Lane Keeping Assistance with Learning-Based Driver Model and Model Predictive Control

Stéphanie Lefèvre, Yiqi Gao, Dizan Vasquez, H. Eric Tseng, Ruzena Bajcsy, Francesco Borrelli

▶ **To cite this version:**

Stéphanie Lefèvre, Yiqi Gao, Dizan Vasquez, H. Eric Tseng, Ruzena Bajcsy, et al.. Lane Keeping Assistance with Learning-Based Driver Model and Model Predictive Control. 12th International Symposium on Advanced Vehicle Control, 2014, Tokyo, Japan. hal-01104458

HAL Id: hal-01104458
https://inria.hal.science/hal-01104458

Submitted on 27 Jan 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Lane Keeping Assistance with Learning-Based Driver Model and Model Predictive Control

Stéphanie Lefèvre[*†], Yiqi Gao[*], Dizan Vasquez[†], H. Eric Tseng[‡], Ruzena Bajcsy[*], Francesco Borrelli[*]

[*] **University of California, Berkeley, CA, USA**
[†] **Inria, Grenoble, Rhône-Alpes, France**
[‡] **Ford Research Laboratories, Dearborn, MI, USA**

Berkeley, CA 94720, USA
Phone: +1 510 6432044
E-mail: slefevre@berkeley.edu

This paper proposes a novel active Lane Keeping Assistance Systems (LKAS) which relies on a learning-based driver model. The driver model detects unintentional lane departures earlier than existing LKAS, and as a result the correction needed to keep the vehicle in the lane is smaller. When the controller has control of the car, the driver model estimates what the driver would do to keep the car in the lane, and the controller tries to reproduce that behavior as much as possible so that the controlled motion feels comfortable for the driver. The driver model combines a Hidden Markov Model and Gaussian Mixture Regression. The controller is a Nonlinear Model Predictive Controller. The results obtained with real data show that our driver model can reliably predict lane departures. The controller is able to keep the car in the lane when there is a risk of lane departure, and does so less intrusively than existing LKAS.

Topics / Active safety and driver assistance systems, Driver modeling

1. INTRODUCTION

Lane Keeping Assistance Systems (LKAS) were introduced a decade ago and are gradually becoming available in passenger cars. They rely on proprioceptive and exteroceptive sensors to predict unintentional lane departures on the highway and try to prevent them by warning the driver or by actively controlling the car. In the second case it is crucial to define a suitable strategy for switching control between the driver and the controller. The main challenge is to be both efficient (avoid lane departures) and as unintrusive as possible (for driver comfort and acceptance). Unintentional lane departures should be detected early, so that the controller is able to keep the vehicle in the lane with small inputs.

Typically, commercial LKAS detect unintentional lane departures based on the Time to Line Crossing (TLC) [1, 2, 3]. The idea is to predict the future trajectory of the vehicle using a dynamical model and to compute the time remaining before the car reaches the lane markings. The TLC can be computed in a number of ways, depending on the assumptions made in the vehicle model and about the shape of the road [4]. A major drawback with TLC-based LKAS is that they require setting a threshold on the TLC, and that they are prone to false alarms [5].

As an alternative to dynamical models and trajectory prediction, it is possible to detect upcoming lane departures using maneuver recognition algorithms. The advantages are twofold. Firstly, they do not require setting a threshold. Secondly, they can learn from data and adapt to the driving style of a specific driver. Therefore lane departure prediction tends to be more reliable [5]. Examples of algorithms for lane change prediction include Support Vector Machines [6, 7], Relevance Vector Machine [8, 9], Hidden Markov Models [10], Hierarchical PieceWise ARX mode [11], Directional Sequence of Piecewise Lateral Slopes [5]. To the best of our knowledge, none of these approaches has been used in combination with a controller in an active LKAS.

In this work we use a Hidden Markov Model (HMM) to predict lane departures. The HMM is driver-specific, i.e. we train one model per driver using driving data from that driver. When the system detects an upcoming unintentional lane change, the control of the vehicle is handed to the controller, which is responsible for keeping the vehicle in the lane. We use a Nonlinear Model Predictive Controller (NMPC) to perform this task [12, 13, 14], with a modified optimization criterion to further reduce the intrusiveness

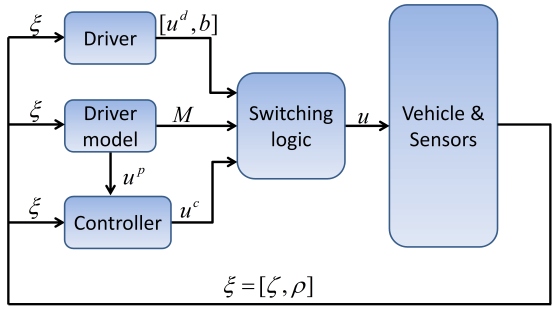

Fig. 1: Architecture of the proposed LKAS.

of the system. The modification is as follows: as soon as the controller gets switched on, we use Gaussian Mixture Regression (GMR) on the HMM to predict the most likely sequence of inputs that the driver would apply to keep the vehicle in the lane. The controller takes that prediction into account in the optimization criterion, and generates control inputs which keep the car in the lane while matching the driver's own driving style as much as possible.

The rest of the paper is organized as follows. Section 2 describes the overall architecture of the proposed LKAS. Section 3 introduces the driver model, and Section 4 the controller. The experiments and results are presented in Section 5, and Section 6 presents conclusions and future work.

2. SYSTEM OVERVIEW

The proposed LKAS architecture is illustrated in Fig. 1 and described below.

2.1 Variables

This section defines the variables used in Fig. 1.

$u^d = [\delta^d, \beta_r^d]$ is the driver input, with δ^d the front steering angle and $\beta_r^d \in [-1, 1]$ the braking ratio. $\beta_r = -1$ corresponds to full braking and $\beta_r = 1$ corresponds to full throttle.

$b \in \{left, right, none\}$ is the state of the turn signal set by the driver.

$u^c = [\delta^c, \beta_r^c]$ is the controller input, with δ^c the front steering angle and β_r^c the braking ratio.

$u = [\delta, \beta_r]$ is the input applied to the vehicle, and corresponds to the driver input or controller input.

$\zeta = [\dot{x}, e_y, \dot{e}_y, e_\psi, \dot{e}_\psi]$ is the vehicle state, with $\dot{x}$ the longitudinal speed, e_y the distance to the lane center, e_ψ the orientation with respect to the lane center, $\dot{e}_y$ the derivative of e_y, and $\dot{e}_\psi$ the derivative of e_ψ.

$\xi = [\zeta, \rho]$ is the driving situation. It contains the state of the vehicle ζ and the road curvature ρ.

$m \in \{m_{LK}, m_{DL}, m_{DR}\}$ corresponds to the control strategy applied to the commands of the vehicle. When the vehicle is driven in lane keeping mode ($m = m_{LK}$), it will stay in its current lane. When the vehicle is driven in lane departure mode ($m = m_{DL}$ or $m = m_{DR}$), it will depart from its current lane to reach the left lane ($m = m_{DL}$) or the right lane ($m = m_{DR}$). The most likely control strategy is denoted as M.

$u^p = [\delta^p, \beta_r^p]$ is the predicted driver input, with δ^p the most likely front steering angle and β_r^p the most likely braking ratio, assuming the control strategy will be $m = m_{LK}$ for the entire prediction horizon.

2.2 Block description

This section defines the modules of the architecture introduced in Fig. 1.

The "Vehicle & Sensors" block is the physical car equipped with proprioceptive sensors and a forward-looking camera fixed on the windshield. The former provide information about the current steering $\delta(t)$, braking ratio $\beta_r(t)$, and longitudinal speed $\dot{x}(t)$ of the vehicle. The camera provides information about the current position and orientation of the vehicle with respect to the lane, i.e. $e_y(t)$ and $e_\psi(t)$, and about the current road curvature $\rho(t)$.

The "Driver" block is the human driver, who uses the history of driving situations $\xi(t_0 : t)$ to generate an input $u^d(t+1)$ and to set the turn signal $b(t+1)$.

The "Driver model" block estimates the most likely current control strategy $M(t)$ applied to the car, from the history of driving situations $\xi(t_0 : t)$. It also predicts the most likely sequence of inputs $u^p(t : t + H_p)$ that the human driver would apply to keep the vehicle in the current lane (i.e. assuming $m(t : t + H_p) = m_{LK}$), with H_p the prediction horizon.

The "Controller" block computes the input $u^c(t+1)$ with the objective to prevent the vehicle from leaving the current lane. $u^c(t+1)$ is computed to be close to $u^p(t+1)$ so that the motion of the car matches the driver's driving style.

The "Switching logic" block manages the sharing of the control of the vehicle between the driver and the controller. The driver controls the vehicle through the steering wheel and the accelerator pedal, the controller controls the vehicle through the electric power steering and the acceleration command on the CAN-bus. The switching logic is implemented as follows:

- If the car is most likely in lane keeping mode ($M(t) = m_{LK}$), the driver is always in charge of controlling the vehicle ($u(t+1) = u^d(t+1)$).
- When the car is most likely in lane departure mode ($M(t) = m_{DL}$ or $M(t) = m_{DR}$), the system decides who should be in charge based on the state of the turn signal. If the turn signal is consistent with the maneuver ($M(t) = m_{DL}$ and $b(t) = left$, or $M(t) = m_{DR}$ and $b(t) = right$), the upcoming lane change is considered intentional and the driver is in charge of controlling the vehicle ($u(t+1) = u^d(t+1)$). Otherwise the upcoming lane change is considered

unintentional and the controller is in charge of the control ($u(t+1) = u^c(t+1)$).

This logic, which relies on the turn signal to distinguish intentional and unintentional lane departures, is similar to commercial LKAS. Control switching occurs only in 3 situations: when lane keeping is followed by unintentinal lane departure (control transferred from the driver to the controller), when unintentional lane departure is followed by intentional lane departure (control transferred from the controller to the driver), and when unintentional lane departure is followed by lane keeping (control transferred from the controller to the driver).

2.3 Override of the LKAS

Similarly to existing LKAS, we want the driver to be able to easily override the system and take control of the vehicle at any time to perform the desired maneuver, with or without activating a turn signal. For this reason, an "LKAS interruption" process runs in parallel to the LKAS described in Fig. 1, and transfers control back to the driver as soon as it detects the driver's intention to override the LKAS. It has been shown in the past that this can be detected by looking at the torque applied by the driver on the steering wheel [1].

3. DRIVER MODEL

Classically in LKAS, lane departures are predicted by propagating forward the position of the vehicle using a dynamical model, and the decision to intervene is based on a threshold on the Time to Line Crossing (TLC) [1]. Instead, we model the driver as a statistical system switching between 3 control strategies $m \in \{m_{LK}, m_{DL}, m_{DR}\}$, and we use observations to iteratively estimate the most likely control strategy currently applied to the car. The proposed driver model is a Hidden Markov Model (HMM) whose structure and parameters are learned in 4 steps from real data. These steps are described below in Sec. 3.1. The inference process to estimate the most likely control strategy and to predict the future driver input is presented in Sec. 3.2.

3.1 Learning the structure and parameters of the HMM

The structure and parameters of the HMM are learned from real data, following the 4 steps illustrated in Fig. 2 and described below.

Step 1 - Labeling real data: The first step of the learning process is to label each individual data point in the training dataset with the corresponding control strategy $m \in \{m_{LK}, m_{DL}, m_{DR}\}$. The instant t_D when a lane change occurs is easy to detect by post-processing the data, since there is a "jump" in the data points e_y (see Fig. 3). Then the standard approach to define the time t_B when the lane change

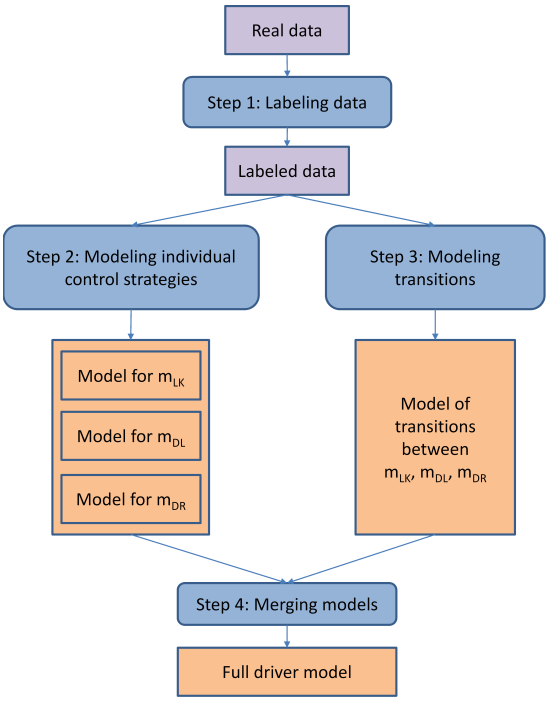

Fig. 2: Steps for learning the structure and parameters of the HMM, as described in Sec. 3.1.

begins and the time t_E when the lane change ends is to define a constant duration Δ_D for lane change maneuvers and to compute $t_B = t_D - \Delta_D/2$ and $t_E = t_D + \Delta_D/2$. Instead we allow each lane change to have a different duration and define t_B (resp. t_E) by fitting a line on the N data points of e_y collected before (resp. after) t_D, as illustrated by in Fig. 3. t_B and t_E are defined as the time when the fitted lines cross the x-axis.

Step 2 - Modeling individual control strategies: Each control strategy $m_i, i \in \{LK, DL, DR\}$ is modeled by a fully connected HMM. Our goal is to model the dependencies between the current driving situation $\xi(t)$ and current driver steering $\delta^d(t)$, therefore we define the following random variables for the HMM representing control strategy m_i:

$q_i(t) \in \{q_i^j\}$, $j = [1, Q_i]$ is the hidden state at time t, with Q_i the number of possible hidden states for control strategy m_i.

$o(t) = [\xi(t), \delta^d(t)] \in \mathbb{R}^7$ are the observations.

The joint probability is written as:

$$P(q_i(0:t), \xi(0:t), \delta^d(0:t)) = P(q_i(0), \xi(0), \delta^d(0)) \times \prod_{k=1:t} \Big[P(q_i(k)|q_i(k-1)) \times P(\xi(k), \delta^d(k)|q_i(k)) \Big] \quad (1)$$

A multivariate Gaussian distribution is assumed for $P(\xi(k), \delta^d(k)|q_i(k))$. The parameters of the HMM (number of states, priors, transition probabilities, means and standard deviations of the Gaussian distributions) are learned from the labeled training data using the Expectation-Maximization (EM) algorithm and the Bayesian Information Criterion (BIC).

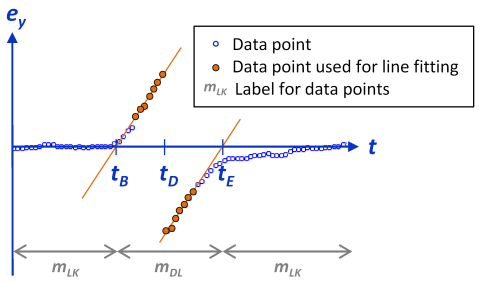

Fig. 3: Automatic labeling of lane changes with $N = 7$, as described in Sec. 3.1.

Step 3 - Modeling transitions between control strategies: The transitions between the different control strategies are modeled by a Markov process with fully connected transition matrix. The priors $P(m(0))$ and the transition probabilities $P(m(k)|m(k-1))$ are learned by counting the number of occurrences and transitions in the labeled training data, and normalizing.

Step 4 - Merging individual models into a full driver model: The full driver model connects the variables introduced in the previous steps:

$$[m(t), q_{LK}(t), q_{DL}(t), q_{DR}(t), \xi(t), \delta(t)] \tag{2}$$

Since there can be only one active control strategy at a time, the joint state introduced in Eq. 2 can be simplified as:

$$[q(t), \xi(t), \delta(t)] \tag{3}$$

with $q(t) \in \{q_i^j\}$, $i = [LK, DL, DR]$, $j = [1, Q_i]$. To define the joint distribution we use a naive approach where the hidden variable $q(t)$ follows a Markov process with fully connected transition matrix and multivariate Gaussian observations:

$$P(q(0:t), \xi(0:t), \delta(0:t)) = P(q(0), \xi(0), \delta(0)) \times \prod_{k=1:t} \Big[P(q(k)|q(k-1)) \times P(\xi(k), \delta^d(k)|q(k))\Big] \tag{4}$$

This new HMM combines the hidden states learned for each individual control strategy. The parameters are learned from the same training data as before, but we introduce a bias in the learning process in order to make use of the HMM structures and parameters learned in Step 2 and Step 3. The parameters of the Gaussian distributions are set a priori using the values computed in Step 2, and are never modified. The priors and the transition probabilities are learned using the EM algorithm, from an initial guess computed using the priors and transition matrices learned in Step 2 and Step 3.

3.2 Inference on the driver model

In the proposed LKAS (see Fig. 1), the driver model is used in two ways: 1) to estimate the most likely control strategy and 2) to predict the future driver inputs. Details of how this is done are provided below.

Mode estimation: The most likely current control strategy $M(t)$ is iteratively selected as the most likely mode given the history of driving situations, by computing:

$$M(t) = \underset{m_i \in [m_{LK}, m_{DL}, m_{DR}]}{\arg\max} \sum_{j=1}^{Q_i} P([q(t) = q_i^j]|\xi(0:t)) \tag{5}$$

with

$$P(q(t)|\xi(0:t)) \propto P(q(0), \xi(0)) \times \prod_{k=1:t} [P(q(k)|q(k-1)) \times P(\xi(k)|q(k))] \tag{6}$$

Driver input prediction: The sequence of future steering inputs $\delta^p(t : t + H_p)$ applied by the driver for lane keeping can be predicted recursively as follows. First, a bicycle vehicle model [15] is used to predict $\xi(t+1)$ from $\xi(t)$ and $\delta^d(t)$. Then, the driver input at time $t+1$ is predicted using Gaussian Mixture Regression [16]:

$$\delta^p(t+1) = E\{P(\delta^d(t+1)|\xi(0:t+1), m(t+1) = m_{LK})\} \tag{7}$$

These two steps are repeated until the prediction horizon H_p is reached.

The sequence of future braking ratios $\beta_r^p(t : t + H_p)$ applied by the driver for lane keeping is predicted using a simple "constant speed" model. A more advanced method, such as Gaussian Mixture Regression, could be used if we had access to more information about the driving situation. More specifically, the acceleration behavior of a driver is strongly influenced by the presence of other vehicles in the nearby lanes, and therefore information about the distance and relative velocity of these vehicles would allow a better prediction. However at this time our vehicle is not equipped with enough sensors to cover 360°and can detect only the vehicle in front.

4. CONTROLLER

In this section we describe the vehicle model and the model predictive controller.

4.1 Vehicle model

The nonlinear bicycle vehicle model used in this work is illustrated in Fig. 4 and described below. We use the following set of differential equations to describe the vehicle motion within the lane:

$$m\ddot{x} = m\dot{x}\dot{\psi} + 2F_{xf} + 2F_{xr}, \tag{8a}$$
$$m\ddot{y} = -m\dot{x}\dot{\psi} + 2F_{yf} + 2F_{yr}, \tag{8b}$$
$$I_z\ddot{\psi} = 2aF_{yf} - 2bF_{yr}, \tag{8c}$$
$$\dot{e}_\psi = \dot{\psi} - \dot{\psi}_d, \tag{8d}$$
$$\dot{e}_y = \dot{y}\cos(e_\psi) + \dot{x}\sin(e_\psi), \tag{8e}$$
$$\dot{s} = \dot{x}cos(e_\psi) - \dot{y}sin(e_\psi), \tag{8f}$$

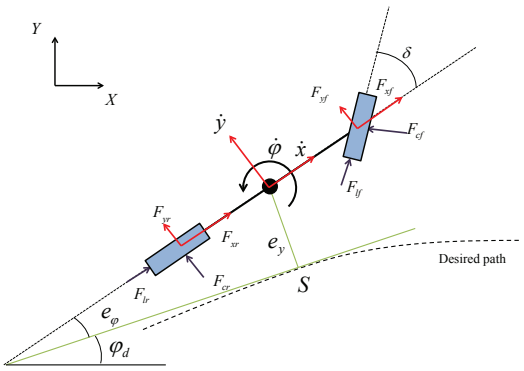

Fig. 4: Modeling notations depicting the forces in the vehicle body-fixed frame ($F_{x\star}$ and $F_{y\star}$), the forces in the tire-fixed frame ($F_{l\star}$ and $F_{c\star}$), and the rotational and translational velocities. The relative coordinates e_y and e_ψ are illustrated on the sketch of the road as well as the the road tangent ψ_d.

where m and I_z denote the vehicle mass and yaw inertia, respectively, a and b denote the distances from the vehicle center of gravity to the front and rear axles, respectively. $\dot{x}$ and $\dot{y}$ denote the vehicle longitudinal and lateral velocities, and $\dot{\psi}$ is the turning rate around a vertical axis at the vehicle's center of gravity. e_ψ and e_y denote the vehicle orientation and lateral position, respectively, in a road aligned coordinate frame and ψ_d is the angle of the tangent to the road centerline in a fixed coordinate frame. s is the vehicle longitudinal position along the lane center. F_{yf} and F_{yr} are front and rear tire forces acting along the vehicle lateral axis, F_{xf} and F_{xr} forces acting along the vehicle longitudinal axis.

The longitudinal and lateral tire force components in the vehicle body frame are modeled as,

$$F_{x\star} = F_{l\star}\cos(\delta_\star) - F_{c\star}\sin(\delta_\star), \tag{9a}$$
$$F_{y\star} = F_{l\star}\sin(\delta_\star) + F_{c\star}\cos(\delta_\star), \tag{9b}$$

where $\star$ denotes either f or r for front and rear tire, $\delta_\star$ is the steering angle at wheel. We introduce the following assumption on the steering angles.

Assumption 1 *Only the steering angle at the front wheels can be controlled. i.e., $\delta_f = \delta$ and $\delta_r = 0$.*

The longitudinal force in the tire frame, $f_{l\star}$ is calculated from the equation

$$f_{l\star} = \beta_r \mu F_{z\star} \tag{10}$$

where $\beta_r \in [-1, 1]$ is referred to as the braking ratio. $\beta_r = -1$ corresponds to full braking and $\beta_r = 1$ corresponds to full throttle. $f_{c\star}$ is computed using a modified nonlinear Fiala tire model [17]:

$$f_{c\star} = \begin{cases} -C_\alpha tan(\alpha_\star) + \frac{C_\alpha^2}{2\eta\mu F_{z\star}}|tan(\alpha_\star)|tan(\alpha_\star) \\ \quad - \frac{C_\alpha^3}{27\eta^2\mu^2 F_{z\star}} tan^3(\alpha_\star), & \textbf{if } |\alpha_\star| < \alpha_{sl} \\ -\eta\mu F_{z\star} sgn(\alpha_\star), & \textbf{if } |\alpha_\star| \geq \alpha_{sl} \end{cases} \tag{11}$$

where $\alpha_\star$ denotes the tire slip angle, μ denotes the friction coefficient, and $Fz_\star$ denotes the vertical load at each wheel. C_α is the tire cornering stiffness and $\eta = \frac{\sqrt{\mu^2 F_z^2 - f_l^2}}{\mu F_z}$ which can be written as $\eta = \sqrt{1-\beta_r^2}$. $\alpha_{sl} = tan^{-1}(\frac{3\eta\mu F_z}{C_\alpha})$ denotes the saturation point of the slip ratio. μ is assumed to be a known constant and is the same for all wheels. We also assume the vertical forces $F_{z\star}$ to be constant and determined by the steady state weight distribution of the vehicle with no acceleration at the center of the gravity.

The slip ratios $\alpha_\star$ are approximated as follows:

$$\alpha_f = \frac{\dot{y} + a\dot{\psi}}{\dot{x}} - \delta, \quad \alpha_r = \frac{\dot{y} - b\dot{\psi}}{\dot{x}} \tag{12}$$

We write the model (8) to (12) in the following compact form:

$$\dot{z}(t) = f(z(t), u(t)) \tag{13}$$

where $z = [\dot{x}, \dot{y}, \dot{\psi}, e_\psi, e_y, s]$ and $u = [\delta, \beta_r]$ are the state and input vectors respectively.

4.2 Safety constraints

The overall objective of the system proposed in this paper is to keep the vehicle in the lane. In this section, we express this objective as constraints on the vehicle state and input. To reduce the complexity, we only constrain the CoG of the vehicle for lane keeping. This can be expressed as:

$$e_{ymin} + w \leq e_y \leq e_{ymax} - w \tag{14}$$

where e_{ymin} and e_{ymax} are bounds determined by the lane width, and w is the vehicle width.

To ensure that the vehicle operates in a state space where a normal driver can easily handle the vehicle, we constrain the tire slip angles to be within the linear region:

$$-\alpha_{sl} \leq \alpha_\star \leq \alpha_{sl} \tag{15}$$

The constraints (14) to (15) are compactly written as:

$$h(z, u) \leq \mathbf{0}, \tag{16}$$

where

$$h(z,u) = \begin{bmatrix} -e_y + e_{ymin} + w \\ e_y - e_{ymax} + w \\ -\alpha_f(z,u) - \alpha_{sl} \\ -\alpha_r(z,u) - \alpha_{sl} \\ \alpha_f(z) - \alpha_{sl} \\ \alpha_r(z) - \alpha_{sl} \end{bmatrix}$$

and $\mathbf{0}$ is the zero vector with appropriate dimension.

4.3 Predictive control problem

In this section we formulate the lane keeping problem as a Nonlinear Model Predictive Control (NMPC) problem. At each sampling time instant an optimal input sequence is computed by solving a constrained finite time optimal control problem. The computed

optimal control input sequence is only applied to the system during the following sampling interval. At the next time step the optimal control problem is solved again, using new measurements.

We discretize the system (13) with a fixed sampling time Δt_s to obtain:

$$z_{k+1} = f_{\mathrm{d}}(z_k, u_k), \tag{18}$$

and formulate the optimization problem being solved at each time instant as:

$$\min_{\bar{\mathbf{u}},\varepsilon} \sum_{k=0}^{H_p-1} ||\eta_{t+k,t} - \eta_{ref}||^2_Q + ||u^{\Delta}_{t+k,t}||^2_R + ||\Delta u^{c}_{t+k,t}||^2_S + \lambda\varepsilon \tag{19a}$$

$$s.t.\ z_{t+k+1,t} = f_{\mathrm{d}}(z_{t+k,t}, u^c_{t+k,t}),\ k = 0, ..., H_p - 1, \tag{19b}$$

$$u^c_{t+k,t} = u^{\Delta}_{t+k,t} + u^p_{t+k,t},\ k = 0, ..., H_p - 1, \tag{19c}$$

$$u^c_{t+k+1,t} = u^c_{t+k,t} + \Delta u^c_{t+k,t},\ k = 0, ..., H_p - 2, \tag{19d}$$

$$h(z_{t+k,t}, u^c_{t+k,t}) \leq \mathbf{1}\varepsilon,\ \ \varepsilon \geq 0,\ k = 1, ..., H_p, \tag{19e}$$

$$u^c_{t+k,t} \in \mathcal{U},\ k = 0, ..., H_p - 1, \tag{19f}$$

$$\Delta u^c_{t+k,t} \in \Delta\mathcal{U},\ k = 0, ..., H_p - 1, \tag{19g}$$

$$z_{t,t} = z(t), \tag{19h}$$

where t denotes the current time instant and $z_{t+k,t}$ denotes the predicted state at time $t + k$ obtained by applying the control sequence $\{u^c_{t,t}, \ldots, u^c_{t+k,t}\}$ to the vehicle model with $z_{t,t} = z(t)$. $\eta_{t+k,t} = [\dot{\psi}, e_\psi, e_y] \subset z_{t+k,t}$ denotes the tracked vehicle states and η_{ref} is the reference, which here corresponds to the lane center. $H_p = 1.2\ s$ is the prediction horizon. (19e) denotes the state constraints imposed by lane keeping and have been imposed as soft constraints, by introducing the slack variable ε. $u^p_{t+k,t}$ is the predicted driver input at timestep k, provided by the driver model with $m(t : t+H_p) = m_{LK}$ (see Sec. 3.2). $u^{\Delta}_{t+k,t}$ is the correction to be added to $u^p_{t+k,t}$ to ensure that the safety constraints are respected. Q, R, S and λ are weights of appropriate dimension penalizing the deviation from the reference, the deviation from the predicted driver input, the change rate of the control input, and violation of the constraints, respectively.

5. Evaluation

The LKAS proposed in this paper makes use of a driver model both for detecting lane departures and for controlling the vehicle (see Fig. 1). In order to evaluate the performance of the approach, the proposed LKAS is compared against two reference LKAS. The first one, called *LKAS1*, uses a driver model for detecting lane departures but not for controlling the vehicle, i.e. the arrow between the "Driver model" and the "Controller" blocks in Fig. 1 is removed. The second one, called *LKAS2*, uses no driver model at all, i.e. the "Driver model" block in Fig. 1 is removed. The comparison between the three LKAS is done using a combination of recorded driving data and simulation.

5.1 Description of *LKAS1*

LKAS1 detects lane departures using the same approach as the proposed LKAS, but differs in the way it controls the vehicle when the controller is switched on. Instead of predicting the steering that the driver would apply to do lane keeping and mimicking this behavior with the controller, *LKAS1* simply brings the vehicle back to the center of the lane as efficiently and smoothly as possible. In practice this means that the only difference with the proposed LKAS is the control optimization problem. Instead of penalizing the deviation from the predicted driver input, *LKAS1* penalizes strong controller inputs, similarly to what we did in previous work [13]. Eq. (19c) therefore becomes: $u^c_{t+k,t} = u^{\Delta}_{t+k,t}$.

5.2 Description of *LKAS2*

In *LKAS2*, decisions to intervene are based solely on the Time to Line Crossing (TLC), similarly to commercial LKAS [1]. The TLC can be computed in many different ways [4]; here we consider the road curvature and assume a straight vehicle trajectory. This solution was shown to perform well in the past [4]. The decision to transfer the control of the car between the driver and the controller is made based on a threshold γ on the TLC. When $TLC \leq \gamma$ the controller is given control of the car. The controller is the same as the one implemented in *LKAS1*: it does not use a driver model, and brings the vehicle back to the center of the lane as efficiently and smoothly as possible. Control is transferred back to the driver when $TLC > \gamma$. The resulting LKAS is representative of commercial LKAS, in the sense that it relies on dynamic equations to predict lane departures and does not adapt to the driver's driving style.

5.3 Testing approach

The three LKAS are evaluated using real data replayed offline, and simulated controller intervention. We collected 35 minutes of driving data on Highway 580 near Berkeley, California (USA). Measurements are collected every $\Delta t = 0.2\,\mathrm{s}$, therefore the dataset contains 10500 data points. The dataset contains 65 lane changes. When the data is replayed offline the turn signal is assumed to be off, therefore all the lane changes are treated as unintentional lane departures which should be predicted and avoided by a LKAS.

During the tests, for each LKAS, the collected real data is replayed until the LKAS detects an unintentional lane departure. When the controller gets switched on, the future vehicle states are simulated using the bicycle vehicle model described in Sec. 4.1.

5.4 Evaluation criteria

The three LKAS are evaluated based on their ability to predict lane departures and to avoid them. The following metrics are used:

- The median, min, and max prediction horizon, respectively denoted by T^p_{med}, T^p_{min}, and T^p_{max}. These metrics characterize how early the LKAS is able to predict an upcoming lane departure. Early predictions generally lead to increased safety and lower intrusiveness.
- The rate of successful interventions: $RSI = \frac{N_A}{N_{LD}} \times 100$, with N_A the number of lane departure instances where the LKAS intervened and was able to avoid the lane departure, and $N_{LD} = 65$ the number of lane departure instances in the collected data. This metric evaluates the safety performance of the system.
- The ratio of false alarms to lane departure instances: $RFL = \frac{N_F}{N_{LD}} \times 100$, with N_F the number of instances where the LKAS intervened unnecessarily. False alarms are a source of annoyance for the driver, and therefore should be low for the system to be accepted by drivers [1].
- The median absolute deviation from the initial driver steering when the controller is active, denoted by $\delta^{\Delta 0}_{med}$, where the deviation at time t is defined as: $\delta^{\Delta 0}(t) = (\delta^c(t) - \delta^d(t_{on}))$ and t_{on} is the timestep at which the controller was switched on. A LKAS which is able to keep the car in the lane with small control inputs is more likely to be accepted by the driver, as it is less intrusive [1].
- The median absolute deviation from the predicted driver steering when the controller is active, denoted by $\delta^{\Delta p}_{med}$, where the deviation at time t is defined as: $\delta^{\Delta p}(t) = (\delta^c(t) - \delta^p(t))$. This provides additional information about the intrusiveness of the system.

Choice of γ: In *LKAS2* the threshold γ needs to be tuned, and the system will perform differently depending on the choice of γ. There is no such parameter in the other two LKAS, which rely on a driver model to predict lane departures. In order to compare the three LKAS, the performance of *LKAS2* is evaluated for a specific threshold γ, which was selected so that the number of false alarms is the same as the one obtained by the other two LKAS.

Cross-validation: *LKAS2* can be directly evaluated on all the data points, but the two other LKAS rely on a learning-based driver model and need to be trained and tested on different partitions of the collected dataset. For that reason we perform 10-fold cross-validation. The collected dataset is partitioned into 10 partitions of equal durations, 9 of which are used for training and the remaining one for testing. This process is repeated 10 times, each time with a different partition for testing. This way, all three LKAS are tested on the entire collected dataset.

	Proposed LKAS	*LKAS1*	*LKAS2*
RFL	16.9 %	16.9 %	16.9 %
T^p_{med}	1.8 s	1.8 s	1.4 s
T^p_{min}	1.2 s	1.2 s	1.0 s
T^p_{max}	3.0 s	3.0 s	2.4 s
RSI	100.0 %	100.0 %	73.9 %
$\delta^{\Delta 0}_{med}$	4.9°	5.3°	6.4°
$\delta^{\Delta p}_{med}$	2.8°	3.3°	3.6°

Table 1: Results obtained by the three tested LKAS.

5.5 Results

The results obtained by the two reference LKAS and the proposed LKAS are provided in Table 1 and commented below.

- RFL: The RFL is the same for the three systems, since the threshold γ was set specifically to achieve that (see Sec. 5.4). The RFL is 16.9%, which is lower than what was reported by other works for the same prediction horizon [5]. However, for the comparison to be meaningful we would need to evaluate all these approaches on the same dataset, similarly to what we did here with *LKAS1*, *LKAS2*, and the proposed LKAS.
- T^p_{med}, T^p_{min}, and T^p_{max}: The use of a driver model for lane departure prediction increases the median, min, and max prediction horizons by 0.4 s, 0.2 s, and 0.6 s respectively, compared with predictions made by the TLC. Early predictions of lane departures allow less aggressive interventions from the controller and higher chances of successful lane keeping, as will be discussed below.
- RSI: Thanks to the early predictions of lane departures, *LKAS1* and the proposed LKAS successfully avoid all 65 lane departure instances. *LKAS2* fails in 17 out of 65 instances, because of late predictions. These late predictions all correspond to lane departures which are detected as they occur, therefore no amount of steering could prevent the vehicle from crossing the lane border. It is of course possible to bring the car back in its original lane after the lane border was crossed, but these are still considered unsuccessful interventions since the car entered another lane.
- $\delta^{\Delta 0}_{med}$: The deviation from the driver's initial steering is 1.1°smaller with *LKAS1* compared to *LKAS2*, thanks to the earlier predictions of lane departures. Using the driver model in the controller, as done in the proposed approach, brings the deviation down by an additional 0.4°. The median strength of the intervention of the proposed LKAS is therefore 23% smaller than a standard LKAS, which should be a very perceptible difference for the driver.

- $\delta_{med}^{\Delta p}$: The proposed LKAS generates control inputs which match better the driver's driving style, compared to the two reference LKAS. The steering inputs applied by the proposed approach are 22% closer to the driver's predicted steering, compared to a standard LKAS.

6. CONCLUSIONS

This paper presented an active LKAS which uses a driver model combined with model predictive control to prevent unintentional lane departures. The driver model is used in two ways: 1) for predicting unintentional lane departures, and 2) for generating control inputs which mimic the driver's driving style.

The results obtained with real driving data and simulated controller interventions show that the proposed LKAS is more efficient and less intrusive compared to a standard commercial LKAS. For an equivalent false alarm rate, our driver model predicts lane departures earlier than TLC-based approaches, therefore the rate of successful controller interventions is higher. In addition, the control inputs needed to keep the car in the lane are smaller and closer to the driver's predicted steering, which makes the interventions less intrusive for the driver.

In the near future, the proposed LKAS will be implemented on our test vehicle and compared with the commercial LKAS already present in the car. We also want to define some metrics to evaluate the similarity between different driving patterns, in order to measure how well our controller is able to reproduce driving styles that it can learn from data.

ACKNOWLEDGEMENTS

This work was partially supported by the National Science Foundation under grant No. 1239323 and by the Hyundai Center of Excellence at UC Berkeley.

REFERENCES

[1] J. Gayko, *Handbook of intelligent vehicles*. Springer, 2012, ch. Lane departure and lane keeping, pp. 689–708.

[2] A. Amditis, M. Bimpas, G. Thomaidis, M. Tsogas, M. Netto, S. Mammar, A. Beutner, N. Mohler, T. Wirthgen, S. Zipser, A. Etemad, M. D. Lio, and R. Cicilloni, "A situation-adaptive lane-keeping support system: overview of the SAFELANE approach," *IEEE Trans. on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 617–629, 2010.

[3] R. Risack, N. Mohler, and W. Enkelmann, "A video-based lane keeping assistant," in *Proc. IEEE Intelligent Vehicles Symposium*, 2000, pp. 356–361.

[4] S. Mammar, S. Glaser, and M. Netto, "Time to line crossing for lane departure avoidance: a theoretical study and an experimental setting," *IEEE Trans. on Intelligent Transportation Systems*, vol. 7, no. 2, 2006.

[5] P. Angkititrakul, R. Terashima, and T. Wakita, "On the use of stochastic driver behavior model in lane departure warning," *IEEE Trans. on Intelligent Transportation Systems*, vol. 12, no. 1, pp. 174–183, 2011.

[6] P. Kumar, M. Perrollaz, S. Lefèvre, and C. Laugier, "Learning-based approach for online lane change intention prediction," in *Proc. IEEE Intelligent Vehicles Symposium*, 2013, pp. 797–802.

[7] H. M. Mandalia and D. D. Salvucci, "Using support vector machines for lane change detection," in *Proc. of the Human Factors and Ergonomics Society 49th Annual Meeting*, 2005.

[8] A. Doshi and M. Trivedi, "On the roles of eye gaze and head dynamics in predicting driver's intent to change lanes," *IEEE Trans. on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 453–462, 2009.

[9] B. Morris, A. Doshi, and M. Trivedi, "Lane change intent prediction for driver assistance: on-road design and evaluation," in *Proc. IEEE Intelligent Vehicles Symposium*, 2011, pp. 895–901.

[10] N. Oliver and A. P. Pentland, "Graphical models for driver behavior recognition in a SmartCar," in *Proc. IEEE Intelligent Vehicles Symposium*, 2000, pp. 7–12.

[11] R. Terada, H. Okuda, T. Suzuki, K. Isaji, and N. Tsuru, "Multi-scale driving behavior modeling using hierarchical PWARX model," in *Proc. IEEE Conf. on Intelligent Transportation Systems*, 2010, pp. 1638–1644.

[12] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Low complexity mpc schemes for integrated vehicle dynamics control problems," *Proc. 9th International Symposium on Advanced Vehicle Control*, 2008.

[13] Y. Gao, A. Gray, J. V. Frasch, T. Lin, E. Tseng, J. K. Hedrick, and F. Borrelli, "Spatial predictive control for agile semi-autonomous ground vehicles," *Proc. 11th International Symposium on Advanced Vehicle Control*, 2012.

[14] A. Gray, M. Ali, Y. Gao, J. Hedrick, and F. Borrelli, "Integrated threat assessment and control design for roadway departure avoidance," *Proc. IEEE Conf. on Intelligent Transportation Systems*, pp. 1714–1719, 2012.

[15] P. Falcone, "Nonlinear model predictive control for autonomous vehicles," Ph.D. dissertation, Universitá del Sannio, Italy, 2007.

[16] S. Calinon, F. D'halluin, E. Sauser, D. Caldwell, and A. Billard, "Learning and reproduction of gestures by imitation," *IEEE Robotics Automation Magazine*, vol. 17, no. 2, pp. 44–54, 2010.

[17] R. Y. Hindiyeh and J. C. Gerdes, "Equilibrium analysis of drifting vehicles for control design," *Proc. Dynamic Systems and Control Conference*, pp. 181–188, 2009.