controller itself has no unstable components, but the joint car-driver model does. On the other hand, the angular trajectory obtained simulating the single point controller (continuous bright line) is not able to match the human response, even though it can steer the car to keep it on the lane according to the stability performed analysis.

An interesting result derived from figure 3 is that there is almost no difference in terms of steering angle prediction between the double proportional controller and the one using a phase lead control of the near point. This could imply that humans, during normal driving, do not predict the rate of change of the near point, but use the far point instead. To further compare these two mechanisms we plotted in figure 4 the parameters obtained from the identification of each driver. The pairs corresponding to a same driver are connected with a continuous line, while the point with a red circle inside corresponds to an unstable closed loop controller and other controllers producing unstable behavior are not plotted. As it can be seen in the figure, the obtained \( K_f \) values for the far controller roughly match for both cases, while the value of \( K_n \) changes with the model. As we saw earlier, the improvement in the average error when including the predictive component is small, instead the capability of these fitted models to generate stable closed loop trajectories decreases.

<table>
<thead>
<tr>
<th>Model</th>
<th>Stable</th>
<th>Av. Err.</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PL</td>
<td>61%</td>
<td>1.4 · 10^{-3}</td>
<td>3.1 · 10^{-3}</td>
</tr>
<tr>
<td>P-P</td>
<td>92%</td>
<td>1.6 · 10^{-6}</td>
<td>4.0 · 10^{-6}</td>
</tr>
<tr>
<td>PL-P</td>
<td>77%</td>
<td>1.5 · 10^{-6}</td>
<td>3.9 · 10^{-6}</td>
</tr>
<tr>
<td>PP-P</td>
<td>0%</td>
<td>2.7 · 10^{-6}</td>
<td>6.8 · 10^{-6}</td>
</tr>
</tbody>
</table>

TABLE II
STABILITY AND ERROR MEASURES OF THE MODELS.
Finally, figure 5 shows the behavior of the models obtained for driver negotiating a curve of radius and length 2500 and 800 meters respectively at 100 Km/h. This configuration was included among the experimental setup. As it can be seen in the figure, the models relying on two points depart from the center of the lane but stay within half a meter of it. On the other hand the model including an integral component is unstable and, as represented in the figure, it cannot keep the car on the lane. Interestingly, even though the single point controller is stable in closed loop when including the car dynamics, its static gain between the road curvature and the lateral position is big enough as to drive the car out of the road too. Since this controller produces a closed loop stable system, the difference with the integral controller is that the distance to the middle of the lane is bounded.

IV. CONCLUSIONS AND FURTHER WORK

This paper presents a comparison between four possible models of human lane keeping, some of them found in the literature but not fitted with real human drivers data. We propose a simpler new model based on the know result of two point steering control of the car, namely a proportional controller for both, the near and far points. Results show that this simple control mechanism fits better to the empirical data obtained from drivers in a simulated environment. Despite being simpler and not giving the best match in terms of mean square error, the model captures better the way humans control cars than other models. Since humans do not keep the car exactly centered on the lane, models following this assumption can predict in the short term the trajectory based on previous inputs but fail to successfully control a car. On the other hand, the underlying hypothesis that drivers use the evolution of the near point to steer the car is unnecessary in view of the results. Also according to our experimental results, a single point controller can be adjusted to keep a car on the lane but they are not suitable to generate a human like trajectory.

Since the near and far distances are part of the car/perception model their values had to be selected beforehand, and therefore the results could change slightly if other values are used. The performed analysis focused on the lane keeping control mechanisms not in the perceptual part. To mitigate this issue we plan to include these parameters inside the controller model, therefore generating a perception/control model that can be fitted to individual drivers. The two point based lane keeping model with the obtained parameters will be integrated into a general driver model that includes other driving behaviors and a decision making model [12]. This will allow having naturalistic simulated drivers our driver simulator. Further extension to our work will include an on-line model learning that can be used to test or improve lane departure warning systems and lane keeping systems adapting them to specific user.

REFERENCES